A Privacy-Preserving QoS Prediction Framework for Web Service Recommendation

Jieming Zhu, Pinjia He, Zibin Zheng, Michael R. Lyu

Shenzhen Research Institute, The Chinese University of Hong Kong, Shenzhen, China
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong
{*jmzhu, pjhe, zbzheng, lyu*}*@cse.cuhk.edu.hk*

Abstract—QoS-based Web service recommendation has recently gained much attention for providing a promising way to help users find high-quality services. To facilitate such recommendations, existing studies suggest the use of collaborative filtering techniques for personalized QoS prediction. These approaches, by leveraging partially observed QoS values from users, can achieve high accuracy of QoS predictions on the unobserved ones. However, the requirement to collect users' QoS data likely puts user privacy at risk, thus making them unwilling to contribute their usage data to a Web service recommender system. As a result, privacy becomes a critical challenge in developing practical Web service recommender systems. In this paper, we make the first attempt to cope with the privacy concerns for Web service recommendation. Specifically, we propose a simple yet effective privacy-preserving framework by applying data obfuscation techniques, and further develop two representative privacy-preserving QoS prediction approaches under this framework. Evaluation results from a publicly-available QoS dataset of real-world Web services demonstrate the feasibility and effectiveness of our privacy-preserving QoS prediction approaches. We believe our work can serve as a good starting point to inspire more research efforts on privacy-preserving Web service recommendation.

Keywords—*Web service recommendation; QoS prediction; collaborative filtering; privacy preservation*

I. INTRODUCTION

Web services are self-contained units of software functionalities (e.g., retrieving currency exchange rates) delivered over the Internet for users to build composite Web applications. Recent advances in cloud computing enable on-demand service delivery and promote the rapid growth of service markets, where more and more Web services are expected to become available. Whereas the abundance of Web services meets the various needs of different users (i.e., Web application providers), it also poses a significant challenge in selecting among a large number of similar services [1]. In this context, Web service recommendation [2], [3], [4] that aims to help users quickly find desirable services has become a hot research issue in the area of service computing in recent years.

Effective service recommendation needs to fulfil both functional and non-functional requirements of users. While functional requirements focus on what a service does, non-functional requirements are concerned with the quality of service (QoS), such as response time, throughput, and failure probability, etc. QoS plays an important role in Web service recommendation, according to which similar services can be ranked and selected for users. Service invocations usually rely on the Internet for connectivity and are heavily influenced by the dynamic network conditions. Therefore, users at different locations typically observe different QoS values even on

the same Web service. To enable personalized Web service recommendation, QoS evaluation from user side is desired. However, it is a challenge to acquire user-perceived QoS values of all the services because each user only has observed QoS values on a few used services. It is also impractical for each user to actively measure these QoS values due to the expensive overhead of invoking a large number of services.

To address this issue, collaborative QoS prediction has recently been proposed, and becomes a key step to QoS-based Web service recommendation. By applying collaborative filtering (CF) techniques [5] that are widely used in commercial recommender systems, unknown QoS can be predicted based on historical usage data collected from users, eliminating the need of additional service invocations. In other words, users can contribute their historical QoS data on the services they have used and receive prediction results on the QoS values of the services that they have never used before. In recent literature, a number of collaborative filtering approaches have been proposed for QoS prediction. Among them, neighbourhood-based CF approaches (e.g., UIPCC [2]) leverage the similarity between users and/or the similarity between services calculated on the observed QoS data for unknown QoS prediction. Model-based approaches (e.g., PMF [6], EMF [7]) fit the observed QoS data with a pre-defined model (e.g., low-rank matrix factorization), and then utilize the trained model for QoS prediction. Recent studies have shown that these approaches achieve high accuracy of QoS predictions and yield encouraging results on Web service recommendation.

Despite the potential benefits provided by Web service recommender systems, a major impediment to the practical deployment of such systems lies in their threats to user privacy. To receive effective recommendations, users are required to supply their observed QoS values. However, there is currently no policy to protect users from privacy issues. Malicious recommender systems, for example, may abuse the data, infer private information from the data, or even resell the data to a competing user for profits [8]. Even if the recommender system is not malicious, 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). This is why application providers are not willing to disclose their private usage data to the public or a third party. Such privacy threats limit the QoS data collection from users and hence degrade the accuracy of Web service recommendation. To encourage broader user participation, it is desired to consider privacy-preserving approaches for Web service recommendation that can be made without revealing private user data. Encryption is a straightforward way to achieve privacy. However, encryption techniques usually involve large computational overhead and

typically work for distributed collaborative filtering problems (e.g., homomorphic encryption used in [9]) where multi-party communication is necessary. This is inapplicable to our problem because user-user communication is infeasible.

In this paper, we propose a simple yet effective privacy-preserving framework for QoS-based Web service recommendation. Specifically, users are enabled to obfuscate their private data by data randomization techniques [10] before they expose the data to a recommender system. In this way, the recommender system can only collect obfuscated QoS data from users, and thus reduce the risk to expose user privacy. Our privacy-preserving framework is generic and can be applied to both the neighbourhood-based collaborative filtering approach, i.e., UIPCC [11], and the model-based collaborative filtering approach, i.e., PMF [6], which are two most common QoS prediction approaches in recent literature. We further revamp these two existing QoS prediction approaches based on our framework, and develop their corresponding privacy-preserving variants: P-UIPCC and P-PMF. We evaluate these approaches on WS-DREAM dataset [12], a publicly-available QoS dataset that has been widely employed for QoS prediction evaluation in the literature. The experimental results show that while preserving user privacy, our proposed approaches (P-UIPCC and P-PMF) can still attain decent prediction accuracy with comparision to the baseline approaches (UEAN and IMEAN) and the counterpart approaches (UIPCC and PMF). We also show the tradeoff between the achieved prediction accuracy and the preserved user privacy. For reproducibility, we release the source code and detailed evaluation results on our project page[1].

