Research Article

Qinying Li*, Wei Lu, Fangli Li, Taotao Wang, and Hao Wang

QCI-WSC: Estimation and prediction of QoS confidence interval for web service composition based on Bootstrap

https://doi.org/10.1515/comp-2025-0028
received March 27, 2023; accepted February 21, 2025

Abstract: In web service composition, the Quality of Service (QoS) prediction applications based on the statistical point estimation method in accuracy consist of many challenges. Aiming at allowing users to select the web service composition based on their requirements, this study proposed a method based on Bootstrap to estimate and predict the QoS confidence interval for web service composition (QCI-WSC). The QCI-WSC first indicates the structure of the web service composition and simplifies the structure model. Apart from that, the QoS estimation interval can be calculated by the historical QoS data, which are invoked by users. Meanwhile, the user similarity is calculated, and the QoS of web service invoked by the similar users is used to predict QCI-WSC. Finally, the results of user-invoked web service composition QoS are verified by the average interval coverage rate, compared to the actual QoS values and prediction values of the other methods, such as adaptive QoS prediction method based on collaborative filtering (QACF) and QoS-Aware web service recommendation (WSRec). Additionally, in this work, dataset1 in WSDream is adopted to estimate and predict the QCI-WSC. Experiments show that the QoS confidence interval estimation results conform to the exponential distribution, and the validity of the QCI-WSC is proved. Furthermore, the average interval probability of the prediction algorithm was more than 75%. The QCI-WSC can accurately cover the actual QoS values of the web service composition and most of the accurate QoS values predicted by QACF and WSRec. It effectively improves the selectivity of service, which provides web service composition featuring better quality for users.

Keywords: web service composition, confidence interval, QoS estimation, QoS prediction, Bootstrap

*** Corresponding author: Qinying Li,** Information Engineering College, Jiangxi University of Technology, Nanchang, 330098, P. R. China, e-mail: 18270839061@163.com
Wei Lu, Fangli Li, Taotao Wang: Information Engineering College, Jiangxi University of Technology, Nanchang, 330098, P. R. China
Hao Wang: Software School, Jiangxi Normal University, Nanchang, 330098, P. R. China

1 Introduction

Web service [1,2] improves interaction and computing efficiency by establishing a distributed platform that can be operated interactively as a way to achieve cross-platform communication between programs. Based on the development of web service technology, a single web service can no longer meet user needs well, and how to combine existing web services to connect and interact in a friendly and collaborative way has received widespread attention. As users are affected by external network congestion, server load, and geographical changes in the process of selecting service [3,4] for web service composition, Quality of Service (QoS) [5,6] is characterized by random dynamics and nonlinearity.

QoS is gradually playing an important role in the process of web service composition. In order to ensure that users can choose the web service composition that meets their needs, QoS prediction is a crucial basis for web service selection and service recommendation. Recently, QoS prediction is a research area that has received extensive attention in the field of web services. The main methods are workflow-based prediction methods [7–9], collaborative filtering (CF)-based prediction methods [10–13], and model-based prediction methods [14–16].

Although existing research has made significant progress in predicting the QoS of web service composition, achieving the use of statistical point estimation methods to accurately predict the QoS of web service composition, the QoS change mode of the web service composition has not been considered. Therefore, the prediction results lack credibility. Therefore, this study proposed an estimation and prediction method of the QoS confidence interval for web service composition (QCI-WSC) based on Bootstrap technology to quantify the uncertainty of QoS prediction values. The contributions of the proposed method are threefold:

(1) Use the list structure of list processor (LISP) language [17] to build a web service composition model to effectively express the complexity of the composition structure. Other than that, the QoS attribute value calculation formulas for non-nested combination structures and nested combination structures are put forward to improve QoS calculation efficiency.
(2) Adopt non-parametric statistical Bootstrap technology to estimate and predict the QCI-WSC with reference to the historical QoS data of a single web service invoked by the user, so that the results meet the dynamic characteristics of the QoS attribute value. Furthermore, user service recommendations and accuracy and credibility of selection are ensured.
(3) Based on the WSDream dataset from the study by Zheng et al. [18] implemented to conduct related experiments, the average interval probability is used to derive the reliability of the prediction results. Besides, it is compared with the classic methods adaptive QoS prediction method based on collaborative filtering (QACF) and QoS-Aware web service recommendation (WSRec) to verify the coverage of the confidence interval and prediction value to ensure the prediction range accuracy.

The remainder of this study is organized as follows: Section 2 introduces the literature review of QoS prediction. Section 3 describes the web service composition structure model based on LISP language and the calculation method of QoS, and presents the estimation and prediction process of the QCI-WSC. Section 4 depicts the experimental design and evaluation of the method, and analyzes the experimental results. In Section 5, the conclusion and future recommendations of the research are given.

2 Related work

The related works of QoS prediction for web service composition have attracted widespread attention. Typically, the workflow-based QoS prediction method is proposed due to the similarity between the workflow and web service composition process. Wu and Yang [19] proposed a QoS prediction method of web service composition with the transaction mechanism, which considers normal and exception handling processes during the execution of web service composition. Liu et al. [20] presented a QoS dynamic prediction approach based on case-based reasoning to improve the reliability of web service composition by predicting the QoS value of candidate service and selecting the best web service node. Wu et al. [21] took a general context-sensitive matrix-factorization approach to enhancing the prediction accuracy of QoS by fusing implicit and explicit factors in QoS data. Although the workflow-based prediction approach is conducive to simplifying the process of web service composition, it mainly focuses on the contextual association of workflows. Therefore, it cannot be applied in a prediction environment as the time factor.

Furthermore, the CF [22] method is the current QoS prediction method with the highest prediction accuracy. CF uses the historical QoS data of similar users as the basis for predicting the QoS values of web service composition invoked by the current users. Zhang et al. [23] designed a QoS prediction framework, which utilizes the historical experiences generated by users invoking web services to construct feature models and provide personalized QoS predictions for users. Chen et al. [24] considered the bias between the user and web service, thus establishing the matrix decomposition model for selecting geographical neighbors with the hierarchical domain clustering algorithm and predicting QoS value by geographic neighbor information. Due to the fluctuation of QoS prediction results caused by environmental factors, Thakker et al. [25] constructed a QoS prediction method based on hybrid CF for movie recommendation, and considered the weight of movie recommendation factors to reduce the accuracy error of prediction results caused by label fluctuations. The CF algorithm solves the problem from a global perspective and can fully consider the relationship between users and services. However, the prediction results of this method are susceptible to the sparsity and cold start of historical QoS data, which leads to prediction results that do not better meet user needs.

