Knowledge-Based Systems 241 (2022) 108062

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

DeepTSQP: Temporal-aware service QoS prediction via deep neural network and feature integration

Guobing Zou [a], Tengfei Li [a], Ming Jiang [a], Shengxiang Hu [a], Chenhong Cao [a], Bofeng Zhang [a,*], Yanglan Gan [b,*], Yixin Chen [c]

[a] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[b] School of Computer Science and Technology, Donghua University, Shanghai, 201620, China
[c] Department of Computer Science and Engineering, Washington University in St. Louis, MO 63130, USA

ARTICLE INFO

Article history:
Received 3 June 2021
Received in revised form 19 December 2021
Accepted 24 December 2021
Available online 30 December 2021

Keywords:
Web service
QoS prediction
Deep neural network
Feature integration
Temporal aggregated feature mining

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 unknown service QoS has become a hot research issue. Although existing researches have been investigated on temporal-aware service QoS prediction, conventional approaches are restricted to a couple of limitations. (1) most of them cannot well mine the time-series relationships and the interaction invocation information among users and services. (2) even although some sophisticated approaches make use of recurrent neural networks for temporal service QoS prediction, they mainly focus on the learning of user-service temporal relationship and have paid less attention to more effectively represent implicit features, resulting in low accuracy on service QoS prediction. To deal with the challenges, we propose a novel deep learning based approach called DeepTSQP to perform the task of temporal-aware service QoS prediction by feature integration. In DeepTSQP, we first present an improved temporal feature representation of users and services by integrating binarization feature and similarity feature. Then, we propose a deep neural network with gated recurrent units (GRU), learning and mining temporal features among users and services. Finally, DeepTSQP model can be trained by parameter optimization and applied to predict unknown service QoS. Extensive experiments are conducted on a large-scale real-world temporal QoS dataset WS-Dream with 27,392,643 historical QoS invocation records. The results demonstrate that DeepTSQP significantly outperforms state-of-the-art approaches for temporal-aware service QoS prediction in terms of multiple evaluation metrics.

1. Introduction

With the wide adoption of service-oriented architecture (SOA), the number of web services is growing exponentially in recent years. Web service significantly accelerates the interoperable machine-to-machine interaction and greatly promotes the advancements on service discovery, optimal selection [1], automated composition and recommendation. By combining them with applications of different functions through interfaces, it realizes the service reuse and extension of functionality. However, as the overwhelming explosion on the number of web services registered on the Internet, the homogenization of service functions has been becoming more and more prevalent. That tends to be a labor-intensive challenging task for service consumers to select their desired services from a large-scale service repository. Thus, recommending superior web services from those functionally equivalent or similar ones has become a critical issue to be addressed.

QoS is usually employed as a key factor for describing non-functional features of web services, differentiating those web services with the same functionality. Due to the discrepant network environments and geographical locations, when different service consumers invoke the same service, they may receive totally different QoS experiences. On the contrary, when a service consumer invokes different web services, which provides diverse QoS experiences, since QoS measurement depends on the contextual information of both service providers and consumers. More importantly, QoS monitoring has highly relevance on temporal feature, when service consumers invoke web services at different time slices. In service-oriented application scenarios, it is a time-consuming and unrealistic task for service providers to monitor QoS of web services invoked at different moments,

* Corresponding authors.
E-mail addresses: gbzou@shu.edu.cn (G. Zou), mrblite@shu.edu.cn (T. Li), jimmyme520@gmail.com (M. Jiang), shengxianghu@shu.edu.cn (S. Hu), caoch@shu.edu.cn (C. Cao), bfzhang@shu.edu.cn (B. Zhang), ylgan@dhu.edu.cn (Y. Gan), chen@cse.wustl.edu (Y. Chen).

https://doi.org/10.1016/j.knosys.2021.108062
0950-7051/

leading to the issue of temporally predicting unknown QoS values of web services. To deal with this challenge, many efforts have been investigated to perform service QoS prediction by leveraging historical service QoS invocations.

It can be divided into traditional and temporal-aware QoS prediction based on whether QoS records in service invocation matrix change or not over time. For traditional service QoS prediction, memory-based [2–5], model-based [6–19] and deep learning based [20,21] approaches are investigated for predicting a vacant QoS value of a specified user invoking a corresponding service within a singleton QoS matrix. With the consideration of temporal feature where a user may have different QoS experiences by invoking a service across multiple time slices, there exist some researches on temporal-aware service QoS prediction in recent years, such as the approaches based on improved collaborative filtering [22,23], matrix factorization [24–27] and latent factor analysis [28,29]. Additionally, autoregressive integrated moving average(ARIMA) model-based approaches [30–33] have also been applied for temporal-aware service QoS prediction. However, in many real-world application scenarios, univariate ARIMA models might be inappropriate for predicting temporal-aware QoS values due to nonlinear QoS time series [34], which implies long-range dependency and contains correlation among users and services. Moreover, an observed tendency on temporal-aware service QoS prediction is the increasing leverage of deep learning techniques, such as approaches for recommender systems or missing QoS prediction based on recurrent neural network (RNN) [35–37] and its variant long short-term memory (LSTM) model [38–40]. These approaches can better mine the nonlinear time-series information and learn the interactive correlation among users and services.

Although these existing approaches can assist and facilitate service QoS prediction, they still cannot reach the satisfaction of service providers as well as service consumers. The underlying reason is that it is a difficult and challenging task of how to precisely represent user and service features at each time slice, and further effectively mine the implicit nonlinear relationship of user-service interactions across sequentially multiple time slices. More specifically, the deficiencies of current temporal-aware service QoS prediction are twofold. Most of conventional approaches cannot well mine the time-series information and the interaction invocation information of users and services. More importantly, although some existing approaches leverage recurrent neural networks for temporal service QoS prediction, they mainly focus on the learning of user-service temporal relationship, by using straightforward feature representation of user and service. That is, they have paid less attention on how to more effectively represent implicit features which are detailedly explained such as user-service invocation information and collaborative relationship on similar users or services, yielding to low accuracy on service QoS prediction.

To address the above issues, this paper proposes a novel deep learning based approach by enforcing feature representation and integration, called DeepTSQP, for temporal-aware service QoS prediction. First, binarization feature and similarity feature are taken into account and integrated as a whole to respectively represent a user's or service's feature, which reflects both dynamic invocation changes and similarity relationship across different temporal slices on user-service interactions. Then, it embeds a user's and service's features, which are reduced as low-dimensional ones and fused by concatenation to be fed into a deep neural network with gated recurrent units. Finally, temporal aggregated feature between a user and service is mined and applied to effectively predict unknown QoS value for a service to be invoked by a user at a temporal slice.

To evaluate the effectiveness of DeepTSQP in temporal service QoS prediction, we conduct extensive experiments on a real-world large scale dataset, which consists of a total number of 27,392,643 from WSDream [41]. The results demonstrate that DeepTSQP outperforms state-of-the-art benchmarking approaches. The main contributions of this paper are summarized as follows:

- We propose a comprehensive temporal-aware service QoS prediction framework via deep neural network and feature integration.
- We propose a novel approach for learning a user or service temporal feature representation. It simultaneously integrates both binarization feature for the user-service invocation changes across different temporal slices and similarity feature for collaborative relationships of users or services.
- Extensive experiments are conducted on a real-world large scale service QoS dataset under multiple temporal slices. The experimental results demonstrate that DeepTSQP receives superior performance, comparing with competing approaches in MAE and RMSE.

The remainder of this paper is organized as follows. Section 2 formulates the research problem. Section 3 illustrates the overall framework of DeepTSQP and elaborates the approach of temporal-aware service QoS prediction. Section 4 shows the experimental results and analyses. Section 5 provides the threats to validity. Section 6 reviews the related work. Finally, we conclude the paper and discuss the future work in Section 7.

2. Problem formulation

In this section, we first focus on the understanding of temporal-aware service QoS prediction problem by a set of formal definitions, which are detailedly explained by user-service QoS invocation matrix.

Definition 1 (*Temporal Service Ecosystem*)**.** A temporal web service ecosystem with temporal feature is defined as a four-tuple $M = \langle U, I, T, R\rangle$, where $U = \{u_1, u_2, ..\}$ is a set of users, $I = \{i_1, i_2, \ldots\}$ is a set of web services, and $T = \{t_1, t_2, \ldots\}$ is a set of temporal slices of service invocations. $R = \{r^t_{u,i}\}$ is a three-dimensional QoS matrix, where each entry $r^t_{u,i}$ represents a QoS value when a service i is invoked by a user u at temporal slice t.

Fig. 1 illustrates a temporal service ecosystem, which consists of a set of two-dimensional service invocation QoS matrices, denoted as $R = R_1 \cup R_2 ... \cup R_n$. Given a temporal slice t_j, its corresponding two-dimensional matrix R_{t_j} represents the QoS invocation records with a set of values when users invoke services during that time interval.

Definition 2 (*QoS Invocation Record*)**.** Given a temporal service ecosystem $M = \langle U, I, T, R\rangle$, a QoS invocation record is defined as a four-tuple $\langle u, i, t, r^t_{u,i}\rangle$, where $u \in U$ is a user, $i \in I$ is a web service, $t \in T$ is a temporal slice, and $r^t_{u,i}$ is the QoS value when u invoked i at t.

