FHC-DQP: Federated Hierarchical Clustering for Distributed QoS Prediction

Guobing Zou, Shiyi Lin, Shengxiang Hu, Shengyu Duan, Yanglan Gan, Bofeng Zhang, and Yixin Chen, *Fellow, IEEE*

Abstract—With the overwhelming explosion of Web services, how to effectively predict unknown QoS has become a key issue of differentiating large-scale similar or functionally equivalent Web services. However, current state-of-the-art QoS prediction approaches based on deep learning still suffer from two deficiencies. First, they mainly focus on predicting vacant QoS in a centralized manner and scarcely take into account distributed QoS prediction, which makes difficult to protect the privacy information of users invoking Web services. Second, they have ignored the hierarchical collaborative relationship to better extract latent features of users and services, reducing the accuracy of QoS prediction. To address these two issues, we propose a novel framework called ***F**ederated **H**ierarchical **C**lustering for **D**istributed **Q**oS **P**rediction* (FHC-DQP). It collaboratively performs distributed federated training on independent users' QoS invocations, and then the extracted federated users' private features are fed to clustering algorithm for partitioning them into a set of clusters. By iteratively federated hierarchical clustering, users are fine-grained partitioned together and those users within the same cluster have stronger collaborative relevance for more effectively learning the latent features of users and services leading to the performance improvement of distributed QoS prediction, where contextual-aware deep neural network is designed for personalized QoS prediction. Extensive experiments are conducted based on a public real-world benchmarking dataset called WS-DREAM with almost 2,000,000 user-service historical QoS invocations. Compared with both centralized and federated competing baselines, the results demonstrate FHC-DQP receives superior performance for distributed QoS prediction, when it provides privacy-preserving of users' QoS invocations.

Index Terms—Web Service, Distributed QoS Prediction, Federated Learning, Hierarchical Clustering, Deep Neural Network

1 Introduction

With the wide applicability of service-oriented architecture (SOA), Web services are deployed by service vendors with exponential growth in the past few years. As one of the crucially implemented techniques of SOA, Web service has provisioned as fundamental components for service reuse and functionality extension in service-oriented downstream tasks, such as service discovery, selection, composition, recommendation, and mashup creation [1], [2], [3], [4]. Due to the rapidly increasing number of Web services pushed on the Internet, it can easily trigger the phenomenon that different service vendors may provide a lot of similar or functionally equivalent Web services. That becomes impractical or difficulty in choosing satisfactory Web services for service consumers in real-world scenarios, such as enterprise application integration and e-commerce. As a result, how to effectively differentiate and recommend Web services is of vital importance in service-oriented applications.

- *G. Zou, S. Lin, S. Hu, S. Duan are with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China. E-mail: gbzou@shu.edu.cn, sylin@shu.edu.cn, shengxianghu@shu.edu.cn, sduan@shu.edu.cn.*
- *Y. Gan is with the School of Computer Science and Technology, Donghua University, Shanghai 201620, China. E-mail: ylgan@dhu.edu.cn*
- *B. Zhang is with the School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai 201209, China. E-mail: bfzhang@sspu.edu.cn.*
- *Y. Chen is with the Department of Computer Science and Engineering, Washington University in St. Louis, MO 63130, USA. E-mail:chen@cse.wustl.edu.*

Quality of Service (QoS), which represents the non-functional criterion of Web services, including response time (RT), throughput (TP), availability, cost, etc, plays an important role in selecting Web services with the same or similar functionality. Due to the dynamic network environment and different geographical locations, users may observe different QoS values when invoking the same Web service [5]. Furthermore, it is extremely time-consuming for service requesters to invoke all Web services and service vendors to monitor QoS values of their provisioned Web services. It has become a hot research issue to predict the missing QoS based on sparse user-service QoS invocation records. Consequently, how to accurately predict the vacant QoS is a challenging and fundamental task for service-oriented downstream application scenarios.

Collaborative filtering (CF) as the most important technique has been widely applied to predict the vacant QoS, which has received many research investigations. CF-based approaches performs the prediction of missing QoS depending on recorded user-service QoS interactive invocations. They can be divided into memory-based and model-based approaches. Memory-based QoS prediction approaches predict QoS values based on the user-service invocation records by similar neighborhood calculation [6], [7]. To alleviate the the sparsity problem of memory-based approaches, model-based CF approaches such as matrix factorization (MF) and its variants [8], [9] project users and services into latent feature space independently, and then combine them together to reveal their linear interactive relationship between users and services for boosting the accuracy of QoS prediction by using inner product operation. With the considerable

applicability of fully-connected deep neural networks [10], correlative efforts based on deep learning techniques [11], [12], [13] have been dedicated to further effectively mining the implicitly complex nonlinear interaction relationships from user-service historical QoS invocations, leading to better extraction of latent feature representations and more satisfactory QoS prediction accuracy.

Although existing model-based CF approaches can improve QoS prediction performance, they still suffer from two deficiencies. First, most of conventional approaches mainly focus on the problem of centralized QoS prediction, where user-service historical QoS invocations are aggregated for model training in a centralized manner, instead of distributed across multiple service users. That is, they have not taken into account distributed QoS prediction, which makes difficult to protect the QoS privacy information of users invoking Web services. Second, even though partial model-based approaches adopt a variety of advanced deep learning models for QoS prediction, they have ignored the hierarchical collaborative relationship to learn and represent latent features of users and services. However, it is observed that the similarity relationship among users at different levels significantly affects the feature extraction of users and services, thereby reducing the accuracy of service QoS prediction. Thus, current model-based approaches are incapable of both considering the privacy-preserving QoS prediction and more effectively learning the latent features of users and service for better prediction performance.

To address the above two issues, we propose a novel federated hierarchical clustering framework for distributed QoS prediction called FHC-DQP, including three mutually correlative procedures. Starting from the original set of users, we collaboratively perform distributed federated training on independent users' QoS invocations, by aggregating and optimizing the federated parameters of centralized QoS prediction model until it reaches the global convergence. Then, the extracted federated users' private features are fed to clustering algorithm for partitioning users into a set of clusters. Subsequently, we hierarchically continue the process of distributed federated training and clustering on each partitioned user cluster, and it terminates when the number of each user cluster or partitioning depth satisfy its upper bound constraints. By several iterative rounds of federated hierarchical clustering, original set of users are fine-grained partitioned together where those users within the same cluster have stronger collaborative relevance for better embedding and learning the latent features of users and services. Finally, we apply the trained personalized QoS prediction model of each user to predict unknown QoS in distributed way, where graph diffusion strategy is taken to combine users and services location information at different levels based on its relevance to QoS prediction. Extensive experiments are conducted on a public large-scale benchmarking dataset called WS-DREAM [14], involving 5,825 real-world Web services from 74 regions and 339 service users from 31 regions. It records the total number of 1,974,675 user-service QoS invocations, which is distributed and partitioned in terms of users. By comparing FHC-DQP with a bunch of centralized as well as federated competing baselines, the results demonstrate its effectiveness on multiple evaluation metrics for distributed QoS prediction.

In general, the main contributions of this paper are summarized as follows:

- We propose a novel privacy-preserving collaborative federated framework for distributed QoS prediction, which leverages federated learning to protect privacy information of user-service QoS invocations in real service-oriented application scenarios, compared to traditionally centralized QoS prediction.
- To better improve the performance of distributed QoS prediction, we propose a novel federated hierarchical clustering algorithm that iteratively performs multi-stage federated collaboration learning and user clustering. It enhances the ability of learning the latent features of users and services by much stronger collaborative relevance within a fine-grained cluster in personalized QoS prediction model, where location contextual information is incorporated into our designed deep neural network.
- We conduct extensive experiments on WS-DREAM. The experimental results validate that FHC-DQP receives superior QoS prediction performance over existing federated baselines, and achieves a trade-off performance compared to start-of-the-art centralized ones.

The remainder of this paper is organized as follows. Section 2 defines and formulates distributed QoS prediction problem. Section 3 elaborates our proposed approach of FHC-DQP. Section 4 shows and analyzes the experimental results. Section 5 reviews the related work. Finally, Section 6 concludes the paper and discusses the future work.

2 Problem Formulation

Definition 1 (Web Service). A Web service is defined as a five-tuple $i =< n, f, d, q, ID, l >$, where n, f and d represent service name, functionality description and domain tag, respectively. q is the QoS criteria and l indicates the location information of a Web service.

For QoS prediction problem, we mainly pay attention to the non-functional features of a Web service including QoS criteria q, identifier ID and location information l.

Definition 2 (Service User). A service user is defined as a two-tuple $u =< ID, l >$, where ID is the identity of a user, and l indicates the respond location information.

Generally, the location information includes the Region, Autonomous System (AS), longitude and latitude.

Definition 3 (User-Service Invocation Record). A user-service invocation record is defined as a three-tuple $< u, i, r_{u,i} >$, where $u \in U$ is a service user, $i \in I$ is a Web service, and $r_{u,i}$ is the QoS value when u invokes i.

By aggregating all the user-service invocation records, we can obtain a user-service invocation matrix R, where one row represents the QoS values of a user invoking Web services, and one column represents the QoS values of a service invoked by service users. Here, if a user u has invoked a service i , we have $< u, i, r_{u,i} > \in R$, otherwise $< u, i, r_{u,i} > \notin R$.

Definition 4 (Centralized QoS Prediction Problem). Given a user set U, a service set I and observed QoS invocation matrix R, centralized prediction problem is defined as a five-tuple $CQP =< U, I, R, u, i >$, where u is a target user, i is a target service, and $< u, i, r_{u,i} > \notin R$.

Given a centralized QoS prediction problem $CQP =< U, I, R, u, i >$, the goal of a centralized prediction model is to calculate the missing QoS $\widehat{r}_{u,i}$. Thus, a corresponding solution to CQP can be denoted as $< u, i, \widehat{r}_{u,i} >$.