To better predict QoS, the model-based QoS prediction method is proposed. The method designs a corresponding model to predict unknown QoS values by learning and training user historical QoS data. Luo et al. [26] introduced a fuzzy neural network QoS prediction method based on adaptive dynamic programming to learn the parameters of fuzzy rules. Anithadevi and Sundarambal [27] adopted neuro-fuzzy logic and combined it with users' personalized needs to intelligently identify untrusted services and improve the accuracy of service classification. Xiong et al. [28], combined deep learning methods in the time-space-sensitive QoS prediction model to extract the hidden features of multi-dimensional factors for QoS value to enhance the accuracy of the prediction results. In the method based on the GCN proposed by Wang et al. [29], potential user preferences for services in user-item interaction information were considered, and abstract preferences were mapped to specific components, thus having the ability to mine potential preferences of users or services. Due to the nonlinear and dynamic nature of QoS, the model-based QoS prediction method has disadvantages such as slow extraction of features and difficulty in feature extraction.

In summary, the current research can effectively predict the QoS attribute value of web service composition, but the attribute values of QoS fluctuate when users invoke services in different environments, leading to a decrease in the accuracy of the prediction results. In Table 1, the related research work is summarized. Aiming at this problem, this study adopts the QoS confidence interval of web service composition estimation and prediction method QCI-WSC to enable the QoS values within the interval, so that users can make choices flexibly according to their own needs.

3 Method

3.1 Structure model of web service composition

To calculate QCI-WSC more efficiently based on Bootstrap, a list structure similar to LISP Language is proposed to express web service composition. The structures of the non-nested list and the nested list are shown below.

Sequence list: The structure of sequence (sequence, s_1, s_2, ..., s_n) is illustrated in Figure 1. In this model, *sequence* means the type of sequence structure, while s_1, s_2, ..., s_n represent the web service node executed in sequence.

Loop list: The structure of loop (loop, k, s_1, s_2, ..., s_n) is shown in Figure 2, where *loop* is the type of loop structure; and k represents the cycle number of the web service nodes s_1, s_2, ..., s_n in the loop body.

Choose list: Structure of the choose is described in Figure 3. In the structure (choose, p_1, s_1, p_2, s_2, ..., p_n, s_n), *choose* represents the type of choose structure; p_i (i = 1, 2, ..., n) means executive probability of web service nodes, and the total probability of web service composition is 1.

Parallel list: The structure of parallel (parallel, s_1, s_2, ..., s_n) is presented in Figure 4. In this model, web service nodes s_1, s_2, ..., s_n in parallel structure execute tasks concurrently.

Nested list: The workflow-based web service composition structure is shown in Figure 5, and this structure is the nested list structure web service composition for the participating experimental design. As it is shown in Figure 5, web service composition is composed by multiple web service nodes on the basis of designated correlation. It can be

Table 1: Related research work

Algorithm	Principle of the algorithm	Advantages	Disadvantages
QoS prediction method based on workflow [7–9]	The method builds a web service composition model through the workflow technique so that the QoS attributes can be constrained. Apart from that, it is able to estimate the predicted QoS attribute values during the processes of service creation and management	The approach simplifies the structure of web service composition and describes the portfolio process rationally, avoiding complex processing tasks during the prediction phase	The approach ignores the dynamic characteristics of QoS properties, and it focuses on the contextual association of workflows, which cannot be applied in a forecasting environment that considers the time factor
CF based QoS prediction method [10–13]	The method collects user historical QoS data for setting user preferences and calculates the similarity degree by the user-item relationship. Based on the historical QoS data of similar users, the QoS attribute values are predicted, and items are recommended for the current users	This method is the most widely used technique at present, and is mainly used for real-time recommendation and small data volume scenarios with high accuracy	The method is not suitable for scenarios with sparse or large data volumes, and is susceptible to the sparsity and cold start of historical QoS data. Therefore, the prediction results cannot better meet user needs
Model-based QoS prediction method [14–16]	The method divides the data into two parts. It designs the model by learning historical user QoS data from the training set, and then uses the test set to verify the accuracy of the model. When the accuracy reaches a specific threshold, unknown QoS values are predicted	The method is mainly used for scenarios with large data volume as well as offline computation	The training time of the method is too long to extract eigenvalues slowly. Other than that, QoS has the characteristics of nonlinearity and dynamics, which makes the method feature extraction difficult

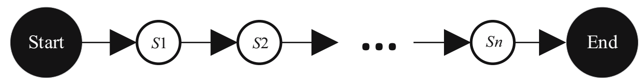

Figure 1: Sequence structure of web service composition.

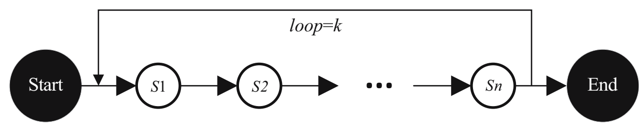

Figure 2: Loop structure of web service composition.

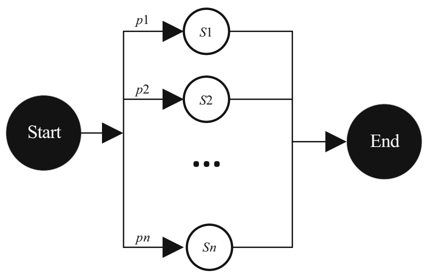

Figure 3: Choose structure of web service composition.

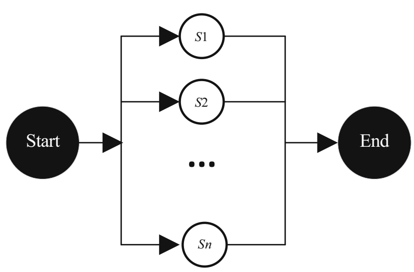

Figure 4: Parallel structure of web service composition.

presented as (sequence, s_1, (loop, k, s_2), (parallel, s_3, ..., (choose, p_1, s_i, p_2, s_{i+1}, ..., p_n, s_{n-1})), s_n). In this structure, (parallel, s_3, ..., (choose, p_1, s_i, p_2, s_{i+1}, ..., p_n, s_{n-1})) and (loop, k, s_2) indicate the nested child nodes of web service composition, while (choose, p_1, s_i, p_2, s_{i+1}, ..., p_n, s_{n-1}) is the nested child nodes of parallel structure.

3.2 QoS calculation of web service composition

The calculation of QoS values for web service composition can be classified into non-nested list structure and nested list structure. Considering that web service composition contains multiple measurements, the value of web service composition QoS is defined as $QoS = (Q_{\text{ava}}, Q_{\text{rel}}, Q_{\text{time}}, Q_{\text{cost}}, Q_{\text{rep}})$, where Q_{ava}, Q_{rel}, Q_{time}, Q_{cost}, and Q_{rep} represent the availability, reliability, response time, service costs, and reputation attributes of QoS.

3.2.1 QoS calculation for non-nested list web service composition

Before QoS of non-nested list web service composition is calculated, the model structure type of web service composition should be determined. Then, the QoS by quality criteria can be calculated.

