FRLN: Federated Residual Ladder Network for Data-Protected QoS Prediction

Guobing Zou, Wenzhuo Yu, Shengxiang Hu, Yanglan Gan*, Bofeng Zhang*, and Yixin Chen, Fellow, IEEE

Abstract—QoS prediction plays an important role in service-oriented downstream tasks. However, most of current state-of-the-art QoS prediction approaches suffer from two limitations. First, traditional approaches typically require collection of user-service historical QoS invocations centrally in order to improve QoS prediction accuracy, which poses a threat to user data privacy. Second, although few of the recent approaches take into account data protection when predicting QoS values, they still cannot effectively capture user-service complex nonlinear invocation relationships, significantly influencing the performance of QoS prediction. To address these two issues, we propose a novel framework of data-protected QoS prediction called Federated Residual Ladder Network (FRLN), which ensures user data protection and effectiveness of predicting missing QoS values. It initially leverages our designed Residual Ladder Network (RLN) to extract latent features of users and services from both low and high dimensional spaces. Then, local QoS prediction models are collaboratively trained by personalized federated learning with the consideration of data heterogeneity. Extensive experiments have been conducted on a real-world large-scale dataset called WS-DREAM, which consists of 5825 Web services from 74 regions and 339 users from 31 regions comprising a total number of 1,974,675 user-service QoS invocations. Experimental results demonstrate the effectiveness of FRLN in multiple evaluation metrics. While the proposed FRLN framework marks a significant step forward for QoS prediction in machine learning, ongoing advancements in ML techniques and expanded datasets are essential for further enhancing its precision and applicability in real-world scenarios.

Index Terms—Web service, Data-protected QoS prediction, Residual ladder network, Personalized federated learning.

1 Introduction

The emergence of Web services in recent years has led to their increasing adoption in various technological domains such as Service-Oriented Architecture (SOA) and Internet of Services (IoS) [1]. It has resulted in a proliferation of the same or similar services, making it challenging to recommend optimal services offering the best invocation experiences to service requesters. Quality of Service (QoS), as a crucial non-functional attribute [2] plays a vital role in discriminating between these same or similar services. It is dependent on various factors such as service deployment conditions, user invocation locations, and network environment adaptation [3]. Therefore, QoS has become a significant criterion in recommending Web services with the same or similar functionality. However, due to the rapidly increasing number of users and services, it is impractical and time-consuming for service requesters to invoke all Web services and service providers to monitor QoS information for each service invocation. To satisfy diverse service-oriented application scenarios, such as service discovery, selection, composition, recommendation and mashup creation, it has become a fundamental yet challenging research issue to accurately perform QoS prediction, because of the remarkable sparsity of historical user-service invocations.

Collaborative filtering has been widely applied in predicting missing QoS values, which can be classified into memory-based and model-based approaches. Memory-based collaborative filtering involves collecting QoS historical invocation records from user devices, calculating similarity among users or services to obtain similar neighborhoods, and predicting vacant QoS values [4], [5], [6], [7]. To alleviate the sparsity problem of memory-based approaches, model-based collaborative filtering is devoted to learning a model from historical QoS invocation records, by extracting the latent semantic features of users and services for QoS prediction [8], [9]. However, these approaches often have limited success in handling sparse data, as they primarily capture low-dimensional and linear features, and may not fully reveal the complex, high-dimensional relationships among users and services. As a solution, recent investigations have applied deep learning techniques [10], [11] to perform QoS prediction task, using shallow neural multilayer perceptrons to learn complex nonlinear interaction relationships among users and services and addressing the poor representations of traditional model-based collaborative filtering.

The model-based approaches mentioned above frequently require local user data, such as IP addresses and historical QoS invocation records to assist the central server in training a unified QoS prediction model. As the QoS prediction task increasingly relies on local user data, privacy concerns have become an important issue to consider when it comes to practical deployment. Specifically, this poses a potential data leakage risk, as malicious central servers could infer sensitive user information from QoS invocation records. Such user privacy threats result in difficulty in

- *G. Zou, W. Yu, S. Hu are with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China. E-mail: {gbzou,yuwenzhuo,shengxianghu}@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.*

learning a QoS prediction model by training all historical invocation records of users. As a result, developing data-protected QoS prediction techniques has become a crucially challenging issue. It must protect user data privacy while still maintaining satisfactory QoS prediction accuracy.

