Front. Eng. Manag. 2025, 12(1): 117–127
https://doi.org/10.1007/s42524-025-4038-x

RESEARCH ARTICLE

Fuzan CHEN, Jing YANG, Haiyang FENG, Harris WU, Minqiang LI

A two-phase learning approach integrated with multi-source features for cloud service QoS prediction

Abstract Quality of Service (QoS) is a key factor for users when choosing cloud services. However, QoS values are often unavailable due to insufficient user evaluations or provider data. To address this, we propose a new QoS prediction method, Multi-source Feature Two-phase Learning (MFTL). MFTL incorporates multiple sources of features influencing QoS and uses a two-phase learning framework to make effective use of these features. In the first phase, coarse-grained learning is performed using a neighborhood-integrated matrix factorization model, along with a strategy for selecting high-quality neighbors for target users. In the second phase, reinforcement learning through a deep neural network is used to capture interactions between users and services. We conducted several experiments using the WS-Dream data set to assess MFTL's performance in predicting response time QoS. The results show that MFTL outperforms many leading QoS prediction methods.

Keywords cloud service, QoS prediction, matrix factorization, deep neural network

1 Introduction

With the emerging popularity of service-oriented computing, a variety of cloud services have gained traction and usage all over the Internet (Xia et al., 2022; Lin and Kumar, 2017). With an increase in service-oriented computing, a competitive space emerges in which the number of functionally similar cloud services increases exponentially. Consequently, Quality of Service (QoS) metrics—for example, cost, response time, throughput, and others—become vital in distinguishing between those functionally equivalent services. Many existing studies, including service selection, service recommendation, and service composition studies (Ahn, 2008; Chen et al., 2014; Jain et al., 2020), operate on the premise that users know all relevant QoS metrics for the services they are considering. Such an assumption is, in many cases, untenable due to a lack of user evaluation on the service side or reports collated on the service of a provider. With the technological intricacies of cloud services, significant downtimes or interruptions may arise due to software malfunctions, network failures, and other faulty systems. Furthermore, the QoS indicators provided by cloud service providers may not always coincide with those that serve users as actual standards. Additionally, providers may lack a combination of the will or the ability to monitor QoS information for every service accurately due to costs or technological restraints. Notwithstanding these issues, QoS data remains critically vital in several scenarios, such as resource allocation, service selection, and service composition. Since missing QoS value computation poses a serious threat in QoS-centric applications, this presents a serious challenge for handling certain applications in the above contexts (Zheng et al., 2011; Xia et al., 2022).

Collaborative filtering (CF) is among the most commonly used techniques to predict the unknown QoS values by leveraging crowd wisdom based on historical QoS records (Zheng et al., 2011). In general, CF methods can be further classified into neighbors based CF and model-based CF. In short, neighborhood-based CF utilizes various similarity measures to select k users or services that are sufficiently similar, whose values will then be employed to estimate the unknown QoS. Most similarity measures rely on the QoS values associated with common invocation records (Breese et al., 2013; Wu et al., 2013), and they tend to perform poorly when faced

Received Jul. 26, 2023; revised Jun. 24, 2024; accepted Sep. 18, 2024

Fuzan CHEN, Jing YANG, Haiyang FENG, Minqiang LI
College of Management and Economics, Tianjin University, Tianjin 300072, China

Harris WU (✉)
Department of Information Technology and Decision Sciences, Old Dominion University, Norfolk, VA 23529, USA
E-mail: hwu@odu.edu

This research was supported by the National Natural Science Foundation of China (Grants Nos. 72394373, 72231004, 72022012, and 71971153).

with sparse data sets and cold-start situations. Model-based CF approaches, such as matrix factorization (MF) methods, have been developed to overcome these drawbacks (He et al., 2014). The MF algorithm transforms an original QoS matrix into the product of two matrices, mapping potential features of users and services into a low-dimensional space. The traditional collaborative filtering methods do not exploit nonlinearity in the relations between users and services thus do not possess this ability to explore complex and nonlinear dependencies. Using a deep neural network is more recent, where complex interactions between users and services were uncovered through the integration of deep learning measurement into QoS prediction (Zou et al., 2020; Wu et al., 2021; Sahu et al., 2021; Choi et al., 2021; Xia et al., 2022). These findings suggest that ANN or deep learning generally improved predictive accuracy compared to conventional CF methods. Nevertheless, much more needs to be done to refine effective and efficient QoS prediction methodologies.

A hybrid two-phase learning framework that combines collaborative filtering with deep learning-based techniques is proposed in order to tackle the matrix completion problem of unknown QoS values in large and sparse data sets. This approach takes into account several features underlying the QoS experiences of users, such as user-side and services-side usage patterns extracted from historical invocation records; categorical or local information acquired from the nearby neighbors; linear interactions modeled via a neighborhood integrated matrix factorization model; and highly complex interactions arising from underlying features identified via deep neural networks. The hybrid two-phase learning approach is thus going to help optimally leverage all these multifaceted features. The primary contributions of this study are as follows:

(1) User collaboration hence handle the problems of sparsity and cold-start in QoS prediction. A novel selection technique based on historical invocation records and categorical information is proposed to identify high-quality neighbors for the target user.

(2) To systematically exploit various features, a two-phase learning framework integrating collaborative filtering and deep learning techniques are proposed so that the model can learn low-order and high-order features. The first involves the Neighbor Integrated Matrix Factorization (NIMF) model to capture linear interactions between users and services (i.e., low-order features). The second will use a deep neural network to understand and capture the underlying potential nonlinear interactions among the features (i.e., high-order features).

(3) To evaluate the proposed two-phase framework on the WS-Dream data set, we present rather extensive experiments, the results of which indicate that our framework outperforms all previously leading QoS prediction methods with respect to predictive accuracy.