Here, when an entry of a QoS invocation record is equal to 0, indicating that a user has not ever invoked a service at that temporal slice. Therefore, its QoS value needs to be further predicted for use, which is defined as below.

Definition 3 (*Temporal-aware Service QoS Prediction Problem, TSQP*)**.** Given a temporal service ecosystem, the TSQP is defined as a five-tuple $\langle M, u, i, t, \widehat{r}^t_{u,i}\rangle$, where

(1) $M = \langle U, I, T, R\rangle$ is a temporal service ecosystem.
(2) u is a target user who desires to invoke a target web service i.

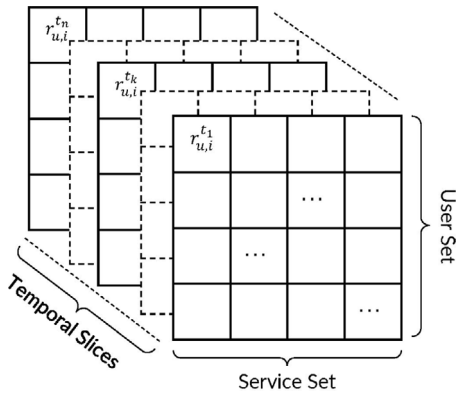


Fig. 1. Temporal-aware user-service QoS invocation matrix.

(3) i is a target web service that can be invoked by a target user u.
(4) t is a temporal slice where u invokes i for satisfying a certain functionality.
(5) $\widehat{r}_{u,i}^{t}$ is the QoS value to be predicted when a user u invokes a web service i at t.

In a temporal service ecosystem M, TSQP problem is dedicated to predicting unknown QoS value $\widehat{r}_{u,i}^{t}$ when u invokes i at t. As shown in Fig. 1, TSQP aims to predict those vacant QoS entries in temporal-aware service QoS invocation matrix. It is observed that there exist differentiated QoS values for the same service invoked by the same user across multiple temporal slices. In such case, how to design a novel approach to dynamically represent temporal features of a user and service over time and effectively learn their implicit nonlinear relationship on interactive invocation across multiple time has become a challenging issue to be solved. We present a deep learning based approach to perform temporal-aware service QoS prediction by feature integration in the subsequent section.

3. Approach

In this section, we first illustrate the overall framework of DeepTSQP. Then, we describe temporal feature representation and integration, temporal aggregated feature mining and QoS prediction, respectively. Subsequently, we elaborate the model training and parameter optimization. Finally, computational complexity is analyzed on DeepTSQP.

3.1. The framework of DeepTSQP

The overall framework of DeepTSQP is illustrated in Fig. 2. The objective of DeepTSQP is to automatically predict a QoS value, when a target user aims at invoking a target web service at a temporal slice. DeepTSQP consists of two independent but correlative components, including temporal feature representation and integration, temporal aggregated feature mining and QoS prediction.

In temporal feature representation and integration, it first characterizes a user's feature from the web services invoked by the specified user at a temporal slice by binarization representation; symmetrically, a service's feature is characterized by the users who has invoked the specified service at a temporal slice by binarization representation. To further take into account the contextual information of a user and service, it then integrates a target user's (or a target service's) binarization feature with a target service's (or a target user's) similarity feature that is mined from the user-service invocation QoS records.

In temporal aggregated feature mining and QoS prediction, it first makes a reduction of high-dimensionally sparse temporal feature of user and service by a fully-connected network, generating low-dimensionally dense feature. Then, they are fused by the concatenation as the inputs of GRU. Finally, the fused temporal features are split across multiple temporal slices and fed into GRU for mining the user-service nonlinear relationship of temporal feature and predicting the vacant service QoS.

3.2. Temporal feature representation and integration of user and service

To initialize the representation of a user or a service, one-hot encoding is widely applied for characterizing the position of a user or a service. Taking a user's feature representation as an example, only the value of the position indicated by the user ID is assigned as 1, whereas the values of remaining positions are set by 0. That is, the dimension of a user feature representation equals to the number of all users with one-hot encoding. Nevertheless, the disadvantage is that the feature representation cannot adapt to the fluctuation along with the variations at different temporal slices.

To reflect the temporal feature, we leverage the binarization representation to initialize temporal feature from user-service invocation QoS records. Conversely, the dimension of binarization feature vector of a user is the number of web services with the consideration of temporal representation, where the values of the positions indicated by all the services that the user has invoked are assigned as 1, while the remaining positions are set by 0. Table 1 illustrates and compares a user's feature representation by one-hot encoding and binarization representation. Since binarization representation contains information about the interactive invocation between a user and a service, it can dynamically reflect the implicit nonlinear relationship over time when a user invokes a service across multiple time slices. Compared with one-hot encoding, it is more beneficial to mine temporal aggregated feature for better service QoS prediction.

To further extend the temporal feature of binarization representation for a user, the preference feature of a target service is integrated to enrich the representation of temporal user feature, by evaluating the relevance between a target service and all web services with similarity calculation. More specifically, since a user's preference feature is affected by the distribution of web services and their invocation relevances, the similarity among web services as the heuristic context information is calculated by Pearson Correlation Coefficient (PCC). Given two web services i and j, the similarity is evaluated by

$$Sim(i,j) = \frac{\sum_{u\in U_c}(r_{u,i}-\bar{r}_i)(r_{u,j}-\bar{r}_j)}{\sqrt{\sum_{u\in U_c}(r_{u,i}-\bar{r}_i)^2}\sqrt{\sum_{u\in U_c}(r_{u,j}-\bar{r}_j)^2}} \quad (1)$$

where $U_c = U_i \cap U_j$ is the intersection of users who have previously invoked both i and j, $r_{u,i}$ indicates QoS value of service i invoked by user u, $\bar{r}_i$ and $\bar{r}_j$ represent average QoS value of i and j invoked by a set of common users in U_c, respectively. Assume that there are a set of services $I = \{i_1, i_2, \ldots\}$, by calculating the similarity between any two services, we can get a similarity matrix, and each row or column of the matrix corresponds to a service's similarity feature vector.

Finally, given a target user, the binarization feature vector $\mathbf{p_u}$ of a user is determined by the invocation information at a temporal slice. Simultaneously, similarity feature vector $\mathbf{q_u}$ of a target service is represented by invocation QoS records. More specifically, the binarization feature of a user indicates which services have been invoked by the target user, while the similarity feature of a user further reflects the similarity relationship

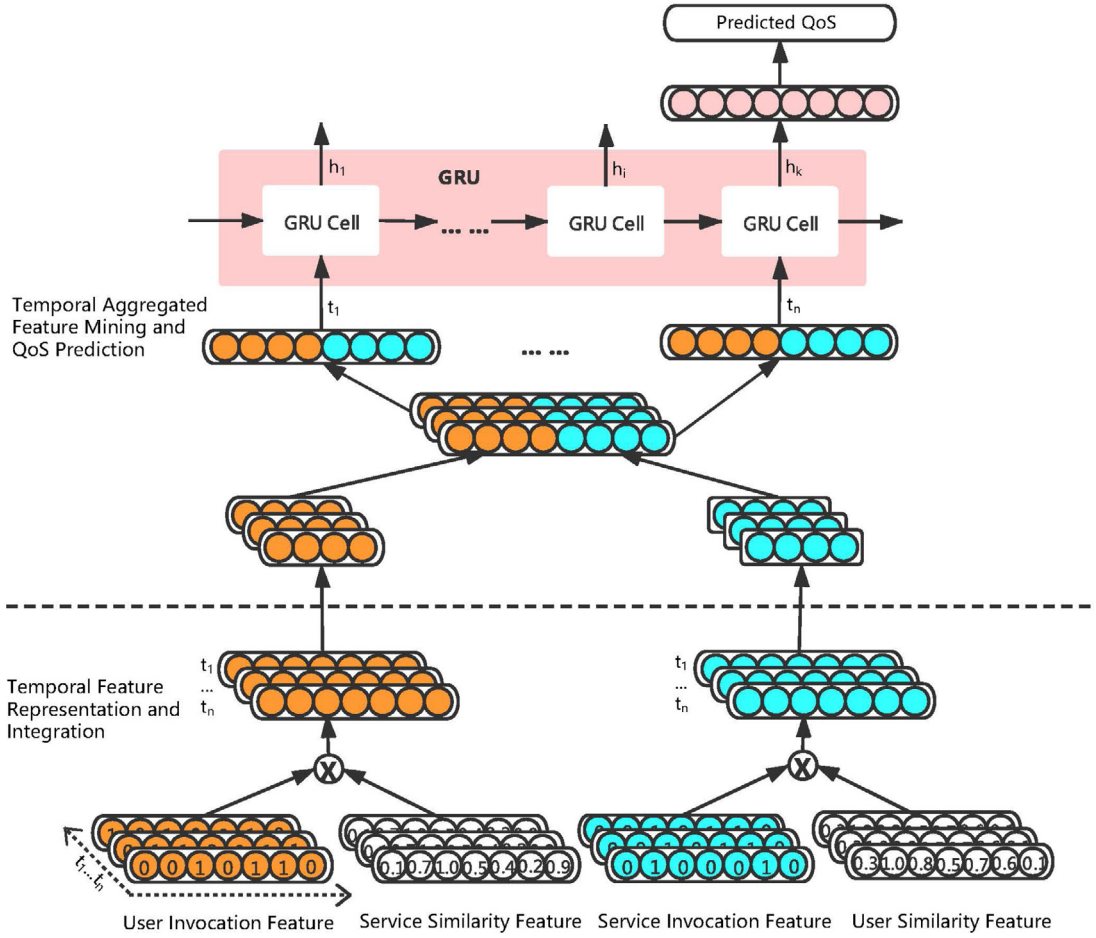


Fig. 2. The overall framework of DeepTSQP for temporal-aware service QoS prediction.