In summary, our paper makes the following contributions:

- This is the first work to cope with the privacy concerns for QoS-based Web service recommendation.
- We propose a simple yet effective privacy-preserving framework, and further develop two representative privacy-preserving QoS prediction approaches, P-UIPCC and P-PMF, under this framework.
- We conduct experiments on a real-world large-scale QoS dataset of Web services to evaluate the effectiveness of privacy-preserving QoS prediction approaches.

The remainder of this paper is organized as follows. Section II introduces the background and related work. Section III presents the framework of privacy-preserving Web service recommendation. Then we describe the detailed QoS prediction approaches in Section IV, and report the evaluation results in Section V. Finally, we conclude this paper in Section VI.

II. BACKGROUND AND RELATED WORK

In this section, we introduce the background of QoS-based Web service recommendation and review two representative collaborative filtering approaches used for QoS prediction. We then discuss the privacy issues and the key techniques for privacy preservation in related work.

A. QoS-based Web Service Recommendation

QoS-based Web service recommendation has recently attracted much attention from the service community, for pro-

	s_1	s_2	s_3	s_4	s_5		s_1	s_2	s_3	s_4	s_5
u_1	1.4	?	1.1	0.7	?	u_1	1.4	0.8	1.1	0.7	0.9
u_2	?	0.3	?	0.7	0.5	u_2	1.0	0.3	1.0	0.7	0.5
u_3	0.4	0.3	?	?	0.3	u_3	0.4	0.3	0.3	0.1	0.3
u_4	1.4	?	1.2	?	0.8	u_4	1.4	0.7	1.2	0.8	0.8

(a) User-Service Invocation Graph   (b) Observed QoS Matrix   (c) Predicted QoS Matrix
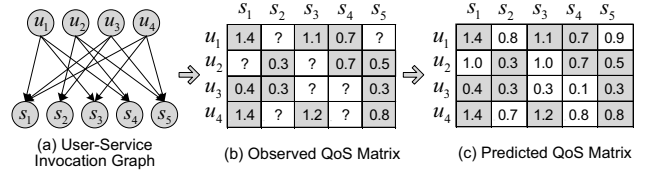
Fig. 1. An Illustrative Example of QoS Prediction

viding a promising way to help users select high-quality services out of all the candidate services according to the user-perceived QoS values. Because it is prohibitively expensive or even infeasible for a user to acquire all the QoS values of the candidate services, the key of QoS-based Web service recommendation is to enable accurate QoS predictions.

Collaborative filtering (CF) [5] has been widely used in commercial recommender systems (e.g., movie recommendation in Netflix, item recommendation in Amazon) for rating prediction, where the observed user ratings are leveraged to learn user preferences on the unrated movies or items and further make predictions on the unknown ratings. In recent literature, CF has been suggested as a promising approach to QoS prediction (e.g., [2], [3], [4]). As with the user-movie rating matrix collected in a movie recommender system, users invoking services can produce a user-service QoS matrix with respect to each QoS attribute. We denote a QoS matrix by R, whose entry R_{ij} represents the observed QoS value (e.g., response time) of user u_i invoking service s_j. Fig. 1(b) illustrates a QoS matrix with four users (u_1, ..., u_4) and five services (s_1, ..., s_5), produced by the user-service invocation graph in Fig. 1(a). In practice, the QoS matrix is very sparse (i.e., most of the entries are unknown), since each user usually invokes only a few services. As shown in Fig. 1(b), the grey entries are observed QoS values (e.g., $R_{11} = 1.4$) and the blank entries are unknown QoS values (e.g., $R_{12} = ?$). As a result, the QoS prediction problem can be modelled as a collaborative filtering problem. Fig. 1(c) shows the predicted QoS matrix from the observed QoS matrix in Fig. 1(b), where the unknown values are approximately reconstructed.

Specifically, two types of CF approaches have been studied for QoS prediction of Web services in recent literature:

1) Neighbourhood-based Collaborative Filtering: This type of CF approaches use the observed QoS data to compute the similarity values between users or services, and further leverage them for QoS prediction. Typical examples include user-based approaches (e.g., UPCC [13]) that leverage the QoS information of similar users for prediction, item-based approaches (e.g., IPCC [14]) that employ the QoS information of similar items (i.e., services) for prediction, and their hybrids (e.g., UIPCC [2], [11]) that combine user-based and item-based approaches together for accuracy improvement. These approaches are easy to implement, but they fail to deal with the data sparsity problem, which limits their performance in practice.

2) Model-based Collaborative Filtering: Model-based CF approaches provide a predefined model to fit the observed QoS data, and then the trained model can be used to predict the unknown QoS values. Matrix factorization (e.g., PMF [15]) is one of the most popular model-based CF approaches, which was first introduced to address the QoS prediction

problem in [7]. Matrix factorization model handles the sparsity problem well and usually achieves better performance than neighbourhood-based approaches.

In this paper, we mainly look into two QoS prediction approaches, UIPCC [2] and PMF [15]. They are representatives of the two types of CF approaches respectively and serve as a basis to develop many more sophisticated approaches. For example, some studies such as CloudPred [16], NIMF [6], and LN-LFM [17] integrate neighbourhood-based and model-based CF approaches, while some others suggest to leverage additional context information such as location information [4] and time information [18], [19] for improving prediction accuracy. Our work focuses on providing a privacy-preserving QoS prediction framework. Therefore, the studies on how to build more sophisticated models for accuracy improvement are orthogonal to our work and fall outside the scope of this paper.

B. Privacy Issues

Privacy is an important issue that has raised particular concerns among many research areas. In the following, we review the privacy studies related to our work.

1) Privacy in Service Computing: In service computing, applications are typically built by composing Web services offered by different service providers. User information often needs to be shared across the providers to fulfil an overall application task. This can raise privacy issues between users and service providers when the selected Web services for composition have privacy policies that are not compliant with users' privacy requirements. In this regard, privacy-aware Web service selection and composition (e.g., [20], [21], [22], [23]) have been studied. For example, Costante et al. [22] propose an approach to rank the candidate Web services with respect to the privacy level they offer. Tbahriti et al. [23] further provide a mechanism to verify and negotiate privacy constraints between users and service providers to enable privacy-compatible service composition. Different from these studies, our work aims to address privacy issues for Web service recommendation.