The paper is arranged as follows: Section 2 gives a review of related work and existing methods for QoS prediction. Section 3 discusses the overall framework of the model in detail, experimental results are analyzed in Section 4, and finally, conclusions are drawn in Section 5.

2 Related work

CF is a widely adopted technique in recommendation systems and has gained traction for predicting missing QoS values of cloud services (Zheng et al., 2013). CF-based approaches first identify similar users and then predict unknown entries based on the QoS values of the target user's neighbors. These approaches are generally classified into three categories: neighborhood-based methods, model-based methods, and deep-learning hybrid methods.

The core premise of neighborhood-based CF is to identify k users or services with the highest similarity as neighbors, using various similarity measures to predict unknown QoS values (Breese et al., 2013). Ding et al. (2014) proposed an item-based approach for CF, while Zheng et al. (2011) and Silic et al. (2014) presented hybrid CF methods that combine traditional user-based approaches with service-based techniques. Generally, neighborhood-based CF algorithms demonstrate strong performance when sufficient QoS values are available; however, they struggle with sparsity issues. Additionally, these algorithms encounter significant cold-start challenges when predicting QoS values for new users or services due to difficulties in locating appropriate neighbors.

To address the issues of sparsity and cold-start problems, researchers have made significant advancements in model-based matrix factorization approaches (He et al., 2014), including singular value decomposition (SVD), principal component analysis (PCA), probabilistic matrix factorization (PMF), and nonnegative matrix factorization (NMF). To further incorporate local features from nearest neighbor information, Zheng et al. (2013) proposed a neighborhood-integrated matrix factorization method. This approach utilizes user collaboration by integrating the neighbors of users into a matrix factorization model through regularization terms. While these CF methods successfully capture linear aspects of users and services, they struggle to account for more complex nonlinear interactions.

Deep learning has emerged as a promising method for modeling complex correlations among multiple elements, with recent studies exploring its application for QoS prediction challenges. Wu et al. (2021) proposed a universal deep neural model that offers a robust framework for integrating various contextual features, thereby enhancing multi-attribute QoS prediction and achieving

superior accuracy as measured by mean absolute error. In their deep learning model, Zhang et al. (2024) designed a feature mapping and inference network to highlight the nuanced relationships between users and services. For QoS prediction in edge computing environments, Yin et al. (2020) proposed a novel neural network model that combines denoising auto-encoding and fuzzy clustering techniques. Choi et al. (2021) employed a generative adversarial imputation network model to predict QoS based on a reconstructed user-service invocation matrix. Additionally, Li et al. (2024) developed a dynamic QoS prediction model for online service systems leveraging a graph neural network to learn representations of current network states.

Moreover, recent studies have sought to integrate traditional CF methods with neural network techniques. These studies use user and service vectors derived from the CF-based decomposition of the user-service invocation matrix as inputs to neural networks (Zou et al., 2020; Sahu et al., 2021; Xia et al., 2022). Findings indicate that this hybrid approach enhances prediction accuracy. Nevertheless, there remains considerable scope for developing efficient and effective hybrid methods to optimize QoS prediction performance, particularly in the context of sparse and large-scale QoS data sets. Significant challenges persist, including addressing the cold-start problem in QoS data, identifying high-quality neighbors, and leveraging user or service collaboration in sparse and extensive data sets, as well as employing neural networks to capture nonlinear and deep interactions among underlying features. These challenges form the core focus of our study.

In conclusion, existing studies have demonstrated that neighbors can provide information similar to that of target users and services. The matrix factorization model can evaluate potential features from a global perspective, while deep neural networks can learn the complex relationships between users and services. Although these techniques are effective when applied individually, few studies have integrated them to enhance QoS prediction accuracy. To address this gap, we propose an integrated approach, termed Multi-source Feature Two-phase Learning (MFTL), that aims for personalized and precise QoS prediction of cloud services. This method leverages multi-source features that impact the user's QoS experience and develops a two-phase learning framework to effectively capture various types of information. The experimental results validate the efficacy of both the two-phase learning framework and the integration of multi-source features.

3 Multi-source QoS influence factors

The cloud service paradigm allows multiple users to share computing resources from a resource pool simultaneously. Consequently, in real-world scenarios, the QoS performance of cloud services for target users is complexly linked to various factors such as user usage patterns, provider policies for QoS guarantees, resource allocation strategies, deployment statuses, and invoking environments. To fully utilize this information and enhance prediction accuracy, this study considers several multi-source features that affect the target user's QoS experience. These features include user-side and service-side usage patterns derived from historical invocation records, categorical or local information sourced from nearby neighbors, linear interactions captured by a neighborhood-integrated matrix factorization model, and higher-order features obtained from these linear interactions. These multi-source features underpin the design of the MFTL.

3.1 Historical invocation records

Historical invocation records provide insights into both users' usage patterns and providers' policies for QoS guarantees. Therefore, the most crucial feature we consider is the complete set of historical invocation records, which will be structured as an invocation matrix in the subsequent section. We will apply a matrix factorization model to identify linear patterns among users and services.

3.2 Local or categorical information of the user

The sparse and cold-start issues pose significant challenges for QoS prediction. Leveraging insights from local neighbors is a crucial technique for addressing sparse QoS records. For instance, users access cloud services via the Internet, and those within the same geographical area are likely to have similar QoS experiences due to their shared network environments. Conversely, users located in different regions with markedly distinct economic conditions (e.g., the United States versus Africa) may encounter vastly different experiences when utilizing the same service. Therefore, for the target user, insights drawn from users within the same region provide considerably more relevant information than those from users in disparate locations. As such, we incorporate local or categorical user information, such as geographical regions, into our QoS prediction models.