Table 1
Binarization feature representation across multiple temporal slices.

User	Service ID	Temporal slice	One-hot encoding	Binarization feature representation
u_1	1,2,3,7	t_1	0100000000	0111000100
u_1	4,5,8,9	t_2	0100000000	0000110011
u_2	3,4,7,9	t_1	0010000000	0001100101
u_2	1,5,6,8	t_2	0010000000	0100011010

between these services and the target service. That is, similarity feature representation to some extent is used as a weighting vector to quantitatively assist in improving the temporal feature representation of a user through qualitative user-service invocation information. After multiplying the two feature vectors by bit, the integrated temporal feature of a user is represented as $\mathbf{x_u}$, which is not only related to a user's invocation preference at a certain temporal slice, but also affected by the context of the relevance of a target service to be invoked. It is expressed as

$$\mathbf{x_u} = \mathbf{p_u} \odot \mathbf{q_u} \tag{2}$$

where $\odot$ means that each corresponding entry in the two feature vectors is multiplied by bit, and the obtained $\mathbf{x_u}$ is the integrated temporal feature vector of a user. In the task of temporal-aware service QoS prediction, we perform above integrated representation to obtain a series of feature vectors for all the users across sequentially multiple temporal slices.

In the same way, for the integrated feature representation of a service, we first obtain the binarization vector $\mathbf{p_i}$ according to the service's invocation information at a certain temporal slice. The length of service binarization vector is the number of all users, where the value of a position is 1, indicating that the corresponding user has invoked the service; otherwise, the rest of the positions are 0. Then, we calculate the similarity matrix among users with PCC, which is expressed by

$$Sim(u, v) = \frac{\sum_{i \in I_c} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_c} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_c} (r_{v,i} - \bar{r}_v)^2}} \tag{3}$$

Thus, we can extract the similarity feature vector of a user $\mathbf{q_i}$ who may desire to invoke the service. Specifically, the binarization feature of a service indicates which users have invoked the target user, while the similarity feature of a service further reflects the similarity relationship between these users and the target user. That is, similarity feature representation is used as a weighting vector to quantitatively assist in improving the temporal feature representation of a service through qualitative user-service invocation information. Finally, the two feature vectors are multiplied by bit to generate an integrated temporal feature representation of a service, which is formalized as

$$\mathbf{x_i} = \mathbf{p_i} \odot \mathbf{q_i} \tag{4}$$

Here, $\mathbf{x_u}$ and $\mathbf{x_i}$ as the integrated temporal feature representation of a user and service are used as inputs and fed into GRU for further temporal aggregated feature mining and QoS prediction.

3.3. Temporal aggregated feature mining and QoS prediction

Due to huge number of users and web services in service-oriented application systems, it is high-dimensional for a temporal feature vector. Moreover, since it is a time-consuming task to

invoke each web service for a user, the temporal feature vector is also very sparse, leading to difficulty in directly applying them as input features for further temporal aggregated feature mining. To transform a high-dimensional and sparse temporal feature vector of a user and service into a densely low-dimensional one, a fully-connected network is used to perform dimensionality reduction, which is formalized as

$$\begin{aligned}\mathbf{x}'_{\mathbf{u}} &= \sigma(\mathbf{W_u} \cdot \mathbf{x_u} + \mathbf{b_u}) \\ \mathbf{x}'_{\mathbf{i}} &= \sigma(\mathbf{W_i} \cdot \mathbf{x_i} + \mathbf{b_i})\end{aligned} \tag{5}$$

where σ is the activation function, and dimensionality reduction *Relu* function is applied for non-linear transformation of feature vector. After performing the dimensionality reduction, embedded user temporal feature $\mathbf{x}'_{\mathbf{u}}$ and embedded service temporal feature $\mathbf{x}'_{\mathbf{i}}$ are further fused as a whole by the concatenation. It is expressed by

$$\mathbf{x} = \mathbf{x}'_{\mathbf{u}} \oplus \mathbf{x}'_{\mathbf{i}} \tag{6}$$

where $\oplus$ represents the concatenation operation of two feature vectors. After the fusion of embedded temporal features of a user and a service, $\mathbf{x}$ is taken as input and fed into a GRU [42] model to mine temporal-aware implicit information on user-service invocation across multiple temporal slices. Given a temporal slice t, the extraction process of temporal aggregated feature mining is expressed by

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \cdot \mathbf{x}_t + \mathbf{U}_r \cdot \mathbf{h}_{t-1}) \tag{7}$$

where $\mathbf{x}_t$ represents the fused temporal feature at t, σ is the activation function, and $\mathbf{r}_t$ is a set of forget gates, which are calculated by the current input $\mathbf{x}_t$, weight coefficients $\mathbf{W}_r$ and $\mathbf{U}_r$, and the output of the previous moment $\mathbf{h}_{t-1}$. At the starting temporal slice t_0, $\mathbf{h}_{t-1}$ is a randomly initialized feature vector. According the obtained $\mathbf{r}_t$, the approximate temporal feature $\tilde{\mathbf{h}}_t$ at t can be calculated by

$$\tilde{\mathbf{h}}_t = tanh(\mathbf{W} \cdot \mathbf{x}_t + \mathbf{U}(\mathbf{r}_t \odot \mathbf{h}_{t-1})) \tag{8}$$

where $\odot$ represents multiplication operation by element. In case of $\mathbf{r}_t$ close to 0, indicating that the approximate temporal feature mainly depends on current state information $\tilde{\mathbf{h}}_t$, while the past state information should be forgotten at t. On the contrary, if $\mathbf{r}_t$ converges to 1, it reflects unreliable current state of the fused temporal feature x_t. Thus, it is necessary to ignore the current state as much as possible and retain the historical temporal features.

Based on the above state information $\tilde{\mathbf{h}}_t$, a weighting factor z_t through further learning is applied to update the temporal feature of user-service invocation $\mathbf{h}_t$, which can be expressed by

$$\begin{aligned}z_t &= \sigma(\mathbf{W}_z \cdot \mathbf{x}_t + \mathbf{U}_z \cdot \mathbf{h}_{t-1}) \\ \mathbf{h}_t &= (1 - z_t) \odot \mathbf{h}_{t-1} + z_t \odot \tilde{\mathbf{h}}_t\end{aligned} \tag{9}$$

where z_t is the updating weight to be learned in model training, and $\mathbf{h}_t$ is the temporal feature vector mined by GRU at temporal slice t.

For the QoS prediction of a service to be invoked by a user at a temporal slice k, we can leverage the temporal feature $\mathbf{h}_k$ mined from the learned GRU model. Finally, $\mathbf{h}_k$ is fed into a fully-connected network to predict the missing QoS value, which is expressed by

$$\hat{y} = Relu(\mathbf{W} \cdot \mathbf{h}_k + \mathbf{b}) \tag{10}$$

where $\hat{y}$ is the predicted service QoS, and $\mathbf{W}$ and $\mathbf{b}$ are the hyperparameters from model training, respectively.

3.4. Model training and parameter optimization

During the two stages of temporal feature representation and integration, and temporal aggregated feature mining and QoS prediction, a set of hyperparameters need to be learned and optimization by model training. It can be expressed by

$$\hat{r}_{u,i} = f(\mathbf{Q_u} \odot \mathbf{u},\ \mathbf{Q_i} \odot \mathbf{i}|\Theta_f) \tag{11}$$

where $\mathbf{Q_u}$ represents the similarity feature matrix among users, and $\mathbf{u}$ is the binarized feature vector of a certain user; $\mathbf{Q_i}$ represents the similarity feature matrix among services, and $\mathbf{i}$ is the binarization feature vector of a certain service; Θ_f consists of all the hyperparameters in the model; the operator $\odot$ represents the vector multiplication by bit. By model training, it aims to optimize the model parameters Θ_f through the learning from the training samples.

Since temporal-aware service QoS prediction is to solve a regression problem, MSE (Mean Squared Error) is taken as the optimization objective. It has been widely used in regression analysis and prediction problems. Here, the objective function of TSQP model training is as follows

$$J = \alpha * \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + (1-\alpha) * \sum_j w_j^2 \tag{12}$$

where $\hat{y}_i$ denotes the predicted service QoS value by the model, y_i represents the original service QoS value, and N is the number of training samples; w_j is a parameter value in the model; $\sum_j w_j^2$ is the regularization term of the model, which is used to avoid overfitting in model training; α is a weighting factor for balancing the importance of the regularization term, which is generally set to a value approximately close to 1 after iterative validation in the experiments.

