Computing (2024) 106:2033–2058
https://doi.org/10.1007/s00607-024-01283-0

REGULAR PAPER

Qos-based web service selection using time-aware collaborative filtering: a literature review

Ezdehar Jawabreh[1,2] · **Adel Taweel**[1]

Received: 30 January 2024 / Accepted: 22 March 2024 / Published online: 9 April 2024

Abstract

The proliferation of available Web services presents a big challenge in selecting suitable services. Various methods have been devised to predict Quality of Service (QoS) values, aiming to address the service selection problem. However, these methods encounter numerous limitations that hinder their prediction accuracy. A key issue stems from the dynamic nature of the service environment, leading to fluctuations in QoS values due to factors like network load and hardware issues. To mitigate these challenges, QoS selection methods have leveraged contextual information from the surrounding environments, such as service invocation time, user, and service locations. Among these methods, Collaborative Filtering (CF) has gained notable importance. In recent years, several CF methods have incorporated service invocation time into their prediction processes, giving rise to what is commonly known as time-aware CF methods. Despite the increasing adoption of time-aware CF methods, there remains a notable absence of a dedicated and comprehensive literature review on this topic. Addressing this gap, this paper conducts an analysis of the literature, reviewing the forty (40) most prominent studies in this domain. It offers a thematic categorization of these studies along with an insightful analysis outlining their objectives, advantages, and limitations. The review also identifies key research gaps and proposes potential directions for future investigations. Overall, this literature review serves as an up-to-date resource for researchers engaged in service-oriented computing research.

Keywords Web service · QoS · Time-aware · Prediction · Collaborative filtering (CF)

✉ Ezdehar Jawabreh
eajawabreh@birzeit.edu

Adel Taweel
ataweel@birzeit.edu

[1] Department of Computer Science, Birzeit University, 1 Marj Street, Ramallah P627, Palestine

[2] Department of Computer Science, Palestine Polytechnic University, Wadi Alharea, Hebron P198, Palestine

Mathematics Subject Classification 68Q85

1 Introduction

Service Oriented Architecture (SOA) has emerged as a promising paradigm in system engineering, where systems are constructed by integrating services as their fundamental building blocks. The proliferation of service providers has led to a vast number of services, making the selection of the most suitable one among many offering equal functionalities a big challenge. To optimize, one approach, services are selected based on their Quality of Service (QoS) attributes (i.e. non-functional properties, such as response time, throughput, etc.). However, certain QoS attributes, being provider-declared, are not inherently stable. For instance, in the widely recognized WSDREAM dataset [1], the response time attribute exhibits fluctuations within a range of [0 s–20 s]. Consequently, the prediction of dynamic QoS attributes has become a critical area of research interest over the past decade.

In dynamic environments, users may encounter varying QoS values from the same service due to fluctuations in service load (number of clients) and network conditions (e.g., congestion) over time. Therefore, time emerges as a critical factor influencing prediction accuracy. Addressing this challenge, time-aware Collaborative Filtering (CF) methods have been proposed to predict QoS in such environments. Recent research has prominently favored these methods for QoS prediction, driven by several reasons. Firstly, they have demonstrated significant enhancements in prediction accuracy by incorporating diverse contextual information about users and services [2–4]. Secondly, their versatility has been evidenced across various applications in the service computing domain, including service selection, composition, adaptation, and fault tolerance [5]. Thirdly, they possess the capability to leverage large historical data for predicting current or future QoS values. Lastly, they exhibit adaptability to the dynamic environment, accommodating changes such as the introduction of new QoS values, new users, or services. Given the considerations mentioned above, it is evident that time-aware CF is emerging as a new trend for achieving accurate QoS predictions. In light of this trend, we present a comprehensive literature review that emphasizes various methods employed in this type of prediction.

Our review focused on time-aware QoS prediction using CF methods. The primary studies included in this review were collected from four known digital scientific libraries, namely IEEExplore, Springer, ScienceDirect, and ACM, spanning the years from 2011 to 2022. The named libraries were chosen as they encompass well-known journals and conferences in this field such as SOCA, TSC, ICWS, ICSOC, and SCC. During our search, the following keywords were used in the search queries: *Time aware, temporal, Collaborative Filtering, CF, QoS, service, predict, recommend, assess*. To limit the scope of our review, two inclusion criteria were applied: first, studies had to propose a time-aware QoS prediction method, predicting either current or future QoS; second, the proposed method had to be a CF method, utilizing data from other users and services in making QoS prediction.

Ultimately, we identified 40 notable studies that represent the current state-of-the-art. These studies were thematically categorized into three groups: (1) time-aware neighborhood CF, (2) time-aware model-based CF, and (3) time-aware hybrid approaches. Remarkably, our literature review is the first dedicated exploration of time-aware CF methods, as previous CF-related work discussed time-aware methods within a broader context. The specialization in our review proves valuable for researchers seeking a comprehensive understanding of state-of-the-art time-aware methods. Additionally, for each primary study, we analyzed its key strengths and weaknesses. We then conducted a comprehensive comparison of time-aware methods within our thematic categorization, offering readers insight into the evolution of research trends over the years. Lastly, we pinpointed key research challenges in time-aware CF and offered potential research directions to guide further exploration in this domain.

The rest of the review is organized as follows: Sect. 2 introduces the background, and Sect. 3 presents the related work. Section 4 describes our classification and the approaches under each category. In Sect. 5 we discuss our findings. Section 6 presents research challenges and directions. Finally, we conclude our work in Sect. 7.

2 Background

Utilizing collaborative filtering for QoS prediction draws inspiration from commercial recommendation systems, such as those employed by Netflix, Amazon, and eBay. The term "collaborative filtering" is defined as the process of filtering information or patterns through techniques involving collaboration among multiple users, agents, and data sources [6]. In their work [7], the authors take the initiative of applying collaborative filtering in QoS prediction. Their approach involved predicting the missing QoS values for a target user by leveraging the existing values from other similar users utilizing the same invoked services. To provide a more precise understanding, we will explain the issue using the formal definition of collaborative filtering methods:

- Let $U = \{ u_1, u_2 , \ldots , u_m \}$ is set of users for Web services, u_i denotes a user, where $(1 \leq i \leq m)$.
- Let $S = \{ s_1, s_2 , \ldots , s_n \}$ is set of Web services, s_j denotes a service, where $(1 \leq j \leq n)$.
- Let $Q_{m \times n}$ is user-service matrix, where q_{ij} represents the QoS value of the i user when he invoked the j service.

Figure 1 shows a toy example of the user-service matrix, which holds values for QoS (it can be any attribute, such as response time, and throughput). The blank cells indicate missing (unknown) QoS values since users have not invoked these services yet. In essence, the CF methods consist of three main types, which are neighborhood CF, model-based CF, and hybrid CF. Neighborhood CF, also recognized as memory-based CF, relies on similarity calculations, typically employing the Pearson Correlation Coefficient (PCC). The user PCC (UPCC)

	S1	S2	S3	S4	S5
U1	0.77		0.45		
U2		0.55		0.48	
U3		0.34			0.99
U4	0.67		0.89	0.39	

Fig. 1 User-service QoS matrix(U:user, S:service)

and item IPCC (IPCC) are commonly utilized methods for calculating user and service similarity, respectively. The model-based CF is introduced to address scalability and QoS sparsity problems, employing a pre-trained model to predict missing QoS values. Lastly, the hybrid methods aim to leverage the advantages of both neighborhood and model-based approaches to improve prediction accuracy.