3.3 Complex interactions of users and services

The cloud service framework allows numerous users to simultaneously share the same resource pool, meaning individual users may engage with various services concurrently. Additionally, factors such as network environment status, bandwidth limitations, and scheduling algorithms can alter a user's QoS experience. This scenario creates complex interactions between users and services. Our proposed MFTL approach aims to capture

these complex interactions, including both low-order features that represent linear interactions and higher-order features derived from those linear interactions.

Initially, we identify linear interactions between users and services through a neighborhood-integrated matrix factorization approach. This process decomposes the extensive invocation QoS matrix into smaller user and service matrices through matrix factorization. The resulting vectors effectively represent the features of the corresponding users and services. By mapping these vector features into a low-dimensional space, we can derive linear characteristics that enable the prediction of unknown QoS values.

However, the neighborhood-integrated matrix factorization model is constrained to capturing simple, linear interactions between users and services. To address this limitation, we harness the capabilities of deep neural networks to capture nonlinear relationships through multilevel processing in the second phase of the MFTL approach. Additionally, to mitigate potential overfitting from linear interactions and to reduce the model's parameters, we incorporate higher-order features derived from linear interactions during the neural network phase.

4 Two-phase QoS prediction approach

Considering historical invocation records with m users and n services, let $U = \{u_1, u_2, \ldots, u_m\}$ indicates the set of service users and $u_i (1 \leqslant i \leqslant m)$ is a user, $S = \{s_1, s_2, \ldots, s_n\}$ indicates the set of services, where $s_j (1 \leqslant j \leqslant n)$ represents a service. The matrix R is made up of all users to invoke services, in which r_{ij} is the QoS value when user i invocates service j, then a service invocation record can be denoted as a tuple (u_i, s_j, r_{ij}). If r_{ij} is null, it indicates that user i has not called service j, and its value will be predicted. The goal of this study is to forecast the absent values in the invocation matrix R.

4.1 The framework of MFTL

MFTL employs a two-phase learning approach that integrates various influential factors, such as historical invocation records, service categories, the influence of similar neighbors, and complex interaction features between services and users. The first phase of the proposed MFTL is neighborhood-integrated matrix factorization, abbreviated as NIMF. In the NIMF phase, K-nearest neighbors are selected based on similarity measures (see procedure S1-1), followed by a matrix factorization operation to derive potential feature vectors related to users and services (see procedure S1-2). The second phase consists of deep neural network-based reinforcement learning, utilizing the obtained feature vectors to estimate the missing values (see procedure S2). Figure 1 illustrates the process of MFTL.

(1) S1-1: Neighbor selection

As highlighted, neighbor selection is a critical component of neighbor-based QoS prediction, significantly influencing prediction accuracy (Wu et al., 2013; Zou et al., 2020). This study proposes a novel method for selecting a set of neighbors for the target user. Instead of relying solely on common invocation records, we analyze the probability distribution of all historical records to calculate user similarity, further incorporating users' locations to identify the K-nearest neighbors.

(2) S1-2: Neighborhood-integrated matrix factorization

Although matrix factorization effectively addresses global information, it often fails to capture the significant associations between target users and their neighbors. Based on the previous analysis, neighbors share similar service invocation experiences. By integrating neighbors' experiences into the matrix factorization model, we can enhance our understanding of the relationships between users and services, as represented by the corresponding potential feature vectors for users and services, respectively.

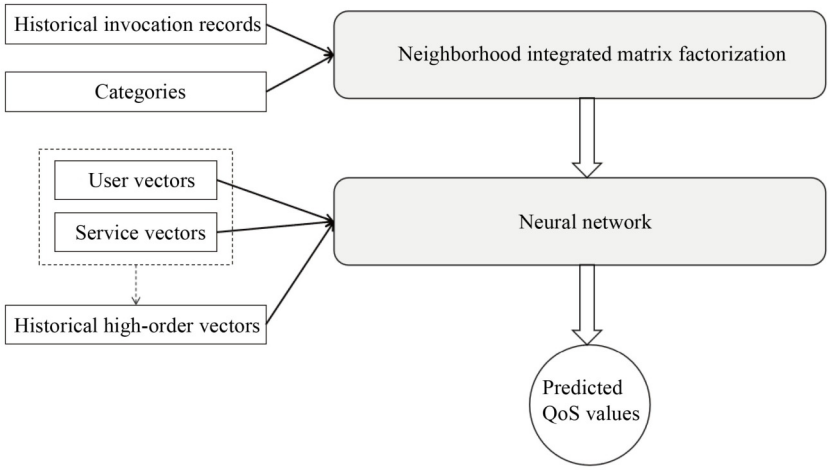

Fig. 1 The framework of MFTL.

(3) S2: Reinforcement QoS prediction based on deep neural network

Due to the limitations of the linear model's representational capabilities, it is essential to explore more nonlinear features to enhance prediction accuracy. Deep neural networks are capable of extracting complex interaction features through multi-level processing. Consequently, we employ this approach to capture the nonlinear and complex relationships between users and services, ultimately yielding more precise prediction results. A neural network is utilized to generate a final prediction by integrating both the low-order features obtained from matrix factorization and the high-order interaction features among them.

4.2 Neighbor selection

To identify the optimal k neighbors for a target user, we have developed a novel strategy to assess user similarity based on historical invocation records and categorical information.

4.2.1 Historical similarity

The similarity among users increases with the similarity of their records. However, traditional measures that rely on the QoS values of common invocations may falter when there are few co-invoked services. In this study, we utilize the Bhattacharyya coefficient to assess similarity based on all historical records. This method enables us to achieve a more accurate measure of similarity and effectively mitigates the issue of data sparsity.