2) Privacy in Recommender Systems: In recommender systems [24], users want to gain useful recommendations without compromising their privacy. To achieve so, a variety of privacy-preserving collaborative filtering approaches [25] have been proposed by using techniques such as randomization [10], cryptography [8], anonymization [26], and so on. Privacy is also of vital importance to the realization of QoS-based Web service recommendation, where users might not be willing to disclose their private usage data. However, there is currently a lack of studies on how to cope with the privacy issues for QoS-based Web service recommendation. Existing privacy-preserving CF approaches are not directly applicable because of the unique challenges posed by Web service recommendation. For example, most of these approaches (e.g., [27], [9], [28]) require multi-party or peer-to-peer collaboration between users, which is inapplicable to service users. To bridge this gap, our paper makes the first attempt to build a privacy-preserving QoS prediction framework for Web service recommendation.

III. FRAMEWORK OF PRIVACY-PRESERVING WEB SERVICE RECOMMENDATION

Fig. 2 presents our privacy-preserving Web service recommendation framework. The workflow of this framework can
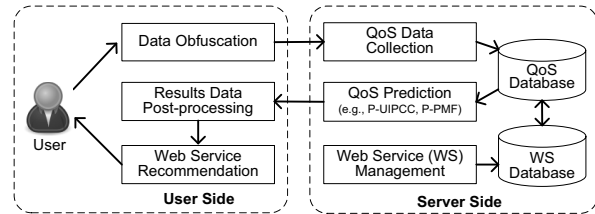

Fig. 2. Framework of Privacy-Preserving Web Service Recommendation

be separated into two parts executed at user side and server side respectively. At user side, the observed QoS data of each user undergo a data obfuscation process in order to protect user privacy as well as preserve the information required for performing collaborative QoS prediction. The obfuscated user data are then submitted to the server for QoS prediction. After receiving the prediction results from the server, a post-processing step is performed to recover the obfuscated results to the true QoS prediction values. At last, according to the recovered QoS values, candidate Web services can be ranked and recommended for the user. On the other hand, at server side, obfuscated QoS data are collected from different users in a collaborative way, through which a obfuscated QoS matrix can be acquired and stored in a QoS database. QoS prediction is then performed on the obfuscated QoS matrix by using our proposed privacy-preserving techniques such as P-UIPCC and P-PMF. At the same time, a list of the available Web services is maintained at a Web service database, which allows for service ranking and recommendation for the users.

User privacy is preserved by our framework because: 1) For each user, user data are obfuscated before being submitted to the server, and the obfuscation settings are only known to the user itself; 2) For the server, collaborative QoS prediction is performed based solely on the obfuscated user data, whereby user-observed real QoS values cannot be inferred. In this way, our framework enables users with greater control on their private usage data and less dependence on the server for privacy preservation. The privacy-preserving framework is generic such that both of the representative QoS prediction approaches (i.e., UIPCC and PMF) can work well without the need of significant modifications.

IV. QoS PREDICTION APPROACH

The above framework enables data obfuscation for preserving privacy, but also poses a challenge in accurate QoS prediction. In this section, we describe the data obfuscation process in detail, and then extend two representative QoS prediction approaches (UIPCC and PMF) into their privacy-preserving variants (P-UIPCC and P-PMF) accordingly.

A. Data Obfuscation

The need for privacy preservation has led to the development of a number of data obfuscation techniques, such as data randomization [10], data encryption [8], data anonymization [29]. Due to the sparse nature of our data, in this paper, we make use of data randomization [10], a simple yet effective way to obfuscate the data.

The basic idea of data randomization is to add a random value (i.e., noise) to the true value so that the resulting value

becomes disguised. In this way, when the obfuscated QoS data undergo further processing, user information regarding real QoS values can be preserved. Fortunately, although each individual QoS value becomes disguised, we find that some approximate computations (e.g., scalar product) on the aggregated data of users can still be done with decent accuracy.

To make it clear, we now describe the scalar product property of data randomization [10] in detail. Let $a = (a_1, ..., a_n)$ and $b = (b_1, ..., b_n)$ be true vectors with a mean of zero. We obfuscate these vectors as $a' = a + \epsilon$ and $b' = b + \delta$, where $\epsilon = (\epsilon_n, ..., \epsilon_n)$ and $\delta = (\delta_n, ..., \delta_n)$ are random noises generated from a uniform distribution in $[-\alpha, \alpha]$. Next, we show that the scalar product between a and b can be approximated by using the obfuscated vectors a' and b': i.e., $\mathbf{a'b'} \approx \mathbf{ab}$. To this end, we have

$$a'b' = \sum_{i=1}^{n}(a_i + \epsilon_i)(b_i + \delta_i) = \sum_{i=1}^{n}(a_ib_i + a_i\delta_i + b_i\epsilon_i + \epsilon_i\delta_i).$$

Because a and δ are independent vectors and each has a zero mean, we have $\sum_{i=1}^{n} a_i\delta_i \approx 0$. Likewise, we have $\sum_{i=1}^{n} b_i\epsilon_i \approx 0$, and $\sum_{i=1}^{n} \epsilon_i\delta_i \approx 0$. Hence, we derive the following approximation:

$$a'b' \approx \sum_{i=1}^{n} a_ib_i = ab. \tag{1}$$

With this observation, we find that data randomization can potentially preserve user privacy as well as the usability of the data for collaborative analysis. Therefore, it is appealing to study how to apply this data obfuscating technique to performing collaborative QoS prediction in a privacy-preserving way. To achieve this goal, we propose a two-step data obfuscation procedure for QoS data processing. We emphasize that, as shown in our framework in Fig. 2, each user performs data obfuscation individually at user side before contributing the QoS data to the server.

