Web Service QoS Prediction via Collaborative Filtering: A Survey

Zibin Zheng, *Senior Member, IEEE*, Xiaoli Li, *Student Member, IEEE*, Mingdong Tang, *Member, IEEE*, Fenfang Xie, *Student Member, IEEE*, and Michael R. Lyu, *Fellow, IEEE*

Abstract—With the growing number of competing Web services that provide similar functionality, Quality-of-Service (QoS) prediction is becoming increasingly important for various QoS-aware approaches of Web services. Collaborative filtering (CF), which is among the most successful personalized prediction techniques for recommender systems, has been widely applied to Web service QoS prediction. In addition to using conventional CF techniques, a number of studies extend the CF approach by incorporating additional information about services and users, such as location, time, and other contextual information from the service invocations. There are also some studies that address other challenges in QoS prediction, such as adaptability, credibility, privacy preservation, and so on. In this survey, we summarize and analyze the state-of-the-art CF QoS prediction approaches of Web services and discuss their features and differences. We also present several Web service QoS datasets that have been used as benchmarks for evaluating the predition accuracy and outline some possible future research directions.

Index Terms—Web service, QoS, prediction, collaborative filtering

1 Introduction

Web services are self-contained reusable Web components designed to support machine-to-machine interactions by programmatic method calls [1]. *Programmableweb.com* reports that there are 20,525 public Web services available on the Web. In addition, the development of cloud computing and mobile computing further accelerates the availability of Web services [2]. Unsurprisingly, many of this huge number of Web services offer similar functionality to users. Among those Web services of similar functionality, Quality-of-Service (QoS), which describes the services' non-functional characteristics, is recognized as an important criterion to differentiate between them.

A number of QoS-aware approaches of Web services have been proposed, such as service recommendation [3], service selection [4], service composition [5], service discovery [6], and so on. Fig. 1 is an example illustrating QoS-aware service selection. The service composition process **S_com** is composed of several abstract services (S_1 to S_5). Each abstract service can be implemented via a set of functionally equivalent concrete services $\{s_{i1}, s_{i2},..., s_{iN_i}\}$. The aim of QoS-aware service selection is to select appropriate services from each of these sets to optimize the composite service. Most previous QoS-aware service selection approaches assume that the QoS values of all concrete service candidates are user-independent and identical for different users. However, some QoS properties, such as response time, throughput, failure probability, and reliability, are user dependent. The response time is the time interval between a service user sending a request to a service and receiving the last byte of the corresponding response from the service, while throughput is defined as the number of successful messages passing through a communication channel per second. Failure probability is defined as the probability that a service user's invocation of a Web service will fail, while reliability refers to the probability that a service will run without failure in a specific environment for a specific time. These properties are closely related to the unpredictability of Internet connections and the heterogeneity of user environments. They may vary significantly from user to user because of dynamic network conditions. In such an environment, Web service evaluation on the client side is likely to obtain more accurate and personalized QoS values for the demanded Web services than a server-side evaluation.

However, it may be too time-consuming or even impractical to acquire QoS values by evaluating all service candidates on the client side because of time or cost constraints. In addition, because QoS performance is quite susceptible to unpredictable Internet connections and heterogeneous user environments, the QoS values of a Web service are unlikely to remain stable continuously. Based on these facts, accurate and personalized Web service QoS prediction becomes a necessity for QoS-aware approaches of Web services.

Collaborative filtering (CF), which is one of the most successful prediction techniques for recommender systems, has been widely applied to Web service QoS prediction. CF-based QoS prediction approaches can make personalized

- *Z. Zheng, X. Li, and F. Xie are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong 510000, China. E-mail: zhzibin@mail.sysu.edu.cn, {lixli27, xieff5}@mail2.sysu.edu.cn.*
- *M. Tang is with the School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, Guangdong 510000, China. E-mail: tangmingdong@gmail.com.*
- *M. R. Lyu is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, China. E-mail: lyu@cse.cuhk.edu.hk.*

Manuscript received 14 July 2019; revised 13 April 2020; accepted 16 May 2020. Date of publication 18 May 2020; date of current version 8 August 2022. (Corresponding author: Xiaoli Li.)
Digital Object Identifier no. 10.1109/TSC.2020.2995571

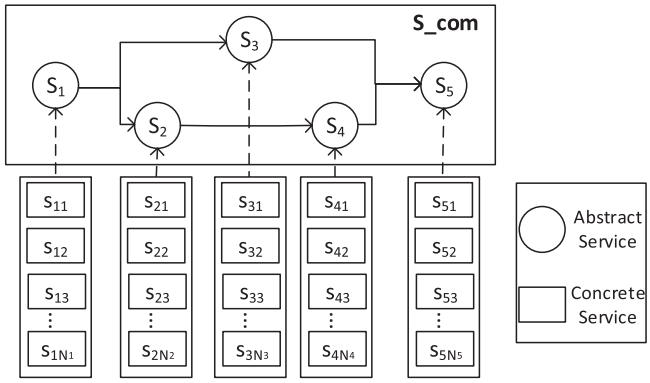

Fig. 1. QoS-aware service selection.

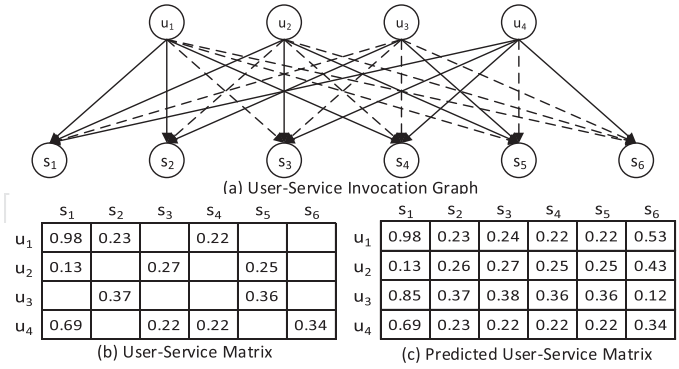

Fig. 3. A motivating example of QoS prediction.

QoS values prediction for service users, because of their great success in modeling the characteristics of users and services. CF approaches can be roughly divided into memory-based, model-based, and hybrid approaches. The core assumption of memory-based CF is that users who have observed similar QoS values in the past are likely to observe common QoS values in the future, whereas model-based CF approaches are based on prediction models that have been trained using previous QoS values. Hybrid CF approaches are a combination of memory-based and model-based CF approaches. There are additional information beyond the QoS values that can be employed to improve the performance further. In this survey, we summarize the state-of-the-art CF-based QoS prediction approaches of Web services, as shown in Fig. 2, and discuss their features and differences. We also present several Web service QoS datasets that have been used for QoS prediction evaluation and outline some possible future research directions.

The remainder of this paper is organized as follows: Section 2 defines the Web service QoS prediction problem. Section 3 introduces the background to CF. Section 4 surveys memory-based CF QoS prediction approaches, including conventional and extended memory-based approaches, which incorporate context information to improve the QoS prediction. Section 5 surveys conventional and extended model-based CF QoS prediction approaches. Section 6 surveys hybrid CF QoS prediction approaches. Section 7 analyzes adaptive, credible, and privacy-preserving CF QoS prediction. Section 8 introduces some Web service QoS datasets. Section 9 discusses some future research directions. Finally, we conclude the survey in Section 10.

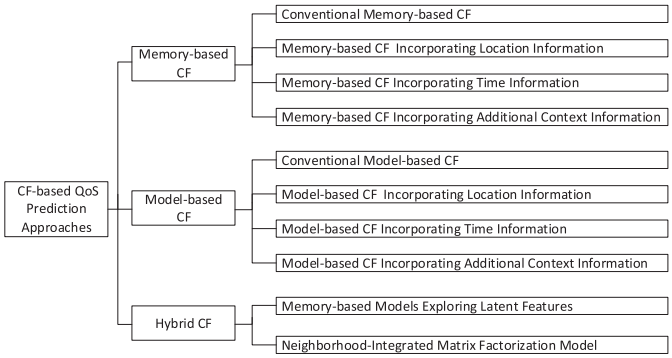

Fig. 2. Overview of CF-based QoS prediction approaches of Web services.

2 The Web Service QoS Prediction Problem

First, we consider the motivating example shown in Fig. 3a. Let $U = \{u_1, u_2, ..., u_4\}$ be the set of Web service users, $S = \{s_1, s_2, ..., s_6\}$ be the set of Web services, and E (solid lines) be the set of invocations between U and S. The edge e_{ij} between $u_i \in U$ and $s_j \in S$ indicates that user u_i has invoked Web service s_j previously. The weight w_{ij} on edge e_{ij} refers to the QoS value (e.g., response time) of that invocation. The user-service invocations can be represented by a matrix P, as shown in Fig. 3b. Each entry p_{ij} in P represents the QoS value of a Web service s_i observed by a service user u_i. The missing QoS values in P indicate that there are no invocations between the corresponding users and Web services. Under this setting, the Web service QoS prediction problem can be defined thus: given a user-service matrix P that represents known QoS values of Web services, predict the missing QoS values in P.

CF approaches have been widely applied to rating-based recommender systems for predicting missing ratings, and the basic assumption is that users who have expressed similar interests in the past will share common interests in the future [7]. The same phenomenon can be observed for Web service QoS prediction. If the QoS values observed by two users in the past are similar, it means that the network conditions of the two users may be similar, so the QoS values observed by the two users in the future will probably again be similar. Therefore, CF is also a popular approach for Web service QoS prediction. CF-based QoS prediction approaches of Web services have made great progress in the past few years, providing ever more automated solutions for cloud computing [8], multimedia services [9], e-commerce [10], and other application domains. Web service QoS prediction is not a trivial task and there are some significant challenges, as follows:

Sparsity: In practice, each user typically invokes only a few services at a time, leading to a limited number of service invocations. With limited training data, it is difficult to make accurate QoS predictions.

Scalability: The proliferation of available Web services and service users makes QoS prediction subject to serious scalability problems.

Objectivity: Web service QoS prediction should be distinguished from ratings prediction. QoS values are objective values perceived by users, whereas ratings are typically subjective values offered by users. QoS values are highly

dependent on the performance of the underlying network, which leads to differences in QoS scales for different users and different environments. In addition, ratings have only one-dimensional data, while QoS values are multidimensional, since Web services have several QoS properties and there are dependencies among these QoS data.

Considering these features of QoS, dozens of research works have attempted to revamp CF to obtain accurate Web service QoS predictions. This paper aims to provide a survey of these works.

3 Background to Collaborative Filtering

This section provide a brief introduction of CF, including memory-based, model-based, and context-aware CF approaches that incorporating additional information.

A typical memory-based CF approach can be described as a three-phase process: similarity computation, neighbor selection, and missing value prediction. The Pearson correlation coefficient (PCC) algorithm [11] and the vector-space similarity (VSS) algorithm [12] are often used for similarity computation. Those users or items with high similarity will be selected as similar neighbors of each other. Having identified the set of neighbors, the next step is to employ the information from similar neighbors to predict the unknown ratings in the user-item matrix. According to the similarity calculated for users or items, memory-based CF approaches can be divided into three categories: item-based approaches, user-based approaches, and their fusion in hybrid approaches. User-based approaches, such as GroupLens [13] and Bellcore video [14], predict the rating of an active user for an item by leveraging the ratings for this item by other users who have similar rating patterns. Ratings for similar users (neighbors) are those that are most correlated to the active user's ratings. Item-based approaches [15], [16], however, predict the rating of an active user for an item based on the ratings of items similar to those chosen by the active user. For such approaches, two items are similar if several users of the system have rated these items in a similar fashion. Because user-based approaches and item-based approaches address different aspects of the data, they may each ignore potentially valuable information. Hybrid approaches [17], [18] combine the prediction results of both approaches, aiming to fully utilize information from similar users and from similar Web services to improve accuracy.

In contrast to memory-based approaches, which use the stored ratings directly, model-based CF approaches usually employ these ratings to construct a predefined model with appropriate parameters. The model will have a certain ability to predict unknown ratings after the learning process and produces good estimations of an overall structure that relates simultaneously to all users and items. Model-based approaches are very popular and include Bayesian models [19], Latent Semantic Analysis [20], and clustering models [21]. Recently, matrix factorization (MF) techniques [22], [23] have attracted considerable attention because of their advantages with respect to scalability and accuracy, as witnessed by the algorithms developed within the Netflix contest[24]. MF models refer to a group of algorithms for which a user-item matrix is factorized into a product of the two latent factor matrices for users and items [25].