Nonetheless, traditional CF types can be expanded by integrating the time of service invocation as an additional contextual factor. This integration, as previously discussed, often leads to improved accuracy in QoS predictions. Methods that incorporate time factors are commonly referred to as time-aware CF. Figure 2, illustrates the QoS matrix used in time-aware methods, which leverage QoS data gathered across various time intervals. This historical data can be augmented in the prediction process through various techniques, such as adjusting time similarity computations in neighborhood methods and training models using specialized time-aware datasets in model-based methods.

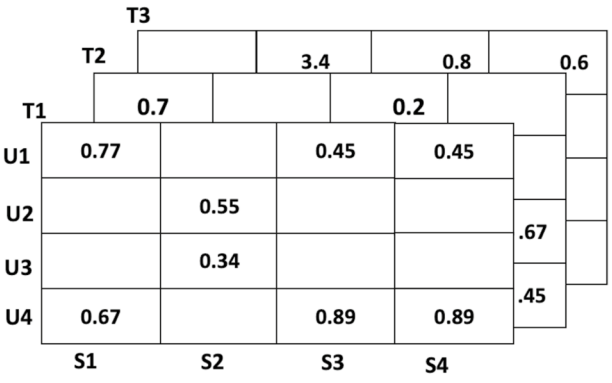

Fig. 2 User-service QoS matrix(U:user, S:service, T:time)

3 Related work

3.1 Time-aware methods for QoS prediction

QoS prediction using CF methods has received the attention of researchers in the last decade. Several studies reviewed and summarized these methods. These studies are general and not dedicated to any specific type, however, this literature review is dedicated only to the time-awareness CF methods in QoS prediction, and to the best of our knowledge there are no reviews that have been conducted in this field. However, we will discuss these general studies according to their relevance to the topic.

In [8], the authors provided a survey about Web service QoS prediction via CF, they categorized these methods in two levels: at the first level, they used the general categorization as neighborhood, model-based, and hybrid, at the second level the methods under each general category were further categorized according to what type of contextual data they incorporated, such as location, time or other. In addition, they discussed the forefront research issues like adaptability, credibility, and privacy-preserving. The work in [9] also provided a survey about QoS Web service prediction methods, the authors categorized methods into the known general categories: neighborhood, model-based, and hybrid. They also dedicated a specific section to time-aware collaborative methods, briefly discussing several popular techniques.

In literature, time-aware CF finds applications in various domains beyond the QoS prediction domain, including service recommendation. For example, in [10] the authors reviewed the time-aware recommender systems (TARS) that are used in various types of services, different types were reviewed such as time-aware CF, time-aware content-based, and time-aware knowledge-based, in their work QoS was used an important criteria for evaluating and recommending a service. In the same line of research, the authors in [11] provided an overview of the Web service recommendation system, they differentiated between recommendations and predictions. They also provided explanations of different types of CF, like user-based, item-based, model-based, personalized, and location-aware.

An alternative strategy that prioritizes time awareness in selecting services is the time series forecasting methodology. This method enables the statistical prediction of QoS values. Prominent methods here are the Moving Average (MA) method, Auto Regressive (AR), and Auto-Regressive Integrated with Moving Average (ARIMA). However, it's important to note that this approach diverges from collaborative filtering-based methods as it operates on a per-user-service basis, placing it beyond the scope of this review. Despite this limitation, we found that some CF methods integrated time series in their prediction, which prompted us to include a recent relevant study in this domain. In [12], a comprehensive survey on QoS time series modeling and forecasting is presented. The authors selected a collection of studies and examined four key aspects in each: the identified problem, the proposed methodology, the performance metrics considered, and the QoS time series dataset used. Additionally, they highlighted the shortcomings observed in these studies.

3.2 General time-aware CF methods

In this section, we focus on reviewing time-aware CF methods applied in areas not directly related to services. The study in [13] highlighted the significance of incorporating time factors to enhance the accuracy of CF recommendation systems. They discussed traditional CF methods and elaborated on how these techniques can be expanded to incorporate time factors using various techniques. In [14] the authors conducted an analysis of time-aware recommendation systems, highlighting the limitations of the evaluation methods used in these recommenders. They proposed a methodological framework aimed at ensuring a fair evaluation process. Additionally, the work in [15] presented a recent systematic review of neural network-based recommender systems. Within this review, they classified recommender systems into different categories, including CF. The authors specifically emphasized the growing trend of employing temporal (sequential) models to enhance the accuracy of recommenders, addressing this as a separate section within their work.

4 Time-aware CF methods: review

Time-aware CF methods are thematically categorized into three categories: time-aware neighborhood methods, time-aware model-based methods, and time-aware hybrid methods. This categorization aligns with the various aspects of time-awareness in QoS prediction. The subsequent subsections provide a literature review of the diverse methods within each category.

4.1 Time-aware neighbourhood collaborative filtering

The methods under this section used the traditional CF computation in both similarity and prediction measurements, however, to be time-aware methods they have to capture the dynamic change of QoS similarity over time. The time-aware similarity can be computed using one of two methods: first, using the time decay function as a weighting major for the effectiveness of QoS values, and second, using the time interval slots method. Next, we provide more details about the studies under each method. Additionally Table 1 highlights the strengths and limitations of each study individually, along with general information such as their publication year and type.

4.1.1 Time decay method

The authors in [16] used an exponential decay function whose value decreases as the time span between two related QoS increases or as the time span between the current time and two related QoS increases. They alleviated the data sparsity problem by using the random walk algorithm, which discovered the indirect user and service similarities. However, authors in [17] argued that using non-linear decay functions alone is not sufficient for evaluating the effectiveness of QoS values, so they

Table 1 Time-aware neighbourhood collaborative filtering

References	Pub. year	Pub. type	Name	Strengths	Limitations
[16]	2014	Journal	TAWSRec	Improved prediction accuracy and provided aggregate rank using multiple QoS prediction	High computation cost caused by random walk algorithm, works offline only
[17]	2013	Conference	FWSE-TCF	Improved the accuracy of prediction	Since it user-based it missed incorporating changes in service similarity over time, works offline only
[18]	2014	Conference	TACF	Provided a way to reduce the search space for similar services (users)	The number of time intervals was static, works offline only
[19]	2016	Journal	TLACF	Improved the prediction accuracy by incorporating context data	Additional time complexity was added for building the clusters, works offline only
[20]	2017	Conference	TACF-DT	Had a reasonable cluster-based algorithm to determine the length of time interval	Clustering approach required extra time computation, it works offline only
[21]	2017	Journal	CluCF	Alleviated sparsity and scalability problems, accommodated changes in users and services	There was a trade-off between prediction accuracy and scalability
[22]	2018	Conference	TSSRec	Distinguished between global and local service similarity	Since it is service based it ignored the temporal user similarity, works offline only
[23]	2017	Journal	CASR-TSE	Using the well-designed exponential function and location reasoning improved predictions	Ignored temporal service similarity, works offline only
[2]	2021	Journal	TUIPCC	Improved prediction accuracy over average similarity measure	Number of time intervals was static, works offline only

designed a hybrid decay function, of both linear and non-linear. Similarly, the study in [23] has used the exponential time decay function, but a novel idea is added, which aims to increase weights for QoS values that seemed to be too small or too large in user similarity calculation. They also modeled the correlation between user and service locations before calculating the similarity in order to increase prediction accuracy.