In recent years, emerging federated learning [12], [13], [14], [15] techniques have been used to develop privacy-preserving QoS prediction techniques that avoid centralizing local user data. However, while these approaches move toward data privacy protection, they work against improving QoS prediction accuracy for two essential and critical reasons. First, existing approaches ignore the significant impact on model performance caused by more limited local users' QoS invocation records. Given that each client only has a fraction of the global QoS invocation records, the shift from centralized to distributed training may compromise the model's ability to accurately learn latent user-service interaction features, which also aggravates the overfitting situation of QoS prediction model. Second, personalized service demands and distinct network environments leave QoS records that deviate from the global QoS invocation distribution on each client. In such case, most existing approaches upload all the parameters of the trained local QoS prediction models to the cloud center for aggregation, exacerbating client-drift [16] and leading to the degradation of prediction performance. Thus, training a single global QoS prediction model for all local users is often insufficient when facing a non-independent and homogeneously distributed QoS invocation records.

To solve the above issues, we propose a novel approach for data-protected QoS prediction called Federated Residual Ladder Network (FRLN), which is designed to overcome the challenges posed by limited local QoS invocation records and the data heterogeneity arising from varying distributions of QoS invocation records among users. Specifically, FRLN consists of two main components. First, we design a new RLN feature extraction network, which effectively captures latent user-service feature representations across the limited local historical QoS invocations. It uses forward and backward residual perception blocks to capture the interaction between users and services from both low and high-dimensional spaces. Second, we train local QoS prediction models collaboratively using federated learning, considering data heterogeneity. It optimizes globally shared parameters as well as locally private ones for each user, towards the personalization of performing QoS prediction. We upload only the global representations in the RLN model parameters to the cloud center for aggregation, while keeping the rest of the personalized parameters locally. That enables each user to train a personalized QoS prediction model with their private local QoS invocation records, thereby achieving the objective of data protection.

To evaluate the effectiveness of our proposed approach, we conduct extensive experiments on a public and large-scale real-world dataset called WS-DREAM, which consists of 5825 real-world Web services from 74 regions and 339 service users from 31 regions. It involves the total number of 1,974,675 user-service QoS invocations, which is partitioned into a set of independent groups of user-service QoS invocations in terms of users. Our experimental results demonstrate that the proposed FRLN achieves the best prediction performance in multiple evaluation metrics for data-protected QoS prediction compared to several state-of-the-art competing baselines.

The main contributions are summarized as follows:

- We propose a novel effective QoS prediction framework that uses local invocation records from user devices, combined with deep neural networks and personalized federated training techniques, to reconcile data-protected and prediction accuracy while predicting missing QoS values.
- We propose a new QoS feature extraction network, RLN, which leverages two-way residual-aware blocks to reveal the complex nonlinear interactions for capturing latent features of users and services more deeply. Considering the heterogeneity of QoS invocation records, we train the RLN models collaboratively in federated learning by a personalized way.
- We reproduce several centralized models under the federated paradigm, and conduct extensive experiments on a large-scale real-world dataset, WS-DREAM. The results indicate that FRLN keeps data protection with still maintaining the superior performance for QoS prediction, compared with existing data-protected approaches.

The rest of this paper is organized as follows. Section 2 formulates the problem of data-protected QoS prediction. Section 3 presents the proposed approach of FRLN in detail. Section 4 shows and analyzes the experimental results. Section 5 reviews the related work. Finally, Section 6 concludes the paper.

2 Problem Formulation

Definition 1 (Service User). Service users mainly refer to the users who have invoked one or more services. Let $U = \{u_1, u_2, ..., u_m\}$ be a set of users. For each $u \in U$, it can be defined as a three-tuple $u =< ID, RE, AS >$. ID is the identifier of u and the rest can be collectively represented as location information.

A service user's location information mainly includes Region (RE) and Autonomous System (AS), respectively.

Definition 2 (Web Service). For data-protected QoS prediction problem, we mainly focus on the nonfunctional features of a Web service. Let $S = \{s_1, s_2, ..., s_n\}$ be a set of Web services. For each $s \in S$, it can be defined as a three-tuple $s =< ID, RE, AS >$. ID is the identifier of s and the rest can be collectively represented as location information.