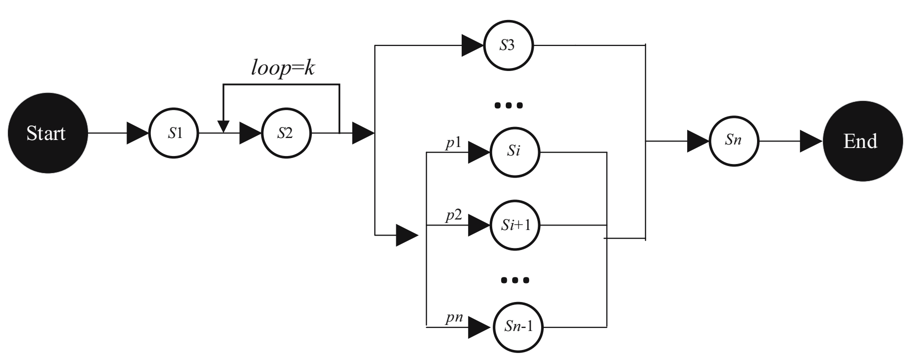

Figure 5: Web service composition based on workflow.

The sequence structure web service composition contains n web service nodes. The QoS for sequence structure under different attributes Q_{seq} can be computed as follows:

$$Q_{\text{seq}} = \prod_{i=1}^{n} \text{qos}_{s_i}(\text{qos}_{s_i} = \{Q_{\text{ava}}^{s_i},\ Q_{\text{rel}}^{s_i},\ Q_{\text{rep}}^{s_i}\}), \tag{1}$$

$$Q_{\text{seq}} = \sum_{i=1}^{n} \text{qos}_{s_i}(\text{qos}_{s_i} = \{Q_{\text{time}}^{s_i},\ Q_{\text{cos}\ t}^{s_i}\}). \tag{2}$$

Assuming that the web service node number of loop structure is n and the cycle-index of service nodes is k, the QoS of loop structure web service composition Q_{loo} can be computed using (3) and (4).

$$Q_{\text{loo}} = \prod_{i=1}^{n} \text{qos}_{s_i}(\text{qos}_{s_i} = \{Q_{\text{ava}}^{s_i\ k},\ Q_{\text{rel}}^{s_i\ k},\ Q_{\text{rep}}^{s_i\ k}\}), \tag{3}$$

$$Q_{\text{loo}} = k\sum_{i=1}^{n} \text{qos}_{s_i}(\text{qos}_{s_i} = \{Q_{\text{time}}^{s_i},\ Q_{\text{cos}\ t}^{s_i}\}). \tag{4}$$

In the choose structure of web service composition, n web service nodes correspond to executive probability p_i, and $\sum_{i=1}^{n} p_i=1$. The QoS of choose structure Q_{cho} can be expressed as

$$Q_{\text{cho}} = \sum_{i=1}^{n} p_i \times \text{qos}_{s_i}(\text{qos}_{s_i} = \{Q_{\text{ava}}^{s_i},\ Q_{\text{rel}}^{s_i},\ Q_{\text{time}}^{s_i},\ Q_{\text{cos}\ t}^{s_i},\ Q_{\text{rep}}^{s_i}\}). \tag{5}$$

There are n web service nodes in parallel structure. According to the execution sequence of the parallel struc-

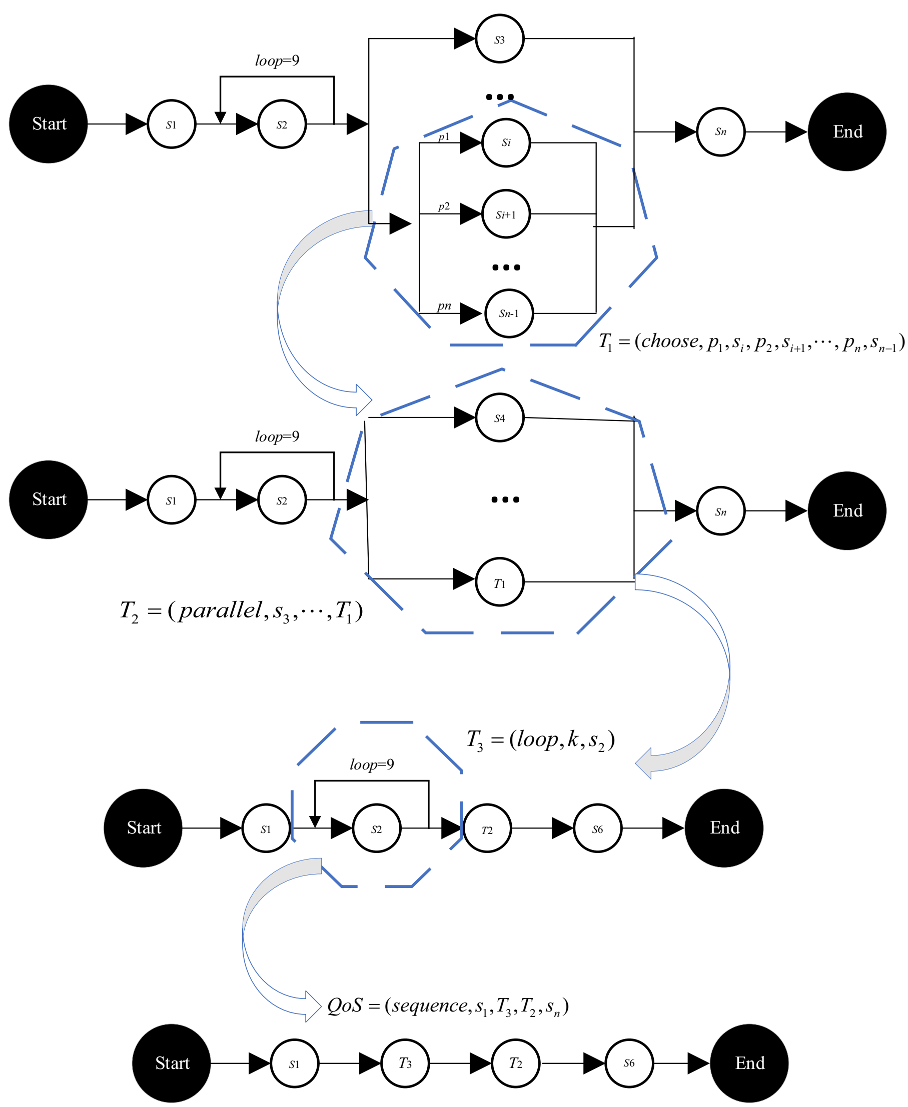

Figure 6: Flow chart of web service composition calculation.

ture, the QoS of the web service composition Q_{par} can be represented as follows:

$$Q_{\text{par}} = \prod_{i=1}^{n} \text{qos}_{s_i}(\text{qos}_{s_i} = \{Q_{\text{ava}}^{s_i},\ Q_{\text{rel}}^{s_i},\ Q_{\text{rep}}^{s_i}\}), \tag{6}$$

$$Q_{\text{par}} = \max\{Q_{\text{time}}^{s_1},\ Q_{\text{time}}^{s_2} \dots Q_{\text{time}}^{s_n}\}, \tag{7}$$

$$Q_{\text{par}} = \sum_{i=1}^{n} Q_{\text{cos }t}^{s_i}. \tag{8}$$

3.2.2 QoS calculation for nested list web service composition

To calculate nested list-structural web service composition QoS, the nested child nodes can be converted to non-nested child nodes by the recursive method. In the process, the first step is to determine the type and nested substructure for calculating the composition of web services. Then, all service nodes for web service composition can be simplified as non-nested child nodes, and the QoS value of web service composition can be computed by the QoS calculation formula.