4.1.2 Time interval method

The time interval method was used with average similarity computation. This method divides the historical QoS data into time slots and creates a matrix of users and services in each slot. It computes similarity in each time slot and the final value of similarity at the current time is the average of similarities in all time slots. In a study done in [18], the authors calculated user and service similarity in a static number of time slots determined by a variable named d, which was a parameter used to reduce the searching space. The same authors extended their work and introduced a time and location-aware method in [19]. Their new method used location-based clusters of users and services in order to alleviate scalability problems. In [20], the authors tried to improve the work done in [18]. They used a clustering approach that determined dynamically the size of time slots instead of being static.

Another work in [21] introduced a novel approach named CluCF. This work extended the studies [18] and [19]. The authors alleviated the data sparsity problem. They converted the sparse user, service, and time tensor into a high-density user-service matrix, this matrix was converted into userCluster-service matrix and user-serviceCluster matrix. The clustering was based on location data. In the end, a hybrid prediction with weighted parameters is computed from both user and service predictions. In this method, the clusters can be updated when new users or services are introduced, however, it had a trade-off between scalability and prediction accuracy.

Later on, [22] and [2] improved the final similarity measure by using weighting functions and this achieved a better improvement over the average similarity measure used in the aforementioned studies. So first, in [22], a new approach was used to calculate the service similarity in the historical data. They used CANDECOMP/PARAFAC (CP) tensor decomposition to alleviate the data sparsity problem, and they assigned weights to global and temporal neighborhood services. Second, in [2], the user and service similarities were measured in a set of time slots, to compute the final similarity, the authors used a weighted decay function, which emphasized the similarity effect of recent time slots. In addition, they introduced a novel approach that searched for the most similar user in each time slot.

4.2 Time-aware model-based collaborative filtering

Time-aware model-based methods represent a large number of studies in CF methods. They depend on training a model with a large set of historical QoS data. The trained model can be used later for predicting QoS. They are further classified into

three subcategories: latent factors methods, clustering, machine learning methods, and deep learning methods.

4.2.1 Latent factors methods

Latent factors methods are based on the assumption that the user-service matrix can be factorized into low-rank latent factor matrices, by utilizing these matrices the missing QoS can be predicted. It's important to note that while latent factorization is the central focus across all studies in this section, some studies exhibit an overlap with other mentioned approaches. The strengths and limitations of each study are delineated individually in Table 2.

In the year 2011, Zhang et al. [1] introduced the first time-aware CF method which was named WSPred. This method created a tensor of three dimensions: user, services, and time. In order to predict missing QoS data, it performed a tensor factorization that learned the latent factors of users, and services in specific time intervals. The main contribution of their work was the data used in the tensor, which was real data that had been collected and used for the first time. It is known now as WSDREAM dataset2 [24] and it has become a well-known benchmark in the research community. Later on, similar work was introduced in [25]. The authors used a Non-negative Tensor Factorization (NTF) approach. The approach used CANDECOMP/PARAFAC (CP) factorization with consideration to the non-negativity property of QoS data. It decomposed the user, service, and time tensor into three non-negative latent matrices to get an approximation for the temporal QoS values. Moreover, the approach was evaluated using their own collected dataset, which was a tensor of size $343 \times 5817 \times 32$ user-service-time.

The same authors introduced another work in [26]. They used a triadic factorization approach on a user, service, and time tensor. The novelty in their approach was providing a mechanism to reduce the memory space needed to store the sparse data in the high dimensional tensor. To do so, they proposed two methods: Tucker Decomposition (TD) and the coordinate approach, the former achieved a remarkable memory space reduction. They evaluated their approach using a tensor of size $408 \times 5473 \times 56$ user-service-time.

One of the main limitations in studies [1, 25, 26] was making predictions offline, which means once the models are trained, they are unable to deal with new incoming QoS data. To overcome this limitation, the study in [27] proposed an Incremental Tensor Factorization (ITF) method, the ITF is based on the incremental approach of Singular Value Decomposition (SVD) and Tucker Decomposition (TD). The new approach could update prediction when new QoS data arrives while preserving the scalability and space efficiency properties. It was evaluated on a tensor of size: 408 users and 5473 Web services at 240 time periods, and it achieved higher accuracy than the offline methods.

In [5], the authors used the Adaptive Matrix Factorization (AMF) method that made QoS prediction for candidate services in run-time service adaptation. A set of well-designed steps were followed to achieve the requirements of accuracy, efficiency, and robustness. The method performed matrix factorization for each

Table 2 Time-aware latent factors collaborative filtering

References	Pub. year	Pub. type	Name	Strengths	Limitations
[1]	2011	Symposium	WSPred	Provided the first real dataset for time-aware QoS data, increased the accuracy of prediction	Didn't model long time dependency between QoS, works offline only
[25]	2014	Conference	NNCP	Provided a real dataset for QoS data, achieved higher accuracy	Didn't model long time dependency between QoS values, works offline only
[26]	2014	Conference	TD	Provided a real dataset for QoS data, achieved reduction in memory storage, increased the prediction accuracy	TD needed extra time computation, didn't model long time dependency in QoS values, works offline only
[27]	2014	Conference	ITF	Achieved higher prediction accuracy than their offline model	Didn't accommodate new users or service, didn't model long time dependency between QoS
[5]	2017	Journal	AMF	Provided up-to-date predictions	Didn't model long time dependency between QoS
[28]	2016	Conference	THC	Combining the two approaches improved the accuracy of prediction	Extra time computation and memory usage, works offline only
[29]	2019	Journal	BNLFT	Increased the prediction accuracy by adding Linear Bias(LB) and multiplicative learning rule	LB enlarged the solution space and increased the training time, works offline only
[30]	2021	Conference	CTF	The first method that made attention to an outlier in the time-aware prediction, improved the prediction accuracy	Had efficiency problems compared to other baseline time-aware methods, works offline only
[31]	2014	Conference	TASR	Provided a good way of inferring the interaction between user, services, and preference according to time	Their dataset reflected the user feedback, not real QoS values, works offline only
[32]	2018	Journal	TMF	Modeled temporal dependency between QoS in different time intervals	Works offline only
[3]	2017	Conference	CARP	Alleviated scalability and sparsity problems by clustering, had online prediction	The prediction model may require periodical retraining to update with new data
[33]	2015	Journal	TBQP	Introduced a unified method that accounted for multi-QoS attributes in specific time and location	The high dimensional tensor required considerable memory storage and high time computation, works offline only
[34]	2019	Journal	QoSHTD	Improved QoS prediction by utilizing location as context data	High time computation because of double tensor decomposition and clustering, works offline only

Table 2 (continued)

References	Pub. year	Pub. type	Name	Strengths	Limitations
[35]	2015	Conference	HDOP	Modeling the interactions between context data (time and location) improved the prediction accuracy	The high dimensional tensor consumed considerable memory storage and high time consumption for prediction, works offline only

time slot, with the ability to learn online and to update its parameters using adaptive weights as new QoS data arrives or as new users and services come.