Definition 5 (Distributed QoS Prediction Problem). Given a set of user-service invocation submatrices $R' = \{R_1, R_2, ..., R_n\}$, where n is the number of users, a distributed QoS prediction problem is defined as a five tuple $DQP =< U, I, R', u, i >$, where u is a target user and i is a target service. The solution to a DQP problem is $< u, i, \widehat{r}_{u,i} >$ of the target user u invoking i.

Here, the significant difference between a CQP and DQP problem is that centralized QoS prediction model can learn the complex nonlinear invocation feature from the aggregated invocation matrix R, while distributed QoS prediction models can only collaborate by their independent submatrices $R' = \{R_1, R_2, ..., R_n\}$.

3 Approach

3.1 Overview

Fig. 1 illustrates the overall framework of FHC-DQP. It consists of three components: Federated User Private Feature Extraction (FUPFE), Federated Hierarchical Clustering (FHC) and Personalised QoS Prediction (PQP). The process of each component in FHC-DQP is described as below.

- In the module of federated user private feature extraction, all users preserve their private QoS data locally and collaborate with each other by transmitting their federated parameters of QoS prediction model to cloud center, which employs federated learning by aggregating all of the propagated model parameters iteratively to train the central model. After the global process satisfies the convergence condition, we can extract the vector representation of user private features.
- In the module of federated hierarchical clustering, it initially applies an effective clustering algorithm (such as K-means++) to partition a set of users into several clusters based on the extracted user private features. For each partitioned user cluster, user hierarchical clustering subsequently carry out the process of federated user private feature extraction and K-means++ clustering recursively in a gradually narrowing range of users, which finally shapes the users like a tree structure as the hierarchical clustering terminates.
- In the module of personalized QoS prediction, a location-aware deep neural network is designed as unknown QoS prediction model, where both the federated and private parameters are trained by the previous process of federated user private feature extraction and federated hierarchical clustering. As a result, it can be leveraged to predict vacant QoS values for a target user when invoking Web services in personalised way.

3.2 Federated User Private Feature Extraction

Based on deep neural networks, a centralized QoS prediction model is trained by historical QoS invocation records of all users. It can be expressed:

$$y = f(x; w) \tag{1}$$

where w is the parameters of centralized neural network. In distributed QoS prediction, w can be optimally learned by federated aggregation strategy after finishing each round local training of each user, which is then propagated back to all selected users' QoS prediction models. Consequently, each user's QoS prediction model can also be expressed as Equation (1) and w is equivalent to all users' model parameters. Nevertheless, different from the applications in object detection [15] and NLP [16], the parameters of personalised models in distributed QoS prediction are divided into two parts, including federated parameters and private parameters. Federated parameters participate in federated aggregation of centralized neural network model, while private parameters jointly train with federated parameters for users' personalised QoS prediction models, but they keep private and independent with users. For a user k, the personalised QoS prediction model can be expressed as:

$$y = f(x; \widetilde{w}, w_k(k > 0)) \tag{2}$$

where $\widetilde{w}$ is federated parameters and w_k is the private parameters of the user k. Fig. 2 shows the division of private and federated parameters, where centralized QoS prediction model is trained by the collaboration of federated users. Their corresponding personalised QoS prediction models extract users' private features in black part.

To optimally learn the federated parameters $\widetilde{w}$ and private parameters w_k of personalised QoS prediction models, they collaborate with centralized QoS prediction model by propagating federated parameters. Their collaborative objective function is expressed by:

$$\mathcal{J} = \min_{\mathcal{L}_k, 1 \le k \le n, k \in P_t} G(\mathcal{L}_1(\widetilde{w}, w_1), \ldots, \mathcal{L}_k(\widetilde{w}, w_k)) \tag{3}$$

where n is the total number of users, P_t indicates a selected subset of users, $\mathcal{L}_k(\widetilde{w}, w_k)$ is the personalised objective function of user k, and $G(\bullet)$ is a global objective function that combines the selected objectives. To further consider the importance of each user, a weighted average of local losses, i.e., $\sum_{k=1}^{|P_t|} p_k \mathcal{L}_k(\widetilde{w}, w_k)$ and satisfy $\sum_{k=1}^{|P_t|} p_k = 1$. By applying FedAvg aggregation algorithm, the parameters of centralized QoS prediction model, are updated with the weighted sum of parameters from all the selected personalised QoS prediction models. It can be expressed by:

$$(\widetilde{w}_{t+1}^k, w_{t+1}^k) \leftarrow BackPropagation(W_t, w_t^k)\ (k \in |P_t|) \tag{4}$$

$$W_{t+1} = \sum_{k \in P_t} \frac{n_k}{n_s} \widetilde{w}_{t+1}^k, n_s = \sum_{k \in P_t} n_k \tag{5}$$

where $\widetilde{w}_t^k$ denotes the federated parameters of the user k in t and W_t denotes federated parameters of centralized model; w_t^k and w_{t+1}^k indicate the private parameters of the user k in t and $t+1$; n_k and n_s denote the training number of the user k and the total training number, respectively.

Based on the federating training between centralized and personalised QoS prediction models, federated user private feature extraction (*FUPFE*) is described in Algorithm 1. Specifically, it iteratively performs the federated training by mutually propagating federated parameters (Lines 2-8). At each iteration round, the personalised federated parameters from each selected user are initially updated by the user's personalised QoS prediction model and the latest shared

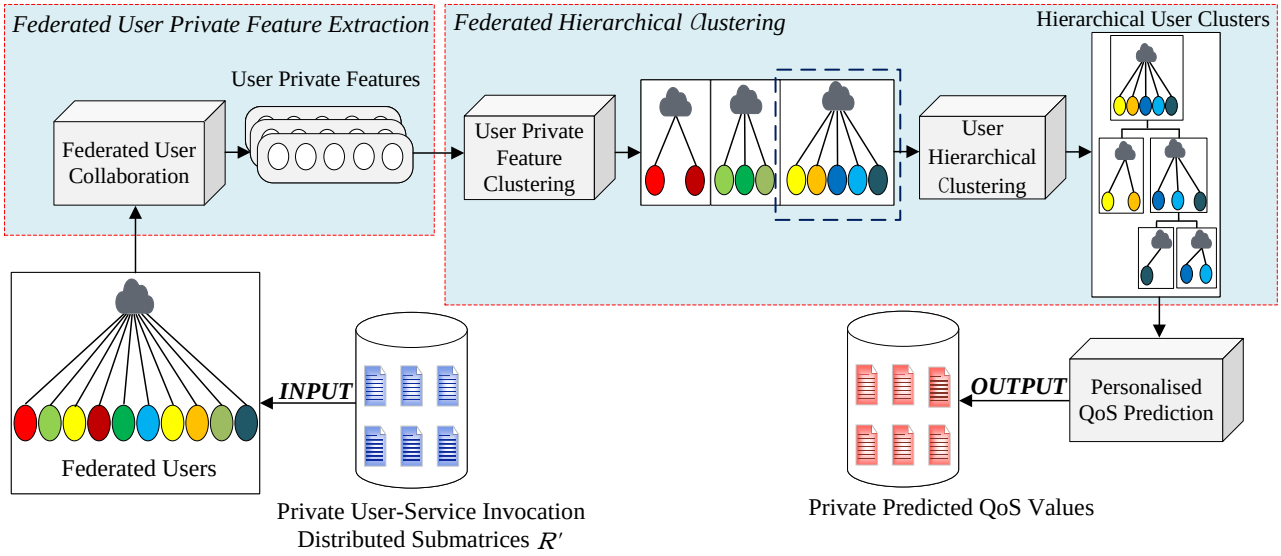


Fig. 1: Overall framework of FHC-DQP. It consists of three parts: Federated User Private Feature Extraction collaboratively trains a global QoS prediction model and extracts users' private embedding features; Federated Hierarchical Clustering recursively partitions users into a set of tree structure clusters in a gradually narrowing range of users; Personalized QoS Prediction designs a location-aware deep neural network to perform QoS prediction in a personalized way.

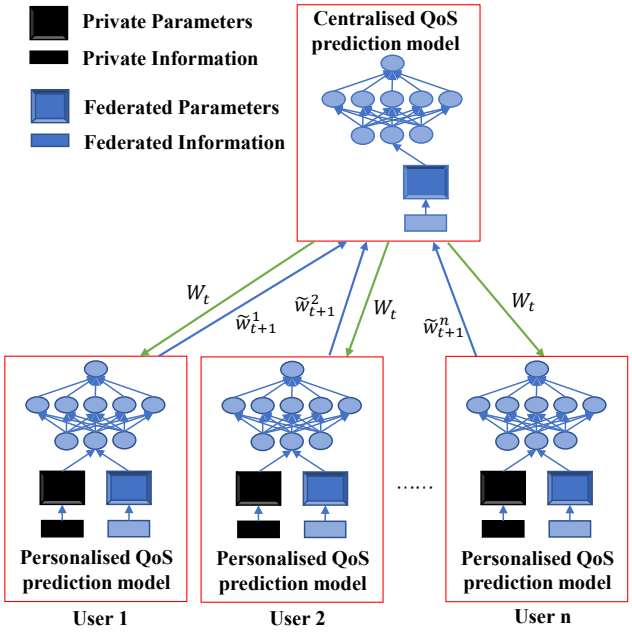


Fig. 2: Federated user collaboration of private feature extraction.

parameters from centralized QoS prediction model (Lines 3-5). Then, all of these updated federated parameters are aggregated together as the newly centralized parameters at the next time (Lines 6-7), until it terminates when the global objective function J reaches its convergence condition in Equation (3) (Line 8). Finally, we extract each user's private feature vector f_u from the corresponding personalised QoS prediction model and combine them into a feature matrix F_U (Lines 9-13). Note that we choose all users as selected user set P_t, which means $|P_t|$ is equivalent to n. Here, the federated user private feature refers to ID embedding or an integrated embedding from ID, Region, AS and location.