PCC and vector space similarity (VSS) are among the most commonly used metrics for estimating user similarity (Zhou and Chellappa, 2006). Yet, these methods may not be applicable to sparse QoS matrices (Patra et al., 2015). To address these challenges, Patra et al. (2015) and Jain et al. (2020) propose using the Bhattacharyya coefficient to calculate similarity, maximizing the use of all invocation records provided by users and increasing the likelihood of identifying similar neighbors in sparse data sets. As indicated in Eq. (1), this study employs the Bhattacharyya coefficient to evaluate the comparability of two users based on all their QoS values.

$$\mathrm{BC}(u,v)=\sum_{i=1}^{I}\sqrt{P_u(i)P_v(i)}. \tag{1}$$

Herein I is the number of bins, $P_u(i)=\dfrac{\mathrm{Num}(Q_u,i-1,i)}{\mathrm{Num}(Q_u,0,I)}$, where $\mathrm{Num}(Q_u,i-1,i)$ denotes the number of QoS values of the service invoked by user u at the ith interval, and $\mathrm{Num}(Q_u,0,I)$ represents the total number of services invoked by the user u. And $\sum_{i=1}^{I}P_u(i)=\sum_{i=1}^{I}P_v(i)=1$. Note that the WS-Dream data set used in this study has a QoS value that varies from 0 to 20, therefore it can be divided into 20 intervals, that is, $I=20$.

4.2.2 Neighbor selection

Given that cloud services are delivered over the Internet, users invoking the same service may experience varying QoS values based on their network environments. Factors such as network distance and bandwidth are closely related to the geographic locations of both the target user and the target service. Nearby users typically share similar network infrastructure and routing protocols, which increases the likelihood of experiencing comparable QoS values when accessing the same services. Therefore, in our neighbor selection process, we also consider geographic location information to enhance the relevance of identified neighbors.

Specifically, an autonomous system is a small unit capable of determining which routing protocol to implement within the system and can be managed by one or more network operators. When users request services, those within the same autonomous system generally experience similar network conditions, with only minor differences in QoS. However, in practical scenarios, there may not be an adequate number of neighbors within the same autonomous system as the target user to effectively represent relevant features. Given the significant variations in network infrastructure across different countries, we incorporate country information to identify more relevant neighbors.

The process of neighbor selection for the target user, denoted as u, involves three steps: (1) Identify all users within the same autonomous system as the target user. If the total number of users is equal to or greater than a predefined threshold K, the most similar users, as determined by Eq. (1), will constitute the final neighbors; (2) If the number of users is less than K, all identified users will form the initial neighborhood. We will then supplement this neighborhood by selecting users from the same country, but belonging to different autonomous systems, following the same criteria as outlined in Step 1; (3) If the total number of neighbors remains below K, we will calculate the similarity between the target user and all other users, selecting the neighbors with the highest similarity until we achieve K neighbors.

4.3 Neighborhood-integrated matrix factorization

Matrix factorization is a widely-used model-based collaborative filtering algorithm. Its fundamental concept is to decompose the initial user-service QoS matrix R into two low-dimensional matrix products, U and S:

$$R\approx U^{T}S, \tag{2}$$

where $U\in R^{d\times m}$, $S\in R^{d\times n}(d<\min\{m,n\})$ representing the feature vectors of users and services, respectively. By mapping the features of users and services into low-dimensional space, we obtain potential linear features that

can be utilized for predicting unknown QoS values. To incorporate neighborhood information into the matrix factorization model, we define the objective function as follows:

$$\begin{aligned}\underset{U,S}{\text{Min}} f = &\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(r_{ij}-U_i^T S_j)^2 + \frac{\lambda_U}{2}\|U\|_F^2 \\ &+\frac{\lambda_S}{2}\|S\|_F^2 + \frac{\alpha}{2}\sum_{i=1}^{m}\sum_{g\in G(i)}\|U_i - U_g\|_F^2.\end{aligned} \tag{3}$$

Therein, if there is a QoS historical record of user i calling service j, the value of I_{ij} is 1; otherwise, the value is 0. r_{ij} is actual observed QoS values, λ_U and λ_S are the parameters that can control the importance of constraints. $\|\cdot\|_F$ denotes the Frobenius norm. $\alpha > 0$ is an adjustable parameter used to control the importance of the gap between the target user and neighbors. $G(i)$ indicate the neighbor set of target user i.

The gradient descent (GD), an iterative strategy, is employed to implement the matrix factorization process, as shown in Eqs. (4)–(7):

$$U_i' = U_i - \gamma_i \frac{\partial f}{\partial U_i}, \tag{4}$$

$$S_j' = S_j - \gamma_j \frac{\partial f}{\partial S_j}, \tag{5}$$

$$\frac{\partial f}{\partial U_i} = \sum_{j=1}^{n} I_{ij}(r_{ij} - U_i^T S_j)(-S_j) + \lambda_U U_i + \alpha\sum_{g\in G(i)}(U_i - U_g), \tag{6}$$

$$\frac{\partial f}{\partial S_j} = \sum_{j=1}^{m} I_{ij}(r_{ij} - U_i^T S_j)(-U_i) + \lambda_S S_j. \tag{7}$$

The final potential feature vectors are obtained through the iterative optimization of these two functions, as outlined in Eqs. (6) and (7).

4.4 QoS prediction based on neural network

The second phase of MFTL involves a fully connected deep neural network, as illustrated in Fig. 2. This network includes an input layer, L hidden layers, and an output layer corresponding to the predicted QoS values. Each layer's operational process will be elaborated upon in the subsequent subsections.