In [28], the authors used a hybrid method of both traditional neighborhood CF and latent factors in order to increase prediction accuracy. In the traditional neighborhood CF part, they used a service-based similarity measure that distinguished between static and temporal QoS attributes. In the latent factor part, they used CANDECOMP/PARAFAC (CP) decomposition on the user, service, and time tenor. The final prediction was a weighted addition of the two parts. In [29], the study used the CP factorization of a user-service-time tensor by applying non-negativity constraint on QoS data. The important contribution of this study was improving the prediction accuracy by several steps including a linear bias for both user, service, and time to model the temporal changes in data, using multiplicative learning rule for parameter optimization, and using of altering direction method in the training process.

In [30], authors provided an outlier resilient prediction method that used Cauchy loss for measuring the prediction errors. However, they extended their method by providing time-aware prediction by using CP factorization approach. Also, they added the non-negativity constraint on QoS data, which caused them to use the Multiplicative Updating (MU) algorithm to optimize the parameters.

In [31], the authors modeled the effect of temporal changes on service recommendation at three levels: users, services, and preferences. They used a latent factor decomposition that had a bias shifting for each one of the mentioned levels. They used the implicit feedback from users, which was collected on their own dataset. In [32], an adaptive matrix factorization approach was used to model the interactions between users and services in a specific time slot. The enhancement, in this approach, was the addition of temporal smoothing of the prediction, which accounted for the dependency between QoS in adjacent time slots. In [3], a model named CARP was proposed, the model can be used for offline and online predictions. The method used K-means clustering to cluster the invocation records, where each cluster represented a specific context and a cluster may contain a set of time slots. In order to alleviate the data sparsity problem, they aggregated invocation records from different time slots in the same cluster. Lastly, a matrix factorization approach was used to predict the final reliability value.

To improve the prediction accuracy, other studies incorporated context data like the location of users and services. Incorporating such context data to cluster users and services may help in alleviating the data sparsity problem. Moreover, it can help in improving the final prediction accuracy due to the implicit correlation between time and location that must be considered when making predictions. An example of these studies is the study in [33], where the authors created a tensor of multi-dimensions(user, service, time, location, and QoS property) and used a tensor decomposition method to predict missing QoS values. Another study is [34], which created local clusters of users and services based on location information, it performed a hierarchical tensor decomposition in two types of tensors: the location-based local tensors and the general global tensors. Finally, in [35], a unified and generalized approach was contributed. The approach created a tensor of five dimensions(user, service, time, location, and QoS property). It used tensor

decomposition to predict QoS. The prediction loss was minimized using iRPROP+ optimization method, which produced accurate prediction results.

4.2.2 Clustering and machine learning methods

Several studies have exploited clustering and machine learning approaches in QoS prediction. Clustering is usually used as a data pre-processing step to alleviate the scalability and data sparsity problems. It is not sufficient alone to perform QoS prediction, so other methods like linear regression and QoS averaging are merged with the approaches in this section. Below is a summary of these studies and Table 3 emphasizes the strengths and limitations of each study.

In [36], a method named CLUS was proposed, it predicted reliability attributes for ongoing services. The method performed a K-means clustering of invocation records into three steps: environmental variable (network load) clustering, user-specific clustering, and service-specific clustering. The final prediction was done by cluster-based computations that used the averaging of the reliability values. In addition, the authors used a linear regression model for making predictions.

In [37], the authors provided a novel method that first predicted the QoS at the current time by calculating the average of the historical QoS data in a pre-determined time interval, then a K-means clustering approach was used to make clusters of similar users and services. The authors used the average value of the resulting clusters to make user and service-based predictions, lastly, a linear weighted addition of the two predictions was used.

In [38], the authors proposed a method of two steps: first, it filled in missing QoS values in the historical QoS time slots. This was done by employing clustering to compute user and service similarity, the missing QoS was then calculated by averaging the weighted similarity for both users and services. Second, it predicted QoS in the current time slot by using the averaging of the calculated historical QoS data. The method in [39] generated temporal patterns that represented a series of user invocations for each service, after smoothing the pattern, a clustering approach was used to cluster the generated temporal patterns. The final prediction of missing QoS was done using a polynomial fitting function.

In [40], a novel approach called lasso was proposed, this method treats the QoS as a general regression problem. It used lasso regularization to overcome the sparsity of the QoS data. In addition, it used the location of users and services to improve prediction accuracy. This model also can accommodate newly incoming QoS and provide up-to-date predictions. In [41], a Weighted Support Vector Machine (WSVM) was used. This approach treated the problem of QoS prediction as a linear regression problem but in a high dimensional space. It used an exponential weighting function to give high weights for recent data. A sliding window approach was used to generate data for training.

4.2.3 Deep learning methods

To distinguish it from traditional machine learning methods, this section describes methods that use deep learning approaches, including neural networks and their

Table 3 Time-aware clustering and machine learning methods

References	Pub. year	Pub. type	Name	Strengths	Limitations
[36]	2014	Journal	CLUS	Alleviated scalability problem, accommodate with new QoS data	Require rebuilding of clusters, had compromises between accuracy and computation complexity
[37]	2016	Conference	TF-KMP	Clustering was more efficient in computing similarity for sparse data	The average value of historical data is not an accurate measure for forecasting QoS, works Offline
[38]	2019	Journal	TWQP	Clustering was more efficient in computing similarity for sparse data	Clustering of a large amount of historical data was time-consuming, predicting the current QoS using the average is not an accurate prediction, works offline
[39]	2016	Conference	TPP	The used pattern can capture the whole time dependency between invocations	Generating temporal pattern time series of historical data is time-consuming, works offline
[40]	2016	Journal	Lasso	Overcame sparsity by using lasso reg, improved accuracy by using location for selecting similar users, succeeded in modeling sudden changes in QoS	Although it provided prediction for new QoS data, it couldn't deal with newly coming users or new services
[41]	2016	Conference	WSVM	Achieved better accuracy than standard SVM and ARIMA	Couldn't model any user-side data, works offline

Table 4 Time-aware deep learning methods

References	Pub. year	Pub. type	Name	Strengths	Limitations
[4]	2018	Conference	PLMF	Accurate model since it incorporated updated QoS data	Used a static window size, suffered from vanishing gradient problem
[42]	2019	Conference	QF-RNN	Captured the long term dependency of user preference	Works offline only
[43]	2020	Conference	RFM	PFM helped in capturing relationships from spare data and GRU solved the problem of vanishing gradient	Works offline only
[44]	2019	Journal	RTF	Overcame the vanishing gradient of LSTM methods	Works offline only
[45]	2019	Journal	STCA-1/2	Improved prediction accuracy by understating the correlation between location and time, had better interpretability	Works offline only
[46]	2021	Journal	QSPC	Can predict multi-attributes, can work with newly incoming data	LSTM layer has the problem of vanishing gradient, the window size is static
[47]	2019	Conference	MtforSRec	Improved prediction accuracy by using attention for important data	LSTM has vanishing gradient problem
[48]	2022	Journal	DeepTSQP	Achieved a higher prediction accuracy, can adapt its parameters dynamically to cope with data sparsity	Works offline only

derivations. Table 4 shows more details about the strengths and limitations of each study in this section.