Algorithm 1 : Federated User Private Feature Extraction (FUPFE)

Input:
P_t: one user cluster
Output:
F_U: federated user private feature matrix

1: **Initialize** round $t = 0$
2: **repeat**
3: **for** each $k \in P_t$ **in parallel do**
4: $\widetilde{w}^k_{t+1} \leftarrow PersonalisedFederatedUpdate(k, W_t)$
5: **end for**
6: $W_{t+1} \leftarrow \sum_{k \in P_t} \frac{n_k}{n_s} \widetilde{w}^k_{t+1}$
7: $t \leftarrow t + 1$
8: **until** $\mathcal{J}$ reaches convergence in Equation (3)
9: **for** each $u \in P_t$ **in parallel do**
10: f_u obtains user u private feature vector
11: $F_U \leftarrow F_U \cup f_u$
12: **end for**
13: **return** F_U

Since the common FL paradigm is applied within FHC-DQP, each client user keeps their user-service QoS invocations locally and does not share them with other users when performing distributed model training of local QoS prediction. That is, the distributed privacy safeguard can be partially achieved by uploading the trained model parameters to the server for federated aggregation, instead of raw user-service QoS invocations. Moreover, advanced and efficient encryption algorithms are expected to be adopted for further enhancing the privacy safeguards by encrypting the shared model parameters for aggregation across multiple client users.

3.3 Federated Hierarchical Clustering

3.3.1 User Private Feature Clustering

Due to the global updating of shared federated parameters and the local optimization of private parameters,

the values of federated parameters in each user model are similar after several rounds of federated training, which can train a general global model. Intuitively, the difference in QoS value of users maps into distinct private feature vector in latent space, similar to word2vec [17] and doc2vec [18]. In the latent space, users who have similar QoS values tend to be closer with each other. After acquiring user private features, it predefines a hyper-parameter k in K-means++ clustering algorithm to partition users into a set of clusters.

We use the Euclidean distance to measure the feature vectors between a user i and a centroid j:

$$dist(u_i, c_j) = \sqrt{\sum_{k=1}^{n} |u_{ik} - c_{jk}|^2} \quad (6)$$

Algorithm 2 : User Private Feature Clustering (UPFC)

Input:
F_U: federated user private feature matrix
k: the number of predefined clusters
Output:
U_{fc}: the partitioned k user clusters
Parameters:
u_i: user i private feature embedding vector

1: **Initialize** k cluster centroids $\{c_0, c_1, ..., c_{k-1}\}$
2: **Initialize** $n \leftarrow$ the number of rows in matrix F_U
3: **repeat**
4: **for** i = 1 **to** n **do**
5: $u_i \leftarrow$ (i - 1)th row in F_U
6: $z_i \leftarrow argmin_k$ ($dist(u_i, c_k)$ in Equation 6)
7: **end for**
8: **for** j = 0 **to** $k-1$ **do**
9: $c_j \leftarrow MEAN(\{u_i, z_i = j\})$
10: **end for**
11: **until** converged
12: **for** j = 0 **to** $k-1$ **do**
13: $U'_{fc} \leftarrow U'_{fc} \cup (\{i, z_i = j\})$
14: **end for**
15: $U_{fc}(U_0, U_1, ..., U_{k-1}) \leftarrow$ construct virtual centralized nodes for each user cluster of $U'_{fc}(U'_0, U'_1, ...U'_{k-1})$
16: **return** U_{fc}

The process of user private feature clustering (*UPFC*) can be shown in Algorithm 2. First, it exploits K-means++ algorithm to divide a father cluster into k son clusters. In K-means++, by applying the above distance calculation, we calculate the distance between two user private features and allocate each user to the corresponding subcluster. Then, k new virtual nodes with federated center are constructed for subclusters, and the central models in those nodes inherit initial parameters from their father cluster's centralized model. For example, given user set $U = \{u_1, \ldots, u_{10}, \ldots\}$ and cluster number k, we can obtain $U_1 = \{u_1, u_4, \ldots\}$, $U_2 = \{u_2, u_5, u_9, \ldots\}$, $U_3 = \{u_3, u_6, u_7, u_8, u_{10}, \ldots\}$ by *UPFC* and satisfy following conditions:

$$U_1 \cap \ldots \cap U_k = \varnothing \quad (7)$$

$$U_1 \cup \ldots \cup U_k = U \quad (8)$$

UPFC clustering algorithm is beneficial to similar users' collaborative signal that can improve QoS prediction accuracy based on implicit collaborative filtering of user private feature vectors. As a result, users partitioned in the same subcluster have closer relationship to collaboratively learn a more effective QoS prediction model than those neighborhoods in a different subcluster.

3.3.2 User Hierarchical Clustering

Despite the *UPFC* algorithm that partitions a father user cluster into a set of subclusters, abundant number of users may still cause the training process of QoS prediction model inadequately because it cannot utilize correlation well among users in the same subcluster. Thus, we set the hyper-parameter threshold where the number of each subcluster should satisfy the following convergence condition:

$$|U_i| <= threshold \quad (9)$$

Algorithm 3 : User Hierarchical Clustering (UHC)

Input:
U_{fc}: a set of user clusters
$depth$: maximum hierarchical tree depth
$threshold$: maximum number of users in one cluster
Output:
U_{res}: leaf-layer hierarchical clusters
Parameters:
$U_{current}$: user cluster set of current layer
U_{next}: user cluster set of next layer

1: **Initialize** $U_{current} \leftarrow U_{fc}$
2: **Initialize** $deep \leftarrow 2$
3: **repeat**
4: **for** each $U_i \subseteq U_{current}$ **in parallel do**
5: **if** $|U_i| \geq threshold$ **then**
6: $F_u \leftarrow$ *FUPFE*(U_i) % *call Algorithm 1*
7: $U_{next} \leftarrow U_{next} \cup$ *UPFC*(F_u) % *call Algorithm 2*
8: **else**
9: $U_{next} \leftarrow U_{next} \cup U_i$
10: **end if**
11: **end for**
12: $U_{current} \leftarrow U_{next}$, $U_{next} \leftarrow NULL$
13: $deep \leftarrow deep + 1$
14: **until** $deep > depth$ **or** $\forall\, U_i \subseteq U_{current}$, U_i *s.t.* Equation 9
15: $U_{res} \leftarrow U_{current}$
16: **return** U_{res}

Based on the above constraint, a tree structure user hierarchical clustering (UHC) is proposed to further subdivide a subcluster into a set of more fine-grained hierarchical clusters, leading to better QoS prediction model training by deeper clustering partition for more precise collaborative relationships. Here, root node is the initially first-layer central node and leaves indicate the final elements of hierarchical clusters. The intermediate tree nodes between root and leaves are all constructed federated virtual nodes for the division of the leaves further.

By invoking the above Algorithms 1 and 2, the process of *UHC* is shown in Algorithm 3. It starts from a set of given clusters generated by *UPFC* (*Line 1*). In subsequent rounds, each of the user clusters $U_i \subseteq U_{current}$ is iteratively partitioned into a set of subclusters in parallel at the same depth of hierarchical clustering, when it satisfies the partitioning condition $|U_i| \geq$ threshold (*Line 5-7*), and then the subclusters of next layer continue Algorithms *FUPFE* and *UPFC* (*Line 3-14*). Consequently, we finish hierarchical clustering and generate a set of fine-grained clusters, where each has less than the predefined upper bound number of users or reaches given depth (*Line 14*). By the *UHC*

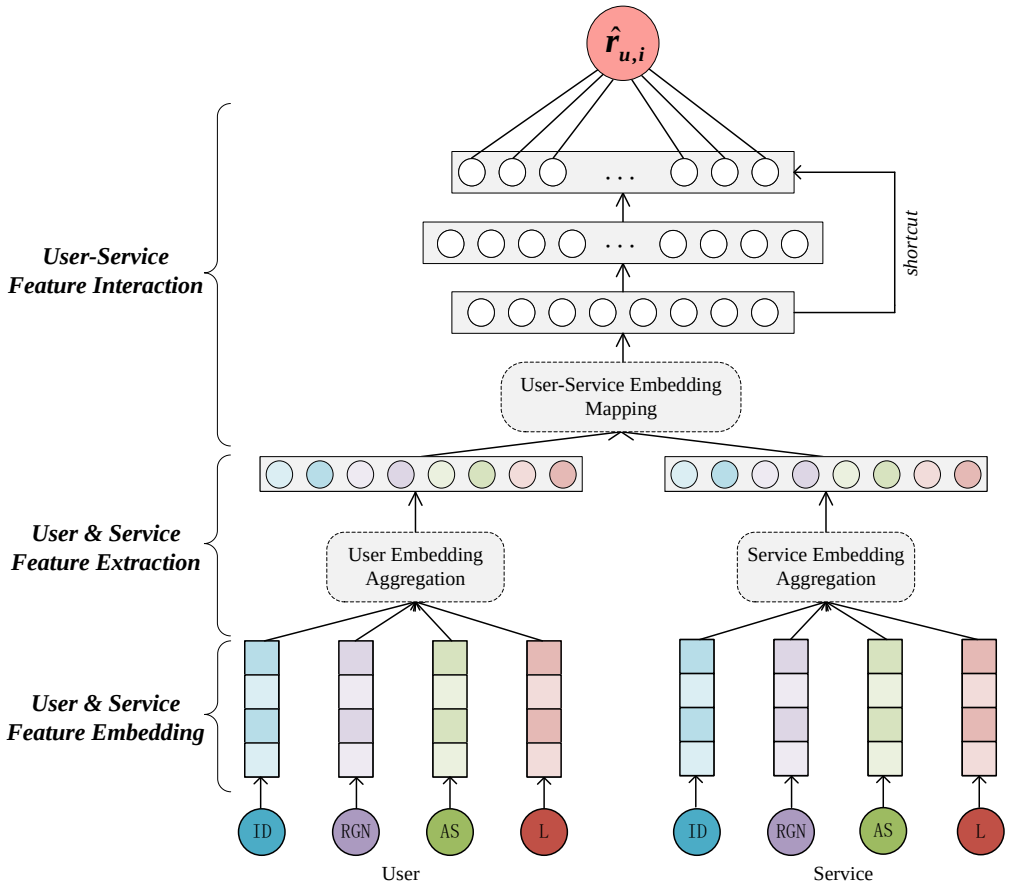


Fig. 3: Location-aware personalised QoS prediction model based on deep neural network.

algorithm, the hyper-parameters of each personalized QoS prediction model are trained and optimized, leading to a better performance of predicting vacant QoS of user-service invocations.