1) Z-score normalization: To facilitate better randomization of the data, we perform z-score normalization on the observed QoS data as the first step. Z-score normalization is a standard normalization method to adjust the data average and data variance. The normalized data have a zero mean and unit variance. More specifically, for user u, we denote $R_u = (R_{u1}, ..., R_{um})$ as a vector of observed QoS values on m Web services. $R_{us} > 0$ indicates that user u has invoked service s; otherwise, $R_{us} = 0$. We compute the mean ($\bar{R}_u$) and standard deviation (σ_u) of this QoS vector R_u:

$$\bar{R}_u = \sum_{s \in I_u} R_{us}/|I_u|, \ \ \sigma_u = \sqrt{\sum_{s \in I_u}(R_{us} - \bar{R}_u)^2/|I_u|}, \tag{2}$$

where $I_u = \{s \mid R_{us} > 0\}$ denotes the set of Web services that has been invoked by user u. Then z-score normalization is performed on the QoS values with the following equation:

$$r_{us} = (R_{us} - \bar{R}_u)/\sigma_u. \tag{3}$$

The normalization step results in a zero-mean data vector that is well suited for the following data randomization process.

2) Data Randomization: As the second step, we perform randomized perturbation on the normalized QoS vector by:

$$r'_{us} = r_{us} + \epsilon_{us}, \tag{4}$$

where ϵ_{us} is a random value generated from a specified distribution, for example, uniform distribution in $[-\alpha, \alpha]$.

Especially when $\alpha = 0$, the overall data obfuscation process reduces to a z-score normalization. We further study the effect of different distributions (e.g., uniform distribution, Gaussian distribution) of random noises on QoS prediction accuracy in Section V-E.

After data obfuscation, users can submit their obfuscated QoS data to the server. Given n users and m services, the server can collect a QoS matrix denoted as $r' \in \mathbb{R}^{n \times m}$ with each entry (r'_{us}) being obtained via Equ. (4). Since such data obfuscation process is performed at user side, the private information such as $\bar{R}_u$ and σ_u are kept at user side. As a result, the server cannot infer the true QoS values of the users, and user privacy is preserved.

Next, we will show how we extend the two representative approaches (UIPCC and PMF) to perform privacy-preserving QoS prediction based on the obfuscated QoS matrix r'. Note that UIPCC and PMF have been carefully reported in the related work [11], [6], so we do not intend to provide the original descriptions but the necessary extensions from them.

B. Privacy-Preserving UIPCC (P-UIPCC)

UIPCC (a.k.a. WSRec), first proposed in [2], has been a widely-studied QoS prediction approach. The key of UIPCC is to compute the similarity between users and the similarity between services, after which QoS values contributed by similar users and similar services can be leveraged to compute the prediction value. Existing work usually employ Pearson correlation coefficient (PCC) as the similarity measure. For example, the PCC similarity between user u and user v is defined as follows:

$$sim(u, v) = \frac{\sum_{s \in J}(R_{us} - \bar{R}_u)(R_{vs} - \bar{R}_v)}{\sqrt{\sum_{s \in J}(R_{us} - \bar{R}_u)^2}\sqrt{\sum_{s \in J}(R_{vs} - \bar{R}_v)^2}}, \tag{5}$$

where $J = I_u \cap I_v$ is the set of Web services that are invoked by both user u and user v. R_{us} is the true QoS value of user u invoking service s. $\bar{R}_u$ and $\bar{R}_v$ are the average QoS values observed by user u and user v, respectively. From this definition, we have $sim(u, v) \in [-1, 1]$, where a larger PCC value indicates higher user similarity.

However, due to the obfuscation of QoS data, at server side we only have obfuscated QoS value r'_{us}, rather than its true value R_{us}. Therefore, we consider to employ r'_{us} to approximately compute the similarity value $sim(u, v)$ as follows:

$$sim(u, v) = \sum_{s \in I_u \cap I_v} r'_{us}r'_{vs}/\sqrt{|I_u||I_v|} \tag{6}$$

$$\approx \sum_{s \in I_u \cap I_v} r_{us}r_{vs}/\sqrt{|I_u||I_v|} \tag{7}$$

$$= \frac{\sum_{s \in I_u \cap I_v}(R_{us} - \bar{R}_u)(R_{vs} - \bar{R}_v)}{\sigma_u\sigma_v\sqrt{|I_u||I_v|}} \tag{8}$$

$$= \frac{\sum_{s \in I_u \cap I_v}(R_{us} - \bar{R}_u)(R_{vs} - \bar{R}_v)}{\sqrt{\sum_{s \in I_u}(R_{us} - \bar{R}_u)^2}\sqrt{\sum_{s \in I_v}(R_{vs} - \bar{R}_v)^2}}. \tag{9}$$

By applying the scalar product property in Equ. (1) to Equ. (6), substituting Equ. (3) to Equ.(7), and substituting Equ. (2) to Equ. (8), we derive Equ. (9), which is exactly the similarity measure used for collaborative filtering in the related work [10], [30]. Note that this similarity measure differs

244

slightly from Equ. (5) in the denominator part, but provides a good approximation to it (as the experiments shown in Section V). Therefore, by using the obfuscated QoS data, we employ Equ. (6) as the approximation of the similarity between user u and v.

After similarity computation between users, we can identify a set of top-k similar neighbours (T_u) for each user u. Then the unknown QoS value, for each entry where $r'_{us} = 0$, can be estimated as the weighted average of the QoS values observed by similar neighbours, i.e.,

$$\hat{r}^U_{us} = \sum_{v \in T_u} sim(u,v) r'_{vs} \Big/ \sum_{v \in T_u} sim(u,v). \quad (10)$$

In a similar way, we can also leverage the information of similar services to make QoS prediction:

$$\hat{r}^S_{us} = \sum_{g \in T_s} sim(s,g) r'_{ug} \Big/ \sum_{g \in T_s} sim(s,g), \quad (11)$$

where T_s is the set of top-k similar services of service s. The similarity $sim(s,g)$ is further calculated by employing the cosine similarity between service s and service g:

$$sim(s,g) = \frac{\sum_{u \in I_s \cap I_g} r'_{us} r'_{ug}}{\sqrt{\sum_{u \in I_s \cap I_g} (r'_{us})^2} \sqrt{\sum_{u \in I_s \cap I_g} (r'_{ug})^2}}, \quad (12)$$

where $I_s \cap I_g$ represents the set of users that invoke both service s and service g. Note that the cosine similarity here equals to the original PCC similarity in UIPCC, because the QoS vectors have already been normalized during data obfuscation.