Many existing CF approaches deal only with two types of entities, namely users and items, and do not consider any contextual information such as time, location, and so on. However, it is likely to be insufficient to consider only users and items. During the past 10–15 years, many context-aware approaches have been developed. Melville *et al.* [26] proposed predicting the missing ratings in the user-item matrix by using the contextual information of items (e.g., categories such as title, genre, and so on). Moshfeghi *et al.* [27] improved item recommendation by incorporating additional contextual information, and Gunawardana *et al.* [28] incorporated item content features into a unified recommendation model. Rong *et al.* [29] leveraged the invocation frequencies of services and Zhang *et al.* [30] utilized the query histories of users to improve the performance of service recommendation.

4 QoS Prediction via Memory-Based CF

4.1 Conventional Memory-Based CF QoS Prediction

A number of approaches employ conventional memory-based CF to make QoS predictions. They use only the user-service QoS matrix and attempt to improve the prediction performance from the four aspects described below: (1) data preprocessing, (2) similarity computation, (3) similar neighbor selection, and (4) QoS-value prediction.

4.1.1 Data Preprocessing

Data preprocessing is an important step in improving the data quality, with a resultant improved performance for Web service QoS prediction.

Considering the Sparsity of QoS Data. A common solution to sparse QoS data is to fill the missing QoS values with default values [12], [15], such as the middle value of the QoS value range or the average user's or service's QoS value. To further improve the data quality, Wu *et al.* [31] clustered all users via the K-means algorithm and employed data-smoothing techniques to fill the missing QoS values. In an alternative approach, Zheng *et al.* [32] and Ma *et al.* [33] claimed that no prediction is better than a bad prediction and restricted the filling of missing QoS values to those having similar users or similar services.

Considering the Normalization of QoS. Ratings are usually integer numbers ranging from 0 to 5, but the various QoS properties can be of different types and may therefore vary across different ranges. For example, reliability is expressed as a ratio, with a range of 0 to 100 percent, whereas the response time is a real number with values likely to vary in the range $[0, 20\ seconds]$. Considering that the normalization of QoS would benefit CF accuracy, Zhang *et al.* [34] adopted the Gaussian normalization approach and 3-sigma rules to convert the QoS values of various types to the range $[0, 1]$.

4.1.2 Similarity Computation

In memory-based CF approaches, similarity plays a double role: filtering dissimilar neighbors (and obtaining similar neighbors for the target users or services) and weighting the importance of similar neighbors for collaborative prediction. Therefore, the similarity computation is one of the most important design decisions in CF, with a good metric often leading to good performance.

Considering the Differences in QoS Scale. Because of its objectivity, the QoS scale for different users is likely to be different. For example, perhaps caused by security needs or a gateway, a user may observe a response time longer than 3000 ms for all services, while another user with a faster network may observe a response time shorter than 200 ms for all services. Therefore, the QoS values of Web services are distributed discretely in different ranges. In such situations, the similarity between two services is impacted by other irrelevant issues instead of by the service itself. Considering this, Chen *et al.* [35] used A-cosine [36] to compute the cosine similarity between services and eliminated the impact of different QoS scales by subtracting the vector of average QoS values for the service. Chen *et al.* [37] designed a similarity model called JacMinMax, which introduces a ratio (MinMax) that represents the overall experiential difference between two Web services invoked by the same user. In this way, JacMinMax avoids the problem of inaccurately describing the similarity between services when they are actually similar but have very different QoS values.

Considering the Overall Experience Difference. Jiang *et al.* [38] found that the more popular services or those services having a more stable QoS from one user to another should contribute less to the user similarity measurement. Conversely, a service that provides a very different QoS for different users but a similar QoS for a user's u and a contributes significantly to the similarity between the user's u and a. The authors therefore proposed an improved PCC similarity measure by introducing the personalized influence of services. In addition, Ma *et al.* [39] found that if two users have a high similarity, then their similarity will fluctuate very little with their growing invocations of Web services. Conversely, if two users have only a low similarity, then their similarity would fluctuate notably with their growing invocations of Web services. Based on these characteristics, they proposed a highly accurate prediction algorithm (HAPA).

Considering the Significance of Weights. Considering sparsity, Zheng *et al.* [40] employed similarity weights to reduce the influence of a small number of similar yet co-invoked services or users. However, the similarity weight may reduce the popular services' similarity, because the number of users invoking the popular services is usually large. Likewise, the similarity among the users who invoke a large number of Web services will be reduced. To address this issue, Zheng *et al.* [41] developed a logistic function. When the number of co-invoked services or users is large, the logistic function is approximately 1 and therefore has little impact on similarity estimation. Tang *et al.* [42] introduced two preset thresholds for adjusting the similarity, mainly determined by the sparsity of the user-service matrix.

Other Considerations. The PCC does not properly handle the QoS style differences between vectors in different vector spaces, and cosine similarity only measures the angle between two vectors and neglects length differences between vectors. To overcome these shortcomings, Sun *et al.* [43] proposed a new similarity measure named normal recovery (NR), which unifies the similarities between scaled user vectors (or service vectors) in different multidimensional vector spaces. Fletcher *et al.* [44] incorporated the satisfaction with users' personalized preferences for nonfunctional attributes in the similarity computation. The aim of these QoS prediction approaches is to predict QoS values as accurately as possible. Zheng *et al.* [45] proposed a QoS-ranking prediction framework to predict the QoS rankings directly. This framework uses the Kendall rank correlation coefficient to evaluate user similarity by considering the number of inversions of service pairs that would be needed to transform one rank order into the other.

4.1.3 Similar Neighbor Selection

Before predicting the missing values, the neighborhoods that include sets of similar users or services need to be identified. Similar neighbor selection is an important step in accurate missing value prediction, because many dissimilar neighbors will decrease the prediction accuracy.

Top-N Filtering. In general, the Top-N users or services, which have larger similarity values than the remainder, will be selected as similar neighbors [46].

Negative Filtering. Traditional Top-N algorithms ignore those cases when the number of similar neighbors may be less than N. Zheng *et al.* [47] enhanced the Top-N algorithm by excluding those neighbors with negative PCC similarities.

Threshold Filtering. Ma *et al.* [33] used a preset threshold to allow only those users or services whose similarity exceeds the threshold to be considered. This threshold-based approach retains only the most significant neighbors and is more flexible than the Top-N algorithm, but an appropriate value for the threshold may be difficult to determine.

Other Filtering. Wu *et al.* [31] proposed a two-phase neighbor-selection strategy to accelerate neighbor selection. This strategy obtains the most similar user or service cluster and takes the users or services in the cluster as candidates for neighbor selection. By employing a process of neighbor preselection, the efficiency of neighbor selection is improved, as only users/services in similar clusters are considered as neighbor candidates. Considering unbalance in the data distribution, Fletcher *et al.* [48] proposed an unbalanced data distribution approach, whereby neighbor identification is achieved via sampling importance resampling. They divided the similar neighbors set equally into N splits and then randomly selected one of these splits to form a new set of neighbors.

4.1.4 QoS Value Prediction

After a set of neighbors has been computed for each user or service, the prediction of QoS values is normally made via these neighbors. User-based approaches employ the QoS values of similar users, and item-based approaches employ the QoS values of similar Web services to predict the QoS values. In combining the user-based and item-based approaches in a hybrid approach, Jiang *et al.* [38] focused on a parameter that determines the ratio of user-based prediction to item-based prediction. However, if the parameter is generated artificially without any prior knowledge, mistakes may occur. Zheng *et al.* [32] proposed a combination of confidence weights and this parameter to balance user-based prediction and item-based prediction automatically.

4.2 Memory-Based CF QoS Prediction Approaches Incorporating Location Information

QoS is highly dependent on the performance of the underlying network. If a user and an invoked service are located in

different networks that are distant from each other on the Internet, network performance is likely to be poor, caused by both data transfer delay and the limited bandwidth of links between different networks. In contrast, when the user and the Web service are located in the same network, high network performance is more likely. Therefore, the locations of users and services are crucial factors affecting QoS. By also considering the locations of the user, the problem of choosing inappropriate neighbors (who happen to have similar QoS experiences on a few Web services for the target user) can be avoided, thereby improving the accuracy of QoS prediction.

According to the method of representation used for the location, we can divide the memory-based CF QoS prediction approaches that incorporate location information into three categories: longitude and latitude coordinates, IP address, and autonomous system.

Longitude and Latitude Coordinates. Tang *et al.* [49] incorporated the geographic location information of users into a data-smoothing procedure. They computed the neighbors of users based on their longitude and latitude coordinates. In contrast to cluster-based data-smoothing techniques [31], [35], which need to be recalculated whenever QoS data in the user-service matrix changes, the location-based data-smoothing technique is more efficient. Besides, Wang *et al.* [50] proposed a distance-based enhanced Top-K selection strategy using the coordinates of latitude and longitude to select similar edge service server set in mobile edge computing. However, even if two users or services are nearby in terms of physical distance, if their computers are located in different networks, they may be very distant in terms of network distance. That is, physical-location neighbors do not necessarily belong to the same network.

IP Address. Chen *et al.* [51] proposed RegionKNN, which groups users into a set of regions according to their IP addresses. The QoS values of users in the same region can then be employed for QoS value prediction even if no similar users are found using historical Web service QoS experience. In this way, the data sparsity problem can be greatly relieved. In addition, instead of searching the entire set for similar users, the approach only needs to search those regions to which the target user belongs, thereby improving the time efficiency of the QoS prediction. However, measuring the distance between two users simply by comparing their IP addresses may not be accurate. IP prefixes (i.e., IP address blocks assigned to networks) are constantly divided into finer granularities because of the IPv4 address shortage and multihoming [52]. For example, addresses 4.67.68.0 to 4.68.247.255 belong to Canada, while 4.67.64.0 to 4.67.67.255 belong to Japan. Therefore, two IP addresses with similar values do not necessarily belong to the same network.

Autonomous System. An autonomous system (AS) is either a single network or a group of networks within the Internet that is controlled by a common network administrator on behalf of a single administrative entity (such as a university or a business enterprise). Each AS has a globally unique ID, called its autonomous system number (ASN) [53]. Tang *et al.* [53] represented the user's location and the Web service's location as triples $(IP_u, ASN_u, CountryID_u)$ and $(IP_s, ASN_s, CountryID_s)$, respectively. This method can obtain the network distance between users or services, because the ASN represents the network region. Mappings of IP to ASN and IP to country can be accomplished by using the dataset publicly available from Internet topology measurement projects such as the Skitter project [1] and the RouteViews project.[2] Data from both Skitter and RouteViews are free to access and are frequently updated, which ensures that the IP to ASN and IP to country mappings remain up-to-date.

4.3 Memory-Based CF QoS Prediction Approaches Incorporating Time Information

The QoS performance of Web services is highly correlated with service invocation time, because the service status (e.g., workload or number of clients) and the network environment (e.g., congestion) change over time. WS-DREAM3,[3] which was published by Zhang [54], includes real-world QoS evaluation results from 142 users and 4,532 Web services in 64 different time slots. For the response time dataset #31, when user #140 invokes service #4497 at time interval #41, the response time was 7.037. But when the same user invokes the same service at the different time interval #25, the response time became 0.307. That is, the first response time is almost 23 times greater than the second. In general, a longer timespan indicates a higher probability that the QoS value will have deviated from its original value. The time factor is therefore a very important factor in predicting QoS.

Time Intervals Set Model. The critical step in these approaches is computing the similarity between services or users. However, because users only invoke one service at a time, the QoS data is very sparse. To increase density, Yu *et al.* [55] used a tunable parameter d to partition the time intervals into several time-interval sets $\mathrm{t_D}$, making the similarity between services at a particular time interval equal to the average similarity between the two services at $\mathrm{t_D}$. Yu *et al.* [56] further increased the density of the user-service-time tensor by clustering all users and services into different groups, thereby converting the user-service matrix $\mathrm{M_{u,s}(t_k, d)}$ into the userCluster-service matrix $\mathrm{M_{uc,s}(t_k, d)}$ and the user-service-Cluster matrix $\mathrm{M_{u,sc}(t_k, d)}$. However, there are trade-offs between scalability and prediction performance in this cluster algorithm. To address this problem, they also proposed a location-aware cluster method, in which the user set and the service set are divided into many clusters according to their location information [57].