3.4 Personalised QoS Prediction

Fig. 3 illustrates the personalised QoS prediction model based on deep neural network, where a user's or a service's ID and multiple location information are taken into account to predict unknown QoS, including Region, AS, longitude & latitude. It consists of three independent but correlative layers. In the layer of User & Service Feature Embedding, it is observed that users with similar network environment often perceive similar QoS values on the same Web service [5]. These influence factors can be expressed by:

$$I^u = [U-ID, U-Region, U-AS, U-Location(longitude\ \&\ latitude)] \quad (10)$$

$$I^s = [S-ID, S-Region, S-AS, S-Location(longitude\ \&\ latitude)] \quad (11)$$

we first generate a user's one-hot embedding vector i_u^e, i_u^r, i_u^a, i_u^l, which separately represent high-dimensional sparse vector on diverse factors of user ID, Region, AS and Location. Similarly, i_s^e, i_s^r, i_s^a, i_s^l represent a service's one-hot embedding vectors on diverse factors of service ID, Region, AS and Location, respectively. Based on the above initial one-hot embedding representations, a user's and service's embedding feature on diverse factors can be expressed by:

$$e_u^1 = M_u^1 i_u^e,\ e_u^2 = M_u^2 i_u^r,\ e_u^3 = M_u^3 i_u^a,\ e_u^4 = M_u^4 i_u^l \quad (12)$$

$$e_s^1 = M_s^1 i_s^e,\ e_s^2 = M_s^2 i_s^r,\ e_s^3 = M_s^3 i_s^a,\ e_s^4 = M_s^4 i_s^l \quad (13)$$

where $M_u^1 \in \mathbb{R}^{n_u \times d_u}, M_u^2 \in \mathbb{R}^{n_r \times d_u}, M_u^3 \in \mathbb{R}^{n_a \times d_u}, M_u^4 \in \mathbb{R}^{n_l \times d_u}$ denote the embedding matrices of users or side information and n_u, n_r, n_a, n_{loc} are the number of ID, Regions, ASs, locations in recorded QoS data of users; $M_s^1 \in \mathbb{R}^{m_s \times d_s}, M_s^2 \in \mathbb{R}^{m_r \times d_s}, M_s^3 \in \mathbb{R}^{m_a \times d_s}, M_s^4 \in \mathbb{R}^{m_l \times d_s}$ denote the embedding matrices of services or side information and m_s, m_r, m_a, m_l are the number of ID, Regions, ASs, locations in services' historical invocation records; d_u and d_s are dimensionality of user and embedding features, respectively. By above Equations (12) and (13), we have transformed one-hot encoding vector of a user or service into dense and low-dimensional embedding feature vector.

In the layer of User & Service Feature Extraction, to incorporate these side information, we adopt weighting coefficients to integrate different kinds of the embeddings, which can be expressed by:

$$p_u = \sum_{k=1}^{m} w_k e_u^k \quad (14)$$

$$q_s = \sum_{k=1}^{m} w_k e_s^k \quad (15)$$

where m is the number of influence factors of integrated embeddings. We adopt $m = 4$ when the influential factors

including ID, Region, AS, longitude & latitude; e_u^k and e_s^k represents the k-th embedding feature of a user and a service on ID, Region, AS, longitude & latitude, and w_k is corresponding weights to e_u^k and e_s^k, subject to $\sum_{k=1}^{m} w_k = 1$ and $w_1 > w_4, w_3 > w_2$.

To effectively reflect the importance of different influence factors and evaluate the weighting coefficients, we borrow the propagation mechanism of graph diffusion network that more relevant information for predicting unknown QoS has closer distance/hop with the center node. Here, ID of a user or service is viewd as 1-hop connectivity information, while AS and longitude & latitude view as 2-hop connectivity information, and Region as remote information is appointed as 3-hop connectivity information. To represent impact of distance simply, we introduce parameter decay strategy of the personalised PageRank (PPR) graph diffusion [19], which is expressed by:

$$\theta_{PPR}^{d_k} = \alpha(1-\alpha)^{d_k} \tag{16}$$

where the value range of decay factor is $\alpha \in (0, 1)$ [20], and d_k represents the distance/hop of the k-th influence factor. Then we normalize each of them by its proportion:

$$w_k = \frac{\theta_{PPR}^{d_k}}{\sum_{k=1}^{m} \theta_{PPR}^{d_k}} \tag{17}$$

where w_i represents the weighting coefficient of each influence factor, and α plays a crucial role in evaluating the weighting coefficient and it needs to be determined in advance.

In the layer of User-Service Feature Interaction, the integrated feature embeddings of users and services are all dense and low-dimensional vectors and they are fused as the input into a deep neural network for interactive learning. Given two integrated embedding features p_u and q_s of a user and a service, the element-wise product operation is performed to obtain the initial interaction feature:

$$h_0 = p_u \odot q_s \tag{18}$$

where $\odot$ denotes the element-wise product of two feature vectors. Subsequently, we feed h_0 into a multi-layer perceptron (MLP) network for learning user-service complex nonlinear invocation relationship. The forwarding procedure in user-service feature interaction layer is expressed as:

$$\begin{aligned} h_1 &= ReLU(W_1 h_0 + b_1) \\ &\cdot \\ &\cdot \\ h_{K-1} &= ReLU(W_{K-1} h_{K-2} + b_{K-1}) \\ \widehat{r}_{u,i} &= W_K(h_{K-1} + h_0) + b_K \end{aligned} \tag{19}$$

where K is the number of hidden layers, W_x and b_x indicate weight matrix and bias vector. $ReLU$ is the activation function. Here, MLP is implemented by the typical tower structure. The output of the user-service feature interaction $\widehat{r}_{u,i}$ is the predicted missing QoS. Especially, we insert shortcut connection [21] in last layer to help training.

Considering the ability of fitting outliers, we use the minimum absolute (L1) loss function, which is more suitable to learn and optimize the parameters for QoS prediction problem in the training procedure. Let $r_{u,i}$ and $\widehat{r}_{u,i}$ are the original and predicted QoS values, the loss function of user u personalised QoS prediction model is defined as:

$$\mathcal{L}_u = \frac{1}{\mathcal{N}} \sum_i |r_{u,i} - \widehat{r}_{u,i}| \tag{20}$$

Stochasticity in the training process is introduced via dropout [22] to avoid model overfitting. We adopt Adaptive Moment Estimation (Adam) [23] as optimizer. To facilitate federated learning training procedure in terms of time and storage expense, several training epochs are set up in each personalised model during a round of the federal learning process.

TABLE 1: Notations of Overhead Analysis

Notation	Description
P_u	the set of participating users
D	the embedding size of user and service
T	the rounds of federated training
$\mathcal{M}$	the overall number of client parameters
E	the number of client epochs
I	the iteration number of K-means++
K	the clusters number of K-means++
H	the layer number of federated hierarchical clustering

3.5 Overhead Analysis

The overhead of FHC-DQP mainly includes computational burden and communication expenses, which can be formally analyzed and expressed by the notations as listed in TABLE 1.

The computational burden of FHC-DQP is mainly determined by two factors. (i) Distributed client model training. In the FHC-DQP client model, the training complexity of both forward and backward propagation is determined by the product of the number of neurons in adjacent layers, where the number of neurons in each layer is the multiple of embedding size. Therefore, the training complexity can be expressed as $O(D^2 * E * T)$. (ii) K-means++ clustering. After federated training convergence, the K-means++ algorithm is utilized to generate subclusters of user latent vectors, the computational cost of which is $O(|P_u| * D * K * I)$. Thus, the total computational burden is the sum of client model training and the K-means++ clustering for H iterative rounds of federated hierarchical clustering, which is expressed as $O((D^2 * E * T + |P_u| * D * K * I) * H)$. In real-world application contexts, it generally satisfies that T, D, E, K are much smaller than $|P_u|$ (*i.e.*, $|P_u| >> T, D, E, K$). In such case, the computational burden can be expressed as $O(|P_u| * I)$.

As for communication expenses, during the first round of FL communication, the server distributes the initial global parameters $\mathcal{M}$ to all participating users P_u. At the end of the round, it downloads the updates with the size of $\mathcal{M}$ client parameters from each user and performs federated aggregation. The above processes are iterated for T communication rounds. Thus, the total communication expenses are calculated as $O((2 * |P_u| * \mathcal{M}) * T + |P_u| * \mathcal{M})$, which can be expressed as $O(|P_u| * \mathcal{M} * T)$.

By the above analyses, the comprehensive overhead of FHC-DQP is $O(|P_u| * I + |P_u| * \mathcal{M} * T)$. When the number

TABLE 2: Statistics of QoS Dataset in WS-DREAM

Item	Value
Users	339
Services	5825
Users' Regions	31
Users' AS	137
Services' Regions	74
Services' AS	992
Services' Providers	2699
Range of RT	0-20
Range of TP	0-1000

of participating users is large enough in real-life service-oriented application contexts, it can be expressed as $O(|P_u|)$ that is linear with the number of participating users, indicating the practicability of employing FHC-DQP.