At last, as with UIPCC, a convex combination between user-based QoS prediction and service-based QoS prediction is employed to enhance the prediction accuracy.

$$\hat{r}_{us} = \lambda \hat{r}^U_{us} + (1 - \lambda) \hat{r}^S_{us}, \quad (13)$$

where λ controls the combination weight between $\hat{r}^U_{us}$ and $\hat{r}^S_{us}$. Especially, when $\lambda = 0$, $\hat{r}_{us} = \hat{r}^S_{us}$, and when $\lambda = 1$, $\hat{r}_{us} = \hat{r}^U_{us}$.

However, this prediction result $\hat{r}_{us}$ is a normalized value that cannot reveal the prediction on the true QoS. When the user receives the prediction results from the server, a post-processing step, which is a re-normalization operation of the z-score normalization, can be taken to get the final prediction value $\hat{R}_{us}$:

$$\hat{R}_{us} = \bar{R}_u + \sigma_u * \hat{r}_{us}. \quad (14)$$

Note that the post-processing step can be only performed at user side because $\bar{R}_u$ and σ_u are only known to the user.

C. Privacy-Preserving PMF (P-PMF)

PMF, or probabilistic matrix factorization [15], as a popular model-based collaborative filtering approach, has been suggested for QoS prediction by prior work [7], [6]. PMF works on an essential assumption of the low-rank structure of the QoS matrix. A matrix has a low rank when the entries of the matrix are largely correlated. In our case, as reported by the related work [11], [6], similar users usually have similar QoS values on the same Web service. The goal of PMF is to map n users and m services into a joint latent factor space with dimensionality d such that each observed entry of the QoS matrix can be captured as the inner product of the corresponding latent factors.

Formally, we denote the latent user factors as $U \in \mathbb{R}^{d \times n}$ whose u-th column represents the latent factor of user u, and the latent service factors as $S \in \mathbb{R}^{d \times m}$ whose s-th column represents the latent factor of service s. Accordingly, we use $U^T_u S_s$ to approximate the observed QoS value R_{us} between user u and service s, i.e., $R_{us} \approx U^T_u S_s$, or more precisely,

$$R_{us} = U^T_u S_s + \delta_{us}, \quad (15)$$

where U^T_u is the transpose of U_u and δ_{us} denotes the approximation error. The goal is to minimize all of the approximation errors. By taking δ_{us} as Gaussian noise [15], the loss function can be formulated as follows:

$$\mathcal{L} = \frac{1}{2} \sum_{u=1}^{n} \sum_{s=1}^{m} I_{us}(R_{us} - U^T_u S_s)^2 + \frac{\gamma}{2}(\sum_{u=1}^{n} \|U_u\|^2 + \sum_{s=1}^{m} \|S_s\|^2). \quad (16)$$

The first part measures the sum of squared approximation errors between R_{us} and $U^T_u S_s$, where I_{us} acts as an indicator that equals to 1 if R_{us} is observed, and 0 otherwise. The second part are regularization terms used to avoid the overfitting problem, where $\|\cdot\|$ denotes the Euclidean norm, and γ is a parameter to control the extent of regularization.

According to the basic PMF model as specified in Equ. (15), the specific QoS of user u invoking service s can be effectively captured by the interaction between U_u and service S_s. However, some other effects known as biases for determining the QoS values are independent of user-service interactions. For example, the users with high network bandwidth tend to experience fast network connections and the services equipped with abundant system resources likely provide short request-processing time. To capture these factors associated with either users or services, there is a suggestion for biased matrix factorization model in [17]:

$$R_{us} = \mu + b_u + b_s + U^T_u S_s + \delta_{us}, \quad (17)$$

where μ is a global bias, and b_u and b_s measure the user bias and service bias respectively.

While preserving user privacy, the application of data obfuscation poses new challenges in modelling the obfuscated QoS data. To compromise the effect of data obfuscation, we set $\mu = 0$ and $b_u = \bar{R}_u$. Accordingly, we derive the following model:

$$r'_{us} = b'_s + {U'_u}^T S' + \delta'_{us} + \epsilon_{us}. \quad (18)$$

For ease of presentation, we further denote it as:

$$r'_{us} = b_s + U^T_u S_s + \delta_{us} + \epsilon_{us}. \quad (19)$$

This model naturally compromise the effect of z-score normalization at user side. By taking both δ_{us} and ϵ_{us} as Gaussian noise [15], the loss function can be expressed as:

$$\begin{aligned} \mathcal{L}' = & \frac{1}{2} \sum_{u=1}^{n} \sum_{s=1}^{m} I_{us}(r'_{us} - b_s - U^T_u S_s)^2 \\ & + \frac{\gamma}{2}(\sum_{u=1}^{n} {b_s}^2 + \sum_{u=1}^{n} \|U_u\|^2 + \sum_{s=1}^{m} \|S_s\|^2). \quad (20) \end{aligned}$$

The minimization of this loss function can typically be solved by the gradient descent algorithm used in [6] or the stochastic gradient descent algorithm used in [17]. Due to space limits, we omit the algorithmic description here and refer interested readers to our supplementary report (see our project page).

QoS	#Users	#Services	Range	Average	Std.
RT (sec)	339	5,825	$0 \sim 20$	0.909	1.973
TP $(kbps)$	339	5,825	$0 \sim 1000$	47.562	110.797

After obtaining the solutions with respect to b_s, U_u, and S_s, we can make the following QoS prediction:

$$\hat{r}_{us} = b_s + U_u^T S_s. \tag{21}$$

At last, as with P-UIPCC, a post-processing step in Equ. (14) is required to recover the prediction result $\hat{r}_{us}$ to the true prediction value $\hat{R}_{us}$. For both P-UIPCC and P-PMF, after obtaining the predicted QoS values of all the available Web services, we can recommend to users those services with top-ranked QoS values.