To efficiently and optimally learn the parameters Θ_f, Adam (Adaptive Moment Estimation) [43] is used to train the model. By simulating the object motion model in classical physics, the learning rate is dynamically updated, yielding to a better local optimal point with more efficient convergence speed in the parameter domain. The learning rate of Adam is updated by

$$\begin{aligned}\hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1-\beta_2^t}\end{aligned} \tag{13}$$

where $\hat{m}_t$ represents the value of the momentum in the changing direction of the current one, and $\hat{v}_t$ represents the value of the speed in the changing direction of the current one. After each round of model training and parameter optimization, the two values of momentum and speed in the direction of the learning rate dynamically change. That incurs the adjustment of learning rate as follows

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \tag{14}$$

where θ_t is the learning rate of the tth batch in the stochastic gradient descent training process, and θ_{t+1} is the learning rate of the $t+1$th batch in model training. After the learning of hyperparameters Θ_f, the QoS prediction model can be formalized as

$$f = Adam(J|\Theta_f) \tag{15}$$

where f is the service QoS prediction model. In real-world applications, by taking unknown service invocation into the learned model as request, the service QoS at an expected certain temporal slice can be predicted for a target user invoking a target service.

Table 2
Statistics of WS-Dream Dataset.

Item name	Value
Users	142
Services	4,500
Service Invocations	27,392,643
Temporal Slices	64
Sparsity	66.98%

3.5. Computational complexity analysis

The time consumption of DeepTSQP is mainly from fully connected layer for user and service embedding and GRU layer for temporal aggregated feature mining. Specifically, let N_u denote the dimension of user temporal feature, which equals to the number of services, N_i denote the dimension of service temporal feature, which equals to the number of users, E_u denote the dimension of embedded user feature, E_i denote the dimension of embedded service feature, $N_{max} = max(N_u, N_i)$, $E_{max} = max(E_u, E_i)$. By calculating in Eqs. (5) and (6), the time computational complexity of the fully connected layer for temporal feature embedding is $O(N_uE_u + N_iE_i) = O(N_{max}E_{max})$. Let d denote the size of GRU hidden unit, ω denote the size of time window. By calculating in Eqs. (7), (8) and (9), time computational complexity of GRU layer is $O((d^2 + E_{max}d) * \omega)$.

From the above analyses, it is observed that time computational complexity of DeepTSQP is $O(\omega d^2 + \omega dE_{max} + N_{max}E_{max})$. Since, the selection of d and E_{max} is proportional to N_{max}, the above time computational complexity can be simplified to $O(\omega N_{max}^2)$. In real-world application scenarios, it generally satisfies that the number of time windows ω is much smaller than the number of users or services, i.e., N_{max}. Finally, the time complexity of DeepTSQP is quadratic to the number of N_{max}.

4. Experiments

4.1. Experimental setup and dataset

Our experiments are carried out on a workstation equipped with Intel Xeon Gold 6132 CPU, NVIDIA Geforce GTX 1080Ti GPU and 192 GB RAM. The modules of DeepTSQP in the experiments are implemented by Python 3.7.4 with Pytorch 1.1.0.

To validate the effectiveness of DeepTSQP, we conduct extensive experiments on a real-world web service QoS dataset called WS-Dream[1] [41], which has been widely used for service QoS prediction. Here, it consists of two kinds of temporal-aware service invocation QoS records, including response time (rtdata) and throughput (tpdata). We use rtdata as the experimental dataset, which has been extremely used for comparisons in the most of conventional approaches for temporal-aware service QoS prediction. Rtdata contains 142 independent users, 4500 web services, and a total number of 27,392,643 QoS invocation records. The user-service QoS invocation records are divided into 64 different temporal slices. The overall sparsity of WS-Dream dataset is approximately 66.98%. The statistics of the experimental dataset is illustrated in Table 2.

Based on the above dataset, we conducted a set of experiments to demonstrate the effectiveness and efficiency of our proposed DeepTSQP for temporal-aware service QoS prediction. In the experiments, we partition the rtdata into a series of subsets across sequentially multiple temporal slices, and generate training datasets under different QoS matrix densities. Specifically, the training QoS datasets are categorized into two groups:

(1) the first group of divided training datasets span from 5% to 20% with a QoS matrix density interval of 5%. Due to large number of users and web services, this kind of partitioning on QoS dataset can simulate the realistic application situation as much as possible, where QoS invocation matrix of users and services behaves with overwhelming sparsity;
(2) the second group of divided training datasets span from 10% to 60% with a QoS matrix density interval of 10%. The main purpose of partitioning QoS dataset into larger span of matrix densities is to test the performance of the temporal-aware service QoS prediction approach at different densities influenced by the values of different time windows.

After partitioning the rtdata with different densities, we denote the QoS invocation matrix at each temporal slice as $R_{train}^{(1)}, R_{train}^{(2)}, \ldots, R_{train}^{(k)}$. For the sampling of the test QoS dataset, we only perform it from the last temporal slice of the original QoS dataset denoted as $R_{test}^{(t)}$ in the experiments.

4.2. Evaluation metrics

Temporal-aware service QoS prediction is essentially a regression problem. Mean absolute error (MAE) and root mean square error (RMSE) are 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 follows.

$$MAE = \frac{\sum_{u,i} |r_{u,i} - \hat{r}_{u,i}|}{N} \tag{16}$$

where $r_{u,i}$ denotes the original QoS value of a target user invoking a service, $\hat{r}_{u,i}$ represents the QoS value predicted by a trained model, and N is the total number of samples to be predicted. From the definition of MAE, when the gap between y_i and $\hat{y}_i$ becomes smaller and smaller, the value of MAE tends to be much smaller accordingly, leading to better accuracy of vacant service QoS prediction.

Since MAE is linear to the deviation of QoS prediction, all the individual differences are weighted equally in the average, which cannot well uncover those predicted QoS having sharp deviations on their corresponding original ones. To this end, RMSE is applied to measure 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 follows.

$$RMSE = \sqrt{\frac{\sum_{u,i}(r_{u,i} - \hat{r}_{u,i})^2}{N}} \tag{17}$$

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

In the experiments, MAE reflects the overall accuracy of temporal service QoS prediction, which averages absolute deviations to the original QoS values. Compared with MAE, RMSE is more sensitive to individual outliers by representing a relatively higher weighting to large errors on predicted temporal-aware service QoS values.

4.3. Competing approaches

To evaluate the effectiveness of DeepTSQP, nine competing approaches, including one baseline, six state-of-the-art approaches, including WSPred [24], CARP [27], CLUS [27], RNCF [37], TUIPCC [23] and PLMF [40], and two GRU-based variants of our self-developed approaches. They are described as below.

[1] https://github.com/wsdream/wsdream-dataset.

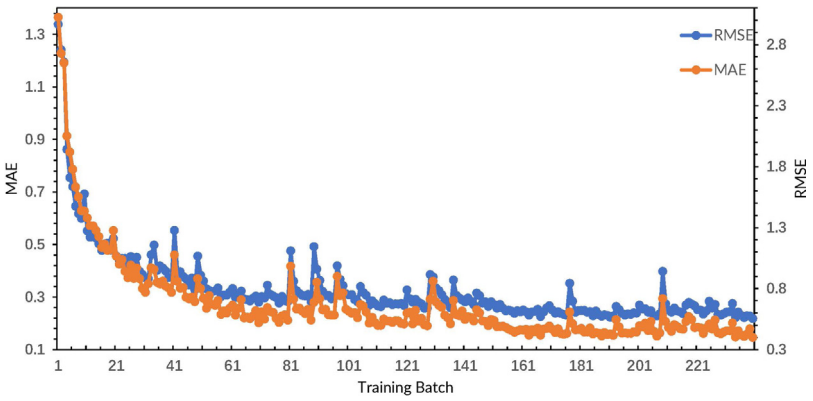


Fig. 3. The trend of training error with training batches.