4 Experiments

4.1 Experimental Setup and Dataset

All the experiments are carried on our workstation equipped with two NVIDIA GTX 1080TI GPU, an Intel(R) Xeon(R) Gold 6130 @2.60GHz CPU and 192GB RAM. The components of FHC-DQP are implemented by Python 3.6 with Pytorch 1.6.0.

To verify the performance of FHC-DQP, we conduct extensive experiments on WS-DREAM [14], which has been widely used for validating the performance of vacant QoS prediction. It has two kinds of user-service QoS invocations, including response time (RT) and throughput (TP), which totally involves 1,974,675 historical QoS invocation records collected from 339 users and 5,825 Web services. For both RT and TP QoS records, they can be formalized as a QoS matrix, where a row consists of a bunch QoS entries indicating a corresponding user who invokes all of the Web services, and a column has a bunch of QoS entries indicating a corresponding Web service that is invoked by all of the users. For the demands of learning and representing features of users and services, location information in WS-DREAM, such as region, latitude and longitude, is also used for the training of personalized QoS prediction model. The statistics of WS-DREAM is shown in TABLE 2.

To simulate the sparsity of user-service invocations in real application scenarios, we partition the QoS records into four matrix densities (MD) in the experiments, including 5%, 10%, 15% and 20% by randomly removing large number of QoS invocation records from WS-DREAM. For the comparisons of QoS prediction accuracy among centralized and federated competing baselines, we respectively choose 90% and 10% as QoS invocations as training set, and 100,000 user-service QoS records that do not belong to both the training and validation set are selected as the testing set.

4.2 Competing Methods and Evaluation Metrics

To evaluate the performance of FHC-DQP, we compare it with eleven widely-used competing baselines, including two centralized memory-based approaches [6], [24], four centralized model-based approaches [8], [9], [25], [26], two hybrid model [27], [28] and three federated approaches [29], [30], [31]. They are described as below.

- **Centralized Methods**:
  - **UIPCC [6]:** It is the most representative memory-based CF for hybrid QoS prediction, which combines UPCC and IPCC together by weighting coefficient.
  - **LACF [24]:** It is a typically memory-based CF for QoS prediction, which takes advantage of users' and services' location contextual information as heuristics to more effectively calculate similar neighborhood.
  - **NMF [8]:** It is a representative non-negative matrix factorization approach, which decomposes user-service QoS invocation matrix to two latent non-negative matrices when original matrix elements are non-negative.
  - **PMF [9]:** It is a model-based representative approach for predicting vacant QoS by probabilistic matrix factorization, which leverages Gaussian distribution to optimize probability model.
  - **NCF [25]:** It is an advanced neural collaborative filtering approach that combines multi-layer perceptron and generalized matrix factorization for recommendation.
  - **LRMF [26]:** It is a location and reputation model-based baseline, which combines both the user's reputation and location information into matrix factorization.
  - **PSO-USRec [27]:** It is a global search optimization model for QoS prediction. It customizes particle swarm optimization and smoothes outlier particles to enhance the performance of QoS prediction.
  - **HAP [28]:** It is a case-based reasoning hybrid model for QoS prediction. It establishes an hybrid model by taking advantage of enhanced PCC to calculate similarity and global case-based reasoning to seek a global prediction.
- **Federated Methods:**
  - **FedMF [29], [30]:** It is the recently representative federated collaborative filtering approach that integrates federated learning and matrix factorization to protect user privacy and improve prediction accuracy.
  - **NCSF-GMF [31]:** It is a privacy-preserving federated GMF approach where users collaboratively upload perturbed updates during server aggregation without affecting the global model.
  - **FedGMF:** It is a self-developed variant as a completing baseline for distributed QoS prediction based on generalized matrix factorization [25] in federated environment, which performs QoS prediction without the consideration of federated hierarchical clustering.

Mean absolute error function (MAE) and root mean squared error (RMSE) are used as our two evaluation metrics to compare the performance of QoS prediction in the experiments. MAE and RMSE are defined as follows:

$$MAE = \frac{\sum_{i,j} |r_{i,j} - \widehat{r}_{i,j}|}{N} \tag{21}$$

$$RMSE = \sqrt{\frac{\sum_{i,j} \left(r_{i,j} - \widehat{r}_{i,j}\right)^2}{N}} \tag{22}$$

where $r_{i,j}$ is the ground-truth QoS value of a user invoking a service and $\hat{r}_{i,j}$ is the predicted QoS; N is the number of test samples of predicted QoS values.

TABLE 3: Results of Performance Comparisons of QoS Prediction on Response Time

Approaches	MD=0.05		MD=0.1		MD=0.15		MD=0.2	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UIPCC	0.625	1.412	0.582	1.330	0.501	1.250	0.450	1.200
LACF	0.630	1.439	0.560	1.338	0.510	1.269	0.477	1.222
NMF	0.546	1.473	0.478	1.283	0.447	1.202	0.427	1.163
PMF	0.569	1.537	0.486	1.316	0.452	1.220	0.430	1.169
LRMF	0.555	1.500	0.485	1.300	0.454	1.210	0.435	1.160
NCF	0.556	1.421	0.500	1.345	0.488	1.245	0.464	1.212
PSO-USRec	0.565	1.358	0.506	1.274	0.471	1.222	0.444	1.186
HAP	0.539	1.458	0.472	1.275	0.438	1.195	0.422	1.157
FedMF	0.727	1.505	0.595	1.326	0.540	1.243	0.522	1.227
NCSF-GMF	0.569	1.557	0.481	1.456	0.453	1.340	0.400	1.286
FedGMF	0.547	1.467	0.468	1.436	0.436	1.245	0.403	1.256
FHC-DQP	0.510	1.400	0.434	1.316	0.395	1.236	0.338	1.205
Gains	**5.69%**	**-3.09%**	**7.83%**	**-3.20%**	**10.38%**	**-3.43%**	**19.23%**	**-4.15%**

TABLE 4: Results of Performance Comparisons of QoS Prediction on Throughout

Approaches	MD=0.05		MD=0.1		MD=0.15		MD=0.2	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UIPCC	26.759	60.799	22.370	54.456	20.219	50.704	18.928	48.295
LACF	22.973	58.786	19.449	52.927	17.589	49.565	16.458	47.411
NMF	21.885	60.530	16.579	49.824	15.856	46.803	14.592	42.440
PMF	19.079	57.888	15.995	48.080	15.082	46.050	13.923	42.164
LRMF	19.109	58.072	15.950	48.272	14.600	45.070	13.920	41.805
NCF	26.656	65.719	19.188	51.527	17.668	51.807	16.053	47.384
PSO-USRec	23.332	60.155	19.740	54.254	17.839	50.209	16.787	47.395
HAP	18.374	54.158	15.529	48.747	14.502	45.147	13.517	42.828
FedMF	25.657	62.255	21.216	55.157	19.259	52.054	16.823	50.828
NCSF-GMF	18.079	63.888	21.995	55.080	17.082	49.050	14.923	47.164
FedGMF	18.014	53.456	16.655	48.527	15.422	46.915	14.719	45.053
FHC-DQP	17.266	51.520	14.347	46.605	13.495	44.916	12.638	42.197
Gains	**4.33%**	**3.76%**	**8.24%**	**3.16%**	**7.46%**	**0.34%**	**6.96%**	**-0.94%**

4.3 Experiment Results and Analyses

To validate the effectiveness and reduce the overhead of our proposed FHC-DQP, we adopt 10 epochs to user's personalized QoS prediction model for one round of federated user collaboration. The user-service interaction layer of deep neural network is set as [32,32,8] and the dropout rate in personalized QoS prediction model is set as 0.1. In model training of FHC-DQP, Adam learning rate is equal to 0.001 and batch size is equal to 64.

In the experiments, we tune the model parameters of competing methods directly as they are suggested with the best performance in the experiments of the references. Furthermore, we implement all of centralized and federated competing baselines by: (1) For UIPCC, we use Pearson Correlation Coefficient to calculate the similarity between users and services; (2) For LACF, NMF, PMF, and LRMF, L2 is applied as the loss function in model training; (3) NCF and FedMF are implemented based on the released code[1] [2]; (4) The results reported in the PSO-USRec are used directly; (5) HAP and NCSF-GMF are implemented by the description of [28], [31]; (6) As the first layer result of FHC-DQP, we implement FedGMF with the same parameters setting of FHC-DQP. All the competing baselines are run on both RT and TP training sets, and QoS prediction performance is evaluated on the test sets by calculating MAE and RMSE. To prevent the deviations, we run FHC-DQP and competing baselines a set of times to calculate the average results for the guarantee of fair comparisons of QoS prediction.

TABLE 3 and TABLE 4 show the experimental results of QoS prediction on RT and TP among both centralized and federated competing baselines. Here, lower MAE and RMSE indicate better performance of QoS prediction. The best and second-best results of each column are marked in dark and light grey, respectively. As matrix density varies from 5% to 20% with an interval of 5% on RT and TP, it is observed that all of the competing baselines have a continuing upward tendency of QoS prediction accuracy, since higher matrix density can provide more sufficient user-service QoS invocations for more effective model training and parameter optimization, which is beneficial to improve the QoS prediction performance across multiple competing baselines.

Specifically, as expected of our FHC-DQP, it becomes gradually better and overall receives superior QoS prediction performance across diverse QoS matrix densities compared with both centralized and federated competing baselines. With regard to the results of centralized QoS prediction, UIPCC as the traditionally representative CF approach performs poorly on both MAE and RMSE, because it mainly depends on limited user-service QoS invocations to find similar neighborhood. Thus, it is extremely sensitive to the sparsity of QoS matrix density. Compared with UIPCC, it takes advantage of geographical context as heuristics information for neighborhood selection, LACF obtains better

1. https://github.com/hexiangnan/neural_collaborative_filtering
2. https://github.com/Di-Chai/FedMF

QoS prediction accuracy than UIPCC. To alleviate the influence of sparsity of QoS density, NMF and PMF as two variants of matrix factorization achieve significant performance improvement compared to memory-based approaches. The primary reason is that they learn a QoS prediction model by user-service linear invocation relationship. LRMF mines latent user reputation and incorporates location context in model training, leading to better performance at high QoS densities compared with other MF-based QoS prediction approaches. To further improve the QoS prediction accuracy, NCF leverages multi-layer perceptron (MLP) that mines nonlinear interaction relationships from the embedded feature vectors of users and services. PSO-USRec adopts a swarm intelligence search for all unknown QoS records among users with outlier values correction, which is more effective than the other CF-based QoS prediction algorithms. HAP makes full use of both user and service information based on the case-based reasoning, and yields better prediction results at low QoS densities among the centralized baselines.