V. EVALUATION

This section describes the experiments and the corresponding results of evaluating our privacy-preserving QoS prediction approaches. In particular, we intend to answer the following research questions.

RQ1: What is the effect of data obfuscation?

RQ2: What is the accuracy of P-UIPCC and P-PMF?

RQ3: What is the tradeoff between accuracy and privacy?

RQ4: What is the effect of distribution of random noises on prediction accuracy?

A. Experimental Setup

In our experiments, we focus mainly on two representative QoS attributes: response time (RT) and throughput (TP). Response time measures the time duration between user sending out a request and receiving a response, while throughput stands for the data transmission rate of a user invoking a service.

The experiments are conducted based on a publicly-available QoS dataset of real-world Web services [12]. The dataset was collected in August 2009, providing a total of 1,974,675 response time and throughput records of service invocations between 339 users and 5,825 Web services. The 339 users are simulated by PlanetLab[2] nodes distributed at 30 countries, while the 5,825 real-world Web services are crawled from the Internet and are deployed at 73 countries. Table I provides a summary of the statistics of the data.

In our experiments, we represent each type of QoS data by a 339-by-5825 QoS matrix with each entry denoting the observed response time/throughput of a specific invocation. In practice, the QoS matrix is very sparse because each user usually invokes only a handful of services. To simulate such data sparsity in our experiments, we randomly remove entries from the full data matrix and only keep a small density of historical QoS values. Data density = 10%, for example, indicates that each user invokes 10% of the services, or each service is invoked by 10% of the users. We leverage the preserved data entries for QoS prediction, and then use the removed QoS values as testing data for accuracy evaluation.

---

[2]PlanetLab (https://www.planet-lab.org) is an open platform for system and networking research, currently consisting of 1341 nodes at 654 global sites.

(a) $\alpha = 0$ (b) $\alpha = 0.5$ (c) $\alpha = 1$
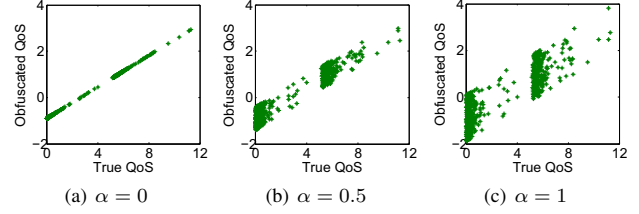
Fig. 3. Obfuscated QoS (r'_{us}) v.s. True QoS (R_{us})

To quantize the accuracy of QoS prediction, we employ a standard error metric, MAE (Mean Absolute Error), which has been widely used in the existing work (e.g., [7], [6]).:

$$MAE = \sum\nolimits_{I_{us}=0} \left| \hat{R}_{us} - R_{us} \right| / N , \tag{22}$$

where R_{us} and $\hat{R}_{us}$ denote the observed QoS value and the corresponding predicted QoS value of the invocation between user u and service s. N is the total number of testing samples to be predicted, i.e., entries with $I_{us} = 0$. A smaller MAE value indicates better prediction accuracy.

B. Effect of Data Obfuscation (RQ1)

The aim of data obfuscation is to perturb the QoS data such that user privacy regarding the true QoS values can be preserved when performing collaborative analysis on the server. To understand the effect of data obfuscation made on QoS data (*RQ1*), we compare the obfuscated QoS (r'_{us}) against the corresponding true QoS data (R_{us}). As an example, we randomly select a user from our dataset and provide three scatter plots by using the response time data of this user. The plots present the relationships between r'_{us} and R_{us} under different α settings. α is a parameter to determine the range of noises ϵ_{us} used to obfuscate the data. Especially, when $\alpha = 0$, the data obfuscation reduces to a z-score normalization process. Thus, Fig. 3(a) shows linear dependence between r'_{us} and R_{us}. Z-score normalization is able to provide basic protection for user data where the mean and variance properties of QoS data are eliminated. The data after z-score normalization have a zero mean and unit variance. As α increases, the obfuscated data become more and more disordered, As shown in 3(a) and (b), the linear correlation between r'_{us} and R_{us} is further eliminated. Consequently, a larger α indicates better protection for user data. Note that we have similar observations on the throughput data and thus omit the details here.

C. Prediction Accuracy (RQ2)

Data obfuscation is useful to perturb the QoS data for preserving user privacy, but it makes no sense without providing accurate prediction results. We evaluate the accuracy of our privacy-preserving QoS prediction approaches (P-UIPCC and P-PMF) based on the obfuscated QoS data, and compare them against the following baselines and counterpart approaches (*RQ2*). We emphasize that these existing approaches require users' true QoS data and do not consider privacy issues.

- **UMEAN** [2]: This is a baseline approach that employs the average QoS value observed by a user (i.e., the row mean of R) to predict the unknown QoS of this user invoking other unused Web services.

246

TABLE II.   PARAMETER SETTINGS

Approach	RT			TP		
UIPCC	$k:10$	$\lambda:0.1$	–	$k:10$	$\lambda:0.9$	–
P-UIPCC	$k:10$	$\lambda:0.9$	$\alpha:0.5$	$k:10$	$\lambda:0.9$	$\alpha:0.5$
PMF	$d:10$	$\gamma:40$	–	$d:10$	$\gamma:800$	–
P-PMF	$d:10$	$\gamma:12$	$\alpha:0.5$	$d:10$	$\gamma:12$	$\alpha:0.5$

TABLE III.   PREDICTION ACCURACY (W.R.T. MAE)

QoS	Approach	Data Density				
		10%	15%	20%	25%	30%
RT	UMEAN	0.875	0.875	0.875	0.875	0.875
	IMEAN	0.688	0.683	0.681	0.680	0.679
	UIPCC	0.582	0.501	0.450	0.427	0.411
	PMF	0.487	0.452	0.431	0.418	0.409
	P-UIPCC	0.569	0.537	0.512	0.495	0.482
	P-PMF	0.540	0.504	0.478	0.458	0.443
TP	UMEAN	53.835	53.816	53.801	53.804	53.799
	IMEAN	26.860	26.716	26.641	26.593	26.571
	UIPCC	22.370	20.219	18.928	17.891	17.080
	PMF	15.994	14.670	13.924	13.405	13.117
	P-UIPCC	23.572	21.324	19.754	18.681	17.953
	P-PMF	20.702	18.451	17.351	16.634	16.063

- **IMEAN** [2]: Likewise, this baseline approach employs the observed average QoS value of a Web service (i.e., the column mean of R) to predict the unknown QoS of other users invoking this Web service.