4.4.1 Input layer

In selecting input features, our objective is to enhance the modeling capability of the neural network while simultaneously minimizing model parameters and mitigating risks of overfitting. Users' experiential QoS values for services are intrinsically linked to the characteristics of users, services, and their interactions. Therefore, user potential features U_i and service potential features S_j derived from the preceding NIMF phase serve as foundational input features. Additionally, we recognize that cross features generated by the interactions between users and services can aid the subsequent nonlinear layers in interpreting complex relationships. Consequently, we also incorporate high-order features of the coarse-grained U_i and S_j, as represented in Eq. (8):

$$\mathrm{H}_{ij} = U_i \odot S_j. \tag{8}$$

Therein $\odot$ is the element-wise product of two vectors, $(U_i \odot S_j)_k = U_{ik} S_{jk} (1 \leqslant k \leqslant d)$.

In this reinforcement learning phase, we consider both the low-order relationships of potential feature vectors obtained in the MIMF phase and the high-order features that capture complex interactions between users and services. The input to the neural network is a synthesis of these features, as shown in Eq. (9):

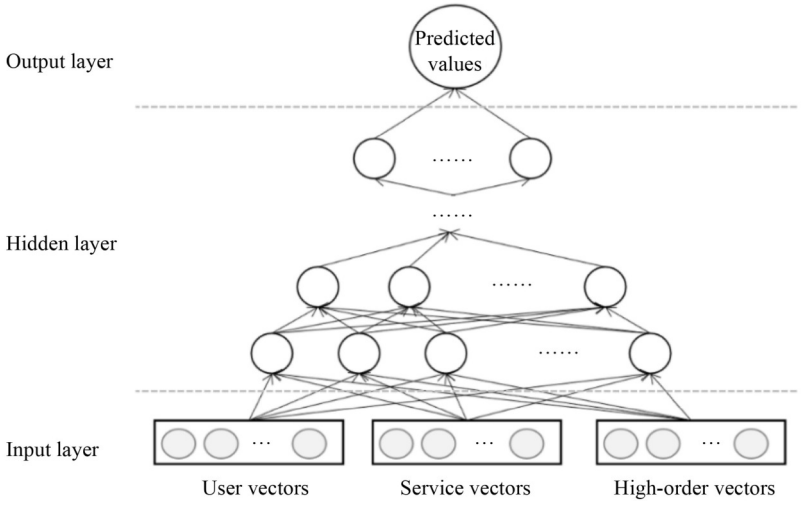

Fig. 2 Neural network framework.

$$\mathrm{E}_{ij} = \{U_i, S_j, H_{ij}\}. \tag{9}$$

4.4.2 Hidden layer

Below stack of fully connected layers is used to learn the deep complex nonlinear relationship between features:

$$\begin{aligned} Z_1 &= \sigma_1(W_1 E + b_1), \\ Z_2 &= \sigma_2(W_2 \mathrm{E} + b_2), \\ &\cdots \\ Z_L &= \sigma_L(W_L E + b_L). \end{aligned} \tag{10}$$

Therein, L represents the number of hidden layers, W_i, b_i, σ_i represent the weight, bias term, and activation function of the i-th layer respectively. Commonly used activation functions include sigmoid, tanh, and rectified linear unit (ReLU). The sigmoid function constrains values between (0,1), which may limit model performance, with tanh partially addressing this limitation. We have chosen ReLU as the activation function due to its suitability for sparse data and its lower propensity for causing model overfitting. The neural network structure is designed in a tower format, with the bottom layer being the widest and the number of neurons in each successive layer progressively decreasing.

4.4.3 Output layer

The output layer delivers the final prediction result. There are two predominant loss functions employed when utilizing a neural network for regression prediction: minimum absolute error ($L1$) and minimum square error ($L2$). The $L1$ loss function is more robust to outliers and aims to minimize the absolute difference between predicted and actual values. Conversely, the $L2$ loss function focuses on minimizing the squared variance, imposing greater penalties on significantly deviating predicted values. For this study, we have selected $L2$ as the loss function, as illustrated in Eq. (11):

$$L2 = \sum_{x \in \chi} (\hat{y}(x) - y(x))^2. \tag{11}$$

Therein, χ represents the training data, $\hat{y}(x)$ is the predicted target value of x and $y(x)$ is the real value.

5 Experiments

In this section, we present several experiments conducted to validate the effectiveness of the proposed MFTL approach. These experiments were performed on Google Collaboratory, with the components of MFTL implemented in Python.

5.1 Data set and evaluation metrics

To evaluate the performance of the proposed MFTL approach, we conducted experiments on WS-Dream, a well-known real-world service invocation record compiled and maintained by Zheng et al. (2013). This data set includes 339 users, 5825 services, and a total of 1,974,675 historical QoS records, which include response time and throughput. For the following experiments, we focus specifically on response time records. The data set also contains additional information regarding the country, autonomous system, latitude, and longitude of both users and services. The statistics of this data set are presented in Table 1.

As for the evaluation metrics, we choose Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), which are widely used in QoS prediction. They are defined as Eqs. (12) and (13).

$$\mathrm{MAE} = \frac{\sum_{ij} \left| r_{ij} - \hat{r}_{ij} \right|}{N}, \tag{12}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{ij} (r_{ij} - \hat{r}_{ij})^2}{N}}. \tag{13}$$

Therein r_{ij} represents the real QoS value and $\hat{r}_{ij}$ is the predicted value of service j invoked by user i,N is the number of QoS values in the testing data set. MAE calculates the mean value of the absolute deviation and indicates the accuracy of the overall prediction. In contrast, RMSE is more sensitive to outliers, making it more effective in scenarios where significant deviations are not expected. Lower values of both MAE and RMSE indicate more accurate predictions.