In [4], the authors proposed a novel method called PLMF. The method improved the prediction accuracy by employing Long Short-Term Memory (LSTM), which is a type of Recurrent Neural Network (RNN). It performed online learning and continuously trained with newly coming QoS data by using a moving sliding window. The model used matrix factorization, where the latent factors of both users and services were learned using a personalized LSTM.

The study in [42], proposed a method that used a matrix called QI, which was generated from integrating invocation records with the QoS observation matrix. The method captured the user preferences and service features matrices by using matrix factorization. An LSTM was used to predict the QoS values at each time slice of 64 time intervals, from which, the top N Web services were recommended to the user. Despite that LSTM can model long-term dependency between QoS data, it has the problem of vanishing gradient, which may stop the learning process in the neural network. In order to overcome this limitation, the study in [43] proposed a method that used a Projected Factorization Machine (PFM) and Gated Recurrent Unit (GRU). The PFM was used to capture the non-linear interaction in a user, service, and time tensor, and the GRU was used to model the long-term dependency between sequential historical QoS records. A combination of the two predictions was adopted. A similar method was proposed in [44] which used Generalized Tensor Factorization (GTF) to model the static relationship between user, services, and time. Indeed, it used a Personalized Recurrent Gated Unit (PRGU) to model the long-term dependency. A maximum activation function was used to combine the two predictions.

Other studies utilized the ability of deep learning in inferring the complex relationships between different input features, they used neural networks to model the correlation between time and location as two important context data in the prediction. In the study [45] two methods named STCA-1 and STCA-2 were proposed. In these methods, the spatial and temporal features of services and users were extracted and entered into hierarchical neural networks. The networks were composed of multiple important layers, for example, an interaction layer was used to identify the first and second-order features. Attention layers were used to assign more weights to spatial features which made this model more interpretable than other models. In [46], a method named QSPC was introduced. It utilized two inputs: the request context and the temporal information. These inputs were fed to a multi-layer neural network. One of the important layers in this network was the LSTM layer, which captured the temporal information into a set of service requests using a static time window. The final output consisted of the prediction of multiple QoS attributes, in their case, response time and throughput. In [47], another method MtforSRec was proposed which accounted for static and dynamic QoS data. It used a factorization machine to model the static feature of QoS and a bi-directional LSTM to model the dynamic features. A softmax layer was used to give the final recommendations from the combined predictions. In [48], a method named DeepTSQP was proposed. It integrated features computed from the traditional similarity measures with binary features. For QoS prediction, it used the GRU model which helped in modeling the temporal

dependency and in mining the implicit features in user-service interactions. This method achieved good prediction accuracy compared with the methods covered in this review.

4.3 Time-aware hybrid collaborative filtering methods

Several recent studies combined the CF methods with other methods, such as time series models and their derivations. Usually, this hybridization is done to improve prediction accuracy, these studies can be summarized as follows, indeed Table 5 highlights the strengths and limitations of these studies.

In [49], the authors proposed a hybrid method that combined autoregressive integrated with moving average (ARIMA) model and traditional CF. ARIMA was used to generate a time series for each Web service, however, ARIMA can't correct itself timely by taking new observations as feedback. To overcome this limitation, KALMAN filtering was used. The authors employed CF to capture user side effects by using user-based similarity. Lastly, they added two predictions for the final output. In [50], a method was presented that also combined CF with ARIMA. The method first applied traditional CF to predict missing QoS for the past and current Point In Time (PIT). This method used two types of user similarities: global similarity with attenuation function, and user invocation similarity with edit distance measure. In the second step of prediction, the ARIMA method was used to forecast QoS for the future PIT. The final Web services recommendation was done using Multi-Criteria Decision Making (MCDM). In [51], the authors proposed a method that combined time series analysis with cloud model theory based on the CF approach to predict unknown QoS. The QoS data was transformed into a time series that represented different cloud models for different time periods. The similarity between models was measured using two novel methods namely, orientation and dimension similarity, which improved the final similarity computation. This method also used weights for every period using the fuzzy analytic hierarchy method.

5 Discussion

Each of the CF time-aware methods presented above possesses its own set of strengths and weaknesses. Neighborhood CF methods are generally characterized by their simplicity, making them easy to comprehend, implement, and increment new data. Moreover, they are more interpretable compared to other methods. In terms of their time-awareness capability, these methods have successfully captured temporal changes in user and service similarities using uncomplicated techniques like time decay and time intervals. However, their effectiveness is constrained to capturing temporal changes within limited time intervals and they cannot account for long-term time dependency between QoS values as a whole. To address this limitation, these methods have been combined with other approaches, such as time series methods [49] and [51], or they have employed tensor decomposition as demonstrated in [28]. Another significant challenge in this category is the scalability issue, wherein

Table 5 Time-aware time-aware hybrid methods

References	Pub.Year	Pub.Type	Name	Strengths	Limitations
[49]	2015	Conference	ARIMA+CF	Captured the temporal dependency of QoS invocation records for each service	Suffer from time-complexity in obtaining time series for each service, ignored service similarity, works offline
[50]	2018	Journal	taSR	CF is used to alleviate data sparsity which improved the accuracy of forecasting	Works offline only
[51]	2017	Journal	QP-VCM	Succeeded in identifying accurate neighbours users	Number of time periods was static, works offline only

computational complexity increases with the growing number of users and services over time.

The methods in latent factor model-based CF models provide better scalability along with better prediction accuracy. They modeled the time factor by embedding it in a tensor and employing tensor decomposition methods like TD or CP. Other simpler methods use matrix factorization in a specific time slot, however, the tensor structure still suffers from data sparsity and scalability problems [3]. However, these methods can only model the interactions between users and services in short-term (limited time) intervals and do not account for long-term dependency [44]. Additionally, to provide up-to-date predictions, they require re-training of the entire model to accommodate the new coming QoS data, which is considered an expensive process. Alternatively, they utilize online adaptive learning through methods like Stochastic Gradient Descent (SGD) [5].