- Mean: It is a basic comparison approach, which directly calculates the average QoS value of a service as its predicted QoS value.
- WSPred [24]: It leverages the traditional user-service two-dimensional matrix factorization, which is upgraded with temporal feature by three-dimensional tensor factorization. Compared with the traditional matrix factorization, it takes into account temporal feature and obtains more reliable service QoS prediction.
- CARP [25]: It sets up different time window sizes and aggregates all user-service invocation matrices in a time window. By aggregating invocation QoS matrices across multiple temporal slices, the vacant QoS values can be predicted by matrix factorization.
- CLUS [27]: It applies K-means to generate multiple dimensional clusters from user, service, and temporal slices. Then, the mean value of the clusters in each dimension is calculated as the axis to construct a feature space. Finally, the known QoS value is predicted by performing weighted calculation within the feature space.
- RNCF [37]: It incorporates a multi-layer GRU structure into neural collaborative filtering framework that shares the service QoS invocation records across different temporal slices.
- TUIPCC [23]: It is a temporal-aware collaborative filtering approach. Firstly, it calculates the average value of filtered historical QoS as temporal value. Then, it uses temporal-aware similarity computation mechanism to select neighborhood users (or services), and further predict CF-based QoS value. Finally, temporal value and CF-based QoS value are combined together to predict the missing QoS.
- PLMF [40]: It is an LSTM-based temporal-aware service QoS prediction approach. Firstly, the three dimensional tensor of user-service-time invocation relationship is represented with one-hot encoding. Then, the encoded feature vector is embedded through a fully-connected network to make dimensionality reduction. Finally, LSTM is applied to extract the implicit temporal feature and predict the missing service QoS.
- OneHotGRU: It is our self-developed variant approach. Here, we use one-hot encoding to generate feature representation of users and services. It is used to compare and verify the effectiveness of temporal feature representation and integration.
- BinGRU: It is our another self-developed variant approach. Unlike the DeepTSQP, it first represents temporal feature of users and services using binarization feature without neighborhood information. Then, we propose a deep neural network by GRU, mining temporal aggregated feature among users and services, and predicting the vacant service QoS.
- DeepTSQP: It is an improved version of BinGRU, and is our main proposed approach for temporal-aware service QoS prediction. Both binarization feature and similarity feature are integrated to represent temporal feature of users and services, which are fed into GRU to learn the nonlinear invocation relationship among users and services for temporal-aware service QoS prediction.

4.4. Experimental results and analyses

To validate the effectiveness of our proposed approach DeepTSQP for temporal-aware service QoS prediction, we compare it with state-of-the-art approaches. In the experiments, we run all these competing approaches with the same training and testing dataset, which are conducted for several times to guarantee the fairness of the performance comparison between our proposed approach and the baselines.

In the iterative training process through back propagation mechanism and Adam stochastic gradient descent algorithm, the performance error on the training set decreases as the number of iterative training increases. Fig. 3 illustrates one of the model training processes of DeepTSQP. From the results, we can find that RMSE gradually declines from 3.0 to around 0.6 to reach a steady state. Simultaneously, MAE varies from 1.4 to approximately 0.2 to keep a steady prediction accuracy.

After the model training is completed, the prediction results on the test set are obtained by feeding the test samples into the model. In order to compare with the existing approaches, three versions of the temporal-aware service QoS prediction approaches, denoted as DeepTSQP, BinGRU and OneHotGRU, are used in the experiments. Table 3 shows the experimental results of accuracy on service QoS prediction in terms of MAE and RMSE compared with state-of-the-art approaches. Here, lower MAE and RMSE indicate better prediction accuracy on service QoS prediction. From the experimental results, RNCF performs relatively well on RMSE, but it becomes significantly worse on MAE at different QoS matrix densities. At a specific density of 0.1, TUIPCC achieves superior QoS prediction accuracy on both MAE and RMSE, while it cannot keep well at remaining densities in service QoS matrix. Overall, PLMF performs the best at different densities on both MAE and RMSE among all state-of-the-art competing approaches. However, it is observed that our proposed approach DeepTSQP outperforms the most effective one PLMF, which has the highest QoS prediction accuracy among all the competing approaches. Specifically, compared with PLMF, the improvement of DeepTSQP on MAE ranges from 3.94% to 29.76% at different densities, where superior performance on QoS prediction can be obtained at higher densities. As for RMSE,

Table 3
The experimental results on temporal-aware service QoS prediction among competing approaches.

	MAE				RMSE			
Density	0.05	0.1	0.15	0.2	0.05	0.1	0.15	0.2
Mean	1.2212	1.1055	1.05	0.9782	2.8706	2.6573	2.5392	2.3252
WSPred	0.7317	0.6894	0.6734	0.6635	1.7074	1.6339	1.6068	1.5931
CLUS	0.7842	0.7542	0.735	0.7185	1.8921	1.903	1.9046	1.8948
CARP	0.8759	0.7709	0.7282	0.6992	2.2437	2.0397	1.9424	1.8556
RNCF	1.048	1.010	0.974	0.958	1.616	1.546	1.503	1.470
TUIPCC	0.7814	0.5767	0.8196	0.6970	1.7761	1.2076	2.0595	1.6358
PLMF	0.7267	0.6786	0.6582	0.6444	1.7059	1.6126	1.5749	1.5525
OneHotGRU	**1.036**	**0.9702**	**0.8887**	**0.8708**	**2.2479**	**2.3846**	**2.2296**	**2.2184**
BinGRU	**0.7101**	**0.701**	**0.5563**	**0.5268**	**1.7622**	**1.5503**	**1.4595**	**1.4116**
DeepTSQP	**0.698**	**0.5794**	**0.5202**	**0.4526**	**1.5937**	**1.4572**	**1.3366**	**1.214**
Gains	**3.94%**	**14.62%**	**20.96%**	**29.76%**	**6.58%**	**9.64%**	**15.13%**	**21.80%**

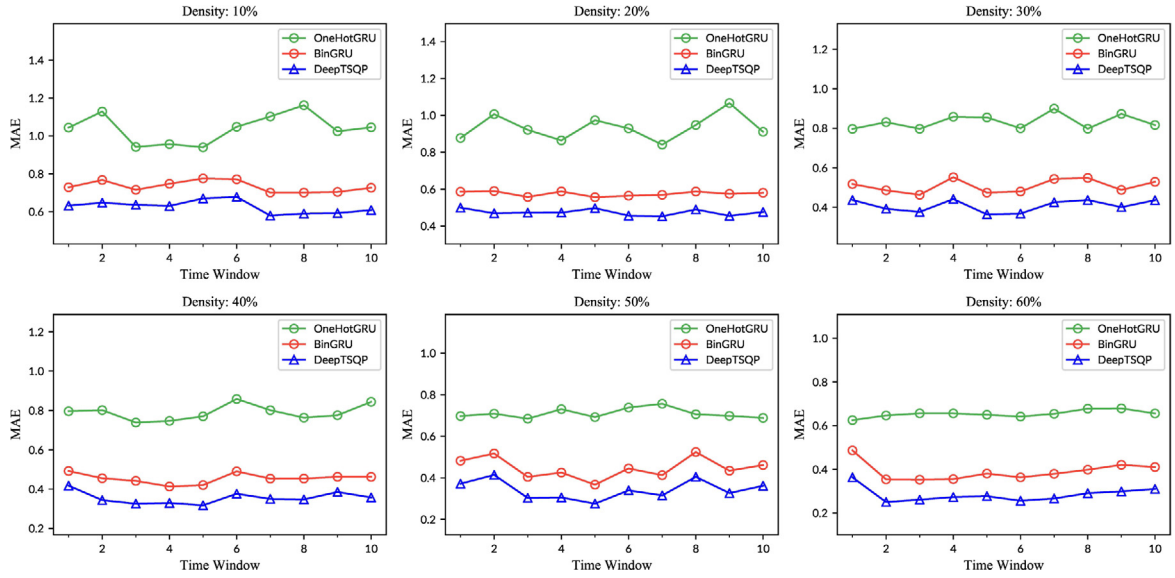


Fig. 4. Performance impact of temporal-aware service QoS prediction among DeepTSQP, BinGRU and OneHotGRU on MAE along with the variations of time window sizes at different matrix densities.

DeepTSQP receives better performance over PLMF from 6.58% to 21.80%. The main reason can be explained by two aspects. First, an integrated temporal feature representation is taken by integrating binarization feature and similarity feature, which leads to better temporal feature of a user and a service by qualitatively and quantitatively reflecting the latest invocation information at a specified temporal slice. Second, DeepTSQP leverages the advanced recurrent neural network GRU for more effectively learning the implicit nonlinear relationship when a user invokes a service across multiple temporal slices.

In order to analyze the performance impact of proposed approach DeepTSQP on MAE and RMSE, a set of experiments are carried out by sampling the rtdata of WS-Dream dataset. The QoS matrix density spans from 10% to 60%, and the interval is set by 10%. Meanwhile, the size of time window is set from 1 to 10, where the interval is set by 1. The accuracy impact of service QoS prediction on MAE and RMSE among DeepTSQP, BinGRU and OneHotGRU along with the variations of time window sizes at different matrix densities are shown in Fig. 4 and Fig. 5, respectively.

It is observed from the results that, DeepTSQP receives more accurate QoS prediction performance compared with our two self-developed variants BinGRU and OneHotGRU at different matrix densities. The possibility on the superiority of DeepTSQP is that it performs better representation of temporal feature than BinGRU and OneHotGRU, where DeepTSQP represents a user's or a service's temporal feature by considering both similar neighborhood information and interactive invocations among users and services. However, BinGRU only considers the binarization representation to characterize temporal feature of a user or a service, resulting in the loss of collaborative similarity among users and services. Meanwhile, OneHotGRU utilizes one-hot encoding to distinguish individual users or services, while it is difficult to capture dynamic changes of interactive invocations among users and services over time.