As for the results of distributed QoS prediction, FedMF is a leading federated regression approach and applies matrix factorization to learn a linear QoS prediction model, which is worse than centralized MF baselines. This phenomenon can be explained by the extensive investigations [32], [33] that have drawn a conclusion that centralized approaches show better performance than corresponding federated ones for prediction and recommendation tasks. Unlike first-order feature interaction by linearly combining latent features of users and services in FedMF, our self-developed federated variant FedGMF exploits high-order cross features to deeply learn complex nonlinear invocation relationship between users and services, leading to better distributed QoS prediction performance. NCSF-GMF achieves stronger privacy protection by uploading perturbed updates, resulting in a reduction of QoS prediction accuracy. Furthermore, our proposed FHC-DQP not only considers hierarchical user clusters for stronger collaborative relationships, but also leverages context-aware deep neural network to more effectively extracting latent features of users and services. As a result, it can outperform FedMF and FedGMF on MAE across two QoS datasets, where it is larger than approximately 10% when MD is set as 15% and 20%, respectively. As for RMSE, it also gains superior performance on RT and TP, although it is slightly worse in some cases, such as MD equal to 10%, 15% and 20% on RT, and 20% on TP. The possibility from [33] is that centralized competing baselines can capture stronger collaborative relationship in higher data density, while federated ones may incline to fall into unpredicted suboptimal points with facing the large data scale.

By systematically comparing multiple advanced distributed and centralized QoS prediction approaches, it is observed that our proposed FHC-DQP achieves the best performance in terms of both MAE and RMSE, demonstrating its advantages for potential applicability in real-world situations. However, it is expected for the improvement of QoS prediction accuracy from the two aspects. First, it is limited by the effectiveness of the personalized QoS prediction model and promises to further boost the QoS prediction accuracy by designing new or plugging in the existing more sophisticated QoS prediction model in our overall framework, while achieving the goal of privacy safeguards of user-service QoS invocations. Second, more advanced deep or heuristic clustering algorithms can be utilized to replace K-means++ in federated hierarchical clustering to find more relevant user subclusters based on their latent representations.

4.4 Performance Impact of Parameters

4.4.1 Impact of Cluster Number

Cluster number k is an important parameter in federated user private feature clustering, which determines the cluster number divided for each set of users during federated hierarchical clustering.

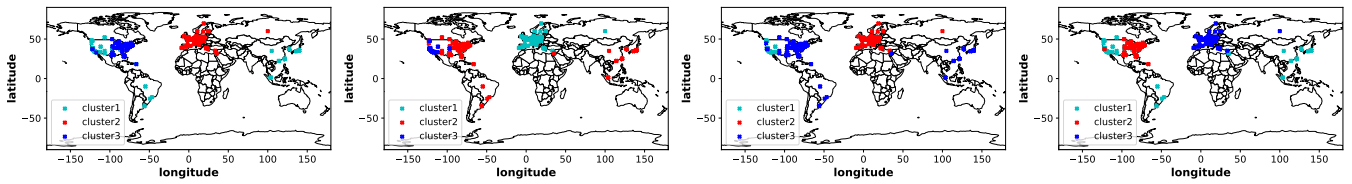
To observe the results of partitioned user clusters, we project each user into 2D-space based on their longitude $\&$ latitude. The 2D visualized user clusters are shown in Fig. 4, where the parameter of cluster number is set to 3 and diverse colors represent three different user clusters. The horizontal and vertical coordinates of the point correspond to the longitude and latitude of a user, respectively. From the results, it is observed that the segmented clusters are more cohesive with closer distance as the QoS densities arise from 0.05 to 0.2, since federated user private features can be better extracted for clustering from higher QoS matrix density.

To further test the performance impact of cluster number on distributed QoS prediction, we set it by $\{1, 2, 3, 4, 5, 6, 7, 8\}$ in the experiments. The results with the changes of cluster number are illustrated in Fig. 5. It can be observed that both MAE and RMSE decrease as the number of partitioned clusters increase initially, and then decline with the increasing cluster number. This phenomenon is reasonably explained that when the cluster number is set too small, those federated users with faint collaborative relationships may incur noisy information to extract unfavorable latent features of users and services, which significantly lowers QoS prediction accuracy. In another extreme case, if the cluster number is set to be a large value, very few of federated users within the same cluster and low density of invocation records both aggravate the situation that fails to effectively take advantage of collaborative relationship for better mining the complex nonlinear interactive invocations, leading to unsatisfactory QoS prediction performance. Based on the above analyses, when the cluster number of federated learning in FHC-DQP is set to 3 or 4 in our experiments, federated hierarchical clustering achieves the best QoS prediction accuracy under different QoS densities.

4.4.2 Impact of Feature Dimensionality

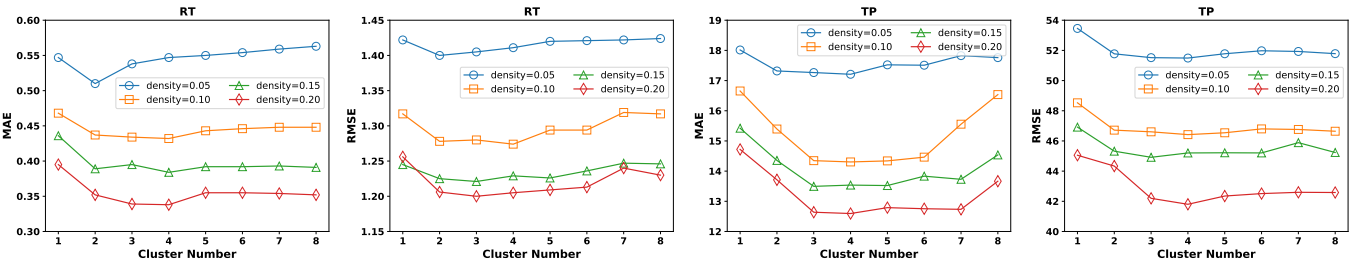
The dimensionality size d means the dimension of embedding vectors that is utilized to characterize the features of user, service and location information. To test the performance impact of dimensionality on QoS prediction, we vary its value d by $\{2, 4, 8, 16, 32\}$ and set matrix density (MD) as $\{0.05, 0.10, 0.15, 0.20\}$, respectively.

Fig. 6 shows the three-dimensional graph of QoS prediction accuracy with different combinations of d and MD. It is observed that both MAE and RMSE show a decreasing trend with the increasing number of matrix density on both RT and TP. Specifically, the QoS prediction accuracy is dramatically improved when the matrix density ranges



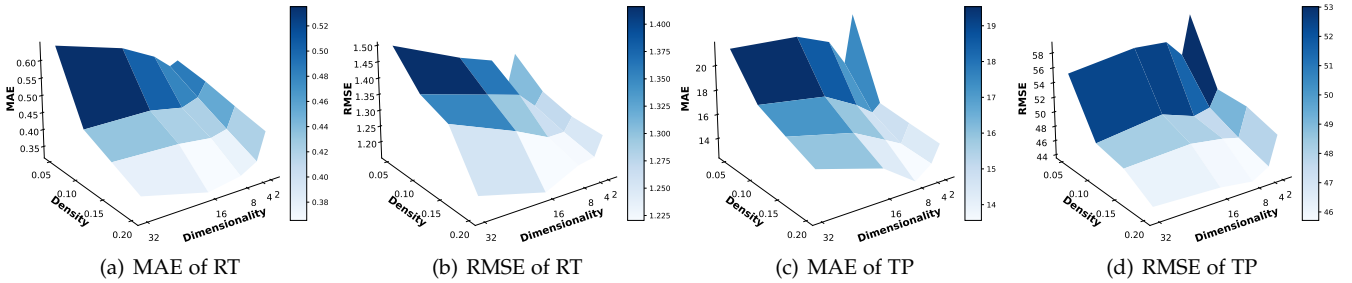
(a) Throughput (MD=0.05) (b) Throughput (MD=0.1) (c) Throughput (MD=0.15) (d) Throughput (MD=0.2)

Fig. 4: The 2D visualized segmented clusters with the changes of diverse QoS densities



(a) MAE of RT (b) RMSE of RT (c) MAE of TP (d) RMSE of TP

Fig. 5: Performance impact of distributed QoS prediction with the changes of cluster number



(a) MAE of RT (b) RMSE of RT (c) MAE of TP (d) RMSE of TP

Fig. 6: Performance impact of distributed QoS prediction with the changes of dimensionality and matrix density

from 5% to 10% and the dimensionality ranges from 2 to 8. It achieves the best QoS prediction performance when the dimensionality size d is set by 8 or 16 across different matrix densities. However, the accuracy of predicting unknown QoS begins to decline with the increasing dimensionality from 16 to 32. The reason for this phenomena is that, when projecting the feature vectors in a low-dimensional vector space, it may result in partial hidden information lost from feature latent vectors, which affects feature representation ability and reduces QoS prediction accuracy. Conversely, when the d is too large, the embedding dimension of feature representation may lead to the risk of over-fitting due to the small number of training samples of per user. It turns out that dimensionality size d can be set between the range from 8 and 16 to receive the superior prediction results under multiple different QoS densities.