The clustering model-based CF methods are recognized for their simplicity and efficiency. They have been employed to address scalability issues by reducing the search space, consequently decreasing computational complexity [19]. Simultaneously, this space reduction has also mitigated the sparsity problem, leading to improvements in final prediction accuracy [21]. However, it is known that the clustering method is less effective when dealing with noisy and outlier data, particularly in cases of highly fluctuating QoS values. To tackle this, data smoothing is necessary before applying clustering, as demonstrated in [39]. In terms of providing up-to-date predictions, these models face the challenge of decreasing prediction accuracy with the accumulation of new data. Consequently, clusters need to be rebuilt periodically to address this issue, as discussed in [21, 36], and [3].

However, with rapid evolution in deep learning methods; a lot of the previously mentioned problems have been mitigated. These methods can accommodate a large number of inputs (users, services, and their context data) without a noticeable decline in computation speed. Also, they succeeded in assembling different methods to gain additional advantages, for example, the PLMF method, in [4], has integrated matrix factorization which captures user, and service interactions with LSTM which captures long-term dependency. This integration has achieved better accuracy results compared with CARP, in [3], and CLUS, in [36]. Regarding their ability to update predictions, two of the approaches, discussed in this review: [4] and [46], have provided up-to-date predictions by using incremental training. They employed algorithms like Stochastic Gradient Descent (SGD) and Adam optimizer to dynamically update their parameters online. Nevertheless, a primary drawback of deep learning methods lies in their lack of interpretability, as they often overlook the reasoning aspect when inferring correlation among input features such as users, services, time, and location.

The hybrid methods, presented in this review, have integrated CF with times series methods, like ARIMA. Specifically, CF utilized the time series in two ways: first, by modeling long-term temporal dependency between QoS data, as done in [49] and [51], resulting in improved prediction accuracy. Second, by forecasting the future values of QoS, as in [50]. However, time series models cannot capture personalized features of users or services. Indeed, generating time series for each Web service is considered a costly and difficult process. For providing up-to-date

predictions, these hybrid models require novel methods for updating with new QoS data; however, none of the studies reviewed have contributed to solving this specific challenge.

In Figure 3, the distribution of studies across different types over time is presented. It is evident that despite their conventional nature, neighborhood methods (NH) continue to be prominent in the literature, particularly when integrated with other types. Likewise, latent factor (LF) methods have maintained consistent usage over the years, serving as fundamental techniques in model-based approaches. However, there has been a notable shift in the trend of QoS prediction towards the adoption of machine learning (ML), with increasing attention towards deep learning (DL) methods in recent years, as illustrated in the aforementioned figure. The reasons behind this shift are illustrated in the preceding discussion.

6 Research challenges and directions

In this section, we explore the challenges that face researchers in this field. Furthermore, we examine potential avenues for new research directions that researchers can seek in future work.

6.1 Research challenges

6.1.1 Data sparsity

In reality, a user usually invokes a limited number of services, so the QoS values of the un-invoked services remain unknown forming what is called the data sparsity

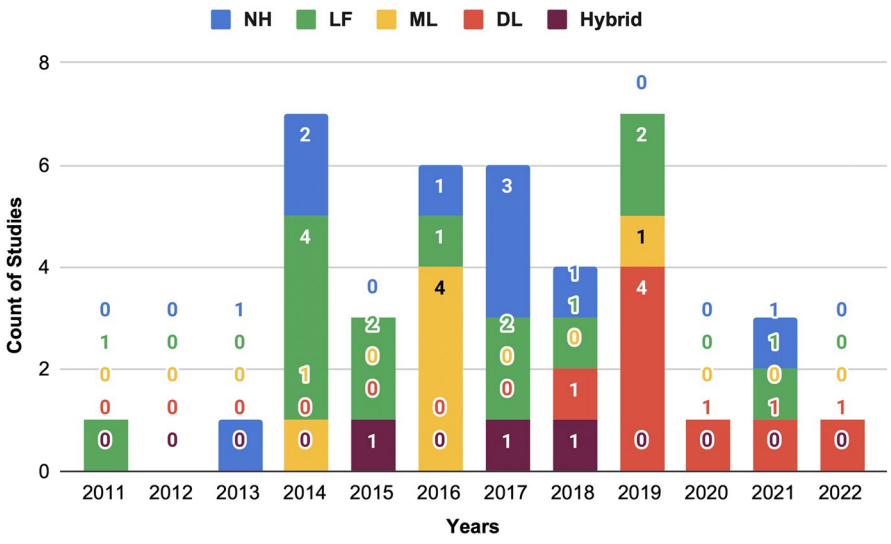

Fig. 3 Distribution of studies over years

problem. This problem becomes more critical when building time-aware methods since it will occur in multi-time slots during user-service interactions. Several studies, in the literature, came up with several sparse-tolerant solutions such as using random walk algorithm [16], using data aggregation [3], or using clustering [21]. However, this challenge is still unsolved and there is room for more innovative ideas to mitigate it.

6.1.2 Deficiency in incorporating other context data correctly

Time is one of the factors that affect prediction accuracy, however, other contextual factors such as the location of users or services, and environmental factors also play a role in prediction accuracy. The important point here is the understanding of the correlation between the time factor and other factors. This is considered a kind of context reasoning that can be inferred by observing and analyzing the historical QoS values in the datasets. Several studies' attempts can help in investigating datasets, on this issue, such as [24] and [52]. In fact, models must be built based on observations and evidence that would interpret context data correlation. This will help in generating true context-aware models that have high prediction accuracy.

6.1.3 The deficiency in providing up-to-date predictions

It is very important for time-aware to be updated continuously as new QoS data is coming. The majority of methods discussed in this review are offline methods (i.e. all QoS data are collected before the training phase). The accuracy of the offline methods deteriorates as time advances since they ignore new QoS observations that may carry changes in users, service similarities, or changes in context. Another important point, here, is that in a dynamic environment, the number of users and services also change over time. In reality, new users or services may appear, or current users and services may be disconnected. However, to address this challenge two solutions exist: first re-training the offline model periodically, re-training is required to accommodate new real-time QoS observations and new users or services. The limitation of this solution is the expensive time spent in re-training and testing the models as in [3, 21, 36]. Second building adaptive online models, these models can adapt to changes timely and can provide accurate up-to-date predictions. The premise of these models is that no need to train the whole model, however, there are limitations to this solution, for example, in online clustering models, there is always a trade-off between accuracy and scalability. Also, the online latent factor and deep learning models need special techniques that use moving sliding window and Adam or SGD optimizer to enable the online incremental training [4, 46]. However, this incremental training is a modern trend that needs further exploration of many issues such as computational complexity, resource consumption, stability, and maintainability.

6.2 Optional research directions

There are several research directions that researchers may work on in order to increase the accuracy of time-aware methods, from these we mention the following:

6.2.1 Creating generalised methods

Most of the research methods attempted to increase their accuracy concerning a limited number of known datasets commonly used in the experiments. However, this may result in creating data-biased methods that produce inaccurate results when they are evaluated on large-scale datasets [53]. Hence, there is room for enhancements here, for example: testing these methods using other different real datasets, applying them in real environments in the industry, or integrating them with real applications that need QoS prediction.