Furthermore, a comprehensive analysis on the variations of both matrix density and time window size is performed with three-dimensional visualization as illustrated in Fig. 6, which shows the accuracy of service QoS prediction among MAE and RMSE for OneHotGRU, BinGRU and DeepTSQP as the variations of matrix density and time window size. From the results of three-dimensional visualization, it can demonstrate that higher matrix densities receive better overall prediction accuracy than those at low densities, no matter how many time windows have been taken into account for service QoS prediction. However, temporal features play an extraordinary role in predicting missing QoS of web services, as the matrix density changes at different levels. We conclude that they interact with each other under three different circumstances. When the matrix density is extremely low with sparse service QoS invocation records, the influence of a larger time window is beneficial to positively provide more temporal invocation information for mining the implicit nonlinear relationship of users and services, yielding to better QoS prediction accuracy. As the matrix density becomes higher and higher, a larger time window gradually drops off on the service

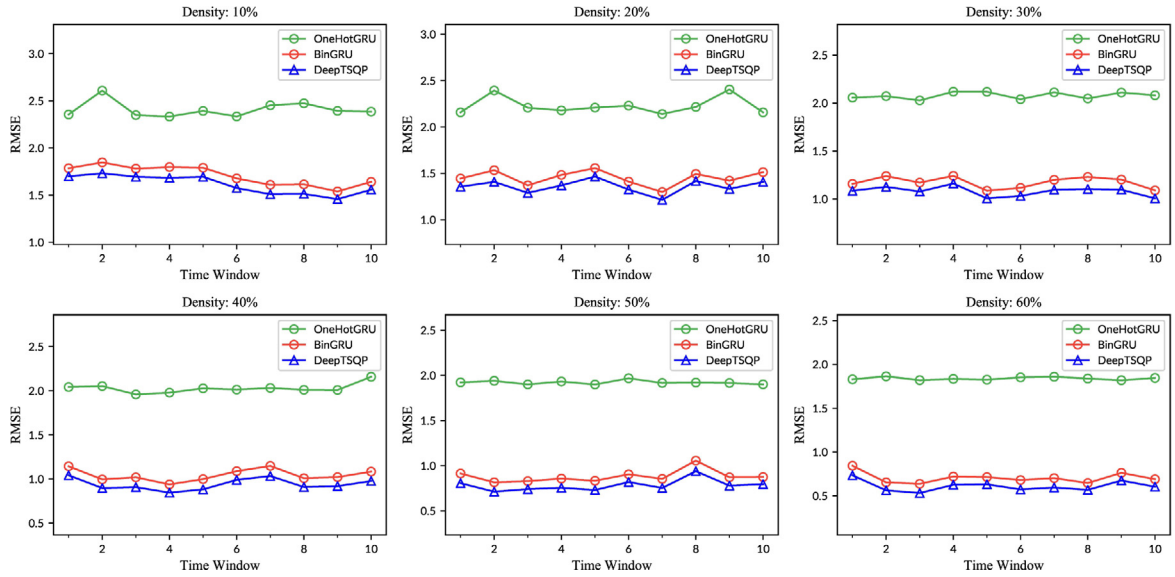


Fig. 5. Performance impact of temporal-aware service QoS prediction between DeepTSQP, BinGRU and OneHotGRU on RMSE along with the variations of time window sizes at different matrix densities.

QoS prediction. Moreover, when the matrix density continues to increase, a larger time window even incur a worse performance on service QoS prediction.

The above phenomena can be more deeply explained from the context of service invocation. The QoS values of web services published on the Internet are more closely related to the current contextual environment. That is, the longer the interval of QoS invocation records, the greater the degree of changes on network application context. Therefore, the historical QoS data closer to the current temporal slice has more relevance to service QoS prediction, while those far away from the current temporal slice have less influence on the prediction of missing service QoS.

Consequently, although service QoS prediction accuracy can be improved by mining implicit temporal relationship from service QoS invocation records over a period of time, it is harmful to apply more historical QoS invocation data with longer time intervals. The reason is that they may partially dilute the importance of those service QoS invocation data closer to the current temporal slice. Therefore, there are two extreme possibilities in application scenarios. One one hand, in case of low QoS matrix density, missing QoS prediction cannot be effectively performed by the current temporal slice due to the insufficient provision of service QoS invocation records, where mining implicit temporal trend on service QoS invocation data as heuristics can be an effective way to obtain more accurate service QoS prediction. On the other hand, when the QoS matrix density is high enough, there are adequate historical QoS data within the current temporal slice for accurately predicting missing QoS values, where the incorporation of temporal features may conversely degrade the purity of service QoS data in the current temporal slice and reduce the accuracy of service QoS prediction. As the theoretical basis from [23,40], they have the similar analytical experiment results on the relationship between the size of time window and the accuracy of service QoS prediction.

From the perspective of real-world application scenarios, since the number of both users and web services are exponentially growing and becoming larger and larger, the corresponding number of user-service QoS invocation records is often very small. In such case, service ecosystem only has high sparse service QoS invocation matrix. Therefore, it is of great practical significance to mine temporal invocation relationship of users and services under the sparsity of the QoS invocation matrix, which has potential applications in effectively improving the accuracy of service QoS prediction.

5. Threats to validity

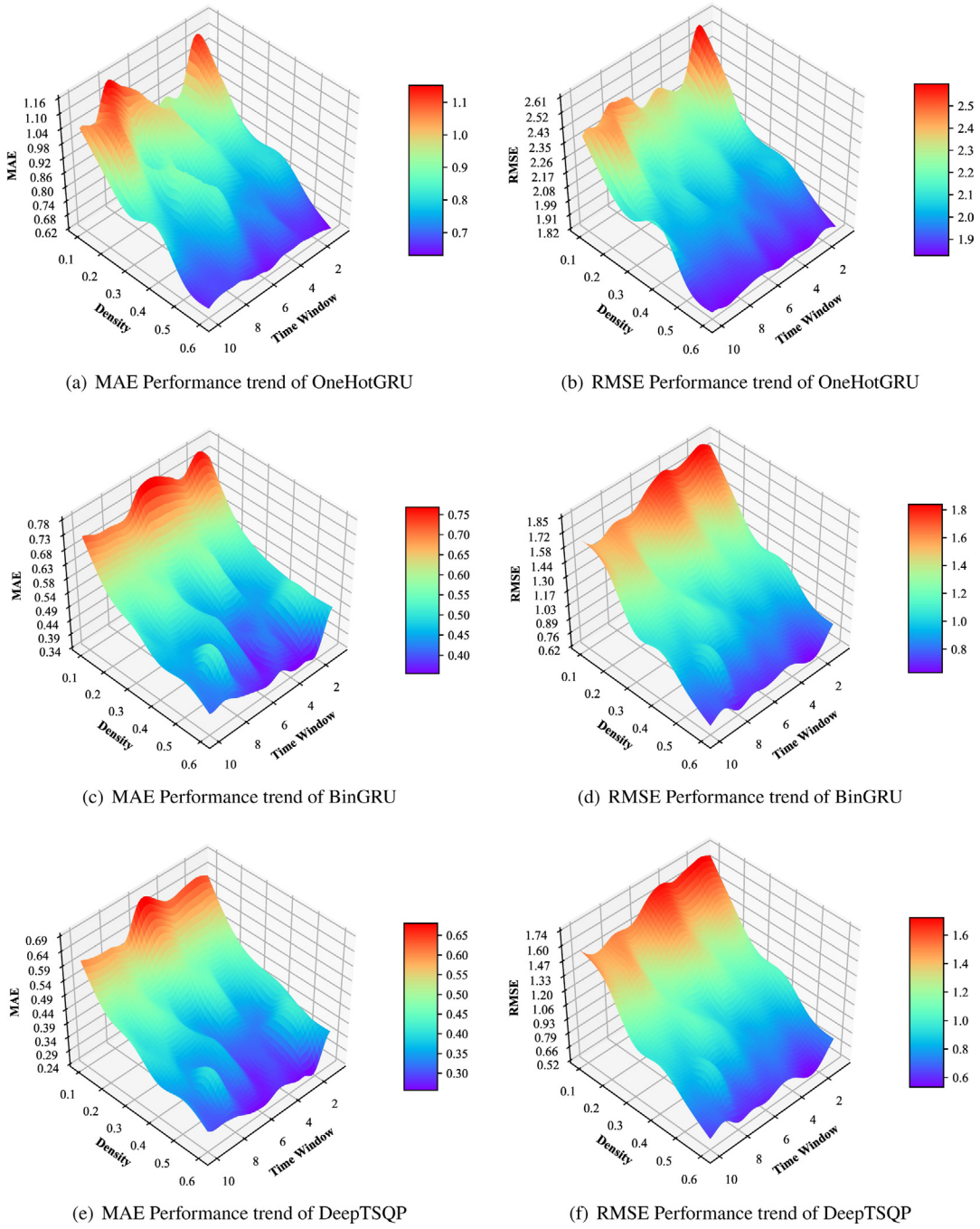
Threats to similarity validity. In real-world scenarios, a user usually experiences only a very small subset of all the available services, resulting in the sparsity of user-service QoS invocation matrix. We use PCC to calculate the similarity between users or services when performing similarity feature representation. However, due to sparse user-service QoS invocation matrix, there are few records of common QoS invocations between users or services, which may lead to inaccurate calculation of similarity and partially affect the accuracy of final missing QoS prediction. From the experimental results, it can be seen that although our approach performs better than the competing ones, the overall prediction results become significantly worse as the density of QoS invocation matrix becomes lower. In recent years, there have been related studies on similarity calculation of sparse matrix [18, 19] that considered the indirect similarity relationship of users or services to find implicit neighborhood information, potentially boosting the QoS prediction accuracy. In future work, we plan to further make improvements on similarity calculation for sparse user-service QoS invocation matrix.