5.2 Impact of parameters

This study examines how prediction accuracy is influenced by two key parameters: 1) the number of neighbors of the target user K; and 2) the dimensionality of the potential characteristics of users and services. We conduct

Table 1 Statistics of the QoS data set

Statistics	Values
Users	339
Services	5825
Service invocations	1,974,675
Users' autonomous systems	137
Users' regions	31
Range of response-time	0–20 s

comparative experiments to assess their influence in this section.

5.2.1 The number of neighbors of target users

To investigate the impact of the number of neighbors, we set its values K as 0, 5, 10, 15, and 20. The corresponding MAE and RMSE values across different matrix densities are illustrated in Fig. 3. As shown in the figure, with a small K, the features of the neighbors are not fully utilized, resulting in higher MAE and RMSE values. As K increases, the prediction accuracy improves. However, after exceeding a certain threshold, the error begins to rise again. This trend occurs because an excessive number of neighbors can introduce higher noise levels, ultimately leading to decreased prediction performance. The findings indicate that an optimal neighborhood size positively affects prediction outcomes. The experimental results demonstrate that the best performance is achieved when K is set to 5 in most matrix densities. Consequently, $K = 5$ is applied in subsequent experiments.

5.2.2 Dimension of potential features of users and services

In the matrix factorization model, the dimension significantly influences the number of potential factors utilized to represent the characteristics of users or services. This experiment will explore the impact of different dimensions by setting their values to 1, 2, 3, 4, and 5. As illustrated in Fig. 4, both MAE and RMSE initially decrease with increasing dimension size before stabilizing beyond a certain threshold (specifically threshold 3 in this study). Similar to the neighborhood size, having an excessive number of dimensions can diminish prediction accuracy due to overfitting. Consequently, the dimension size for potential user and service features is experimentally set to 3.

5.3 Component analysis

As noted, one of the key contributions of the proposed MFTL (Matrix Factorization Transfer Learning) model is its capacity to leverage a wide array of features, such as the influence of neighbors, invocation records, categories, as well as low-order and high-order interactions among users, services, and their relationships. These features are incorporated within a two-phase framework. This section presents two experiments designed to assess the effectiveness of each pivotal component within MFTL: (1) the neighborhood-integrated matrix factorization model, which utilizes neighbor influence and introduces a novel strategy for neighbor selection based on a variety of characteristics, and (2) a reinforcement learning strategy that employs a neural network to capture both low-order linear interactions and high-order complex interactions.

5.3.1 Impact of strategy for neighbor selection

To evaluate the effectiveness of the neighbor selection

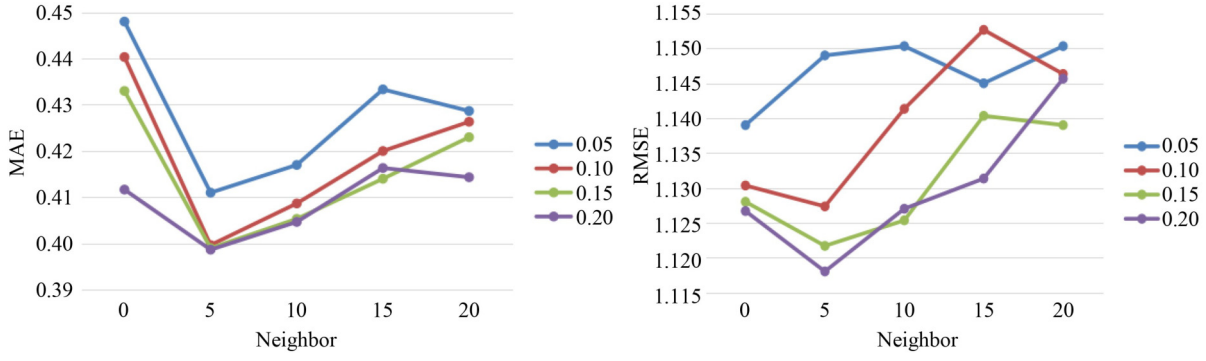

Fig. 3 Impact of the number of neighbors.

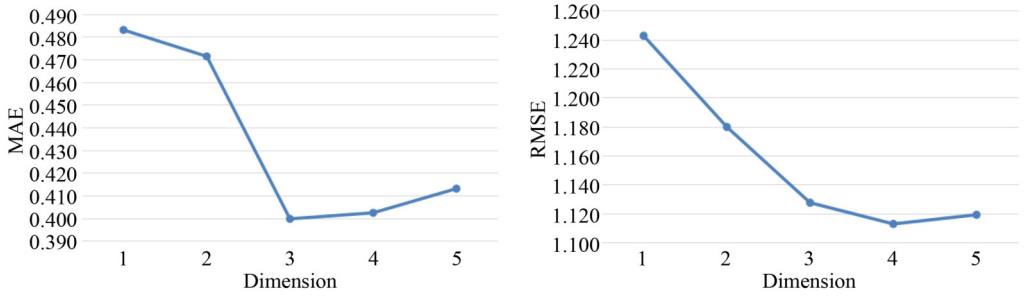

Fig. 4 Impact of dimension.

strategy, we compared the MFTL model using only the first phase (denoted as MFTL-1) with other neighborhood-integrated matrix factorization approaches, including NIMF (Zheng et al., 2013) and NAMF (Tang et al., 2016). The experimental results, reflected in terms of MAE and RMSE, are summarized in Table 2.

As shown in Table 2, MFTL-1 consistently exhibits lower MAE and RMSE values compared to other neighborhood-integrated matrix factorization approaches. Notably, in sparse data sets (data sets characterized by low density), the prediction performance of MFTL-1 shows a significant improvement compared to competing methods. The enhanced performance of MFTL-1 can be attributed to its innovative neighbor selection strategy. When there are sufficient common services that users invoke simultaneously, this strategy effectively identifies very similar neighbors. In cases where the data set is sparse, it continues to perform well by utilizing not only invocation records but also the location information of users. This novel selection strategy adeptly addresses the challenges associated with data sparsity.