6.2.2 Creating unified methods

The majority of the current methods incorporated one or two contextual data, like being location-aware, time-aware, or both. The more contextual data used by the prediction method the higher accuracy it provides [54]. To this end, some methods are oriented toward building a unified framework, which can be extended to include new contextual data without changing the model's internal structure. In fact, this will release researchers from updating or creating models to support new types of contextual data. In these models, contextual data, like service semantic or load, environmental conditions, user-specific context, etc. can be combined into one unified model. In addition, these unified models may be extended to support multi-QoS factor predictions, such as predicting response time, throughput, and reliability at the same time, which is expected to increase prediction accuracy [35].

6.2.3 Creating new datasets

The majority of studies in this review utilized the WSDREAM dataset [24]. Although this dataset is a real dataset, it has several limitations. First, the used Web services are SOAP-based, so it would be helpful to include other recent types of Web service, such as Restful API, providing other types may bring other research challenges in QoS prediction for cloud, mobile, and IoT fields. Second, the size of this dataset is considered small, so creating a larger dataset is an important need to keep up with the huge increase in the number of Web services in the real world. Third, this dataset records QoS values such as response time and throughput independently in different datasets, this forms a limitation to research that attempts to conduct multi-predictions. Including QoS attributes in a synchronous manner will bring about new research issues.

6.2.4 Performing empirical studies

Performing empirical studies in the field of time-aware CF methods is considered an important need. However, until the time of writing this review, there are no empirical studies in this field. In fact, most of the studies, in this review, have deficiencies in selecting the baseline methods for comparison, they may compare their methods with non-time-aware methods or with a small number of time-aware methods. So an empirical study is needed to provide a clear picture of the performance of these methods at computational and prediction accuracy levels. Moreover, most of the studies, in this review, discussed the accuracy of their approach without reporting any information about their computational complexity. Researchers who are interested in this direction can benefit from empirical studies that have been conducted in the time series field, like [55], where the authors compared (23) methods and proved that Genetic Programming (GP) had better accuracy than ARIMA. Similarly, [56] compared time series methods with some machine learning methods. Another less comprehensive one is in [57] where authors compared less complicated time series methods. However, for CF, one may compare several well-known deep learning methods, several online methods, or any other combinations. Surely, this comparison will help in selecting the right method either in academic or industrial fields.

7 Conclusion

This paper presents a comprehensive review of time-aware Web service QoS prediction using CF methods, encompassing a total of forty studies. The reviews are thematically classified into time-aware neighborhood CF, time-aware model-based CF, and time-aware hybrid approaches. Each classification is thoroughly analyzed, revealing research challenges such as data sparsity, inadequate exploration of crucial service prediction context data, and the absence of up-to-date service predictions. Identified limitations include the challenges in providing up-to-date predictions, high computation costs for offline re-training methods, limited details on computational complexity, interpretability issues, particularly in deep learning methods, and insufficient exploration of service and user context along with QoS factors. The review proposes potential research directions aimed at addressing these challenges and advancing the field.

Author Contributions Authors' contribution will be provide later or added to the paper

Funding Not applicable.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

1. Zhang Y, Zheng Z, Lyu MR (2011) Wspred: a time-aware personalized qos prediction framework for web services. In: 2011 IEEE 22nd International Symposium on Software Reliability Engineering, pp 210–219 . IEEE
2. Tong E, Niu W, Liu J (2021) A missing qos prediction approach via time-aware collaborative filtering. IEEE Trans Serv Comput 15(6):3115–3128
3. Zhu J, He P, Xie Q, Zheng Z, Lyu MR (2017) Carp: context-aware reliability prediction of black-box web services. In: 2017 IEEE International Conference on Web Services (ICWS), pp 17–24. IEEE
4. Xiong R, Wang J, Li Z, Li B, Hung PC (2018) Personalized lstm based matrix factorization for online qos prediction. In: 2018 IEEE International Conference on Web Services (ICWS), pp 34–41. IEEE
5. Zhu J, He P, Zheng Z, Lyu MR (2017) Online qos prediction for runtime service adaptation via adaptive matrix factorization. IEEE Trans Parallel Distrib Syst 28(10):2911–2924
6. Goldberg D, Nichols D, Oki BM, Terry D (1992) Using collaborative filtering to weave an information tapestry. Commun ACM 35(12):61–70
7. Shao L, Zhang J, Wei Y, Zhao J, Xie B, Mei H (2007) Personalized qos prediction forweb services via collaborative filtering. In: Ieee International Conference on Web Services (icws 2007), pp 439–446. IEEE
8. Zheng Z, Xiaoli L, Tang M, Xie F, Lyu MR (2020) Web service qos prediction via collaborative filtering: A survey. IEEE Trans Serv Comput 15(4):2455–2472
9. Ghafouri SH, Hashemi SM, Hung PC (2020) A survey on web service qos prediction methods. IEEE Trans Serv Comput 15(4):2439–2454
10. Mezni H, Fayala M (2018) Time-aware service recommendation: taxonomy, review, and challenges. Softw: Pract Exp 48(11):2080–2108
11. Puri AS, Bhonsle M (2015) A survey of web service recommendation techniques based on qos values. Int J Adv Res Comput Commun Eng 4(12)
12. Syu Y, Wang C-M (2021) Qos time series modeling and forecasting for web services: a comprehensive survey. IEEE Trans Netw Serv Manage 18(1):926–944
13. Vinagre J, Jorge AM, Gama J (2015) An overview on the exploitation of time in collaborative filtering. Wiley Interdis Rev: Data Min Knowl Disc 5(5):195–215
14. Campos PG, Díez F, Cantador I (2014) Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Model User-Adap Inter 24:67–119
15. Wu L, He X, Wang X, Zhang K, Wang M (2022) A survey on accuracy-oriented neural recommendation: from collaborative filtering to information-rich recommendation. IEEE Trans Knowl Data Eng 35(5):4425–4445
16. Hu Y, Peng Q, Hu X, Yang R (2014) Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering. IEEE Trans Serv Comput 8(5):782–794
17. Yin G, Cui X, Dong H, Dong Y (2013) Web service evaluation method based on time-aware collaborative filtering. In: International Conference on Intelligent Data Engineering and Automated Learning, pp 76–84. Springer
18. Yu C, Huang L (2014) Time-aware collaborative filtering for qos-based service recommendation. In: 2014 IEEE International Conference on Web Services, pp 265–272. IEEE
19. Yu C, Huang L (2016) A web service qos prediction approach based on time-and location-aware collaborative filtering. SOCA 10(2):135–149
20. Li J, Wang J, Sun Q, Zhou A (2017) Temporal influences-aware collaborative filtering for qos-based service recommendation. In: 2017 IEEE International Conference on Services Computing (SCC), pp 471–474. IEEE
21. Yu C, Huang L (2017) Clucf: a clustering cf algorithm to address data sparsity problem. SOCA 11(1):33–45
22. Meng S, Li Q, Chen S, Yu S, Qi L, Lin W, Xu X, Dou W (2018) Temporal-sparsity aware service recommendation method via hybrid collaborative filtering techniques. In: International Conference on Service-oriented Computing, pp 421–429. Springer
23. Fan X, Hu Y, Zheng Z, Wang Y, Brézillon P, Chen W (2017) Casr-tse: context-aware web services recommendation for modeling weighted temporal-spatial effectiveness. IEEE Trans Serv Comput 14(1):58–70