- **UIPCC** [2], [11]: This is a hybrid approach that combines both user-based CF approach (UPCC) and item-based CF approach (IPCC) to make full use of the historical information from similar users and services for QoS prediction. UIPCC typically performs better than either UPCC or IPCC.

- **PMF** [6]: This is a widely-used implementation of the matrix factorization model [15], which have been introduced to QoS prediction in [6].

For fair comparisons, we use the original parameters for the counterpart approaches, as specified in the related work, because we experiment on the same dataset. To make it consistent with these settings, most parameters of our approaches are set the same with them (e.g., $k = 10$ for top-k neighbours in UIPCC and P-UIPCC). However, since both P-UIPCC and P-PMF work on obfuscated (normalized) data, we set different λ and γ values. The detailed parameters are specified in Table II. We use $\alpha = 0.5$ in this experiment and study the effect of α in Section V-D. Additionally, we vary the data density from 10% to 30% at a step increase of 5%. Each approach is performed 20 times under each data density (with different random seeds), and the average MAE results are reported.

Table III provides the results of prediction accuracy with comparisons among different approaches. The results show that, while both of our approaches preserve decent privacy by data obfuscation ($\alpha = 0.5$), they still perform much better than the baselines including UMEAN and IMEAN, and achieve comparable accuracy with the counterpart approaches including UIPCC and PMF. In particular, P-UIPCC sometimes performs better than UIPCC (e.g., 0.569 vs 0.582), which can be attributed to the use of z-score normalization. Moreover, we observe that even working on obfuscated data, P-PMF mostly performs better than UIPCC. These encouraging results indicate the effectiveness of privacy-preserving approaches. In addition, we can see that the accuracy of these QoS prediction approaches improves with the increase in data density.
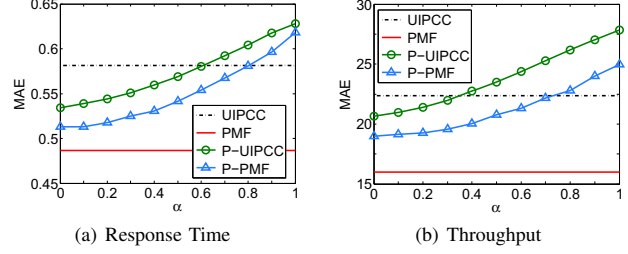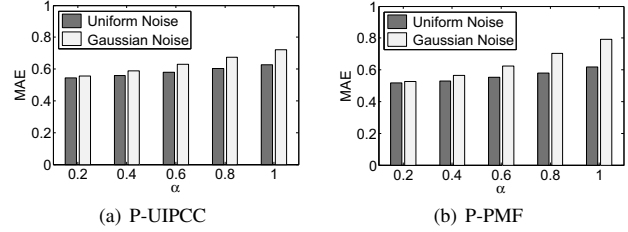

Fig. 4.   Tradeoff between Accuracy and Privacy

Fig. 5.   Impact of Matrix Density

D. Tradeoff between Accuracy and Privacy (RQ3)

Whereas the goal of our work is to achieve both accuracy and privacy, there is indeed a tradeoff between them. At one extreme, users can provide true QoS data to obtain the most accurate QoS prediction results yet they lose privacy. At another extreme, users can submit totally false QoS data to preserve privacy but bad prediction results will be returned. To study such tradeoff between accuracy and privacy (*RQ3*), we consider the effect of noise range α on prediction accuracy, because a larger α indicates better protection of privacy. Specifically, in this experiment, we set data density = 10% and vary α from 0 to 1 at a step increase of 0.1. Accordingly, we obtain the prediction accuracy under each α value.

Fig. 4 presents the experimental results corresponding to response time and throughput, respectively. We can observe that both P-UIPCC and P-PMF degrade in accuracy (i.e., MAE increases) when α becomes larger, because the utility of data is less preserved. However, when α is small, e.g., less than 0.6 in Fig. 4(a), our privacy-preserving approaches are more accurate than UIPCC. Even α is as large as 1, which is the variance of data after z-score normalization, the prediction accuracy is much better than the baselines (UMEAN and IMEAN). As a result, a balance needs to be made between the accuracy and privacy that a user wants to achieve. Additionally, we find that PMF and P-PMF consistently outperform UIPCC and P-UIPCC. This suggests the superior effectiveness of model-based approaches in capturing the latent structure of the QoS data, which conforms to the results reported in [6].

E. Effect of Distribution of Random Noises (RQ4)

In addition to the impact of noise range, a data randomization scheme is also subject to the choice of the distribution of random noises that are used for data obfuscation. In all of the above experiments, the random noises are generated from a uniform distribution located in $[-\alpha, \alpha]$. In contrast, in this experiment, we consider a Gaussian distribution $\mathcal{N}(0, \alpha)$ with a mean of zero and a standard deviation of α. Compared to a uniform distribution, random noises generated from a Gaussian

247

distribution are unevenly distributed. To investigate the effect of distribution of random noises (*RQ4*), we vary the α value and compare the prediction accuracy of P-UIPCC and P-PMF with different settings on the distribution of random noises.

Fig. 5 presents the results of the accuracy comparison. We can observe that, for both P-UIPCC and P-PMF, the randomization scheme with uniform noises performs better than the scheme with Gaussian noises. In particular, the performance differs significantly between the two randomization schemes under a large α setting. The results imply that the distribution of random noises is a crucial factor for determining the performance of our privacy-preserving approaches.