4.4.3 *Impact of Decay Factor*

The value of decay factor reflects the importance of different influence factors of users and services when extracting their implicit features by integrating the embeddings of ID and multi-granularity geographical information. To investigate reasonable value of decay factor, we set α as $\{0.1, 0.4, 0.7, 1\}$, respectively. TABLE 5 summarizes the performance impact of decay factor on TP. We can find from the results that when it is set as $\alpha = 1$, FHC-DQP receives the worst performance on MAE and RMSE across most of

TABLE 5: Performance Impact of Decay Factor on TP.

	MD=0.05		MD=0.10		MD=0.15		MD=0.20	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
$\alpha = 0.1$	17.156	51.674	**14.193**	**46.595**	13.743	45.145	12.261	42.366
$\alpha = 0.4$	**17.096**	**51.511**	14.310	47.022	**13.487**	**45.046**	**12.130**	**41.886**
$\alpha = 0.7$	17.108	51.624	14.376	46.871	13.583	45.091	12.349	42.354
$\alpha = 1$	17.155	51.727	14.416	46.870	13.496	45.253	12.187	42.030

QoS densities. The reason is that it indicates an extreme special case where only ID information of users and services is considered, while other influence factors are omitted for implicit feature extraction. On the other hand, if the decay factor is set as a small value, FHC-DQP cannot effectively reflect the importance of key influence factor, leading to better representing implicit features. Thus, a medium value of decay factor is generally beneficial to receive the lowest MAE and RMSE for QoS prediction accuracy, which regulates the situations of combining all influence factors together by balancing these influence factors of users and services. As a result, when decay factor α is set to 0.4 in our experiments, FHC-DQP achieves the best distributed QoS prediction accuracy under most different QoS densities.

5 Related work

5.1 Centralized QoS Prediction

5.1.1 Memory-Based QoS Prediction

This kind of approaches has been widely investigated to predict unknown QoS by first performing users or services similarity calculation to obtain the neighbors of users or services, and then predicting unknown QoS through user-service historical invocation records based on average QoS and deviation migration calculation. Depending on the objective for calculating similarity, it can be categorized as user-based, service-based approaches, and their combination. Zheng et al. [6] propose a combination strategy that employs a adjusting parameter to coordinate the weighting of user-based and service-based predicted QoS for better prediction performance than each of them. Tang et al. [24] propose a novel approach called LACF, which takes location context into account when calculating similar users and services. To further improve the effectiveness of similarity calculation for QoS prediction, Sun et al. [34] propose a new similarity calculation method named normal recovery (NR), which normalizes the QoS values of users invoking services to the same range and then unifies the similarity of the scaled user/service vectors in different multi-dimensional vector spaces. Wu et al. [35] propose a ratio-based approach to calculate the similarity between users or services. Compared with cosine similarity and NR, it is more precise for predicting unknown QoS. Based on the ratio-based similarity calculation, Zou et al. [7] reduce partial of invoked services (or users) dissimilar with target service (or target user) when calculating average QoS and deviation migration for significant improvement on QoS prediction.

However, memory-based approaches are extremely vulnerable to the sparsity of user-service historical QoS invocations, which significantly affects the QoS prediction performance in real-world application scenarios.

5.1.2 Model-Based QoS Prediction

Matrix factorization (MF) and its variants, as the basic model-based method, directly embed user/service ID as a vector and model their linear interactions with inner product. Zhang et al. [8] design a non-negative matrix factorization (NMF) model, where non-negativity constraint is enforced in the linear model and can be applied for predicting vacant QoS values. Mnih et al. [9] propose probabilistic matrix factorization (PMF) that introduces probability model to optimize matrix factorization model, beneficial to improving QoS prediction performance compared to the traditional MF. By the combination of similar users and services, Zheng et al. [36] propose a hybrid model called NIMF, which integrates the neighborhood by similarity calculation based on user-service historical QoS invocations into matrix factorization model to achieve superior QoS prediction accuracy. To reinforce the feature representation from more heuristics of users and services, LRMF [26] combines both users' reputation and location information into matrix factorization for QoS prediction, which calculates the reputation of all users and then identifies the neighborhood based on user's reputation and geographical information. Compared to memory-based approaches, MF and its variants can relatively well predict vacant QoS by learning linear interaction relationships between latent features of users and services, which to some extent improves the QoS prediction performance. However, they cannot effectively mine the implicitly complex nonlinear interaction relationships from user-service historical QoS invocations, which still easily results in unsatisfactory accuracy of QoS prediction.

To model complicated nonlinear interactions, He et al. [25] propose neural collaborative filtering (NCF), which leverages a multi-layer perceptron (MLP) to learn the interactive function of nonlinear relationships. Based on NCF model, Zou et al. [11] propose neighborhood-integrated deep matrix factorization (NDMF), which fuses neighborhood selection loss term to L2 function, leading to better QoS prediction accuracy on both MAE and RMSE. Wu et al. [12] propose a deep neural model (DNM) to consider multiple attributes information of users and services for QoS prediction, where contextual features are mapped into latent space and their high-order interactions are captured through multi-layer perceptron network. Xia et al. [13] introduce implict and explict features into initial dense vector representation and utilize convolutional neural network (CNN) compress and optimize the procedure of feature extraction for QoS prediction. Li at el. [37] propose topology-aware neural (TAN) model, which introduces introduce network topology structure to assist in solving QoS prediction problem. Although multiple deep learning models have been well investigated to improve the performance of QoS prediction, they mainly focus on the designing of a centralized QoS prediction model that has ignored the importance of privacy protection information of user-service QoS invocations.

5.2 Distributed Recommendation

Federated learning (FL) as a new computing paradigm has been recently introduced into the realm of distributed recommendation. Ammad-ud din et al. [38] firstly introduce collaborative filtering into FL called federated collaborative filtering (FCF) and demonstrate its applicability. Based on FCF, Muhammad et al. [33] further employ clustering algorithm to initialize user group and sample from a diverse set of participating clients for faster training of federated recommender systems. Lin et al. [39] introduce a novel federated matrix factorization approach to reduce the model scale for rating prediction in recommender systems. Zhang et al. [30] apply matrix factorization as the approach of FCF, which maintains user latent vector locally and upload gradients of MF to cloud center. To further enhance the reliability, Zhang et al. [40] propose differential privacy and reputation mechanism for more secure prediction of vacant QoS in federated learning computing paradigm, when propagating local private information to remote cloud for collaborative aggregation. By applying deep learning to FCF, Bui et al. [41] propose federated user representation learning approach, which divides neural model parameters into federated and private parameters for personalised recommendation tasks.

Even though some of existing FL-based novel approaches have been initialized investigated for distributed recommendation, they still suffer from difficulty in effectively yielding embeddings for federated collaborative filtering, since the federated group exists dissimilar or low-relevant users.

6 Conclusion and Future Work

In this paper, we propose a novel framework of distributed QoS prediction based on federated hierarchical clustering, named FHC-DQP. First, we collaboratively perform distributed federated training on independent users' QoS invocations until global convergence, to optimize the federated parameters of centralized QoS prediction model. Then, user private feature clustering algorithm is designed to divide users into a set of clusters. Subsequently, user hierarchical clustering that consists of distributed federated training and user private feature clustering is iteratively performed on each partitioned user cluster, leading to fine-grained user collaborative relevance. Finally, predicting an unknown QoS for a distributed target user can be achieved by the trained personalized QoS prediction model. Extensive experiments are conducted on a large-scale real QoS dataset and the results demonstrate the effectiveness of the proposed FHC-DQP.

In the future work, we plan to introduce a pre-tailoring user clustering stage integrated into FHC-DQP before federated training based on user preferences and behaviors, leading to tailoring QoS predictions. Furthermore, it is feasible to design a personalized user-service deep interaction hypernetwork for server parameter aggregation, where it can learn and propagate personalized shared parameters for each user and promotes the performance of tailoring QoS predictions.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62272290, 62172088) and Shanghai Natural Science Foundation (No. 21ZR1400400).

References

[1] K. Benouaret, J. Agoun, I. Benouaret, and F. Charoy, "Call limit-based composite service selection," in *IEEE International Conference on Web Services (ICWS)*, 2022, pp. 37–46.

[2] M. Li, H. Xu, Z. Tu, T. Su, X. Xu, and Z. Wang, "A deep learning based personalized QoE/QoS correlation model for composite services," in *IEEE International Conference on Web Services (ICWS)*, 2022, pp. 312–321.

[3] X. Wang, P. Zhou, Y. Wang, X. Liu, J. Liu, and H. Wu, "Servicebert: A pre-trained model for Web service tagging and recommendation," in *International Conference on Service-Oriented Computing (ICSOC)*, 2021, pp. 464–478.

[4] B. Cao, X. F. Liu, M. M. Rahman, B. Li, J. Liu, and M. Tang, "Integrated content and network-based service clustering and Web APIs recommendation for mashup development," *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 99–113, 2020.

[5] M. Tang, Z. Zheng, G. Kang, J. Liu, Y. Yang, and T. Zhang, "Collaborative Web service quality prediction via exploiting matrix factorization and network map," *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 126–137, 2016.

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

[7] G. Zou, M. Jiang, S. Niu, H. Wu, S. Pang, and Y. Gan, "QoS-aware Web service recommendation with reinforced collaborative filtering," in *International Conference on Service-Oriented Computing (ICSOC)*, 2018, pp. 430–445.

[8] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *SIAM International Conference on Data Mining (SDM)*, 2006, pp. 549–553.

[9] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 1257–1264.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[11] G. Zou, J. Chen, Q. He, K.-C. Li, B. Zhang, and Y. Gan, "NDMF: Neighborhood-integrated deep matrix factorization for service QoS prediction," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2717–2730, 2020.

[12] H. Wu, Z. Zhang, J. Luo, K. Yue, and C.-H. Hsu, "Multiple attributes QoS prediction via deep neural model with contexts," *IEEE Transactions on Services Computing*, vol. 14, no. 4, pp. 1084–1096, 2021.