Temporal Decay Model. For the time intervals set models, it is difficult to determine a suitable time interval because the prediction accuracy fluctuates with the changing time intervals, whereas temporal decay models can adaptively reduce the similarity with an increased timespan. Hu *et al.* [58] proposed two intuitive principles. First, temporally closer QoS experiences from two users on the same Web service contribute more to the user similarity value. Second, the more recent QoS experiences from two users on the same Web service contribute more to the user similarity value. Based on these two principles, they integrated time information into the PCC similarity measurement by adopting an exponential decay function. In addition, they proposed using transitive

1. http://www.caida.org
2. http://www.routeviews.org
3. http://inpluslab.com/wsdream/

similarity to alleviate the data sparsity problem. However, Hu *et al.* [58] simply considered each QoS value equally and neglected the weighted-rating effect from different QoS values. Fan *et al.* [59] proposed the context-aware services recommendation method based on temporal–spatial effectiveness (CASR-TSE), which models the effect of the correlations between a user's spatial context and a service's spatial context on user-preference expansion. They then presented an enhanced temporal decay model by introducing the weighted rating effect into the traditional temporal decay model.

Time Series Models. The methods described above do not consider the impact of a time series. In practice, the QoS values of a user in a particular time interval will not only be affected by similar users' QoS values, but also by the QoS values in previous time intervals. Hu *et al.* [60] proposed considering the QoS as composed of two parts: the local QoS performance of each Web service and the personalized part of each user. They argued that the local QoS performance of Web services is highly volatile over time in the dynamic Internet environment. To capture the temporal dynamics of QoS, they utilized time series forecasting based on autoregressive integrated moving average (ARIMA) models to forecast future QoS values. Ding *et al.* [61] proposed a time-aware service recommendation (taSR) approach that integrates the time-aware similarity-enhanced CF and the ARIMA model. The taSR approach involves user-invocation similarity, which can identify not only the same invocation but also the same un-invocation. For example, two users are regarded as similar if they either invoke or do not invoke a service at a given time. ARIMA is then applied to predict the QoS values at a future point in time under QoS instantaneity.

4.4 Memory-Based CF QOS Prediction Approaches Incorporating Other Context Information

The algorithms outlined in Sections 4.2 and 4.3 exploit the spatial and temporal information associated with users or services to enable QoS prediction. In practice, the performance of the CF algorithm will be improved if more contextual information is incorporated.

Indeed, Web services hosted by the same provider exhibit similar characteristics and are more likely to be similar neighbors. Cao *et al.* [62] proposed a three-dimensional cube that can explicitly describe the relationship among providers, consumers, and Web services. Based on the cube data model, they presented standard-deviation-based hybrid collaborative filtering (SD-HCF), which considers the impact of service providers when selecting nearest neighbors for a Web service. Silic *et al.* [63] proposed LUCS, based on service load, user location, service class, and service location. According to each of these four LUCS parameters, four different availability predictions are obtained. LUCS then uses a linear combination of these four factors to calculate the final prediction of the expected service availability for a particular user-service pair. Chen *et al.* [64] considered the physical environment of Web services, and partitioned Web services into service clusters involving the same physical environment. As services in the same cluster have a very similar physical environment, they can be substituted for each other without impacting the overall QoS performance and can even be treated as a single service. This enables the calculation of similarity between users to be based on these service clusters instead of on individual services.

TABLE 1
Context Information Used by Memory-Based QoS Prediction Approaches

Methods	Spatial		Temporal	Others
	User	Service		
RegionKNN [51]	✓	×	×	×
COFILL[49]	✓	✓	×	×
LACF [53]	✓	✓	×	×
YUTACF [55]	×	×	✓	×
CluCF [56]	×	×	✓	×
TLACF [57]	✓	✓	✓	×
TAWSRec + HRW [58]	×	×	✓	×
CASR-TSE [59]	✓	✓	✓	×
ARIMA [60]	×	×	✓	×
taSR [61]	×	×	✓	×
SD-HCF [62]	×	×	×	✓
LUCS [63]	✓	✓	×	✓
SCQP [64]	×	✓	×	✓

In Table 1, we summarize the context information used by the various memory-based CF algorithms described in Sections 4.2, 4.3, and 4.4.

5 QoS Prediction Via Model-Based Collaborative Filtering

Memory-based CF algorithms are easy to implement and are highly effective, but they are greatly affected by sparsity of data and have other issues such as a cold start and poor scalability. The key step in memory-based CF approaches is to identify similar neighbors for each user or service by leveraging users' historic QoS values and context information. These approaches make good use of local information but can lose sight of the global structure. Because model-based CF algorithms employ all QoS values in the user-service matrix (global information) to construct a global model for making QoS value prediction, they produce good estimations of the overall structure that relates simultaneously to all users or services. In this section, we summarize the model-based QoS prediction approaches of Web services.

5.1 Conventional Model-Based CF QoS Prediction

A variety of conventional model-based CF approaches have been proposed for QoS prediction. For example, Luo *et al.* [65] proposed the kernel least-mean-square algorithm (KLMS), which analyzes the hidden relationships between all known QoS data and the corresponding QoS data with the highest similarities. It then applies the derived coefficients to the prediction of missing Web service QoS values. Wu *et al.* [66] proposed an embedding-based factorization machine approach, which embeds the user id and service id to vectors and employs a factorization machine to predict the QoS for users.

Latent factor models are used as the most popular model-based CF approaches to predict missing QoS values. The MF model, which can learn the latent factors of users and services, is by far the most powerful tool for undertaking the task of predicting missing QoS values.

By considering an $m \times n$ user-service matrix P, an MF model attempts to find two matrices: W (a users' latent

factor matrix of m rows and l columns) and H (a services' latent factor matrix of l rows and n columns) such that $P \approx W \times H$, where l is the number of factors. The matrices W and H are unknown and need to be estimated by using the available ratings in the user-item matrix P and identifying the optimal ratings by minimizing the distance between $W \times H$ and P. After obtaining the matrices W and H, the product of these two matrices can be employed to predict the missing QoS values in P.

Zheng *et al.* [41] constructed an objective function for MF-based QoS prediction:

$$\min_{W,H} \mathcal{L}(W,H) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{P}(P_{ij} - W_i H_j)^2 + \frac{\gamma}{2}\|W\|_F^2 + \frac{\gamma}{2}\|H\|_F^2, \tag{1}$$

where I_{ij}^{P} indicates that if Web service j has been invoked by user i previously, W_i represents the user-specific coefficients of user i, and H_j represents the factor vector of Web service j. The parameter γ controls the extent of regularization for penalizing large values of W and H to avoid the overfitting problem, and $\|\bullet\|_F^2$ denotes the Frobenius norm. The QoS prediction problem is modeled as an optimization problem that minimizes the sum-squared-errors objective function with quadratic regularization terms.

Considering that the given QoS data in QoS prediction are all positive values, Luo *et al.* [67] proposed the nonnegative latent factor (NLF) model by training the relevant features subject to nonnegativity constraints. Most MF-based models adopt either the gradient descent (GD) or the stochastic GD (SGD) method to find a local minimum of the objective function. To accelerate the model convergence, Luo *et al.* [68] introduced the principle of the alternating direction method (ADM) into an alternating-least-squares-based training process.

5.2 Model-Based CF QoS Prediction Approaches Incorporating Location Information

Most of the conventional model-based CF QoS prediction approaches can be extended to consider location information. For example, Hu *et al.* [69] combined Bayesian inference with user location and Chen *et al.* [70] extended Wu *et al.*'s approach [66] to embed the user id, service id, service location, and user location information in vectors. Tang *et al.* [71] revamped the classic factorization machine model by incorporating the locations of service users. Yang *et al.* [72] proposed a location-based factorization machine (LBFM) model that leverages the location information of users and services to predict the unknown QoS values. Zhou *et al.* [73] proposed a multilayered neural network model that represented all spatial features of users and services as input vectors and established a spatial features interaction layer for capturing the second-order spatial features. However, MF has drawn the most attention from researchers. To incorporate loction information, the MF model must complete two tasks. First, similarities are refined to incorporate location information, after which similar neighbors can be selected. Second, the QoS values of similar neighbors are integrated into the MF model.

5.2.1 Location-Aware Similar Neighbor Selection

To select neighbors, Lo *et al.* [74] computed the distance between each pair of users based on their longitude and latitude coordinates. Yin *et al.* [75] proposed using two-level neighborhood selection, which first employs the traditional PCC method to calculate the user similarity and then selects high-quality neighbors based on the set of local users generated at the first level, as proposed by Lo *et al.* [74]. These methods incorporate users' location information in a simple way, whereas He *et al.* [76] used K-means to cluster both user nodes and service nodes into different groups based on their longitude and latitude information to build robust neighborhoods. To alleviate the sparsity problem, Lee *et al.* [77] first grouped the users and services based on their location information, then adopted preference propagation, which models a bipartite graph between users and items, and constructed a random walk on this graph to infer more data for the similarity computation.

5.2.2 Revamped Objective Function

After obtaining the neighborhood information of users or services, MF-based QoS prediction approaches incorporating location information integrate information about the neighbors of users or services into the traditional MF model. The revamped MF models have extended objective functions of two types: location-based error functions and location-based regularization terms.

Location-Based Error Functions. A location-based error function integrates similar neighbors' information with the error function to revamp the objective function. Lo *et al.* [78] proposed LoNMF, a local neighborhood matrix factorization (NMF) QoS prediction application , which modifies the objective function as follows:

$$\min_{W,H} \mathcal{L}(W,H) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\Big(I_{ij}(P_{ij} - \alpha W_i^T H_j - (1-\alpha)\sum_{k\in N(i)} \omega_{ik} W_k^T H_j)^2\Big) + \frac{\gamma}{2}\|W\|_F^2 + \frac{\gamma}{2}\|H\|, \tag{2}$$

where $\mathcal{N}(u)$ is the set of similar neighbors of the user u, α is a balance parameter to control the impact of the users' neighbors. The relative weight coefficient ω_{ik} specifies the individual importance of the neighbor user u_k of user u_i. This error function is constructed to minimize the global difference within different neighborhoods. LoNMF only incorporates users' location information, whereas He *et al.* [76] proposed a hierarchical matrix factorization model (HMF) to make use of the location of both users and services. HMF first clusters several local user-service matrices and performs MF on these local matrices separately. The QoS values obtained by the local MF are then incorporated into the error function to modify the objective function.

Location-Based Regularization Terms. Location-based regularization terms integrate similar neighbors' information with the regularization terms to revamp the objective function. After generating a list of user-similarity neighborhoods, Yin *et al.* [75] constructed the diverse location-based regularization term:

$$\min \sum_{k \in \mathcal{N}(i)} Lo\_Sim(i,k) \|W_i - W_k\|_F^2, \tag{3}$$

where $Lo\_Sim(i,k)$ presents the local similarity between u_i and u_k as a monotonic decrease. If two users live close to each other, then $Lo\_Sim(i,k)$ would be greater and the influence would be more important. The regularization term is then added to revamp the MF model as follows:

$$\begin{aligned} \min_{W,H} \mathcal{L}(W,H) = & \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (P_{ij} - W_i^T H_j)^2 \\ & + \frac{\gamma}{2} \|W\|_F^2 + \frac{\gamma}{2} \|H\| \\ & + \frac{\alpha}{2} \sum_{i=1}^{m} \sum_{k \in \mathcal{N}(i)} Lo\_Sim(i,k) \|W_i - W_k\|_F^2, \end{aligned} \tag{4}$$

where $\alpha \succ 0$ is a factor controlling the involvement of the diverse location-based regularization. The constraint is constructed to penalize those neighbors with large differences. These methods incorporate users' location information in a simple way, whereas Lee *et al.* [77] converted similar group information into separate user-side and service-side regularization terms and built a unified framework by fusing the two regularization terms.

5.3 Model-Based CF QoS Prediction Approaches Incorporating Time Information

As discussed in Section 4.3, time is a very important factor in predicting QoS values. In this subsection, we will summarize state-of-the-art model-based CF QoS prediction approaches that incorporate time information.