VI. CONCLUSION

Privacy is a practical issue to be addressed for QoS-based Web service recommendation. This paper makes an initial effort to deal with the privacy-preserving Web service recommendation problem. We propose a generic privacy-preserving framework with the use of data obfuscation techniques, under which users can gain greater control on their data and rely less on the recommender system for privacy protection. We further develop two privacy-preserving QoS prediction approaches based on this framework, namely P-UIPCC and P-PMF, as representatives of neighbourhood-based CF approaches and model-based CF approaches respectively. To evaluate the effectiveness of P-UIPCC and P-PMF, we conduct experiments on a publicly-available QoS dataset of real-world Web services. The experimental results show that our privacy-preserving QoS prediction approaches can still descent prediction accuracy compared with the counterpart approaches. We hope that the encouraging results achieved in this initial work can inspire more research efforts on privacy-preserving Web service recommendation.

REFERENCES

[1] A. Klein, F. Ishikawa, and S. Honiden, "Towards network-aware service composition in the cloud," in *Proc. of the 21st ACM World Wide Web Conference (WWW)*, 2012, pp. 959–968.

[2] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WSRec: A collaborative filtering based web service recommender system," in *Proc. of the IEEE International Conference on Web Services (ICWS)*, 2009, pp. 437–444.

[3] W. Zhang, H. Sun, X. Liu, and X. Guo, "Temporal QoS-aware web service recommendation via non-negative tensor factorization," in *Proc. of the 23rd International World Wide Web Conference (WWW)*, 2014, pp. 585–596.

[4] X. Chen, Z. Zheng, Q. Yu, and M. R. Lyu, "Web service recommendation via exploiting location and QoS information," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1913–1924, 2014.

[5] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artificial Intellegence*, vol. 2009, 2009.

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

[7] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu, "An extended matrix factorization approach for QoS prediction in service selection," in *Proc. of the 9th IEEE International Conference on Services Computing (SCC)*, 2012, pp. 162–169.

[8] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, and D. Boneh, "Privacy-preserving matrix factorization," in *Proc. of the ACM SIGSAC Conference on Computer & Communications Security (CCS)*, 2013, pp. 801–812.

[9] J. F. Canny, "Collaborative filtering with privacy," in *Proc. of the IEEE Symposium on Security and Privacy*, 2002, pp. 45–57.

[10] H. Polat and W. Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques." in *Proc. of the IEEE Conference on Data Mining (ICDM)*, 2003, pp. 625–628.

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

[12] Z. Zheng, Y. Zhang, and M. R. Lyu, "Investigating QoS of real-world web services," *IEEE T. Services Computing*, vol. 7, no. 1, pp. 32–39, 2014.

[13] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction for web services via collaborative filtering," in *Proc. of the IEEE International Conference on Web Services (ICWS)*, 2007, pp. 439–446.

[14] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. of the 10th International World Wide Web Conference (WWW)*, 2001, pp. 285–295.

[15] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. of the 21st Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.

[16] Y. Zhang, Z. Zheng, and M. R. Lyu, "Exploring latent features for memory-based QoS prediction in cloud computing," in *Proc. of the 30th IEEE Symposium on Reliable Distributed Systems (SRDS)*, 2011, pp. 1–10.

[17] D. Yu, Y. Liu, Y. Xu, and Y. Yin, "Personalized QoS prediction for web services using latent factor models," in *Proc. of the IEEE International Conference on Services Computing (SCC)*, 2014, pp. 107–114.

[18] C. Yu and L. Huang, "Time-aware collaborative filtering for QoS-based service recommendation," in *Proc. of the IEEE International Conference on Web Services (ICWS)*, 2014, pp. 265–272.

[19] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "Towards online, accurate, and scalable qos prediction for runtime service adaptation," in *Proc. of the IEEE 34th International Conference on Distributed Computing Systems (ICDCS)*, 2014, pp. 318–327.

[20] W. Xu, V. Venkatakrishnan, R. Sekar, and I. Ramakrishnan, "A framework for building privacy-conscious composite web services," in *Proc. of the IEEE International Conference on Web Services (ICWS)*. IEEE, 2006, pp. 655–662.

[21] A. Squicciarini, B. Carminati, and S. Karumanchi, "A privacy-preserving approach for web service selection and provisioning," in *Proc. of the IEEE International Conference on Web Services (ICWS)*. IEEE, 2011, pp. 33–40.

[22] E. Costante, F. Paci, and N. Zannone, "Privacy-aware web service composition and ranking," in *Proc. of the IEEE International Conference on Web Services (ICWS)*, 2013, pp. 131–138.

[23] S. Tbahriti, C. Ghedira, B. Medjahed, and M. Mrissa, "Privacy-enhanced web service composition," *IEEE T. Services Computing*, vol. 7, no. 2, pp. 210–222, 2014.

[24] N. Ramakrishnan, B. J. Keller, B. J. Mirza, A. Y. Grama, and G. Karypis, "Privacy risks in recommender systems," *IEEE Internet Computing*, vol. 5, no. 6, pp. 54–62, 2001.

[25] A. Bilge, C. Kaleli, I. Yakut, I. Gunes, and H. Polat, "A survey of privacy-preserving collaborative filtering schemes." 2013, pp. 1085–1108.

[26] T. R. Hoens, M. Blanton, A. Steele, and N. V. Chawla, "Reliable medical recommendation systems with patient privacy," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 4, p. 67, 2013.

[27] J. Canny, "Collaborative filtering with privacy," in *Proc. of the IEEE Symposium on Security and Privacy*, 2002, pp. 45–57.

[28] J. Zhan, C. Hsieh, I. Wang, T. Hsu, C. Liau, and D. Wang, "Privacy-preserving collaborative recommender systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 40, no. 4, pp. 472–476, 2010.

[29] W. Klosgen, "Anonymization techniques for knowledge discovery in databases," in *Proc. of the International Conference on Knowledge Discovery and Data Mining (KDD)*, 1995, pp. 186–191.

[30] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1999, pp. 230–237.