Figure 6 illustrates the calculating processes of web service composition QoS based on the workflow. The processes are shown below:

(1) Separate the structure of web service composition. In the nested list structure (parallel, s_3, ..., (choose, p_1, s_i, p_2, s_{i+1}, ..., p_n, s_{n-1})), the QoS of nested child nodes (choose, p_1, s_i, p_2, s_{i+1}, ..., p_n, s_{n-1}) is calculated preferentially and represented as T_1.
(2) Substitute nested child nodes by QoS value, and the QoS of the nested list structure in the previous step is expressed as T_2 = (parallel, s_3, ..., T_1).
(3) For each nested structure, simplify them to sequence structure of web service composition, and calculate loop structural QoS value T_3 = (loop, k, s_2).
(4) Calculate the QoS for simplified web service composition QoS = (sequence, s_1, T_3, T_2, s_n).

3.3 Estimation of QCI-WSC based on Bootstrap

In this section, the non-parametric statistical Bootstrap technology is adopted to estimate the QCI-WSC, and the key components are elaborated as follows:

(i) The QoS value (e.g., response time) $Q^s = \{Q^{s_1},\ Q^{s_2},\ \dots,\ Q^{s_n}\}$ of n web service nodes and web service composition $WS = \{s_i \mid \forall s_i \in s,\ i = 1, 2, \dots, n\}$ are original sample data invoked by User_l ($l = 1, \dots, m$).

- QoS contains two types: one is independent of the user and its value is uniformly provided by the service provider, such as service cost; the other type is related to the user and the service environment, and the QoS values are susceptible to the network environment and the level of equipment, such as response time. However, it is unrealistic to require each user to invoke all services to obtain QoS values. Therefore, in the study, a common attribute response time in QoS was used.
- The QoS value of web service node s_k invoked by User_l ($l = 1, \dots, m$) is $Q_{\text{User}_l}^{s_k} = \{Q_{\text{User}_1}^{s_k},\ Q_{\text{User}_2}^{s_k},\ \dots,\ Q_{\text{User}_m}^{s_k}\}$.
- When current user $User_x$ invoked web service node s_k, the QoS value is $Q_{\text{User}_x}^{s_k} = \{\text{qos}_1,\ \text{qos}_2, \dots, \text{qos}_a\}$.

(ii) Replace n web service nodes in a nested list structure web service composition and sample historical QoS values, and obtain Bootstrap sample data of web service composition as $Q^{s*} = \{Q^{s_1*},\ Q^{s_2*}, \dots, Q^{s_n*}\}$, $Q^{s*} \in Q^s$.

(iii) About the Bootstrap sample data, QoS of web service composition named Qws was calculated by the current user as "Bootstrap estimated value." The formulation is shown as (9).

$$Q_{ws} = Q^{s_1*} + T_3 + T_2 + Q^{s_n*}, \tag{9}$$

where Q^{s_1*} and Q^{s_n*} indicate the response time of web service child nodes s_1 and s_n, respectively. T_2 and T_3 are QoS value of web service nested child nodes, which can be calculated using (10) and (11).

$$T_3 = k \times Q^{s_2*}, \tag{10}$$

$$T_2 = \max\{Q^{s_3*}, \dots, T_1\}, \tag{11}$$

where k represents the cycle index of web service nodes; Q^{s_2*} and Q^{s_3*} are response times of web service child nodes s_2 and s_3, and T_1 denotes QoS value of web service nested child nodes that can be represented as (12).

$$T_1 = \sum_{x=i}^{n-1} p_x \times Q^{s_x*}, \tag{12}$$

where p_x represents executive probability of web service nodes; and Q^{s_x*} refers to the response time of web service child node s_x.

(iv) Calculate B-time (B = 10,000) web service composition estimated values based on Bootstrap, and it is represented as $Q_{ws} = \{Q_{ws_1},\ Q_{ws_2}, \dots, Q_{ws_B}\}$.

(v) The B Bootstrap estimated values are sorted as $Q_{ws}[1] \le Q_{ws}[2] \le \dots \le Q_{ws}[B]$, calculated in ascending order by percentile Bootstrap. Hence, the QCI-WSC based on workflow is [low(Q_{ws}), up(Q_{ws})] when the confidence level is $1 - \alpha$ ($\alpha = 0.05$). The lower bound of the confidence interval is $\text{low}(Q_{ws}) = Q_{ws}(|B \times \alpha/2|)$, and the upper bound is $\text{up}(Q_{ws}) = Q_{ws}(|(B \times (1 - \alpha/2))|)$.

The procedure of estimating the QCI-WSC based on the study by Zheng et al. [18] is described in Algorithm 1.

Algorithm 1: Estimation of confidence interval for Web service composition QoS based on Bootstrap

Input: WS: Web service composition; Q^s: historical QoS data of Web service node s
Output: Estimation of QCI-WSC
Q_{ws}←[]
for i = 1 to B do //Calculate B-time Web service composition estimated values
    for j = 1 to length(WS) do //Read Web service nodes of composition
        if WS[[j]] in nested list structure
            Add WS[[j]] to WS_new //Add nested nodes to newly Web service composition
            Q^{s^*} = random(Q^s) //Randomly extract QoS value of WS_new
            Add Q^{s^*} to Q_{ws} //Calculate QoS attribute value of WS_new by formula (9)
        else
            Add Q^{s^*} to Q_{ws} //Calculate QoS attribute value of WS_new by formulas (1) to (8)
    end for
end for
Sort(Q_{ws})
Return [low(Q_{ws}), up(Q_{ws})] //Estimate QCI-WSC when confidence level is $1 - \alpha$

3.4 Prediction of QCI-WSC based on Bootstrap

The prediction method utilizes QoS sample data to calculate user similarity, predicts QoS value of current user invoking web service according to similar users, and then calculates the QCI-WSC invoked by the current user based on the method of percentile interval estimation.