Latent Factorization of Tensors. Zhang *et al.* [54] proposed WSPred, which extends the MF model to three dimensions through the use of an $m \times n \times c$ QoS tensor involving m users, n services, and c time intervals. By performing tensor factorization, user-specific, service-specific, and time-specific latent features can be extracted from the user-service-time-based three-dimensional matrix. In the real world, the Web service QoS values are always nonnegative, enabling Zhang *et al.* [79] to extend their work by presenting the temporal QoS value tensor as a nonnegative three-way tensor. They utilized the CANDECOMP/PARAFAC (CP) model [80] to represent the triadic relations among users, services, and time, and then added the nonnegativity restriction to the CP decomposition model to obtain a nonnegative CP decomposition model (NNCP). To address the fluctuations in QoS data with time, Luo *et al.* [81] proposed a biased nonnegative latent factorization of tensors (BNLFTs) model. BNLFTs models the linear bias (LB) for users, services, and time points, and takes the LB vectors with the same dimension to form rank-one tensors of LBs. It applies additive GD with respect to each single latent factor and LB and manipulates the learning rates to cancel the negative terms with the initial status of the corresponding parameter. To achieve a highly efficient ADM-based training scheme, it decomposes the original optimization task into multiple interdependent subtasks and solves each subtask sequentially. To obtain high prediction accuracy, Cheng *et al.* [82] proposed a QoS prediction approach based on hierarchical tensor decomposition (QoSHTD), which first introduces a local three-order tensor to model the clustering of local users, services, and the access time, and then adopts a good approximation of the Tucker decomposition method to decompose the local three-order tensor. The global tensor-decomposition model blends the local QoS-attribute predictive value into global tensor decomposition by using linear weighting, thereby obtaining high prediction accuracy. To exploit the structural relationships among the multidimensional QoS data, Wang *et al.* [83] presented a five-dimensional QoS data structure (i.e., location, QoS property, user, service, and time period) for Web services. They then proposed an integrated QoS prediction approach (HDOP), which unifies the modeling of the multidimensional QoS data via multilinear algebra based on the tensor concept. This enables accurate QoS prediction via tensor decomposition and reconstruction optimization algorithms.

Statistical Time Series Models. The latent factorization of tensors does not consider the impact of time series. In practice, the QoS values of a user in a specific time interval will not only be affected by similar users' QoS values but also by the QoS values for previous time intervals. Amin *et al.* [84] proposed a forecasting approach that integrates ARIMA with the generalized autoregressive conditional heteroscedastic (GARCH) model to forecast future QoS values for Web services. With the QoS values exhibiting mostly nonlinear behavior, Amin *et al.* [85] then proposed an automated forecasting approach combining linear and nonlinear time series models, in which ARIMA and the self-exciting threshold autoregressive moving average (SETARMA) are used as the nonlinear and linear models respectively. In an alternative approach, Li *et al.* [86] proposed a time-aware matrix factorization (TMF) model that provides two-phase QoS predictions. It first uses adaptive MF to predict missing QoS values and then employs the time series to smooth the predicted curve.

Neural Network-Based Models. Accompanying the vigorous development of deep learning technology, research work based on neural networks has been carried out recently in the field of QoS prediction. Wang *et al.* [87] used motifs-based dynamic Bayesian networks to represent the conditional dependency among time intervals and yield near-future time series predictions. They adopted the concept of motifs to describe the patterns of historical QoS value time series and then employed first-order Markov-chain rules to capture the causal relationships among different QoS value time series. These relationships are represented as conditional probability tables, which are used to make predictions based upon the updated QoS value time series. Wang *et al.* [88] extended their previous work [87] by developing a new method for QoS time series prediction based on long short-term memory (LSTM) [89], aiming to avoid the problem of long-term dependencies and the vanishing gradient when the length of sequences grows. Taking advantage of LSTM in capturing temporal dependencies, Xiong *et al.* [90] proposed a novel personalized LSTM-based matrix factorization approach (P-LSTM), which uses a user-side P-LSTM and a service-side P-LSTM to learn the users' and services' latent factor matrices for MF, respectively. These models only consider the temporal dependence of end-to-end paths, which makes it hard to describe the state of the whole network at various times. Considering this,

Zhou *et al.* [73] proposed a spatiotemporal context-aware collaborative multilayered neural network model, which characterizes each time slice by a latent feature vector to describe the state of the whole network at different times.

Other Models. Zhu *et al.* [91] argued that the time-dimensional characteristics can be typically captured by a finite set of context conditions, each of which is an abstract representation of the underlying factors such as service workloads and network conditions. Therefore, a specific context condition is likely to determine the QoS values for a specific time slice. Based on this observation, they proposed a context-aware model $r(u, s, c)$, where c denotes the specific context condition under which the invocations $inv(u, s, t)$ are performed. In particular, $r(u, s, t) \approx r(u, s, c)$. To characterize and identify different context conditions, they employed K-means clustering to cluster the QoS data R with T time slices into C clusters, where each cluster represents a specific context and different time slices grouped into one cluster belong to the same context. Wang *et al.* [92] assumed that the QoS prediction residual is a zero mean Laplace prior distribution and modeled the QoS prediction as a least absolute shrinkage and selection operator (Lasso) regression problem. Then, they employed the geolocation of users and Web services to detect neighbors for building the sparse representation, thus reducing the searching range while improving prediction accuracy.

5.4 Model-Based CF QoS Prediction Approaches Incorporating Additional Context Information

The model-based CF QoS prediction approaches described in Sections 5.2 and 5.3 only consider the impact of spatial and temporal information. In real service invocations, the QoS performance would also be greatly affected by additional contextual information. In this subsection, we will summarize and analyze state-of-the-art model-based CF QoS prediction approaches that incorporate this additional context information.

Xu *et al.* [93] claimed that the services provided by the same company are likely to share the same running environment and resources, such as network bandwidth, CPU performance, storage size, and so on. They proposed using a linear ensemble MF model (LE-MF) that combined the geographical information from the user side and the company affiliation from the service side. LE-MF uses the company affiliation (i.e., which company runs the service) to identify the context of the service when running. If a service has insufficient neighbors, it randomly selects a second set of neighbors for service j from each company that is in the same country as the company that runs service j.

Wu *et al.* [94] claimed that QoS values depend on the configuration of hosts, the status of servers, and the network conditions. They proposed a general context-sensitive matrix factorization approach (CSMF), which models the interactions of users to services and environment to environment simultaneously and makes full use of implicit and explicit contextual factors. In practice, a service will be invoked by different users, and those users having similar behavioral patterns can be grouped in terms of similar services. Service collaboration is used to model the users' behavioral interactions among services. By considering the context of service collaboration, Guo *et al.* [95] proposed a service-oriented tensor (SOT) model, which incorporates service collaboration from other similar services and relevant users by using a three-dimensional user-service-service tensor. They then employed CP decomposition to find an optimized rank approximation of the tensor.

These methods use only a few types of contextual information. However, to fully enhance the precision, multidimensional context data must be exploited as much as possible. Xiong *et al.* [96] integrated dynamic multidimensional context tracklets systematically and used the LSTM model to transform this complex multidimensional context into more useful feature expressions. They then extracted hidden features from the context and computed the similarity between users and services based on the hidden features. Wu *et al.* [97] proposed a multilayered neural network for making multiple QoS predictions with multiple contextual features. In the input layer, users, services, and all contextual factors are represented in a feature vector. Next, in the embedding layer, each feature is mapped into a dense vector to capture the implied semantics. The interaction layer then generates more useful cross features and reduces the model parameters by pooling operations on features. The perception layer learns the higher-order interactions between features. Finally, the task-specific layer separates the perception modules to provide distinct prediction tasks with corresponding feature selection and weighting functionality.

In Table 2, we summarize the context information used by the various model-based CF algorithms described in Sections 5.2, 5.3, and 5.4.

6 Hybrid CF QoS Prediction Approaches

Memory-based approaches utilize the local information of similar users or services in the user-service matrix to detect neighborhood relationships. But these approaches often ignore the vast majority of QoS values by a user, thus they are unable to capture the totality of weak signals encompassed in all of a user's QoS values. Model-based approaches construct a global model based on the observed QoS data, they are generally effective at estimating overall structure that relates simultaneously to most or all services. However, these approaches are poor at detecting strong associations among a set of closely related users or services. The memory-based and model-based approaches address quite different levels of structure in the data, so none of them is optimal on its own [98]. Hybrid CF approaches combine the memory-based and model-based approaches to solve the limitations of the aforementioned CF approaches and improve prediction performance.

6.1 Memory-Based Models Exploring Latent Features

Users that share common latent environmental factors would be expected to receive and deliver similar QoS values and can therefore be grouped together. The same would apply to services. Memory-based models that explore latent features first factorize the user-service matrix P into W and H. They then use W and H to identify similar neighbors through similarity computations. Finally, they predict the missing values based on the similar neighbors' QoS values.

TABLE 2
Context Information Used by Model-Based QoS Prediction Approaches

Methods	Spatial		Temporal	Others
	User	Service		
LBFM [72]	✓	✓	×	×
LBR [74]	✓	×	×	×
Colbar[75]	✓	×	×	×
HMF [76]	✓	✓	×	×
LMF-PP [77]	✓	✓	×	×
LoNMF [78]	✓	×	×	×
WSPred [54]	×	×	✓	×
NNCP [79]	×	×	✓	×
BNLFTs [81]	×	×	✓	×
QoSHTD [82]	✓	✓	✓	×
HDOP [83]	✓	×	✓	×
ARIMA+GARCH [84]	×	×	✓	×
ARIMA+SETARMA [85]	×	×	✓	×
TMF [86]	×	×	✓	×
LSTM [87]	×	×	✓	×
P-LSTM [90]	×	×	✓	×
STCA[73]	✓	✓	✓	×
CARP [91]	×	×	✓	×
LASSO [92]	✓	✓	✓	×
LE-MF [93]	✓	✓	×	✓
CSMF [94]	×	×	×	✓
SOT [95]	×	×	×	✓
LMDC [96]	✓	✓	✓	✓
DNM [97]	✓	✓	✓	✓

Zhang *et al.* [99] proposed CloudPred. They first utilized nonnegative NMF to factorize the sparse user-service matrix into the user latent factor matrix W and the service latent factor matrix H. Then they calculated the neighborhood similarities based on latent factor vectors and finally identified similar neighbors. Note that W and H are dense matrices with all entries available, enabling all missing values to be predicted. CloudPred utilizes both local information about similar users and global information about all available QoS values in the user-service matrix, thereby achieving better prediction accuracy in cases involving data sparsity. In an alternative approach, Yu *et al.* [100] proposed nonnegative matrix tri-factorization (NMTF) to factorize the QoS matrix P, which results in three matrices, W, H, and R, i.e., $P = W \times R \times H$. More specifically, $W \in \mathbb{R}^{n \times k}$ is the user-cluster indicator matrix, $H \in \mathbb{R}^{m \times l}$ is the service cluster indicator matrix, and $R \in \mathbb{R}^{k \times l}$ is the cluster-association matrix that captures the relationship between user clusters and service clusters. In this way, NMTF simultaneously clusters the m users into k disjoint user groups and n services into l disjoint service groups. The method is able to obtain similar neighbors for users and services and is therefore more effective.

6.2 Neighborhood-Integrated MF Models

Services that have similar historical QoS values tend to have similar factors that can influence the QoS values. Therefore, these services also tend to have similar latent features. Those users whose QoS values are similar also tend to encounter similar influencing factors that impact the QoS values for Web services. Unlike memory-based models that explore latent features, neighborhood-integrated MF models (NIMFs) first identify similar neighbors through similarity computations. They then utilize revamped MF models to predict the missing values based on the neighborhood information. There are two kinds of NIMF objective functions: neighborhood-integrated error objective functions and neighborhood-integrated regularization terms.

Neighborhood-Integrated Error Functions. The neighborhood-integrated error objective function integrates the relationship between similar neighbors to minimize both the global difference within different neighborhoods and the feature differences between each user or service and its neighbors. Zheng *et al.* [3] proposed a NIMF whose core idea is that whenever factorizing a QoS value, it will be treated as the ensemble of a user's information and the user's neighbors' information. Xu *et al.* [101] took this a step further by building a service neighborhood-based MF model (SN-MF) and a user neighborhood-based MF model (UN-MF). Multivariate linear regression was then used to combine the UN-MF and SN-MF.

Neighborhood-Integrated Regularization Terms. The neighborhood-integrated regularization term integrates a regularization term into the objective function. The term aims to minimize the latent difference between each user or service and its neighbors to facilitate personalized quality prediction. Lo *et al.* [102] proposed extending the MF framework with novel relational regularization terms for user regularization and service regularization, aiming to revamp the classic MF model into a unified framework. Zhang *et al.* [103] proposed a covering-based NMF (CNMF) to enable high-quality predictions. CNMF first employs a covering-based clustering algorithm to partition similar users and similar Web services into clusters. The clustering method does not require the number of clusters or the cluster centroids to be prespecified. It then uses users' and services' neighborhood information to perform user-integrated MF and service-integrated MF, respectively. Finally, to utilize the neighborhood information fully, CNMF combines the predicted values from the user-integrated MF and the service-integrated MF.