5.3.2 Impact of reinforcement learning strategy

As previously mentioned, traditional CF methods primarily focus on learning linear interactions between users and services. To capture more complex nonlinear relationships, the MFTL approach employs reinforcement learning. This integration utilizes potential feature vectors obtained from an initial phase, allowing the model to consider both linear interactions (low-order features) and nonlinear interactions (high-order features), represented by an element-wise product of two vectors. In our ablation studies, we compare MFTL with MFTL-1, which executes only the first stage of matrix factorization, and MFTL-2, which focuses solely on deep neural network training. Additionally, we examine MFTL-L, a two-phase approach that utilizes only user and service feature vectors. To assess the impact of the reinforcement learning strategy in MFTL relative to the ablated MFTL methods and other CF-NN hybrid approaches, we conducted a series of experiments across various matrix densities, summarizing the prediction performance in Table 3.

As presented in Table 3, MFTL demonstrates superior MAE and RMSE across all matrix configurations when compared to MFTL-1, which relies solely on matrix factorization, and MFTL-2, which employs only deep neural network learning. Furthermore, the prediction performance of MFTL exceeds that of MFTL-L, which utilizes only the user and service potential vectors. The reductions in both MAE and RMSE suggest that the dual-phase learning strategies employed by MFTL significantly enhances predictive performance. Moreover, compared to NDMF and JDNMFL, which also combine matrix factorization and neural network models but do not account for neighbor influences or nonlinear interactions between users and services, MFTL performs better in terms of MAE and RMSE metrics. Notably, the improvement in RMSE is considerable, indicating a substantial reduction in the probability of substantial discrepancies between predicted and actual values. Additionally, the predictive performance of the proposed MFTL model is less affected by data sparsity, demonstrating effectiveness even with highly sparse data sets. This success can be attributed to the incorporation of a wide variety of features and the implementation of effective strategies to leverage these features.

Table 2 Performance comparisons of MFTL-1 and other similar approaches

Method	5%		10%		15%		20%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
NIMF	0.554	1.479	0.480	1.295	0.443	1.208	0.421	1.159
NAMF	0.538	1.385	0.485	1.259	0.453	1.207	0.435	1.144
MFTL-1	**0.443**	**1.171**	**0.440**	**1.156**	**0.440**	**1.155**	**0.431**	**1.147**

Table 3 Performance comparisons of CF-NN hybrid methods

Method	5%		10%		15%		20%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
NDMF	0.488	1.350	0.430	1.235	0.385	1.157	0.367	1.129
JDNMFL	**0.375**	1.378	**0.349**	1.353	–	–	–	–
MFTL-1	0.443	1.171	0.440	1.156	0.440	1.155	0.431	1.147
MFTL-2	0.478	1.401	0.469	1.419	0.459	1.370	0.457	1.362
MFTL-L	0.448	1.151	0.418	1.152	0.401	1.139	0.411	1.117
MFTL	0.410	**1.149**	0.400	**1.127**	**0.399**	**1.122**	**0.399**	**1.118**

5.4 Effectiveness analysis

To evaluate the prediction performance of MFTL across different matrix densities, we compared it with nine state-of-the-art QoS prediction methods. This included two CF methods: NIMF (Zheng et al., 2013), and NAMF (Tang et al., 2016); two neural network (NN) methods: DNM (Wu et al., 2021) and GAIN-QoS (Choi et al., 2021); and two CF-NN hybrid methods: NDMF (Zou et al., 2020) and JDNMFL (Xia et al., 2022). The corresponding MAE and RMSE results are presented in Table 4.

As illustrated in Table 4, with the increase in matrix density, the QoS prediction can utilize more data, resulting in a decrease in MAE and RMSE across all methods. However, in scenarios characterized by very sparse data, such as with matrix densities of 5% and 10%, MFTL outperforms typical state-of-the-art methods, demonstrating significant improvements in both MAE and RMSE. In cases of relative sparsity, specifically at matrix densities of 15% and 20%, while there is no notable change in MAE, there is an observable enhancement in RMSE, suggesting a reduced risk of substantial prediction deviation. Furthermore, it is evident that the accuracy of MFTL's predictions remains relatively stable as matrix density increases, even when the QoS matrix is extremely sparse, indicating that the proposed MFTL is less vulnerable to the challenges posed by data sparsity.

In contrast to traditional CF and NN methods, the proposed MFTL incorporates reinforcement learning following a CF model to further investigate the complex nonlinear interactions between users and services. This hybrid approach yields significant performance improvements in terms of both MAE and RMSE, particularly at lower matrix densities. Additionally, compared to similar CF-NN hybrid methods like NDMF and JDNMFL, MFTL places a greater emphasis on multi-source features and user collaboration while training a deep neural network more effectively. The superior performance of the proposed MFTL can be attributed to two key factors: (1) the neighbor selection method utilizes not only historical invocations but also various categorical information about users, and (2) the reinforcement learning performed by the deep neural network considers complex interactions between users and services rather than merely linear relationships.