3.4.1 Similarity calculation

Due to the diversity of QoS influencing factors, different users have different QoS values for invoking service. Therefore, original sample data and adjusted cosine are shown in (13) to calculate the similarity between users.

$$\mathrm{sim}(\mathrm{test}_i, \mathrm{train}_i) = \frac{\sum_{w\in W_{\mathrm{test}_i,\mathrm{train}_i}}(Q_{w\,\mathrm{test}_i} - \overline{Q_{\mathrm{test}_i}})(Q_{w\,\mathrm{train}_i} - \overline{Q_{\mathrm{train}_i}})}{\sqrt{\sum_{w\in W_{\mathrm{test}_i}}(Q_{w\,\mathrm{test}_i} - \overline{Q_{\mathrm{test}_i}})^2}\sqrt{\sum_{w\in W_{\mathrm{train}_i}}(Q_{w\,\mathrm{train}_i} - \overline{Q_{\mathrm{train}_i}})^2}}, \tag{13}$$

where $W_{\mathrm{test}_i,\ \mathrm{train}_i}$ represents the web service set that both user test$_i$ and user train$_i$ have invoked. W_{test_i} and W_{train_i} refer to the web service sets invoked by user test$_i$ and user train$_i$, respectively. $Q_{w\ \mathrm{test}_i}$ and $Q_{w\ \mathrm{train}_i}$ are historical QoS data for web service w invoked by user test$_i$ and user train$_i$, respectively. $\overline{Q_{\mathrm{test}_i}}$ and $\overline{Q_{\mathrm{train}_i}}$ are average web service QoS data used by users described as follows:

$$\overline{Q_{\mathrm{test}_i}} = \frac{1}{W}\sum_{w\in W_{\mathrm{test}_i}} Q_{w\ \mathrm{test}_i}, \tag{14}$$

$$\overline{Q_{\mathrm{train}_i}} = \frac{1}{W}\sum_{w\in W_{\mathrm{train}_i}} Q_{w\ \mathrm{train}_i}, \tag{15}$$

where W represents the number of all web services that current users invoke.

3.4.2 Prediction of web service composition QoS

Define Top(test) = {top$_1$, top$_2$, …, top$_k$} to represent the similar users of top k with current user test$_i$. The original sample data are sampled from historical data of web service composition, and it is $Q_{\mathrm{top}}^s = \{Q_{\mathrm{top}}^{s_1}, Q_{\mathrm{top}}^{s_2}, \ldots, Q_{\mathrm{top}}^{s_n}\}$, $Q_{\mathrm{top}}^s \in Q^s$, invoked by similar users.

Hence, the QoS in original sample data that current web service node Sk invoked by similar users Top(test) is $Q_{\mathrm{top}}^{S_k}$ is $Q_{\mathrm{top}}^{S_k} = \{Q_{\mathrm{top}_1}^{S_k}, Q_{\mathrm{top}_2}^{S_k}, \ldots, Q_{\mathrm{top}_k}^{S_k}\}$, and $Q_{\mathrm{top}_x}^{S_k} = \{\mathrm{qos}_1, \mathrm{qos}_2, \ldots, \mathrm{qos}_a\}$.

Based on $Q_{\mathrm{top}}^{S_k}$, the data is sampled with replacement to obtain the "Bootstrap sample value" $Q_{\mathrm{top}}^{s*} = \{Q_{\mathrm{top}}^{s_1*}, Q_{\mathrm{top}}^{s_2*}, \ldots, Q_{\mathrm{top}}^{s_n*}\}$.

The sample data Q_{top}^{s*} is presented to predict QoS value $\widehat{Q_{\mathrm{test}_i}}$, and Bootstrap predicted sample data are computed by user test$_i$ invoking current web service node s_j. The calculation is carried out using the below expressions:

$$\widehat{Q_{\mathrm{test}_i}} = \overline{Q_{\mathrm{test}_i}} + \frac{\sum_{i=1}^{k}\mathrm{sim}(\mathrm{test}_i, \mathrm{train}_i)(Q_{\mathrm{top}}^{s_j*} - \overline{Q_{\mathrm{User}_i}^{s_j}})}{\sum_{i=1}^{k}\mathrm{sim}(\mathrm{test}_i, \mathrm{train}_i)}, \tag{16}$$

$$\overline{Q_{\mathrm{User}_i}^{s_j}} = \frac{1}{n}\sum_{k=1}^{n} Q_{\mathrm{User}_i k}^{s_j}, \tag{17}$$

where n represents the total number of User_i who invokes current web service s_j, and $Q^{s_j}_{\mathrm{User}_i k}$ is the historical QoS value that User_i invokes current web service at kth time.

Given the above process, the QoS value of web service composition predicted for user $test_i$ can be represented as $\widehat{Q_{\mathrm{test}_i}} = \{\widehat{Q_{\mathrm{test}_1}}, \widehat{Q_{\mathrm{test}_2}}, \ldots, \widehat{Q_{\mathrm{test}_n}}\}$, and the QoS value based on workflow for web service composition calculated by (9) is $\widehat{Q_{ws}}$.

3.4.3 Prediction of confidence interval for web service composition QoS

The percentile interval estimation method is utilized, and B (B = 10,000) times are calculated repeatedly following the steps in this section, so that B Bootstrap prediction QoS values of web service composition obtained by user $test_i$ are presented as $\widehat{Q_{ws_1}}, \widehat{Q_{ws_2}}, \ldots, \widehat{Q_{ws_B}}$. Then, the B predicted Bootstrap values are sorted in ascending order and denoted as $\widehat{Q_{ws}}[1] \le \widehat{Q_{ws}}[2] \le \ldots \le \widehat{Q_{ws}}[B]$; the QCI-WSC invoked by current users is calculated as $[\mathrm{low}(\widehat{Q_{ws}}),\ \mathrm{up}(\widehat{Q_{ws}})]$ at confidence level of $1 - \alpha$ (α = 0.05). Wherein, the lower bound of the confidence interval is $\mathrm{low}(\widehat{Q_{ws}}) = \widehat{Q_{ws}}(|B \times \alpha/2|)$, and the upper bound of the confidence interval is $\mathrm{up}(\widehat{Q_{ws}}) = \widehat{Q_{ws}}(|(B \times (1 - \alpha/2))|)$.

3.4.4 Confidence interval evaluation

To verify the authenticity of the prediction method for the QCI-WSC, the interval coverage rate (ICR) and average interval coverage rate (AICR) methods were proposed for the main purpose of determining whether the calculated actual QoS value QoSws is within the predicted range of the QCI-WSC. The formula of AICR is shown as follows:

$$\mathrm{AICR} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{ICR}_{\mathrm{test}_i}, \tag{18}$$

$$\mathrm{ICR}_{\mathrm{test}_i} = \begin{cases} 1, & \mathrm{low}(\widehat{Q_{ws}}) \le Q_{ws} \le \mathrm{up}(\widehat{Q_{ws}}) \\ 0, & \text{else.} \end{cases} \tag{19}$$

where $\mathrm{low}(\widehat{Q_{ws}})$ represents the lower bound of the QoS prediction confidence interval, and $\mathrm{up}(\widehat{Q_{ws}})$ refers to the upper bound. Besides, ICR denotes the interval probability. When the QoS real values of the web service composition invoked by the current user are within the QoS confidence interval, the interval probability is taken as 1. Otherwise, interval probability is taken as 0. Therefore, AICR is a measure of the probability that is the real value in the confidence interval.

In brief, the description of QoS confidence interval prediction for web service composition is shown in Algorithm 2.