A Web service's location information mainly includes Region (RE) and Autonomous System (AS), respectively.

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

Through Web service invocations, the user-service invocation records can be represented as a QoS matrix, denoted as R. Each row of the matrix represents the QoS of a user who invokes all Web services, and each column represents the QoS of a Web service that is invoked by all users. From the user's perspective, R can be partitioned into sub-matrices, denoted as $R' = \{R_1, R_2, ..., R_m\}$, where each R_i

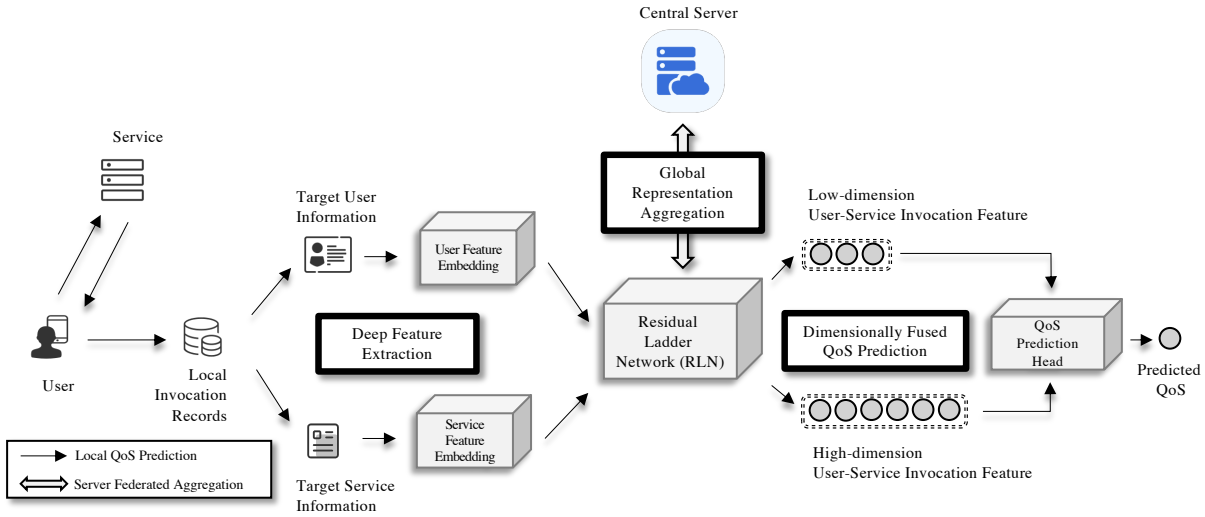


Fig. 1. The Framework of FRLN: It is built on the residual ladder network (RLN) and consists of two main stages, including local QoS prediction and server federated aggregation, which are marked by the black single arrow and the gray two-way arrow, respectively.

represents the QoS record sub-matrices of user u_i invoking all services. If a user u_i has invoked a service s_j, we have $< u_i, s_j, r_{u_i,s_j} > \in R_i$, otherwise $< u_i, s_j, r_{u_i,s_j} > \notin R_i$.

Definition 4 (Data-Protected QoS Prediction Problem). Given a set of users U, a set of Web services S and all observed QoS invocation records R, a QoS prediction problem can be defined as a five-tuple $\Omega =< U, S, R', u, s >$, where $u \in U$ is a target user, $s \in S$ is a target service, $R' = \{R_1, R_2, ..., R_m\}$, and $< u, s, r_{u,s} > \notin R$.

The solution to a data-protected QoS prediction problem Ω is $< u, s, \hat{r}_{u,s} >$. It indicates the predicted QoS value when a target user invokes a target service, by exploiting the provided information of invocation records among users and services, from the perspective of data-protected objective.

3 Approach

3.1 The Framework of FRLN

Figure 1 illustrates the of two main stages, including local QoS prediction and server federated aggregation.

- In the stage of local QoS prediction, the target user and target service information are mapped into low-dimensional dense embedding vectors. Residual Ladder Network is first designed to learn complex nonlinear user-service interaction features from the dense vectors. Then, the generated low and high dimensional latent features, output by the RLN, are fused and fed to the QoS Prediction Head, which is a shallow multilayer perceptron to predict finally unknown QoS values.
- In the stage of server federated aggregation, the RLN models are trained by federated learning in a personalized manner. That is, each user's corresponding RLN model uploads the global representation to the central server for parameter aggregation, while the rest of the parameters are kept locally away from central aggregation. Thus, each client holds a personalized QoS prediction model.