These hybrid CF approaches that combine the advantages of memory-based CF methods and model-based CF methods can obtain better prediction results, particularly for new users and new services. However, they rely on external information that is often not available and implementation complexity may be an issue.

In Table 3, we summarize and assess the main advantages and shortcomings of the memory-based, model-based, and hybrid CF approaches.

7 Recent Challenges

The approaches discussed above focus on how to improve the accuracy of prediction under the challenges of sparsity, scalability, and objectivity. However, in recent times, three additional challenges have emerged: (a) In a dynamic environment, existing QoS values will be continuously updated with newly observed values. (b) Some user-contributed QoS values could be untrustworthy, caused by malicious users submitting incorrect QoS values. (c) Because users' private information might be deduced from submitted QoS values, policies should exist to protect users' privacy.

TABLE 3
Main Advantages and Shortcomings of the Approaches

CF categories		Main advantages	Main shortcomings
Conventional	Memory-based	*better understanding of the reasoning *ease of adding new data incrementally	*limited scalability *low performance with sparse data
	Model-based	*higher prediction accuracy *better performance under the challenges of sparsity and scalability	*poor at detecting associations among a set of closely related users or services *expensive retraining with new data
	Hybrid	*higher prediction accuracy *better performance under the challenge of sparsity	*requirements for external information *the complexity of implementation increased
Incorporating Location Information	Memory-based	*higher prediction accuracy *better performance under the challenges of sparsity and scalability *better understanding of the reasoning	*limited scalability for large datasets *low performance with sparse data
	Model-based	*higher prediction accuracy *better performance under the challenge of sparsity	*the complexity of implementation increased *expensive retraining with new data
Incorporating Time Information	Memory-based	*better understanding of the reasoning *ease of adding new data incrementally	*limited scalability *low performance with sparse data
	Model-based	*better performance under the challenge of sparsity	*expensive retraining with new data
Incorporating Other Context Information	Memory-based	*higher prediction accuracy *better understanding of the reasoning	*limited scalability for large datasets *low performance with sparse data *requirements for external information
	Model-based	*higher prediction accuracy *better performance under the challenges of sparsity	*requirements for external information

In this section, we discuss approaches to these three challenges via adaptive CF QoS prediction, credible CF QoS prediction, and privacy-preserving CF QoS prediction, respectively.

7.1 Adaptive CF QoS Prediction Approaches

QoS values of Web services are dynamic in that existing QoS values are continuously updated with newly observed values. QoS prediction approaches should enable continuous and incremental updating using sequentially observed QoS data if they are to adapt to continuous QoS changes. To adapt to QoS fluctuations over time, adaptive QoS prediction studies have been proposed for each of the memory-based, model-based, and hybrid CF approaches.

Memory-based CF must update similar neighbors when a certain number of QoS data come in. However, the online time complexity for both user-based and item-based CF approaches is $O(mn)$, where m represents the number of users and n represents the number of services. This high computational complexity makes it difficult for memory-based CF algorithms to handle large amounts of performance data if timely prediction is required. Some extended memory-based approaches, such as those that employ clustering, are more efficient and well suited for large datasets. However, when the QoS values change, the corresponding user clusters and service clusters should be updated. Yu *et al.* [56] proposed an online updating approach, claiming that the clusters would be updated quickly by employing representative users and services. However, the cluster quality would decrease with an increase in the execution time for the online updating process. Therefore, the users or services should be reclassified whenever the quality of clusters significantly decreases.

The MF-based CF approaches need to train a factor model before making missing value predictions. Retraining the factor model with new data is quite expensive, particularly when retraining needs to be performed regularly. GD can be used to solve the minimization problem, but it typically works offline (with all data assembled) and cannot easily adapt to time-varying QoS values. Zhu *et al.* [104] proposed using adaptive matrix factorization, which employs SGD and adaptive weights to control the step size when training the model to perform well with rapidly changing QoS values for users and services. For example, if service $s1$ has an inaccuracy of 10 percent and service $s2$ has an inaccuracy of 1 percent, the step size should be less when updating service $s1$ than when updating service $s2$. The neural-network-based models [87], [88], [90] use a sliding window to deal with new data in a timely fashion. A long sliding window can help to incorporate more data in each update, but it may also result in an excessive training time. To update the prediction model more rapidly, Zhang *et al.* [105] proposed LA-LMRBF, a novel online QoS-forecasting approach that uses advertisement and the Levenberg-Marquardt (LM) improved radial basis function (RBF). LA-LMRBF first uses affinity propagation clustering to calculate the number of hidden-layer nodes of the RBF neural network and then employs the LM algorithm to learn the weights in the implicit layer and the output layer via iterative training. The model can produce prediction results with the new QoS values but, if the error function of new QoS values is larger than the threshold value, the originally trained model parameters may no longer meet the requirements of the new data. Therefore, it becomes necessary to use the improved LM algorithm again to calculate updated weights that meet the requirements of the dynamic prediction model.

Hybrid CF such as the neighborhood-integrated MF approaches [3], [102] need a series of experiments to find satisfactory parameters for use in QoS prediction. This is time-consuming and impractical in a dynamic environment. Luo *et al.* [106] designed an optimal online parameter-tuning method based on approximate dynamic programming that aims to find satisfactory parameters through online optimization. The tuner uses a neural network to compute the optimal solution to a multistage dynamic decision-making process. It can adapt to the changes in the network environment without requiring prior knowledge or identification of the prediction model. Considering that a change of any contextual factor will cause change of services' QoS, Liu *et al.* [107] proposed a context-aware real-time multi-QoS prediction method, which constructs a QoS case model combining the multi-QoS attributes and four contextual factors (i.e., task

type, task volume, network speed, and service workload). They first predicted the service workload by an optimized support vector machine, and then took the predicted workload and other contextual factors as input data to calculate the similarity between the target QoS case and a historical QoS case, finally predicted the multi-QoS of services.

The classical CF methods for adaptive QoS prediction are not incremental approaches and usually face heavy computational overheads. As discussed above, memory-based approaches can accommodate to QoS changes by simply storing the new data, however, it is difficult to handle large amounts of performance data in timely prediction. And retraining the factor model with new data is quite expensive in model-based approaches, especially if retraining needs to be performed regularly. A practical prediction system will be required to operate with rapidly changing QoS values. For example, an existing Web service may be discontinued by its provider or expire after a certain time in the absence of updating. However, these changes need a certain amount of information for them to be evaluated. During this time, the predicted QoS values may remain unchanged, and users may then assume that the prediction accuracy is low and may not agree to provide more QoS values. These approaches therefore have a trade-off between accuracy and the speed of the trend shifts. One possible approach is to evaluate the amount of information by measuring the difference between the predicted QoS values before and after the changes.

7.2 Credible CF QoS Prediction Approaches

Predictions are made based on the historical QoS values contributed by different users. Therefore, the prediction accuracy of CF approaches will be highly influenced by the trustworthiness of the user-contributed QoS values. However, existing CF prediction approaches are based on the hypothesis that all user-contributed values on services are trustworthy. In reality, some user-contributed QoS values can be untrustworthy for the following reasons: (1) malicious users may submit wrong values in their service QoS evaluation data; (2) some users may always give maximal/minimal values for their unevaluated services; (3) service users who are also service providers may give high QoS values for their services and low values to their competitors. Therefore, it is important to consider data credibility if robust Web service QoS value prediction is sought.

Registries that certify the QoS performance of all available Web services via third-party agents are used to improve the trustworthiness of the Web service QoS in Manikrao *et al.* [108] and Ran *et al.* [109]. However, this can lead to overload of the registry servers if there are large numbers of QoS feedback responses and the collection of massive amounts of real-time QoS data. To resolve the trustworthiness issue for QoS and to provide accurate prediction results, several approaches have been developed, including reputation-aware algorithms, feedback-based algorithms, and clustering-based algorithms.

Reputation-Aware Approaches. Reputation mechanisms can provide an incentive for honest feedback behavior and help users to make decisions about whom to trust. Qiu *et al.* [110] proposed a memory-based reputation-aware prediction approach that calculates the reputation of each user based on the difference of their contributed QoS values and the weighted average of other users' QoS values. Untrustworthy users with low reputation are then excluded. Xu *et al.* [111] proposed a reputation-aware MF-based CF approach in which the user reputation is integrated into the MF model. They applied a weighting parameter to reduce the influence of low-reputation users and gave more attention to the QoS values observed by users with a high reputation. However, these two reputation-aware algorithms are sensitive to their parameter settings, with inappropriate parameter values leading to inaccurate reputation evaluation and a resultant low prediction accuracy. To address this problem, Su *et al.* [112] presented a trust-aware QoS prediction approach that employs unsupervised K-means clustering and a beta-distribution-based method to calculate the reputation of users. They first employed the K-means clustering algorithm to cluster the QoS data provided by the different users and assigned the cluster containing the most users as the honest cluster. They then calculated the reputation of users by evaluating their deviation from the honest cluster.

Feedback-Based Approaches. Chen *et al.* [113] presented a feedback-based trust model, which uses prediction feedback to improve the performance of QoS prediction. First, they collected the feedback to the prediction results from similar neighbors, including the times similar neighbors were satisfied with the prediction results, the times similar neighbors were not satisfied with the prediction results, and the total number of similar neighbors making predictions. They then used a Bayesian-average voting algorithm to evaluate the trust in the user. Finally, they combined this user trust with the user similarity to generate the trust-based similarity. The trust model is effective and complementary to the memory-based CF prediction approaches, but it is sometimes difficult to collect feedback because users may not always give feedback on the prediction results.

Clustering-Based Approaches. Clustering-based approaches employ an unsupervised clustering algorithm to identify untrustworthy users based on the clustering information. Wu *et al.* [114] proposed a novel credibility-aware QoS prediction method that uses two-phase K-means clustering to identify the untrustworthy users. In the first phase, they clustered the QoS values for each service, and defined those users belonging to the cluster who had the minimum number of elements as candidates for being untrustworthy. In the second phase, they clustered all users based on their untrustworthy index and identified the cluster containing the users with the highest untrustworthy index as a set of untrustworthy users. To address the overloading problem for the QoS-certifier server caused by large-scale QoS data, Tao *et al.* [115] first collected QoS values using event-driven adaptive Poisson sampling and then adopted the partitioning around medoids clustering algorithm to obtain trustworthy QoS data. Liu *et al.* [116] divided trust into two main types, namely local trust and global trust. Being the degree of trust between two users, local trust is measured by averaging the prediction error on co-invoked services, whereas global trust can be computed by averaging the local trust values from neighbors. They then removed those users who did not satisfy a trust threshold to reconstruct the trusted network.

Reputation-aware approaches try to calculate the user reputation precisely, feedback-based approaches need to

collect QoS feedback, and clustering-based employ clustering algorithm to identify untrustworthy users. These credible CF QoS prediction approaches can detect some malicious users. However, if the attackers are aware of such detection strategies being applied, their attack could be modified to avoid detection. This flaw make these approaches vulnerable to sophisticated attacks. For example, some malicious users collude to propose QoS values that are significantly different from others, thereby allowing the QoS values proposed by another malicious user to be considered trustworthy. More research is needed in this area.

7.3 Privacy-Preserving CF QoS Prediction Approaches

To collect QoS values on the client side, users are required to supply their observed QoS values. However, there is a risk of the users' privacy leaking. Malicious recommender systems may abuse the data, infer private information from the data, or even resell the data to a competing user for profit [117]. An unintentional leakage of such data can expose users to a broad set of privacy issues (e.g., QoS data may reveal the underlying application configurations). Some researchers point out that QoS properties such as response time and availability are highly correlated with the users' physical locations [53], [118], which means that a user's location could be deduced from the QoS information. Consequently, it is essential to design privacy-preserving QoS prediction approaches. There are three main privacy-preserving techniques. First, randomized perturbation [119] adds randomness from a specific distribution to the original data to prevent information leakage. Next, homomorphic encryption [120] allows computations to be carried out directly on ciphertexts. Finally, differential privacy [121] can make the inference of private user data from the output difficult by adding random noise to every user's observed QoS data according to a Laplace mechanism. We summarize these three privacy-preserving QoS prediction approaches in the following.