Algorithm 2: Confidence interval prediction based on Bootstrap of web service composition

Input: WS: Web service composition, test: testing users set, train: training users set
Output: Predicted value of QCI-WSC

$\widehat{Q_{ws}} \leftarrow []$
for e = 1 to length(test) do
    for a = 1 to length(train) do
        Sim(e,a) //Calculate similarity between test user and train user by formula (13)
        Add top k users to Top(test) //Choose the top users based on similarity
    end for
end for
for i = 1 to B do //Calculate B-time Web service composition prediction
    for j = 1 to length(WS) do
        Q^{s*}_{top} = random(Q^{s}_{top}) //Randomly extract the QoS values of similar user invoked service nodes
        Add Q^{s*}_{top} to $\widehat{Q_{\mathrm{test}_i}}$ //Calculate the prediction value of current service node by formula (16)
        Add $\widehat{Q_{\mathrm{test}_i}}$ to $\widehat{Q_{ws}}$ //Calculate QoS attribute value of *WS_new* by formula (9)
    end for
end for
Sort($\widehat{Q_{ws}}$)
Return $[\mathrm{low}(\widehat{Q_{ws}}),\ \mathrm{up}(\widehat{Q_{ws}})]$ //Predict QCI-WSC when confidence level is $1 - \alpha$

4 Experiment

In this section, workflow-based QCI-WSC estimation and prediction experiments were conducted using Bootstrap. In addition, the WSDream dataset 1 published in the study by Wang et al. [30] was used. This dataset contains historical response time data of 107 users calling 99 web services at different times. In addition, the experimental environment was Rstudio Version 1.2; the experimental machine was configured with 16G memory, and the CPU was an i7-7700 3.6GHZ processor.

4.1 Analysis of QoS confidence interval estimation results

According to Algorithm 1, the estimated results of 107 users invoking nested list web service composition with the 95% confidence level were calculated. From Figure 7, it was discovered that the upper bound of the QoS confidence was mostly between 0 and 4,000 ms; a small portion of users were between 10,000 and 20,000 ms, and the rest users were between 10,000 and 20,000 ms. However, the lower bound of the QoS confidence was almost distributed between 0 and 20,000 ms, partially between 20,000 and 40,000 ms, and rarely between 40,000 and 50,000 ms. Figure 8 illustrated exponential distribution of the confidence interval, and the distribution of the upper and lower bound for the confidence interval was mainly adjacent straight lines. Therefore, the QoS confidence interval estimation result of the web service composition fitted the exponential distribution, which proved the effectiveness of the computation of the QCI-WSC.

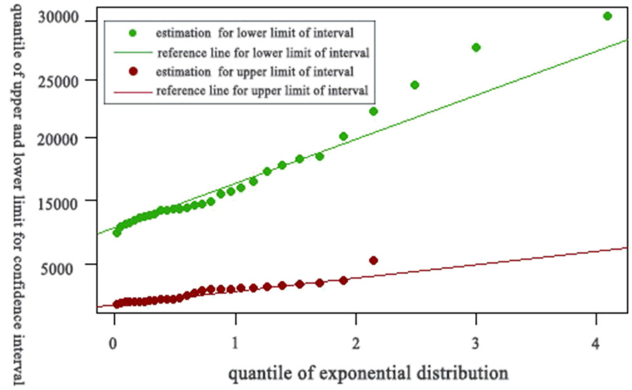

Figure 8: QoS confidence interval exponential distribution.

4.2 Analysis of QoS confidence interval prediction results

4.2.1 Prediction result evaluation

In this experiment, the users of the dataset were randomly classified into 10, 15, 20, 25, 30, 35, and 40% as the test user dataset, and other users as the training user dataset. Then, the top 10, 20, and 30 similar training users were selected to predict the QCI-WSC invoked by the testing users. Finally, the reliability and authenticity of QoS confidence interval prediction results are verified through AICR.

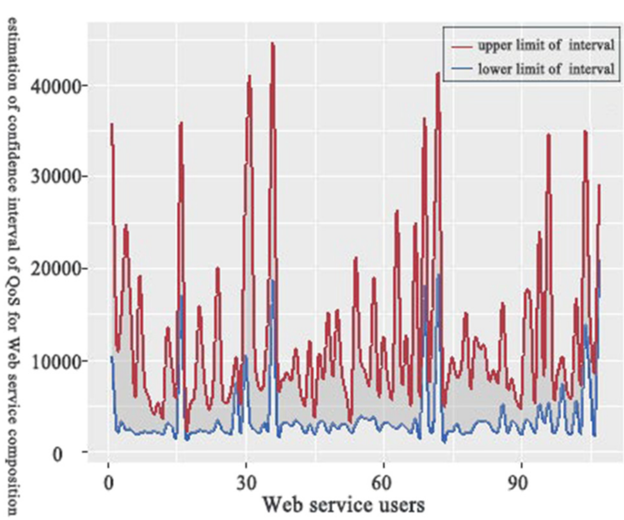

Figure 7: Confidence interval estimation of QoS.

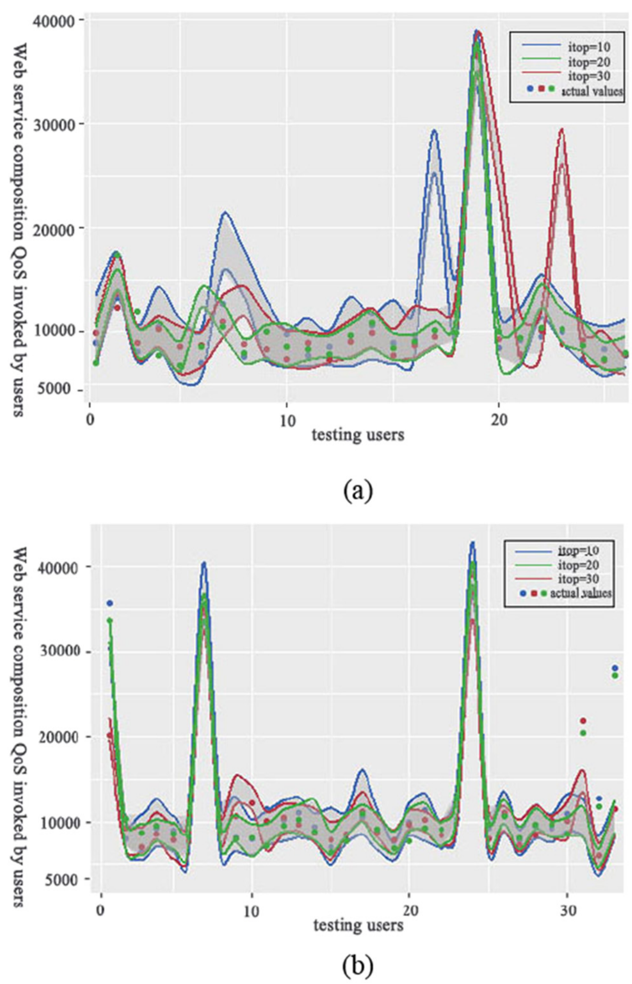

Figure 9: Distribution of QCI-WSC invoked by testing users. (a) The proportion of testing users is 25%. (b) The proportion of testing users is 30%.