Threats to dataset validity. In the experiments, WS-Dream dataset is used as training and testing samples, which are collected from real-world web services and contain full elemental characteristics for service QoS prediction. Nevertheless, as IT evolves and network environment changes, WS-Dream may not reflect the latest web service QoS invocation data. Due to the outdated QoS records, that would possibly affect the effectiveness of our proposed approach for temporal-aware service QoS prediction to a certain extent. Therefore, it is necessary to keep track of the latest advancements on dataset and conduct more experiments to further validate and optimize the prediction accuracy of DeepTSQP, once the new released dataset of temporal QoS invocation records can be available collected by researchers on the Internet.

6. Related work

6.1. Traditional service QoS prediction

Predicting missing service QoS in a traditional way can be categorized by three types of approaches, including memory-based, model-based and deep learning based service QoS prediction.



(a) MAE Performance trend of OneHotGRU (b) RMSE Performance trend of OneHotGRU

(c) MAE Performance trend of BinGRU (d) RMSE Performance trend of BinGRU

(e) MAE Performance trend of DeepTSQP (f) RMSE Performance trend of DeepTSQP

Fig. 6. Experiment results of changing trend on QoS prediction with the variations of both matrix density and time window.

Memory-based approaches mainly leverage collaborative filtering (CF) for service QoS prediction, which consists of average QoS value calculation and deviation migration. This kind of approach is usually classified into user-based, service-based, and their linear combination with a weight coefficient [2]. The core idea of memory-based service QoS approach is to find a set of similar users or services as neighborhood by similarity calculation, which is used to perform deviation migration and integrated with average QoS value. In recent years, some researchers have mainly focused on approaches that can more accurately quantify the similarity among users and services [3,4]. In addition, some efforts have been imposed on enhancing the accuracy of QoS prediction by introducing external heuristic information from users or services [5].

To alleviate the computing overhead and further boost the QoS prediction accuracy, model-based approaches have been fully investigated by matrix factorization (MF) and its variants [7,9], as well as the improved factorization-based machines [10,11]. The basic idea of predicting the unknown service QoS is to transform a sparse higher-order matrix into an equivalent multiplication of two lower-order matrices. After the two low-order matrices are determined separately, those missing QoS values in the original sparse matrix can be obtained by multiplying the two low-order matrices together. Based on matrix factorization, the latent factor model and its multiple variants have been widely investigated to predict missing QoS values due to high scalability and prediction accuracy [12–17].

With the wide application of deep learning, researchers have leveraged diverse deep neural networks to train effective prediction model for solving service QoS prediction problem [20,21]. Wu et al. [20] proposed contextual features that are mapped into a shared latent space to semantically characterize users and services in the embedding layer, which effectively boosts the prediction accuracy of missing service QoS. Furthermore, they further extend from [20] to propose a topology-aware neural (TAN) model for highly-accurate QoS prediction [21], where bi-directional long short-term memory is applied to capture the forward and backward dependence of a path and merge the features of the nodes on the path into a single vector representation for explicit modeling service invocations.

6.2. Temporal-aware service QoS prediction

Service QoS prediction with the consideration of temporal feature has received many attentions in recent years. Temporal-aware service QoS prediction can be summarized by the following three aspects, including the integration of a temporal dimension to the traditional approaches, ARIMA model-based approaches, and deep learning-based approaches.

Hu et al. [22] integrated time information into the similarity measurement and the service QoS prediction of the traditional neighborhood-based CF, where they proposed a random walk algorithm to select indirect similar users and services from user and service graph, respectively. Tong et al. [23] proposed an improved collaborative filtering based missing QoS prediction approach. Firstly, it filtered out the historical QoS values with appropriate timeliness. Then, it integrated the similarity computation results at different temporal slices and selected similar users or services as neighborhood information. Finally, it predicted missing QoS values through hybrid collaborative filtering. Li et al. [26] proposed a three-dimensional tensor to represent the relationship among users, services and temporal features. Then, the traditional matrix factorization algorithm is upgraded to adapt to three-dimensional tensor factorization for more accurate service QoS prediction and recommendation. Meng et al. [28] proposed a temporal-aware hybrid collaborative approach of cloud service recommendation, which analyzed the changes of service QoS and user interests across various time series. Luo et al. [29] proposed a biased non-negative latent factorization of tensors (BNLFTs) model, which extracted temporal latent factors from dynamic QoS data for predicting missing ones. During the process of model training, it integrated the principle of single LF-dependent, non-negative, and multiplicative update (SLF-NMU) and alternating direction method (ADM) to promote the effectiveness of service QoS prediction.

Due to the correlation between the tasks of temporal-aware service QoS prediction and sequence prediction analysis, correlative research exploits the ARIMA model to predict unknown service QoS. Hu et al. [30,31] formulated an ARIMA model of service QoS values, and applied Kalman filter algorithm to predict temporal-aware service QoS. Amin et al. [32] proposed a hybrid approach that integrates ARIMA and generalized autoregressive conditional heteroskedasticity model to effectively solve the problem of time-series service QoS prediction. Ding et al. [33] combined ARIMA model with a memory-based collaborative filtering approach, where the nearest neighbor collaborative filtering algorithm is used to predict a target user's personalized service QoS value based on the fundamental value of ARIMA model.

With the advancements of deep learning techniques, researchers have investigated deep learning based service QoS prediction with temporal feature. In recent years, RNN and its variant LSTM model are commonly applied in temporal-aware service QoS prediction. Ko et al. [35], Wu et al. [36] and Liang et al. [37] applied RNN models to recommender systems, where time-series information is fed into RNN to fully extract feature representation for further recommendation task. By applying an improved recurrent neural network model LSTM, Wang et al. [38] and Xiong et al. [39] considered temporal feature and received remarkable effectiveness for service QoS prediction. After that, Xiong et al. [40] proposed a personalized LSTM based matrix factorization approach PLMF, which can dynamically capture the latent representations of users and services for temporal-aware service QoS prediction.

Motivated by the above investigations, we aim at focusing on the issue of temporal-aware service QoS prediction by deep learning and feature integration. To solve this problem, we propose a novel deep learning based approach DeepTSQP, which can significantly improve the accuracy of service QoS prediction, by learning the nonlinear invocation relationship among users and services with GRU and promoting the feature representation with the combination of binarization and similarity features.

7. Conclusion and future work

In this paper, we focus on the issue of temporal-aware service QoS prediction by novel feature representation of users and services. Binarization feature and neighborhood similarity feature are integrated together to reflect the dynamic temporal feature of a user and a service along with the variations of interactive invocations over time. Moreover, GRU as an advanced recurrent neural network is applied to mine temporal aggregated features across multiple temporal slices, which can more effectively capture the implicit nonlinear relationship among users and services leading to better performance of service QoS prediction. Extensive experiments have been conducted on a real-world temporal QoS invocation dataset. The results demonstrate that DeepTSQP can receive superior accuracy of service QoS prediction compared with state-of-the-art benchmarking approaches.

In the future, we plan to further explore the integration of external heuristic information such as geographical locations of users and services into DeepTSQP for boosting the accuracy of temporal-aware service QoS prediction.

CRediT authorship contribution statement

Guobing Zou: Investigation, Conceptualization, Methodology, Writing – review & editing. **Tengfei Li:** Methodology, Software, Data curation, Formal analysis, Validation, Writing – original draft. **Ming Jiang:** Methodology, Software, Data curation, Formal analysis, Validation. **Shengxiang Hu:** Visualization, Methodology, Software, Formal analysis, Validation. **Chenhong Cao:** Visualization, Writing – review & editing. **Bofeng Zhang:** Supervision, Resources, Funding acquisition, Writing – review & editing. **Yanglan Gan:** Supervision, Validation, Resources, Funding acquisition. **Yixin Chen:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 61772128, 62172088), and Shanghai Natural Science Foundation, China (No. 21ZR1400400).

References

[1] Qiang He, Jun Han, Feifei Chen, Yanchun Wang, Rajesh Vasa, Yun Yang, Hai Jin, QoS-Aware service selection for customisable multi-tenant service-based systems: Maturity and approaches, in: IEEE International Conference on Cloud Computing, 2015, pp. 237–244.

[2] Zibin Zheng, Hao Ma, Michael R. Lyu, Irwin King, Qos-aware web service recommendation by collaborative filtering, IEEE Trans. Serv. Comput. 4 (2) (2011) 140–152.

[3] Huifeng Sun, Zibin Zheng, Junliang Chen, Michael R. Lyu, Personalized web service recommendation via normal recovery collaborative filtering, IEEE Trans. Serv. Comput. 6 (4) (2013) 573–579.

[4] Xiaokun Wu, Bo Cheng, Junliang Chen, Collaborative filtering service recommendation based on a novel similarity computation method, IEEE Trans. Serv. Comput. 10 (3) (2015) 352–365.