Randomized Perturbation. Zhu *et al.* [122] presented a randomized-perturbation-based privacy-preserving QoS prediction framework. They first performed randomized perturbation on the QoS values via random noise generated from a specified distribution. They then applied memory-based CF and model-based CF to predict QoS values based on the obfuscated QoS values. Later, Zhu *et al.* [123] designed a similarity-maintaining privacy preservation (SPP) strategy from the user-level perspective (vector-level instead of element-level). With this SPP strategy, the similarities among the obfuscated data is the same as that among the true data. They then proposed a location-aware low-rank matrix factorization (LLMF) schema to predict the missing QoS values. However, LLMF only considers the services' location information, without utilizing the users' location information. Meng *et al.* [124] combined a region-aggregation strategy with a randomized data obfuscation technique, for which the region-aggregation strategy expands the target region, thereby blurring the specific location of users and protecting the users' location information. These randomized-perturbation-based approaches trade off privacy against accuracy because the accuracy of the prediction is inversely correlated with the magnitude of the noise. It is difficult to set an appropriate noise parameter that can maintain reasonable levels of user privacy and prediction accuracy at the same time. Moreover, Kargupta *et al.* [125] questioned the security of randomized perturbation technology.

Homomorphic Encryption. Badsha *et al.* [126] proposed a privacy-preserving QoS prediction framework via Yao's garbled circuit and homomorphic encryption. It first filters the nearby users by computing the encrypted distance information using the homomorphic property of the Paillier cryptosystem. Then, two service providers, a recommender server (RS) and a privacy server (PS), are incorporated to enable privacy-preserving QoS prediction. RS is responsible for generating recommendations and PS is responsible for generating keys and the decryption of the public key. First, PS generates public and private keys, with all users using the public key to encrypt their QoS values. This encrypted information is then stored in RS. After sending the query, RS predicts the missing QoS values based on the encrypted QoS values of nearby users by using the homomorphic properties of the public-key encryptions and sends the resultant ciphertexts to PS. Then, PS decrypts the prediction results. Having completed this protocol, the RS and PS retain no private information relating to any of the users. Homomorphic encryption-based methods can not only protect the private data of users but can also produce the same prediction results as non-private methods. However, this comes at the cost of computational overheads, making the method suitable mainly for offline prediction.

Differential Privacy. Liu *et al.* [131] proposed a privacy-preserving solution for Web service QoS prediction via differential privacy, which adds noise via a Laplace mechanism to ensure that the prediction result is insensitive to the removal or addition of any QoS record. They designed two methods, namely differential privacy on simple data (DPS) and differential privacy on aggregated data (DPA). DPS adds noise to the users' QoS data directly, while DPA adds noise to the aggregated user's QoS data to improve the utility of the disguised QoS data. In contrast to homomorphic encryption, this solution is not only lightweight but also offers theoretically guaranteed security because differential privacy follows a rigorous and quantitative definition of privacy leakage. However, the differential-privacy-based method also suffers from the privacy-accuracy trade-off issue and is assumed to involve a one-time computation; recalculation may lead to privacy leakage when more data is available. This is because maintaining privacy in multiple computations requires increasing the amount of introduced noise, which will lead to decreased accuracy.

Randomized-perturbation-based and differential-privacy-based approaches trade off privacy against accuracy, while homomorphic encryption-based methods come at the cost of computational overheads. These privacy-preserving research assume that complete privacy is unrealistic, and that privacy may only come at the expense of accuracy or some other trade-off such as computational overhead. It is important to examine this trade-off carefully and make compromises that minimize the invasion of privacy. Moreover, as the rigorous theoretical understanding of the degree of privacy protection is quite limited, an alternative is to define different privacy levels, such as the k-identity [132], and to analyze the sensitivity of algorithms under different privacy levels.

TABLE 4
Released Web Service QoS Datasets

Dataset	Number of users	Number of services
Shao[46]	136	20
Silic[127]	50	49
QWS [128]	1	2507
Vieira[129]	1	300
XiongLuo[65]	200	500
WS-DREAM1[47]	150	100
WS-DREAM2[130]	21,358	339
WS-DREAM3[54]	4,532	142

8 Real-World QoS Datasets

With various QoS prediction approaches of Web services having been comprehensively studied, a large-scale real-world Web service QoS dataset is needed to compare their prediction performance. Some approaches, such as that of Karta [133], simply employ a movie-rating dataset for experimental studies, which is insufficiently convincing. Shao *et al.* [46] collected invoking records from 136 consumers for 20 real Web services, where each volunteer submitted 200 invoking records for each service, on average. Silic *et al.* [127] implemented RESTful services involving various levels of computational complexity and placed these services in different geographic locations worldwide. Luo *et al.* [65] collected data that involved several properties such as response time and throughput from 200 users for 500 Web services. Al-Masri *et al.* [128] released a Web service QoS dataset named Quality of Web Service (QWS), which involved only one service user of 2,507 Web services. The fact that different users will observe quite different QoS for the same Web service limits the applicability of this dataset. Vieira *et al.* [129] conducted an experimental evaluation of security vulnerabilities for 300 publicly available Web services. However, security vulnerabilities usually exist at the server side and are user independent (different users will undergo the same security vulnerabilities on the target Web service). Zheng *et al.* [47] monitored 100 Web services by using 150 distributed computer nodes located all over the world. This dataset contains 150 files, where each file involves 10,000 Web service invocations on 100 Web services by a service user. Altogether, there are more than 1.5 million Web service invocations. Zheng *et al.* [130] also conducted evaluations on the real-world user-observed QoS for 5,825 Web services from distributed locations. In this dataset, the response time and the throughput performance are evaluated by 339 distributed-service users. Zhang *et al.* [54] included response time values and throughput values for 4,532 Web services invoked by 142 service users in 64 time intervals. These datasets can be employed in experiments designed to evaluate the prediction accuracy of different prediction approaches. Table 4 summarizes some of these available Web service QoS datasets.

9 Future Research Possibilities

Previous works show that CF approaches have gained substantial momentum for Web service QoS prediction. However, CF approaches are yet to offer satisfactory solutions under some conditions. In this section, we consider several promising directions for further research.

QoS Prediction for Other Emerging Services. The research of Web service QoS prediction should be extended to non-WSDL-described services, because a large proportion of modern Web services are non-WSDL-described, such as cloud-based services, mobile services, and Internet of Things (IoT) services. First, the popularity of cloud computing has promoted the rapid growth of cloud-based services. Millions of cloud-based services provide multiple real-time functions. Second, smart mobile devices such as smartphones, tablets, wearable devices, and autonomous vehicles are becoming more and more popular. In the era of mobile devices, millions of mobile services are downloadable from the app stores. Third, in an IoT environment, a huge number of heterogeneous devices have led to QoS concerns, and QoS approaches have been proposed in each layer of the IoT architecture and take into consideration different QoS properties. It is a critical challenge to predict the QoS values of such large-scale and highly dynamic services to fulfill user requirements.

Distributed QoS Prediction Approaches. A typical QoS prediction system collects the QoS data of its own users. This results in the historical QoS data distributed in different platforms. some of these platforms may not have enough user data to achieve high prediction accuracy. Due to data privacy, these platforms may be willing to but dares not share their data with other platforms. Although some researches have been proposed to solve the data privacy in a distributed situation [134], [135], the performance and privacy-preserving are still challenging in a distributed scene. On the other hand, the prediction is based on the contributed QoS data of users. However, there is currently no incentive mechanism for encouraging users to contribute. A fair incentive mechanism is required, whereby the more users contribute, the greater the reward they receive. Moreover, the authenticity of participants should be verifiable, because some participants may behave abnormally or even exhibit deliberately obscure or potentially adversarial behavior to maximize their financial interest. Designing blockchain-based QoS prediction methods, which could encourage users to participate by guaranteeing freedom from fraud, will be an important research direction for the future.

Novel Approaches to QoS Prediction. Although recent approaches achieve good performance to some extent, there is still much room to improve. Employing new technologies to further improve the prediction accuracy is one promising direction. For example, Graph Neural Network (GNN), as a new emerging recommendation model, has recently been used for recommender systems. GNN aggregates the feature information of neighbor nodes to obtain the feature information of the target node, and then capture the structure information of the whole graph through layer by layer fusion. Applying GNN to QoS prediction, the feature information of neighbor nodes and the structure information of the whole graph can be used for prediction, which will alleviate the problems of data sparsity and cold start. As another example, there are a variety of objects and rich relationships in the context-aware service network, such as spatial and temporal information, that naturally form a heterogeneous information network. The rich heterogeneous information can be

incorporated into QoS prediction to address the data sparsity and cold-start problem.

Case Studies of Industrial Implementations. QoS prediction plays an increasingly important role in service-based system development. But there is still a lack of research on the industrial implementations of QoS prediction. On the one hand, in real-world implementations, millions of services provide real-time functionality to millions of users. It is a difficult mission to predict the QoS values of such large-scale, highly dynamic services to meet the needs of users. On the other hand, due to user data privacy concerns, industrial companies may be reluctant to disclose information about how QoS prediction methods are used. However, it is very important to explain how the CF-based QoS prediction has then been used in the industry, which will increase the significance of this work. Case studies of industrial implementations may provide a promising direction, which needs urgent attention.

10 Conclusion

QoS prediction plays an increasingly important role in service-based system development. In this survey, we have presented a comprehensive review of CF-based QoS prediction approaches. We have summarized and analyzed conventional memory-based and model-based CF QoS prediction approaches and their extended versions that incorporate spatial, temporal, and other context information. In terms of hybrid CF QoS prediction approaches, we divided them into memory-based models exploring latent features and neighborhood-based MF models. We also summarized three new challenges for QoS prediction, namely adaptive CF, credible CF, and privacy-preserving CF. In addition, we described several Web service QoS datasets for QoS prediction evaluation. Finally, we suggested some future research directions for QoS prediction.

Acknowledgments

The work described in this article was supported by the National Key Research and Development Program (2016YFB1000101), the National Natural Science Foundation of China (61722214, 61976061), and the Pearl River S&T Nova Program of Guangzhou (201710010046).

References

[1] M. P. Papazoglou, "Service-oriented computing: Concepts, characteristics and directions," in *Proc. 4th Int. Conf. Web Inf. Syst. Eng.*, 2003, pp. 3–12.

[2] Q. Duan, Y. Yan, and A. V. Vasilakos, "A survey on service-oriented network virtualization toward convergence of networking and cloud computing," *IEEE Trans. Netw. Service Manage.*, vol. 9, no. 4, pp. 373–392, Dec. 2012.

[3] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative web service QoS prediction via neighborhood integrated matrix factorization," *IEEE Trans. Services Comput.*, vol. 6, no. 3, pp. 289–299, Third Quarter 2013.

[4] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for web services selection with end-to-end QoS constraints," *ACM Trans. Web*, vol. 1, no. 1, 2007, Art. no. 6.

[5] D. A. Menasce, "Composing web services: A QoS view," *IEEE Internet Comput.*, vol. 8, no. 6, pp. 88–90, Nov./Dec. 2004.

[6] R. Phalnikar and P. A. Khutade, "Survey of QoS based web service discovery," in *Proc. World Congress Inf. Commun. Technol.*, 2012, pp. 657–661.

[7] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[8] Z. ur Rehman, O. K. Hussain, F. K. Hussain, E. Chang, and T. S. Dillon, "User-side QoS forecasting and management of cloud services," *World Wide Web J.*, vol. 18, no. 6, pp. 1677–1716, 2015.

[9] V. Deora, J. Shao, W. A. Gray, and N. J. Fiddian, "A quality of service management framework based on user expectations," in *Proc. 1st Int. Conf. Service-Oriented Comput.*, 2003, pp. 104–114.

[10] Z. Zheng and M. R. Lyu, *QoS Management of Web Services*. Berlin, Germany: Springer, 2013.

[11] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. Supported Cooperative Work*, 1994, pp. 175–186.

[12] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. 14th Conf. Uncertainty Artif. Intell.*, 1998, pp. 43–52.

[13] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "Grouplens: Applying collaborative filtering to usenet news," *Commun. ACM*, vol. 40, no. 3, pp. 77–87, 1997.

[14] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in a virtual community of use," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 1995, pp. 194–201.

[15] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, 2004.

[16] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, no. 1, pp. 76–80, Jan./Feb. 2003.

[17] R. Bell, Y. Koren, and C. Volinsky, "Modeling relationships at multiple scales to improve accuracy of large recommender systems," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 95–104.