Figure 9 shows the QCI-WSC prediction results for 25 and 30% of test users when selecting the top 10, 20, and 30 similar users. It can be seen that the proportion of test users and the number of similar users have little impact on the results of QoS confidence interval prediction. Furthermore, the actual QoS coverage of users tested by calling the web service combination is relatively high within the predicted range.

To demonstrate the practicality and reliability of the prediction method, the predicted QCI-WSC was validated using AICR, and the results are shown in Figure 10. It can be observed that when the average interval probability is 75%, there are many actual values in the confidence interval. In addition, the average interval probability increases with the increase in the proportion of test users. When the proportion of test users is 30% of the total users, the average interval probability is the highest.

4.2.2 Comparison results with other methods

To verify the scalability of the prediction results of the proposed method, the prediction interval results were compared with the QACF [31] and WSRec [12], when the proportion of test users was 25 and 30%, respectively. The results are shown in Figure 11.

When the proportion of test users is 25%, QACF and WSRec have higher prediction accuracy. In addition to that, the error between the predicted and accurate values of the two methods is relatively small, which can help users choose suitable services. However, when the proportion of test users rose to 30%, the deviation between the predicted and accurate values of the above two methods increased. Therefore, the QCI-WSC can accurately cover the actual QoS values composed of web services and most of the accurate QoS values predicted by QACF and WSRec. The QCI-WSC can provide users with more services or service combination choices, which helps them choose services that meet their needs.

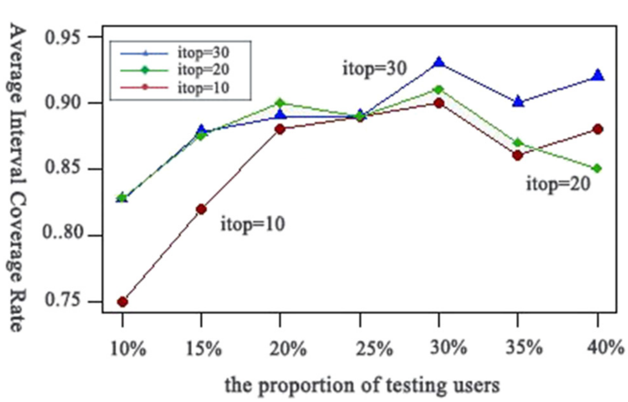

Figure 10: Average interval probability results of testing users with different division ratios.

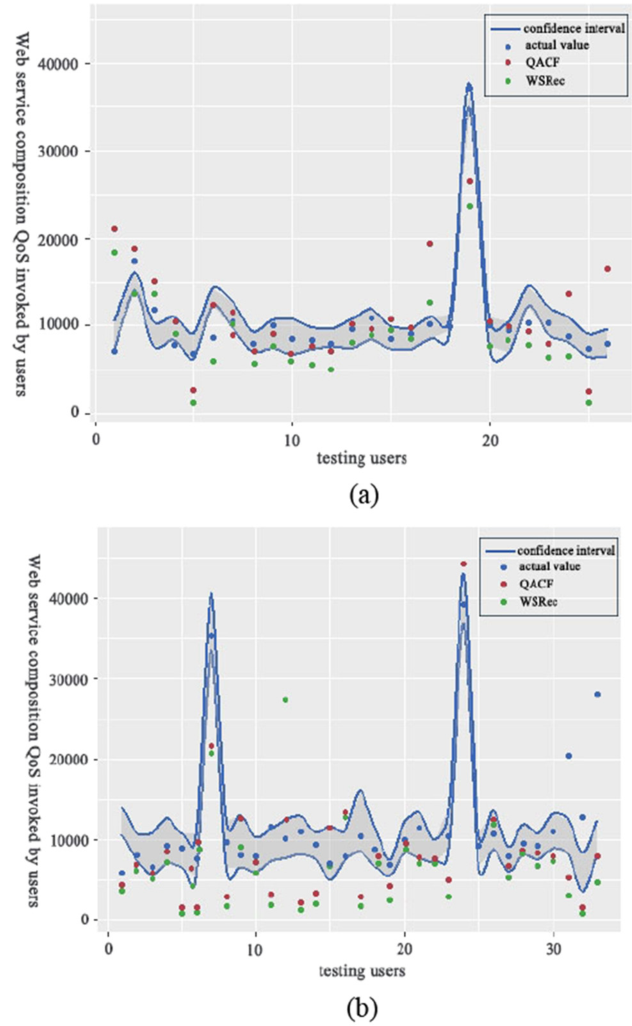

Figure 11: Prediction value of QACF, WSRec, and QCI-WSC. (a) The proportion of testing users is 25%. (b) The proportion of testing users is 30%.

5 Conclusion

To solve the problem that the existing web service composition QoS prediction methods cannot better reflect the characteristics of stochastic dynamics of QoS, this study adopts a method for estimating and predicting the QCI-WSC based on Bootstrap technology. In this method, the composition structure model of web services is simplified, and the historical QoS data of web services are employed to estimate the QCI-WSC invoked by users; besides, the QCI-

WSC invoked by current users is predicted based on the historical QoS data of similar users. The experimental results show that the QoS confidence intervals estimated and predicted in this study can cover the actual QoS attribute values more accurately, and the AICR used to measure the predicted confidence interval is above 75%. The method results can be replicated in the future to design QoS-aware web service composition optimization, service recommendation, and selection, and analyze the impact of changes in the confidence intervals on web service composition and service performance, thereby optimizing the combination method and making service selection and recommendation more stable. It can also make reasonable service selections and recommendations for service combinations under the Restful architecture, simplifying the development and design of future web systems and mobile applications.

In this work, five general QoS attributes were described in detail. However, in the actual network environment, the quality requirements of web services by different users and service providers are affected by the actual application scenarios, and the values of QoS attributes possess individual characteristics. Future research should further introduce more QoS attribute values, consider QoS confidence interval estimation and prediction based on QoS influencing factor weights in dynamic web service composition, and provide stronger basis for users to choose web service composition.

Acknowledgments: This work was supported by the Teaching Reform Project of Jiangxi Provincial Education Department (Nos. JXJG-21-24-6 and JXJG-24-24-2), Jiangxi University of Technology Collaborative Innovation Center Platform Open Project (XTCX2105), the National Natural Science Foundation of China (No. 61262014), the Science and Technology Project of Jiangxi Provincial Department of Education (Nos. GJJ202017, GJJ2402521 and GJJ212023), and Jiangxi University of Technology Teaching Reform Project (Nos. JY2104 and JY2004).

Funding information: This work was supported by the Teaching Reform Project of Jiangxi Provincial Education Department (Nos. JXJG-21-24-6 and JXJG-24-24-2), Jiangxi University of Technology Collaborative Innovation Center Platform Open Project (XTCX2105), the National Natural Science Foundation of China (No. 61262014), the Science and Technology Project of Jiangxi Provincial Department of Education (Nos. GJJ202017, GJJ2402521 and GJJ212023), and Jiangxi University of Technology Teaching Reform Project (Nos. JY2104 and JY2004).