[5] Wei Lo, Jianwei Yin, Shuiguang Deng, Ying Li, Zhaohui Wu, Collaborative web service QoS prediction with location-based regularization, in: IEEE International Conference on Web Services, 2012, pp. 464–471.

[6] Pinjia He, Jieming Zhu, Zibin Zheng, Jianlong Xu, Michael R. Lyu, Location-based hierarchical matrix factorization for web service recommendation, in: IEEE International Conference on Web Services, 2014, pp. 297–304.

[7] Yueshen Xu, Jianwei Yin, Shuiguang Deng, Neal N. Xiong, Jianbin Huang, Context-aware QoS prediction for web service recommendation and selection, Expert Syst. Appl. 53 (2016) 75–86.

[8] Zibin Zheng, Hao Ma, Michael R. Lyu, Irwin King, Collaborative web service QoS prediction via neighborhood integrated matrix factorization, IEEE Trans. Serv. Comput. 6 (3) (2013) 289–299.

[9] Chao Chen, Dongsheng Li, Qin Lv, Junchi Yan, Li Shang, Stephen M. Chu, GLOMA: Embedding global information in local matrix approximation models for collaborative filtering, in: AAAI Conference on Artificial Intelligence, 2017, pp. 1295–1301.

[10] Yaoming Wu, Fenfang Xie, Liang Chen, Chuan Chen, Zibin Zheng, An embedding based factorization machine approach for web service QoS prediction, in: International Conference on Service-Oriented Computing, 2017, pp. 272–286.

[11] Buqing Cao, Jianxun Liu, Yiping Wen, Hongtao Li, Qiaoxiang Xiao, Jinjun Chen, QoS-aware service recommendation based on relational topic model and factorization machines for IoT mashup applications, J. Parallel Distrib. Comput. 132 (2019) 177–189.

[12] Di Wu, Xin Luo, Mingsheng Shang, Yi He, Guoyin Wang, MengChu Zhou, A deep latent factor model for high-dimensional and sparse matrices in recommender systems, IEEE Trans. Syst. Man Cybernet. Syst. 51 (7) (2019) 4285–4296.

[13] Xin Luo, MengChu Zhou, Shuai Li, Di Wu, Zhigang Liu, Mingsheng Shang, Algorithms of unconstrained non-negative latent factor analysis for recommender systems, IEEE Trans. Big Data 7 (1) (2019) 227–240.

[14] Xin Luo, Zidong Wang, Mingsheng Shang, An instance-frequency-weighted regularization scheme for non-negative latent factor analysis on high-dimensional and sparse data, IEEE Trans. Syst. Man Cybernet. Syst. 51 (6) (2019) 3522–3532.

[15] Di Wu, Mingsheng Shang, Xin Luo, Zidong Wang, An L_1-and-L_2-norm-oriented latent factor model for recommender systems, IEEE Trans. Neural Netw. Learn. Syst. (2021) http://dx.doi.org/10.1109/TNNLS.2021.3071392.

[16] Xin Luo, Ye Yuan, MengChu Zhou, Zhigang Liu, Mingsheng Shang, Non-negative latent factor model based on β-divergence for recommender systems, IEEE Trans. Syst. Man Cybernet. Syst. 51 (8) (2019) 4612–4623.

[17] Di Wu, Xin Luo, Robust latent factor analysis for precise representation of high-dimensional and sparse data, IEEE/CAA J. Autom. Sin. 8 (4) (2020) 796–805.

[18] Di Wu, Qiang He, Xin Luo, Mingsheng Shang, Yi He, Guoyin Wang, A posterior-neighborhood-regularized latent factor model for highly accurate web service QoS prediction, IEEE Trans. Serv. Comput. (2019) http://dx.doi.org/10.1109/TSC.2019.2961895.

[19] Di Wu, Xin Luo, Mingsheng Shang, Yi He, Guoyin Wang, Xindong Wu, A data-characteristic-aware latent factor model for web services qos prediction, IEEE Trans. Knowl. Data Eng. (2020) http://dx.doi.org/10.1109/TKDE.2020.3014302.

[20] Hao Wu, Zhengxin Zhang, Jiacheng Luo, Kun Yue, Ching-Hsien Hsu, Multiple attributes QoS prediction via deep neural model with contexts, IEEE Trans. Serv. Comput. 14 (4) (2021) 1084–1096.

[21] Jiahui Li, Hao Wu, Jiapei Chen, Qiang He, Ching-Hsien Hsu, Topology-aware neural model for highly accurate QoS prediction, IEEE Trans. Parallel Distrib. Syst. 33 (7) (2022) 1538–1552.

[22] Yan Hu, Qimin Peng, Xiaohui Hu, A time-aware and data sparsity tolerant approach for web service recommendation, in: IEEE International Conference on Web Services, 2014, pp. 33–40.

[23] Endong Tong, Wenjia Niu, Jiqiang Liu, A missing QoS prediction approach via time-aware collaborative filtering, IEEE Trans. Serv. Comput. (2021) http://dx.doi.org/10.1109/TSC.2021.3103769.

[24] Yilei Zhang, Zibin Zheng, Michae Lyu, WSPred: A time-aware personalized QoS prediction framework for Web services, in: IEEE International Symposium on Software Reliability Engineering, 2011, pp. 210–219.

[25] Marin Silic, Goran Delac, Sinisa Srbljic, Prediction of atomic web services reliability for QoS-aware recommendation, IEEE Trans. Serv. Comput. 8 (3) (2014) 425–438.

[26] Zhi Li, Jian Cao, Qi Gu, Temporal-aware QoS-based service recommendation using tensor decomposition, Int. J. Web Serv. Res. 12 (1) (2015) 62–74.

[27] Jieming Zhu, Pinjia He, Qi Xie, Zibin Zheng, Michael R. Lyu, CARP: Context-aware reliability prediction of black-box web services, in: IEEE International Conference on Web Services, 2017, pp. 17–24.

[28] Shunmei Meng, Zuojian Zhou, Taigui Huang, Duanchao Li, Song Wang, Fan Fei, Wenping Wang, Wanchun Dou, A temporal-aware hybrid collaborative recommendation method for cloud service, in: IEEE International Conference on Web Services, 2016, pp. 252–259.

[29] Xin Luo, Hao Wu, Huaqiang Yuan, MengChu Zhou, Temporal pattern-aware QoS prediction via biased non-negative latent factorization of tensors, IEEE Trans. Cybern. 50 (5) (2019) 1798–1809.

[30] Yan Hu, Qimin Peng, Xiaohui Hu, Rong Yang, Web service recommendation based on time series forecasting and collaborative filtering, in: IEEE International Conference on Web Services, 2015, pp. 233–240.

[31] Rodrigo N. Calheiros, Enayat Masoumi, Rajiv Ranjan, Rajkumar Buyya, Workload prediction using ARIMA model and its impact on cloud applications' QoS, IEEE Trans. Cloud Comput. 3 (4) (2014) 449–458.

[32] Ayman Amin, Alan Colman, Lars Grunske, An approach to forecasting QoS attributes of web services based on ARIMA and GARCH models, in: IEEE International Conference on Web Services, 2012, pp. 74–81.

[33] Shuai Ding, Yeqing Li, Desheng Wu, Youtao Zhang, Shanlin Yang, Time-aware cloud service recommendation using similarity-enhanced collaborative filtering and ARIMA model, Decis. Support Syst. 107 (2018) 103–115.

[34] Yang Syu, Chien-Min Wang, QoS time series modeling and forecasting for web services: a comprehensive survey, IEEE Trans. Netw. Serv. Manag. 18 (1) (2021) 926–944.

[35] Young Jun Ko, Lucas Maystre, Matthias Grossglauser, Collaborative recurrent neural networks for dynamic recommender systems, in: Asian Conference on Machine Learning, 2016, pp. 366–381.

[36] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, How Jing, Recurrent recommender networks, in: ACM International Conference on Web Search and Data Mining, ACM, 2017, pp. 495–503.

[37] Tingting Liang, Manman Chen, Yuyu Yin, Li Zhou, Haochao Ying, Recurrent neural network based collaborative filtering for QoS prediction in IoV, IEEE Trans. Intell. Transp. Syst. (2021) http://dx.doi.org/10.1109/TITS.2021.3099346.

[38] Hongbing Wang, Zhengping Yang, Qi Yu, Online reliability prediction via long short term memory for service-oriented systems, in: IEEE International Conference on Web Services, 2017, pp. 81–88.

[39] Wei Xiong, Zhao Wu, Bing Li, Qiong Gu, A Learning approach to QoS prediction via multi-dimensional context, in: IEEE International Conference on Web Services, 2017, pp. 164–171.

[40] Ruibin Xiong, Jian Wang, Zhongqiao Li, Bing Li, Patrick CK Hung, Personalized LSTM based matrix factorization for online QoS prediction, in: IEEE International Conference on Web Services, 2018, pp. 34–41.

[41] Zibin Zheng, Yilei Zhang, Michael R. Lyu, Distributed QoS evaluation for real-world web services, in: IEEE International Conference on Web Services, 2010, pp. 83–90.

[42] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, Yoshua Bengio, On the properties of neural machine translation: Encoder-decoder approaches, 2014, arXiv preprint arXiv:1409.1259.

[43] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.