[13] Y. Xia, D. Ding, Z. Chang, and F. Li, "Joint deep networks based multi-source feature learning for QoS prediction," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 2314–2327, 2022.

[14] Z. Zheng, Y. Zhang, and M. R. Lyu, "Investigating QoS of real-world Web services," *IEEE Transactions on Services Computing*, vol. 7, no. 1, pp. 32–39, 2012.

[15] Y. Liu, A. Huang, Y. Luo, H. Huang, Y. Liu, Y. Chen, L. Feng, T. Chen, H. Yu, and Q. Yang, "FedVision: An online visual object detection platform powered by federated learning," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 13 172–13 179.

[16] D. Sui, Y. Chen, J. Zhao, Y. Jia, Y. Xie, and W. Sun, "FedED: Federated learning via ensemble distillation for medical relation extraction," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 2118–2128.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[18] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning (ICML)*, 2014, pp. 1188–1196.

[19] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," in *International Conference on Machine Learning (ICML)*, 2002, pp. 315–322.

[20] F. Chung, "The heat kernel as the pagerank of a graph," *Proceedings of the National Academy of Sciences*, vol. 104, no. 50, pp. 19 735–19 740, 2007.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] M. Tang, Y. Jiang, J. Liu, and X. Liu, "Location-aware collaborative filtering for QoS-based service recommendation," in *IEEE International Conference on Web Services (ICWS)*, 2012, pp. 202–209.

[25] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *International Conference on World Wide Web (WWW)*, 2017, pp. 173–182.

[26] S. Li, J. Wen, F. Luo, T. Cheng, and Q. Xiong, "A location and reputation aware matrix factorization approach for personalized quality of service prediction," in *IEEE International Conference on Web Services (ICWS)*, 2017, pp. 652–659.

[27] J. Chen, C. Mao, and W. W. Song, "QoS prediction for Web services in cloud environments based on swarm intelligence search," *Knowledge-Based Systems*, vol. 259, 110081, 2023.

[28] J. Liu and Y. Chen, "HAP: a hybrid QoS prediction approach in cloud manufacturing combining local collaborative filtering and global case-based reasoning," *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 1852–1864, 2021.

[29] D. Chai, L. Wang, K. Chen, and Q. Yang, "Secure federated matrix factorization," *IEEE Intelligent Systems*, vol. 36, no. 5, pp. 11–20, 2020.

[30] Y. Zhang, P. Zhang, Y. Luo, and J. Luo, "Efficient and privacy-preserving federated QoS prediction for cloud services," in *IEEE International Conference on Web Services (ICWS)*, 2020, pp. 549–553.

[31] Z. Xu, J. Lin, W. She, J. Xu, Z. Xiong, and H. Cai, "Neighbor collaboration-based secure federated QoS prediction for smart home services," in *IEEE International Conference on Services Computing (SCC)*, 2022, pp. 71–85.

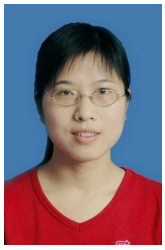
[32] T. Qi, F. Wu, C. Wu, Y. Huang, and X. Xie, "Privacy-preserving news recommendation model learning," *arXiv preprint arXiv:2003.09592*, 2020.
[33] K. Muhammad, Q. Wang, D. O'Reilly-Morgan, E. Tragos, B. Smyth, N. Hurley, J. Geraci, and A. Lawlor, "FedFast: Going beyond average for faster training of federated recommender systems," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2020, pp. 1234–1242.
[34] H. Sun, Z. Zheng, J. Chen, and M. R. Lyu, "Personalized Web service recommendation via normal recovery collaborative filtering," *IEEE Transactions on Services Computing*, vol. 6, no. 4, pp. 573–579, 2012.
[35] X. Wu, B. Cheng, and J. Chen, "Collaborative filtering service recommendation based on a novel similarity computation method," *IEEE Transactions on Services Computing*, vol. 10, no. 3, pp. 352–365, 2017.
[36] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative Web service QoS prediction via neighborhood integrated matrix factorization," *IEEE Transactions on Services Computing*, vol. 6, no. 3, pp. 289–299, 2013.
[37] J. Li, H. Wu, J. Chen, Q. He, and C.-H. Hsu, "Topology-aware neural model for highly accurate QoS prediction," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 7, pp. 1538–1552, 2022.
[38] M. Ammad-Ud-Din, E. Ivannikova, S. A. Khan, W. Oyomno, Q. Fu, K. E. Tan, and A. Flanagan, "Federated collaborative filtering for privacy-preserving personalized recommendation system," *arXiv preprint arXiv:1901.09888*, 2019.
[39] Y. Lin, P. Ren, Z. Chen, Z. Ren, D. Yu, J. Ma, M. d. Rijke, and X. Cheng, "Meta matrix factorization for federated rating predictions," in *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2020, pp. 981–990.
[40] Y. Zhang, P. Zhang, Y. Luo, and L. Ji, "Towards efficient, credible and privacy-preserving service QoS prediction in unreliable mobile edge environments," in *International Symposium on Reliable Distributed Systems (SRDS)*, 2020, pp. 309–318.
[41] D. Bui, K. Malik, J. Goetz, H. Liu, S. Moon, A. Kumar, and K. G. Shin, "Federated user representation learning," *arXiv preprint arXiv:1909.12535*, 2019.

Guobing Zou is a full professor and dean of the Department of Computer Science and Technology, Shanghai University, China. He received his PhD degree in Computer Science from Tongji University, Shanghai, China, 2012. He has worked as a visiting scholar in the Department of Computer Science and Engineering at Washington University in St. Louis from 2009 to 2011, USA. His current research interests mainly focus on services computing, edge computing, data mining and intelligent algorithms, recommender systems. He has published more than 100 papers on premier international journals and conferences, including IEEE Transactions on Services Computing, IEEE Transactions on Network and Service Management, IEEE ICWS, ICSOC, IEEE SCC, AAAI, IJWSR, IJWGS, Information Sciences, Expert Systems with Applications, Knowledge-Based Systems, Applied Intelligence, etc. He served as organization and publicity chair of the International Conference on Service Science, vice chair of IEEE International Conference on Big Data (IEEE BigData 2021), chair of "Service Computing Top Conference Top Journal Forum" of China Digital Service Conference (2021-2023), and guest editor of International Journal of Services Technology and Management.

Shiyi Lin is currently a master student in the School of Computer Engineering and Science, Shanghai University, Shanghai, China. Before that, he received a Bachelor degree in Computer Science and Technology at Shanghai University, 2021. His research interests include service quality management, federated learning and deep learning. He has led a research and development group to successfully design and implement a service-oriented enterprise application big data platform, which can intelligently classify and recycle, cultivate citizens' habit of throwing recyclables, and produce significant economic and social benefits by providing high QoS.

Shengxiang Hu is currently a PhD candidate in the School of Computer Engineering and Science, Shanghai University, Shanghai, China. He received a Bachelor degree in Communication Engineering in 2018 and Master degree in 2021 in Computer Science and Technology at Shanghai University, respectively. His research interests include QoS prediction, graph neural network and natural language processing. He has published two papers on Knowledge-Based Systems, International Conference on Parallel Problem Solving from Nature (PPSN), and submitted a paper on International Conference on Service-Oriented Computing (ICSOC).

Shengyu Duan received the first B.Eng. degree in telecommunication engineering from the Huazhong University of Science and Technology, China, the second B.Eng. degree in electronic and electrical engineering from the University of Birmingham, U.K. in 2013, and the M.Sc. and Ph.D. degrees from the University of Southampton, U.K., in 2014 and 2019, respectively. He is currently working as an Assistant Professor in the School of Computer Engineering and Science, Shanghai University, Shanghai, China. His research interests include design for machine learning, IC reliability, hardware security, and hardware acceleration.



Yanglan Gan is a full professor in the School of Computer Science and Technology, Donghua University, Shanghai, China. She received her PhD in Computer Science from Tongji University, Shanghai, China, 2012. Her research interests include bioinformatics, service computing, and data mining. She has published more than 50 papers on premier international journals and conferences, including Bioinformatics, Briefings in Bioinformatics, BMC Bioinformatics, IEEE/ACM Transactions on Computational Biology and Bioinformatics, IEEE Transactions on Services Computing, IEEE Transactions on Network and Service Management, IEEE ICWS, ICSOC, Neurocomputing, and Knowledge-Based Systems.

Bofeng Zhang is a full professor and dean of the School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai, China. He received his PhD degree from the Northwestern Polytechnic University (NPU) in 1997, China. He experienced a Postdoctoral Research at Zhejiang University from 1997 to 1999, China. He worked as a visiting professor at the University of Aizu from 2006 to 2007, Japan. He worked as a visiting scholar at Purdue University from 2013 to 2014, US. His research interests include personalized service recommendation, intelligent humancomputer interaction, and data mining. He has published more than 200 papers on international journals and conferences.

Yixin Chen received the PhD degree in computer science from the University of Illinois at Urbana Champaign, in 2005. He is currently a full professor of Computer Science at Washington University in St. Louis, MO, USA. His research interests include artificial intelligence, data mining, deep learning, and big data analytics. He has published more than 210 papers on premier international journals and conferences, including AIJ, JAIR, IEEE TSC, IEEE TPDS, IEEE TC, IEEE TKDE, IEEE TII, IJCAI, AAAI, ICML, KDD, etc. He won the Best Paper Award at AAAI and a best paper nomination at KDD. He received an Early Career Principal Investigator Award from the US Department of Energy and a Microsoft Research New Faculty Fellowship. He was an Associate Editor for the ACM Transactions on Intelligent Systems and Technology, IEEE Transactions on Knowledge and Data Engineering, and Journal of Artificial Intelligence Research. He is a Fellow of the IEEE.