Author contributions: Qinying Li conceptualized and designed the study, developed QCI – WSC, drafted and revised the paper, and handled communication; Wei Lu collected data from WSDream, analyzed structure simplification, and verified results; Fangli Li performed user similarity and QoS prediction experiments, and organized data; Taotao Wang analyzed Bootstrap – based QoS interval estimation and compared methods; Hao Wang validated result distribution, and polished language and logic.

Conflict of interest: The authors state no conflict of interest.

Data availability statement: The data presented in this study are available on request from the corresponding author.

References

[1] Z. P. He, P. C. Zhang, Y. Jiang, S. H. Ji, and W. R. Li, "Time-aware web service QoS monitoring approach under dynamic environments," *J. Softw.*, vol. 29, no. 12, pp. 3716–3732, 2018.

[2] C. Yu and L. Huang, "A Web service QoS prediction approach based on time- and location-aware collaborative filtering," *Serv. Oriented Comput. Appl.*, vol. 10, no. 2, pp. 135–149, 2016.

[3] F. Seghir, A. Khababa, and F. Semchedine, "An interval-based multi-objective artificial bee colony algorithm for solving the web service composition under uncertain QoS," *J. Supercomput.*, vol. 75, no. 9, pp. 5622–5666, 2019.

[4] Q. L. Wu and T. H. Zhou, "Research on quality of service-based dynamic web service composition," *Appl. Softw.*, vol. 33, no. 5, pp. 20–23, 2016.

[5] Y. W. Zhang, T. Xiang, and X. Guo, "Quality prediction for services based on SOM neural network," *J. Softw.*, vol. 29, no. 11, pp. 3388–3399, 2018.

[6] B. T. Jiang, Z. G. Hu, and H. Ma, "A cloud service QoS prediction method based on time-aware ranking," *Comput. Eng. Sci.*, vol. 40, no. 7, pp. 1173–1179, 2018.

[7] M. C. Jaeger, G. Rojec-Goldmann, and M. Gero, "QoS aggregation in web service compositions," 2005 IEEE International Conference on e-Technology, e-Commerce, and e-Services, 2005.

[8] J. Cardoso, A. Sheth, and J. Miller, "Quality of service for workflows and web service processes," *J. Web Semant.*, vol. 1, no. 3, pp. 281–308, 2004.

[9] X. Wang, J. K. Zhu, and Y. H. Shen, "Network-aware QoS prediction for service composition using geolocation," *IEEE Trans. Serv. Comput.*, vol. 8, no. 4, pp. 630–643, 2015.

[10] S. Manochandar and M. Punniyamoorthy, "A new user similarity measure in a new prediction model for collaborative filtering," *Appl. Intell.*, vol. 51, no. 1, pp. 586–615, 2021.

[11] K. Halime and R. Jafar, "Survey of similarity functions on neighborhood-based collaborative filtering," *Expert. Syst. Appl.*, vol. 185, p. 115482, 2021.

[12] Z. Zheng, H. Ma, M. R. Lyu, and I. King, “QoS-aware web service recommendation by collaborative filtering,” *IEEE Trans. Serv. Comput.*, vol. 4, no. 2, pp. 140–152, 2011.
[13] Z. Zhou, B. Wang, and G. Jie, “QoS-aware web service recommendation using collaborative filtering with PGraph,” *2015 IEEE International Conference on Web Services*, New York, NY, USA, 2015, pp. 392–399.
[14] L. Chen, F. F. Xie, Z. B. Zheng, Y. M. Wu, and J. X. Li, “Predicting quality of service via leveraging location information,” *Complexity*, vol. 2019, no. 1, p. 4932030, 2019.
[15] G. Kang, J. Liu, and Y. Xiao, “Neural and attentional factorization machine-based Web API recommendation for mashup development,” *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 4, pp. 4183–4196, 2021.
[16] E. Ak and B. Canberk, “Forecasting quality of service for next-generation data-driven WiFi6 campus networks,” *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 4, pp. 4744–4755, 2021.
[17] L. Zeng, B. Benatallah, and A. H. H. Ngu, “QoS-aware middleware for web services composition,” *IEEE Trans. Softw. Eng.*, vol. 30, no. 5, pp. 311–327, 2004.
[18] Z. B. Zheng, Y. L. Zhang, and M. R. Lyu, “Distributed QoS evaluation for real-world web service (ICWS),” *2010 IEEE International Conference on IEEE*, 2010, pp. 83–90.
[19] J. X. Wu and F. C. Yang, “QoS prediction of web service composition with transaction mechanism,” *J. Electron. Inf. Technol.*, vol. 30, no. 3, pp. 703–706, 2008.
[20] Z. Z. Liu, Z. P. Jia, and X. Xue, “Reliable Web service composition based on QoS dynamic prediction,” *Soft Comput*, vol. 19, no. 5, pp. 1409–1425, 2015.
[21] H. Wu, K. Yue, and B. Li, “Collaborative QoS prediction with context-sensitive matrix factorization,” *Future Gener. Comput. Syst.*, vol. 82, pp. 669–678, 2018.
[22] A. D. Wang, *Research on Prediction Method of Web Services Quality Based on QoS*, Master’s thesis. North China Electric Power University, 2020.
[23] Y. Zhang, Z. Zheng, and M. R. Lyu, “WSPred: A time-aware personalized QoS prediction framework for Web Services,” *2011 22nd IEEE International Symposium on Software Reliability Engineering*, 2011.
[24] Z. Chen, L. M. Shen, and F. Li, “Exploiting Web service geographical neighborhood for collaborative QoS prediction,” *Future Gener. Comput. Syst.*, vol. 68, pp. 248–259, 2017.
[25] U. Thakker, R. Patel, and M. Shah, “A comprehensive analysis on movie recommendation system employing collaborative filtering,” *Multimed. Tools Appl.*, vol. 80, no. 19, pp. 28647–28672, 2021.
[26] X. Luo, Y. Lyv, and R. Li, “Web service QoS prediction based on adaptive dynamic programming using fuzzy neural networks for cloud services,” *IEEE Access*, vol. 3, pp. 2260–2269, 2015.
[27] N. Anithadevi and M. Sundarambal, “A design of intelligent QoS aware web service recommendation system,” *Clust. Comput.*, vol. 22, no. 6, pp. 14231–14240, 2019.
[28] W. Xiong, B. Li, and Z. Wu, “A spatial-temporal-aware QoS prediction approach,” *Chin. J. Comput.*, vol. 42, no. 4, pp. 98–111, 2019.
[29] X. Wang, R. Wang, and C. Shi, “Multi-component graph convolutional collaborative filtering,” *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 6267–6274.
[30] H. Wang, J. M. Xiao, and H. Long, “An approach to estimate and predict the confidence interval of web service QoS based on bootstrap,” *Acta Electron. Sin.*, vol. 46, no. 3, pp. 665–671, 2018.
[31] Y. Zhuang, Z. C. Guo, and X. Y. Huang, “An adaptive web service QoS prediction method based on collaborative filtering,” *Comput. Technol. Dev.*, vol. 30, no. 3, pp. 93–97, 2020.