3.2 Local QoS Prediction

3.2.1 User-Service Deep Feature Extraction

Figure 2 shows the layers of the designed RLN for the process of extracting user-service deep feature representations. It begins with the interaction information between users and services, propagating through user-service input layer, user-service embedding layer, and user-service residual ladder layer. By the sequential transformations, it outputs both low and high-dimensional deep features of user-service nonlinearly complex interactive relationships.

User-Service Input Layer. It is employed to initialize the original representation of a user and service. To improve the effectiveness of extracting deep features, we integrate location information such as autonomous system (AS) and region (RE) in addition to the identifiers (ID) of a user and service. The original user and service features can be expressed as multi-dimensional vectors:

$$u_n = [i_u, r_u, a_u] \tag{1}$$

$$s_n = [i_s, r_s, a_s] \tag{2}$$

Where i, r, a denote ID, RE and AS, while u and s represent a user and service, respectively. We use a global mapping system to assign identical non-negative integers to users or services in the same region or autonomous system.

User-Service Embedding Layer. In this layer, the initially high-dimensional and sparse feature vectors of users and services are mapped to low-dimensional and dense embedding feature vectors. The mapping function for a user's ID, RE, and AS can be formulated as follows:

$$I_u = \sigma\left(W_u^T i_u\right) \tag{3}$$

$$R_u = \sigma\left(W_u^T r_u\right) \tag{4}$$

$$A_u = \sigma\left(W_u^T a_u\right) \tag{5}$$

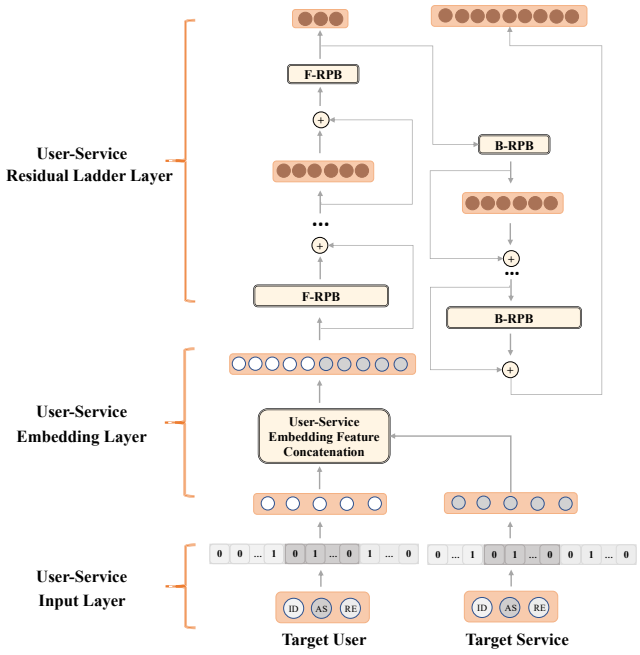


Fig. 2. The architecture of Residual Ladder Network (RLN) for extracting low and high-dimensional deep features of a user and service.

Where W_u represents the user's embedding weight matrix and σ represents the ReLU activation function of embedding layer. I_u, R_u and A_u represent the embedding output of a user's ID, RE and AS, respectively. Similarly, we can obtain a service's embedding output I_s, R_s and A_s, respectively.

These embeddings are then concatenated to correspondingly generate embedding feature vectors E_u and E_s, which are further combined to produce the user-service interaction feature representation $X_{u,s}$. It is expressed by the following formula where Φ is the operation of feature concatenation.

$$E_u = \Phi\left(I_u, R_u, A_u\right) = \begin{bmatrix} I_u \\ R_u \\ A_u \end{bmatrix} \tag{6}$$

$$E_s = \Phi\left(I_s, R_s, A_s\right) = \begin{bmatrix} I_s \\ R_s \\ A_s \end{bmatrix} \tag{7}$$

$$X_{u,s} = \Phi(E_u, E_s) = \begin{bmatrix} E_u \\ E_s \end{bmatrix} \tag{8}$$

User-Service Residual Ladder Layer. The output of embedding layers is integrated into residual ladder layer to extract the nonlinear interaction relationships between users and services.