[18] J. Wang, A. P. De Vries, and M. J. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2006, pp. 501–508.

[19] J. Rong and S. Luo, "A bayesian approach toward active learning for collaborative filtering," in *Proc. Conf. Uncertainty Artif. Intell.*, 2004, pp. 278–285.

[20] T. Hofmann, "Collaborative filtering via gaussian probabilistic latent semantic analysis," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2003, pp. 259–266.

[21] S. Gong and H. Ye, "Joining user clustering and item based collaborative filtering in personalized recommendation services," in *Proc. Int. Conf. Ind. Inf. Syst.*, 2009, pp. 149–151.

[22] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 426–434.

[23] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.

[24] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[25] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 791–798.

[26] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Proc. 18th Nat. Conf. Artif. Intell. 14th Conf. Innovative Appl. Artif. Intell. Edmonton*, 2002, pp. 187–192.

[27] Y. Moshfeghi, D. Agarwal, B. Piwowarski, and J. M. Jose, "Movie recommender: Semantically enriched unified relevance model for rating prediction in collaborative filtering," in *Proc. Eur. Conf. Inf. Retrieval*, 2009, pp. 54–65.

[28] A. Gunawardana and C. Meek, "A unified approach to building hybrid recommender systems," in *Proc. 3rd ACM Conf. Recommender Syst.*, 2009, pp. 117–124.

[29] W. Rong, K. Liu, and L. Liang, "Personalized web service ranking via user group combining association rule," in *Proc. IEEE Int. Conf. Web Services*, 2009, pp. 445–452.

[30] Q. Zhang, C. Ding, and C. H. Chi, "Collaborative filtering based service ranking using invocation histories," in *Proc. IEEE Int. Conf. Web Services*, 2011, pp. 195–202.

[31] J. Wu, L. Chen, Y. Feng, Z. Zheng, M. C. Zhou, and Z. Wu, "Predicting quality of service for selection by neighborhood-based collaborative filtering," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 2, pp. 428–439, Mar. 2013.

[32] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware web service recommendation by collaborative filtering," *IEEE Trans. Services Comput.*, vol. 4, no. 2, pp. 140–152, Apr.-Jun. 2011.

[33] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2007, pp. 39–46.

[34] M. Zhang, X. Liu, R. Zhang, and H. Sun, "A web service recommendation approach based on QoS prediction using fuzzy clustering," in *Proc. IEEE 9th Int. Conf. Services Comput.*, 2012, pp. 138–145.

[35] L. Chen, Y. Feng, J. Wu, and Z. Zheng, "An enhanced QoS prediction approach for service selection," in *Proc. IEEE Int. Conf. Services Comput.*, 2011, pp. 727–728.

[36] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, 2001, pp. 285–295.

[37] Z. Chen, L. Shen, and F. Li, "Your neighbors are misunderstood: On modeling accurate similarity driven by data range to collaborative web service QoS prediction," *Future Gener. Comput. Syst.*, vol. 95, pp. 404–419, 2019.

[38] Y. Jiang, J. Liu, M. Tang, and X. F. Liu, "An effective web service recommendation method based on personalized collaborative filtering," in *Proc. IEEE Int. Conf. Web Services*, 2011, pp. 211–218.

[39] Y. Ma, S. Wang, P. C. Hung, C.-H. Hsu, Q. Sun, and F. Yang, "A highly accurate prediction algorithm for unknown web service QoS values," *IEEE Trans. Services Comput.*, vol. 9, no. 4, pp. 511–523, Jul./Aug. 2016.

[40] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Wsrec: A collaborative filtering based web service recommender system," in *Proc. IEEE Int. Conf. Web Services*, 2009, pp. 437–444.

[41] Z. Zheng and M. R. Lyu, "Personalized reliability prediction of web services," *ACM Trans. Softw. Eng. Methodol.*, vol. 22, no. 2, 2013, Art. no. 12.

[42] M. Tang, W. Liang, B. Cao, and X. Lin, "Predicting quality of cloud services for selection," *Int. J. Grid Distrib. Comput.*, vol. 8, no. 4, pp. 257–268, 2015.

[43] H. Sun, Z. Zheng, J. Chen, and M. R. Lyu, "Personalized web service recommendation via normal recovery collaborative filtering," *IEEE Trans. Services Comput.*, vol. 6, no. 4, pp. 573–579, Oct.-Dec. 2013.

[44] K. K. Fletcher and X. F. Liu, "A collaborative filtering method for personalized preference-based service recommendation," in *Proc. IEEE Int. Conf. Web Services*, 2015, pp. 400–407.

[45] Z. Zheng, X. Wu, Y. Zhang, M. R. Lyu, and J. Wang, "QoS ranking prediction for cloud services," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1213–1222, Jan. 2013.

[46] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction forweb services via collaborative filtering," in *Proc. IEEE Int. Conf. Web Services*, 2007, pp. 439–446.

[47] Z. Zheng and M. R. Lyu, "Collaborative reliability prediction of service-oriented systems," in *Proc. 32nd Int. Conf. Softw. Eng.*, 2010, pp. 35–44.

[48] W. Xiong, B. Li, L. He, M. Chen, and J. Chen, "Collaborative web service QoS prediction on unbalanced data distribution," in *Proc. IEEE Int. Conf. Web Services*, 2014, pp. 377–384.

[49] M. Tang, T. Zhang, J. Liu, and J. Chen, "Cloud service QoS prediction via exploiting collaborative filtering and location-based data smoothing," *Concurr. Comput., Pract. Exp.*, vol. 27, no. 18, pp. 5826–5839, 2015.

[50] S. Wang, Y. Zhao, L. Huang, J. Xu, and C. Hsu, "QoS prediction for service recommendations in mobile edge computing," *J. Parallel Distrib. Comput.*, vol. 127, pp. 134–144, 2019.

[51] X. Chen, X. Liu, Z. Huang, and H. Sun, "Regionknn: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation," in *Proc. IEEE Int. Conf. Web Services*, 2010, pp. 9–16.

[52] G. Huston, "BGP routing table analysis reports," 2011. [Online]. Available: http://bgp. potaroo. net/

[53] M. Tang, Y. Jiang, J. Liu, and X. F. Liu, "Location-aware collaborative filtering for QoS-based service recommendation," in *Proc. IEEE 19th Int. Conf. Web Services*, 2012, pp. 202–209.

[54] Y. Zhang, Z. Zheng, and M. R. Lyu, "Wspred: A time-aware personalized QoS prediction framework for web services," in *Proc. IEEE 22nd Int. Symp. Softw. Rel. Eng.*, 2011, pp. 210–219.

[55] C. Yu and L. Huang, "Time-aware collaborative filtering for QoS-based service recommendation," in *Proc. IEEE Int. Conf. Web Services)*, 2014, pp. 265–272.

[56] C. Yu and L. Huang, "CluCF: A clustering CF algorithm to address data sparsity problem," *Service Oriented Comput. Appl.*, vol. 11, no. 1, pp. 33–45, 2017.

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

[58] Y. Hu, Q. Peng, and X. Hu, "A time-aware and data sparsity tolerant approach for web service recommendation," in *Proc. IEEE Int. Conf. Web Services*, 2014, pp. 33–40.

[59] X. Fan, Y. Hu, Z. Zheng, Y. Wang, P. Brezillon, and W. Chen, "CASR-TSE: Context-aware web services recommendation for modeling weighted temporal-spatial effectiveness," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2017.2782793.

[60] Y. Hu, Q. Peng, X. Hu, and R. Yang, "Web service recommendation based on time series forecasting and collaborative filtering," in *Proc. IEEE Int. Conf. Web Services*, 2015, pp. 233–240.

[61] S. Ding, Y. Li, D. Wu, Y. Zhang, and S. Yang, "Time-aware cloud service recommendation using similarity-enhanced collaborative filtering and arima model," *Decis. Support Syst.*, vol. 107, pp. 103–115, 2018.

[62] J. Cao, Z. Wu, Y. Wang, and Y. Zhuang, "Hybrid collaborative filtering algorithm for bidirectional web service recommendation," *Knowl. Inf. Syst.*, vol. 36, no. 3, pp. 607–627, 2013.

[63] M. Silic, G. Delac, I. Krka, and S. Srbljic, "Scalable and accurate prediction of availability of atomic web services," *IEEE Trans. Services Comput.*, vol. 7, no. 2, pp. 252–264, Apr.-Jun. 2014.

[64] F. Chen, S. Yuan, and B. Mu, "User-QoS-based web service clustering for QoS prediction," in *Proc. IEEE Int. Conf. Web Services*, 2015, pp. 583–590.

[65] X. Luo, J. Liu, D. Zhang, and X. Chang, "A large-scale web QoS prediction scheme for the industrial internet of things based on a kernel machine learning algorithm," *Comput. Netw.*, vol. 101, pp. 81–89, 2016.

[66] Y. Wu, F. Xie, L. Chen, C. Chen, and Z. Zheng, "An embedding based factorization machine approach for web service qos prediction," in *Proc. Int. Conf. Service-Oriented Comput.*, 2017, pp. 272–286.

[67] X. Luo, M. Zhou, Y. Xia, Q. Zhu, A. C. Ammari, and A. Alabdulwahab, "Generating highly accurate predictions for missing QoS data via aggregating nonnegative latent factor models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 524–537, Mar. 2016.

[68] X. Luo, M. Zhou, Z. Wang, Y. Xia, and Q. Zhu, "An effective scheme for QoS estimation via alternating direction method-based matrix factorization," *IEEE Trans. Services Comput.*, vol. 12, no. 4, pp. 503–518, Jul./Aug. 2019.

[69] Y. Hu, X. Fan, R. Zhang, and W. Chen, "Context-aware web services recommendation based on user preference expansion," in *Proc. Asia-Pacific Services Comput. Conf.*, 2015, pp. 108–120.

[70] L. Chen, F. Xie, Z. Zheng, and Y. Wu, "Predicting quality of service via leveraging location information," *Complexity*, vol. 2019, pp. 4932030:1–4932030:16, 2019.

[71] M. Tang, W. Liang, Y. Yang, and J. Xie, "A factorization machine-based QoS prediction approach for mobile service selection," *IEEE Access*, vol. 7, pp. 32 961–32 970, 2019.

[72] Y. Yang, Z. Zheng, X. Niu, M. Tang, Y. Lu, and X. Liao, "A location-based factorization machine model for web service QoS prediction," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2018.2876532.

[73] Q. Zhou, H. Wu, K. Yue, and C. Hsu, "Spatio-temporal context-aware collaborative QoS prediction," *Future Gener. Comput. Syst.*, vol. 100, pp. 46–57, 2019.

[74] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu, "Collaborative web service QoS prediction with location-based regularization," in *Proc. IEEE 19th Int. Conf. Web services*, 2012, pp. 464–471.

[75] J. Yin, W. Lo, S. Deng, Y. Li, Z. Wu, and N. Xiong, "Colbar: A collaborative location-based regularization framework for QoS prediction," *Inf. Sci.*, vol. 265, pp. 68–84, 2014.

[76] P. He, J. Zhu, Z. Zheng, J. Xu, and M. R. Lyu, "Location-based hierarchical matrix factorization for web service recommendation," in *Proc. IEEE Int. Conf. Web Services*, 2014, pp. 297–304.

[77] K. Lee, J. Park, and J. Baik, "Location-based web service QoS prediction via preference propagation for improving cold start problem," in *Proc. IEEE Int. Conf. Web Services*, 2015, pp. 177–184.

[78] W. Lo, J. Yin, Y. Li, and Z. Wu, "Efficient web service QoS prediction using local neighborhood matrix factorization," *Eng. Appl. Artif. Intell.*, vol. 38, pp. 14–23, 2015.