6 Conclusion and future work

This work presents an innovative, MFTL algorithm for QoS prediction in cloud services. MFTL integrates feature sources, including historical invocation records, user categories, geographical locations, distractive influences, linear, and nonlinear interactions between users and services. To merge such diverse features, a two-phase learning framework has been developed. The first phase carries out coarse-grained learning employing a neighborhood-integrated matrix factorization model through which the linear potential vector is derived. This process is followed by the second phase, which involves reinforcement learning using a deep neural network that aids in investigating further the complicated interactions between users and services. Several experiments using the WS-Dream data set were performed to evaluate the performance of MFTL. Experimental results indicate that our proposed MFTL outperforms the present state-of-the-art methods, with significantly reduced Mean Absolute Error and Root Mean Square Error. Several sources of features together are expected to turn a whole quality of incoming cloud service QoS predictably good; such a prediction is crucial for service selection or composition. While a variety of features are taken into consideration here, the time series data are yet another interesting point for future research. Another interesting opportunity for improving the accuracy of this field would be to construct deeper layers of neural network models.

Competing Interests The authors declare that they have no competing interests.

Table 4 Prediction performance comparisons on data sets with different densities

Type	Method	5%		10%		15%		20%	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CF	NIMF	0.554	1.479	0.480	1.295	0.443	1.208	0.421	1.159
	NAMF	0.538	1.385	0.485	1.259	0.453	1.207	0.435	1.144
NN	DNM	0.415	1.427	0.363	1.357	–	–	–	–
	GAIN-QoS	0.509	1.488	0.484	1.447	–	–	0.451	1.426
CF-NN hybrid	NDMF	0.488	1.350	0.430	1.235	0.385	1.157	0.367	1.129
	JDNMFL	**0.375**	1.378	**0.349**	1.353	–	–	–	–
	MFTL	0.410	**1.149**	0.400	**1.127**	**0.399**	**1.122**	**0.399**	**1.118**

References

Ahn H J (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Information Sciences, 178(1): 37–51

Breese J S, Heckerman D, Kadie C (2013). Empirical analysis of predictive algorithms for collaborative filtering. Uncertainty in Artificial Intelligence, 98(7): 43–52

Chen X, Zheng Z, Yu Q, Lyu M R (2014). Web service recommendation via exploiting location and QoS information. IEEE Transactions on Parallel and Distributed Systems, 25(7): 1913–1924

Choi J, Lee J, Ryu D, Kim S, Baik J (2021). GAIN-QoS: A novel QoS prediction model for edge computing. Journal of Web Engineering, 21(01): 27–52

Ding S, Yang S, Zhang Y, Liang C, Xia C (2014). Combining QoS prediction and customer satisfaction estimation to solve cloud service trustworthiness evaluation problems. Knowledge-Based Systems, 56: 216–225

He P, Zhu J, Zheng Z, Xu J, Lyu M R (2014). Location-based hierarchical matrix factorization for Web service recommendation. In: 2014 IEEE International Conference on Web Services. Anchorage, AK, USA

Jain A, Nagar S, Singh P K, Dhar J (2020). EMUCF: Enhanced multistage user-based collaborative filtering through non-linear similarity for recommendation systems. Expert Systems with Applications, 161: 113724

Li J, Wu H, He Q, Zhao Y, Wang X (2024). Dynamic QoS prediction with intelligent route estimation via inverse reinforcement learning. IEEE Transactions on Services Computing, 17(2): 509–523

Lin J, Kumar U (2017). IN2CLOUD: A novel concept for collaborative management of big railway data. Frontiers of Engineering Management, 4(4): 428–436

Patra B K, Launonen R, Ollikainen V, Nandi S (2015). A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. Knowledge-Based Systems, 82: 163–177

Sahu P, Raghavan S, Chandrasekaran K (2021). Ensemble deep neural network based quality of service prediction for cloud service recommendation. Neurocomputing, 465: 476–489

Silic M, Delac G, Krka I, Srbljic S (2014). Scalable and accurate prediction of availability of atomic web services. IEEE Transactions on Services Computing, 7(2): 252–264

Tang M, Zheng Z, Kang G, Liu J, Yang Y, and Zhang T (2016). Collaborative Web service quality prediction via exploiting matrix factorization and network map. IEEE Transactions on Network and Service Management, 13(1): 126–137

Wu H, Zhang Z, Luo J, Yue K, Hsu C (2021). Multiple attributes QoS prediction via deep neural model with contexts. IEEE Transactions on Services Computing, 14(4): 1084–1096

Wu J, Chen L, Feng Y, Zheng Z, Zhou M C, Wu Z (2013). Predicting quality of service for selection by neighborhood-based collaborative filtering. IEEE Transactions on Systems, Man, and Cybernetics. Systems, 43(2): 428–439

Xia Y, Ding D, Chang Z, Li F (2022). Joint deep networks based multi-source feature learning for QoS prediction. IEEE Transactions on Services Computing, 15(4): 2314–2327

Yin Y, Cao Z, Xu Y, Gao H, Li R, Mai Z (2020). QoS prediction for service recommendation with features learning in mobile edge computing environment. IEEE Transactions on Cognitive Communications and Networking, 6(4): 1136–1145

Zhang P, Ren J, Huang W, Chen Y, Zhao Q, Zhu H (2024). A deep-learning model for service QoS prediction based on feature mapping and inference. IEEE Transactions on Services Computing, 17(4): 1311–1325

Zheng Z, Ma H, Lyu M R, King I (2011). QoS-aware web service recommendation by collaborative filtering. IEEE Transactions on Services Computing, 4(2): 140–152

Zheng Z, Ma H, Lyu M R, King I (2013). Collaborative web service QoS prediction via neighborhood integrated matrix factorization. IEEE Transactions on Services Computing, 6(3): 289–299

Zhou S K, & Chellappa R (2006). From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel hilbert space. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(6): 917–929

Zou G, Chen J, He Q, Li K C, Zhang B, Gan Y (2020). NDMF: neighborhood-integrated deep matrix factorization for service QoS prediction. IEEE Transactions on Network and Service Management, 17(4): 2717–2730