24. Zheng Z, Zhang Y, Lyu MR (2012) Investigating qos of real-world web services. IEEE Trans Serv Comput 7(1):32–39
25. Zhang W, Sun H, Liu X, Guo X (2014) Temporal qos-aware web service recommendation via non-negative tensor factorization. In: Proceedings of the 23rd International Conference on World Wide Web, pp 585–596
26. Zhang W, Sun H, Liu X, Guo, X (2014) Incorporating invocation time in predicting web service qos via triadic factorization. In: 2014 IEEE International Conference on Web Services, pp 145–152. IEEE
27. Zhang W, Sun H, Liu X (2014) An incremental tensor factorization approach for web service recommendation. In: 2014 IEEE International Conference on Data Mining Workshop, pp 346–351. IEEE
28. Meng S, Zhou Z, Huang T, Li D, Wang S, Fei F, Wang W, Dou W (2016) A temporal-aware hybrid collaborative recommendation method for cloud service. In: 2016 IEEE International Conference on Web Services (ICWS), pp 252–259. IEEE
29. Luo X, Wu H, Yuan H, Zhou M (2019) Temporal pattern-aware qos prediction via biased non-negative latent factorization of tensors. IEEE Trans Cybern 50(5):1798–1809
30. Ye F, Lin Z, Chen C, Zheng Z, Huang H (2021) Outlier-resilient web service qos prediction. In: Proceedings of the Web Conference 2021, pp 3099–3110
31. Tian G, Wang J, He K, Hung PC, Sun C (2014) Time-aware web service recommendations using implicit feedback. In: 2014 IEEE International Conference on Web Services, pp 273–280. IEEE
32. Li S, Wen J, Luo F, Ranzi G (2018) Time-aware qos prediction for cloud service recommendation based on matrix factorization. IEEE Access 6:77716–77724
33. You M, Xin X, Shangguang W, Jinglin L, Qibo S, Fangchun Y (2015) Qos evaluation for web service recommendation. China Commun 12(4):151–160
34. Cheng T, Wen J, Xiong Q, Zeng J, Zhou W, Cai X (2019) Personalized web service recommendation based on qos prediction and hierarchical tensor decomposition. IEEE Access 7:62221–62230
35. Ma Y, Wang S, Yang F, Chang RN (2015) Predicting qos values via multi-dimensional qos data for web service recommendations. In: 2015 IEEE International Conference on Web Services, pp 249–256. IEEE
36. Silic M, Delac G, Srbljic S (2014) Prediction of atomic web services reliability for qos-aware recommendation. IEEE Trans Serv Comput 8(3):425–438
37. Wu C, Qiu W, Wang X, Zheng Z, Yang X (2016) Time-aware and sparsity-tolerant qos prediction based on collaborative filtering. In: 2016 IEEE International Conference on Web Services (ICWS), pp 637–640. IEEE
38. Jin Y, Guo W, Zhang Y (2019) A time-aware dynamic service quality prediction approach for services. Tsinghua Sci Technol 25(2):227–238
39. Chen L, Ying H, Qiu Q, Wu J, Dong H, Bouguettaya A (2016) Temporal pattern based qos prediction. In: International Conference on Web Information Systems Engineering, pp 223–237. Springer
40. Wang X, Zhu J, Zheng Z, Song W, Shen Y, Lyu MR (2016) A spatial-temporal qos prediction approach for time-aware web service recommendation. ACM Trans Web (TWEB) 10(1):1–25
41. Kai D, Bin G, Kuang L (2016) A time-aware weighted-svm model for web service qos prediction. In: International Conference on Collaborative Computing: Networking, Applications and Worksharing, pp 302–311. Springer
42. Wu X, Fan Y, Zhang J, Lin H, Zhang J (2019) Qf-rnn: Qi-matrix factorization based rnn for time-aware service recommendation. In: 2019 IEEE International Conference on Services Computing (SCC), pp 202–209. IEEE
43. Zhou J, Guo X, Yin C (2020) Recurrent factorization machine with self-attention for time-aware service recommendation. In: 2020 6th International Conference on Big Data Computing and Communications (BIGCOM), pp 189–197. IEEE
44. Zhang Y, Yin C, Lu Z, Yan D, Qiu M, Tang Q (2019) Recurrent tensor factorization for time-aware service recommendation. Appl Soft Comput 85:105762
45. Zhou Q, Wu H, Yue K, Hsu C-H (2019) Spatio-temporal context-aware collaborative qos prediction. Future Gener Comput Syst 100:46–57
46. Li B, Ye C, Yu X, Zhou H, Huang C (2021) Qos prediction based on temporal information and request context. SOCA 15(3):231–244
47. Li M, Lu Q, Zhang M, Liang X (2019) A multi-task service recommendation model considering dynamic and static qos. In: 2019 IEEE Intl Conf on Parallel & Distributed Processing with

Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), pp 760–767. IEEE
48. Zou G, Li T, Jiang M, Hu S, Cao C, Zhang B, Gan Y, Chen Y (2022) Deeptsqp: temporal-aware service qos prediction via deep neural network and feature integration. Knowl-Based Syst 241:108062
49. Hu Y, Peng Q, Hu X, Yang R (2015) Web service recommendation based on time series forecasting and collaborative filtering. In: 2015 Ieee International Conference on Web Services, pp 233–240. IEEE
50. Ding S, Li Y, Wu D, Zhang Y, Yang S (2018) Time-aware cloud service recommendation using similarity-enhanced collaborative filtering and arima model. Decis Support Syst 107:103–115
51. Ma H, Zhu H, Hu Z, Tang W, Dong P (2017) Multi-valued collaborative qos prediction for cloud service via time series analysis. Future Gener Comput Syst 68:275–288
52. Syu Y, Wang CM (2019) An empirical investigation of real-world qos of web services. In: International Conference on Services Computing, pp 48–65. Springer
53. Chen Z, Sun Y, You D, Li F, Shen L (2020) An accurate and efficient web service qos prediction model with wide-range awareness. Future Gener Comput Syst 109:275–292
54. Shen L, Pan M, Liu L, You D, Li F, Chen Z (2020) Contexts enhance accuracy: on modeling context aware deep factorization machine for web api qos prediction. IEEE Access 8:165551–165569
55. Syu Y, Kuo J-Y, Fanjiang Y-Y (2017) Time series forecasting for dynamic quality of web services: an empirical study. J Syst Softw 134:279–303
56. Hussain W, Hussain FK, Saberi M, Hussain OK, Chang E (2018) Comparing time series with machine learning-based prediction approaches for violation management in cloud slas. Future Gener Comput Syst 89:464–477
57. Cavallo B, Di Penta M, Canfora G (2010) An empirical comparison of methods to support qos-aware service selection. In: Proceedings of the 2nd International Workshop on Principles of Engineering Service-Oriented Systems, pp 64–70