[79] W. Zhang, H. Sun, X. Liu, and X. Guo, "Temporal QoS-aware web service recommendation via non-negative tensor factorization," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 585–596.
[80] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *Siam Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[81] X. Luo, H. Wu, H. Yuan, and M. Zhou, "Temporal pattern-aware QoS prediction via biased non-negative latent factorization of tensors," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1798–1809, May 2020.
[82] T. Cheng, J. Wen, Q. Xiong, J. Zeng, W. Zhou, and X. Cai, "Personalized web service recommendation based on QoS prediction and hierarchical tensor decomposition," *IEEE Access*, vol. 7, pp. 62 221–62 230, 2019.
[83] S. Wang, Y. Ma, B. Cheng, R. Chang, and R. N. Chang, "Multi-dimensional QoS prediction for service recommendations," *IEEE Trans. Services Comput.*, vol. 12, no. 1, pp. 47–57, Jan./Feb. 2019.
[84] A. Amin, A. Colman, and L. Grunske, "An approach to forecasting QoS attributes of web services based on ARIMA and garch models," in *Proc. IEEE 19th Int. Conf. Web Services*, 2012, pp. 74–81.
[85] A. Amin, L. Grunske, and A. Colman, "An automated approach to forecasting QoS attributes based on linear and non-linear time series modeling," in *Proc. 27th IEEE/ACM Int. Conf. Automated Softw. Eng.*, 2012, pp. 130–139.
[86] S. Li, J. Wen, F. Luo, and G. Ranzi, "Time-aware QoS prediction for cloud service recommendation based on matrix factorization," *IEEE Access*, vol. 6, pp. 77 716–77 724, 2018.
[87] H. Wang, L. Wang, Q. Yu, Z. Zheng, A. Bouguettaya, and M. R. Lyu, "Online reliability prediction via motifs-based dynamic bayesian networks for service-oriented systems," *IEEE Trans. Softw. Eng.*, vol. 43, no. 6, pp. 556–579, Jun. 2017.
[88] H. Wang, Z. Yang, and Q. Yu, "Online reliability prediction via long short term memory for service-oriented systems," in *Proc. IEEE Int. Conf. Web Services*, 2017, pp. 81–88.
[89] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[90] R. Xiong, J. Wang, Z. Li, B. Li, and P. C. Hung, "Personalized LSTM based matrix factorization for online QoS prediction," in *Proc. IEEE Int. Conf. Web Services*, 2018, pp. 34–41.
[91] J. Zhu, P. He, Q. Xie, Z. Zheng, and M. R. Lyu, "CARP: Context-aware reliability prediction of black-box web services," in *Proc. IEEE Int. Conf. Web Services*, 2017, pp. 17–24.
[92] X. Wang, J. Zhu, Z. Zheng, W. Song, Y. Shen, and M. R. Lyu, "A spatial-temporal QoS prediction approach for time-aware web service recommendation," *ACM Trans. Web*, vol. 10, no. 1, 2016, Art. no. 7.
[93] Y. Xu, J. Yin, S. Deng, N. N. Xiong, and J. Huang, "Context-aware QoS prediction for web service recommendation and selection," *Expert Syst. Appl.*, vol. 53, pp. 75–86, 2016.
[94] H. Wu, K. Yue, B. Li, B. Zhang, and C.-H. Hsu, "Collaborative QoS prediction with context-sensitive matrix factorization," *Future Gener. Comput. Syst.*, vol. 82, pp. 669–678, 2018.
[95] L. Guo, D. Mu, X. Cai, G. Tian, and F. Hao, "Personalized QoS prediction for service recommendation with a service-oriented tensor model," *IEEE Access*, vol. 7, pp. 55 721–55 731, 2019.
[96] W. Xiong, Z. Wu, B. Li, and Q. Gu, "A learning approach to QoS prediction via multi-dimensional context," in *Proc. IEEE Int. Conf. Web Services*, 2017, pp. 164–171.
[97] H. Wu, Z. Zhang, J. Luo, K. Yue, and C.-H. Hsu, "Multiple attributes QoS prediction via deep neural model with contexts," *IEEE Trans. Services Computi.*, to be published, doi: 10.1109/TSC.2018.2859986.
[98] R. M. Bell and Y. Koren, "Lessons from the netflix prize challenge," *ACM Sigkdd Explorations Newslett.*, vol. 9, no. 2, pp. 75–79, 2007.
[99] Y. Zhang, Z. Zheng, and M. R. Lyu, "Exploring latent features for memory-based QoS prediction in cloud computing," in *Proc. 30th IEEE Symp. Reliable Distrib. Syst.*, 2011, pp. 1–10.
[100] Q. Yu, Z. Zheng, and H. Wang, "Trace norm regularized matrix factorization for service recommendation," in *Proc. IEEE 20th Int. Conf. Web Services*, 2013, pp. 34–41.
[101] Y. Xu, J. Yin, Z. Wu, D. He, and Y. Tan, "Reliability prediction for service oriented system via matrix factorization in a collaborative way," in *Proc. IEEE 7th Int. Conf. Service-Oriented Comput. Appl.*, 2014, pp. 125–130.
[102] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu, "An extended matrix factorization approach for QoS prediction in service selection," in *Proc. IEEE 9th Int. Conf. Services Comput.*, 2012, pp. 162–169.
[103] Y. Zhang *et al.*, "Covering-based web service quality prediction via neighborhood-aware matrix factorization," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2019.2891517.
[104] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "Online QoS prediction for runtime service adaptation via adaptive matrix factorization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 10, pp. 2911–2924, Oct. 2017.
[105] P. Zhang, H. Jin, H. Dong, W. Song, and L. Wang, "LA-LMRBF: Online and long-term web service QoS forecasting," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2019.2901848.
[106] X. Luo, H. Luo, and X. Chang, "Online optimization of collaborative web service QoS prediction based on approximate dynamic programming," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 8, pp. 452–492, 2015.
[107] Z. Liu, Q. Z. Sheng, W. E. Zhang, D. Chu, and X. Xu, "Context-aware multi-QoS prediction for services in mobile edge computing," in *Proc. IEEE Int. Conf. Services Comput.*, 2019, pp. 72–79.
[108] U. S. Manikrao and T. Prabhakar, "Dynamic selection of web services with recommendation system," in *Proc. Int. Conf. Next Gener. Web Services Practices*, 2005, pp. 5–11.
[109] S. Ran, "A model for web services discovery with QoS," *ACM Sigecom Exchanges*, vol. 4, no. 1, pp. 1–10, 2003.
[110] W. Qiu, Z. Zheng, X. Wang, X. Yang, and M. R. Lyu, "Reputation-aware QoS value prediction of web services," in *Proc. IEEE Int. Conf. Services Comput.*, 2013, pp. 41–48.
[111] J. Xu, Z. Zheng, and M. R. Lyu, "Web service personalized quality of service prediction via reputation-based matrix factorization," *IEEE Trans. Rel.*, vol. 65, no. 1, pp. 28–37, Mar. 2016.
[112] K. Su, B. Xiao, B. Liu, H. Zhang, and Z. Zhang, "Tap: A personalized trust-aware QoS prediction approach for web service recommendation," *Knowl.-Based Syst.*, vol. 115, pp. 55–65, 2017.
[113] L. Chen, Y. Feng, and J. Wu, "Collaborative QoS prediction via feedback-based trust model," in *Proc. IEEE 6th Int. Conf. Service-Oriented Comput. Appl.*, 2013, pp. 206–213.
[114] C. Wu, W. Qiu, Z. Zheng, X. Wang, and X. Yang, "QoS prediction of web services based on two-phase k-means clustering," in *Proc. IEEE Int. Conf. Web Services*, 2015, pp. 161–168.
[115] Q. Tao, H.-Y. Chang, C.-Q. Gu, and Y. Yi, "A novel prediction approach for trustworthy QoS of web services," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3676–3681, 2012.
[116] J. Liu and Y. Chen, "A personalized clustering-based and reliable trust-aware QoS prediction approach for cloud service recommendation in cloud manufacturing," *Knowl.-Based Syst.*, vol. 174, pp. 43–56, 2019.
[117] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, and D. Boneh, "Privacy-preserving matrix factorization," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 801–812.
[118] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, "Personalized QoS-aware web service recommendation and visualization," *IEEE Trans. Services Comput.*, vol. 6, no. 1, pp. 35–47, First Quarter 2013.
[119] H. Polat and W. Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques," in *Proc. 3rd IEEE Int. Conf. Data Mining*, 2003, pp. 625–628.
[120] C. Gentry and D. Boneh, *A Fully Homomorphic Encryption Scheme*. vol. 20, Stanford, CA, USA: Stanford Univ. Stanford, 2009.
[121] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptography Conf.*, 2006, pp. 265–284.
[122] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "A privacy-preserving QoS prediction framework for web service recommendation," in *Proc. IEEE Int. Conf. Web Services*, 2015, pp. 241–248.
[123] X. Zhu *et al.*, "Similarity-maintaining privacy preservation and location-aware low-rank matrix factorization for QoS prediction based web service recommendation," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2018.2839741.
[124] S. Meng, L. Qi, Q. Li, W. Lin, X. Xu, and S. Wan, "Privacy-preserving and sparsity-aware location-based prediction method for collaborative recommender systems," *Future Gener. Comput. Syst.*, vol. 96, pp. 324–335, 2019.
[125] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "Random-data perturbation techniques and privacy-preserving data mining," *Knowl. Inf. Syst.*, vol. 7, no. 4, pp. 387–414, 2005.
[126] S. Badsha *et al.*, "Privacy preserving location-aware personalized web service recommendations," *IEEE Trans. Services Comput.*, to be published, doi: 0.1109/TSC.2018.2839587.

[127] M. Silic, G. Delac, and S. Srbljic, "Prediction of atomic web services reliability based on K-means clustering," in *Proc. 9th Joint Meeting Foundations Softw. Eng.*, 2013, pp. 70–80.
[128] E. Al-Masri and Q. H. Mahmoud, "Investigating web services on the world wide web," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 795–804.
[129] M. Vieira, N. Antunes, and H. Madeira, "Using web security scanners to detect vulnerabilities in web services," in *Proc. IEEE/IFIP Inte. Conf. Dependable Syst. Netw.*, 2009, pp. 566–571.
[130] Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed QoS evaluation for real-world web services," in *Proc. IEEE Int. Conf. Web Services*, 2010, pp. 83–90.
[131] A. Liu *et al.*, "Differential private collaborative web services QoS prediction," *World Wide Web J.*, vol. 22, no. 6, pp. 2697–2720, 2019.
[132] D. Frankowski, D. Cosley, S. Sen, L. G. Terveen, and J. Riedl, "You are what you say: privacy risks of public mentions," in *Proc. 29th Annu. Int.ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2006, pp. 565–572.
[133] K. Karta, "An investigation on personalized collaborative filtering for web service selection," Honours Programme Thesis, Univ. Western Australia, Brisbane, Citeseer, 2005.
[134] L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, and J. Chen, "A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment," *Future Gener. Comput. Syst.*, vol. 88, pp. 636–643, 2018.
[135] L. Qi, R. Wang, C. Hu, S. Li, Q. He, and X. Xu, "Time-aware distributed service recommendation with privacy-preservation," *Inf. Sci.*, vol. 480, pp. 354–364, 2019.

Zibin Zheng (Senior Member, IEEE) received the PhD degree from the Chinese University of Hong Kong, in 2011. He is currently a professor with the School of Data and Computer Science, Sun Yat-sen University, China. He has published more than 120 international journal and conference papers, including three ESI highly cited papers. According to Google Scholar, his papers have more than 9,100 citations, with an H-index of 46. His research interests include blockchain, services computing, software engineering, and financial big data.

Xiaoli Li (Student Member, IEEE) received the master's degree in computer architecture from the University of Electronic Science and Technology of China, Chengdou, China, in 2011. She is currently working toward the PhD degree with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. Her research interests include services computing, software engineering, cloud computing, machine learning, and deep neural networks.

Mingdong Tang (Member, IEEE) received the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, China, in 2010. He is currently a professor with the School of Information Science and Technology, Guangdong University of Foreign Studies, China. He has published more than 100 peer-reviewed scientific research papers in various journals and conference proceedings. His research interests include service-oriented computing, software engineering, and data mining.

Fenfang Xie (Student Member, IEEE) received the master's degree in computer science and technology from the Hunan University of Science and Technology, Xiangtan, China, in 2016. She is currently working toward the PhD degree with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. Her research interests include services computing, software engineering, cloud computing, mobile computing, machine learning, and deep neural networks.

Michael R. Lyu (Fellow, IEEE) received the BSc degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1981, the MSs degree in computer engineering from the University of California, Santa Barbara, CA, in 1985, and the PhD degree in computer science from the University of California, Los Angeles, CA, in 1988. He is currently a professor with the Department of Computer Science and Engineering, Chinese University of Hong Kong, where he is also the director of the Video over Internet and Wireless Technologies Laboratory. His research interests include software reliability engineering, distributed systems, fault-tolerant computing, mobile networks, web technologies, multimedia information processing, and e-commerce systems. He has fellowships with the ACM, AAAS, and Croucher Senior Research.