Intuitively, high-dimensional spaces are crucial for deep feature extraction, as existing investigations have shown that low-dimensional representations are often insufficient for capturing user-service interaction features, resulting in significant losses in QoS prediction accuracy [17], [18], [19], [20]. Therefore, in our user-service residual ladder layer, low-dimensional output of the forward pyramid is continuously fed as input to the reverse pyramid structure to generate high-dimensional features. The generated low and high-dimensional features are then combined together to further perform local QoS prediction.

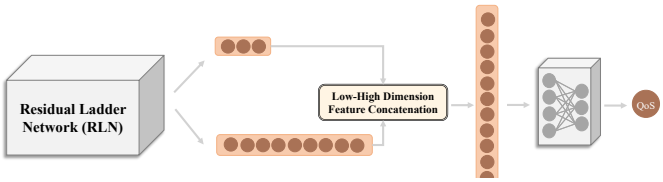


Fig. 3. QoS prediction head.

By extending from ResMLP [21], we design the main unit of user-service residual ladder layer, called the residual perceptron block (RPB), to extract low and high-dimensional deep features of users and services. Here, each RPB consists of two affine layers and dimensions of the affine layer are consistent within an RPB, but decreasing with forward propagation in the F-RPB while increasing in the B-RPB. To reduce the overfitting, a Dropout layer instead of Batch Normalization (BN) layer is added between the two affine layers. The reason is that in a federated setting of our local QoS prediction problem, where each user only holds a small and non-IID set of user-service QoS invocation records, the personalized historical QoS invocations of users are distributed differently across multiple batches. The formalizations of F-RPB and B-RPB are as follows.

$$Aff(X, \sigma) = \sigma\left(W^T X + b\right) \tag{9}$$

$$y_f^i = Aff(D(Aff(x_f^i, ReLU)), ReLU) \tag{10}$$

$$x_f^{i+1} = y_f^i + x_f^i \tag{11}$$

$$y_b^i = Aff(D(Aff(x_b^i, ReLU)), ReLU) \tag{12}$$

$$x_b^* = y_f^* \tag{13}$$

$$x_b^i = y_b^{i+1} + x_b^{i+1} \tag{14}$$

Where Aff represents affine layer, σ represents ReLU activation function, and D represents Dropout layer. x_f^i and y_f^i represents the input and output of the i-th F-RPB, respectively. Similarly, x_b^i and y_b^i represents the input and output of the i-th B-RPB, respectively. x_b^* and y_f^* represent the top-level F-RPB and B-RPB, which means that the output of the last F-RPB is utilized as the initial input for the foremost B-RPB.

3.2.2 *Dimensionally Fused QoS Prediction*

As Figure 3 shows QoS prediction head is to predict the unknown QoS when a target user desires to invoke a target service. We concatenate the low-dimensional x_f and high-dimensional x_b latent user-service interaction features extracted by RLN and feed them into a fully connected layer. It is formalized as follows:

$$X_o = \Phi(x_f, x_b) = \begin{bmatrix} x_f \\ x_b \end{bmatrix} \tag{15}$$

$$\hat{r}_{u,s} = I\left(W_h^T X_o + b_h\right) \tag{16}$$

Where $\hat{r}_{u,s}$ represents the predicted QoS value and I represents a standard identity function as the activation function for the QoS prediction head layer.

3.3 Server Federated Aggregation

3.3.1 Global Representation Aggregation

The federated learning paradigm relies on storing historical QoS invocation records on user-side devices rather than exposing them to a central server. Although it results in personalized non-IID QoS invocation records that closely relate to user behavior and preferences, the heterogeneity of QoS invocation records from multiple clients may have common latent feature representations that can be shared across clients. To this end, we propose an extension of the traditional federated learning where clients cooperate to learn a global model using all parameters of each client and then simply replace this global model on each client.

Figure 4 illustrates our proposed scheme of federated training, which involves global update, local update, as well as their corresponding server update and client update. The objective is to develop a bunch of personalized RLN local QoS prediction models, by both collaboratively learning shared global feature representation of heterogeneous user-service invocation QoS across multiple clients and keeping their own personalized local features for each client.

3.3.2 Personalized Federated Training of RLN

RLN fully extracts the deep features with multiple forward and backward residual perception blocks (RPBs), which may contain the common feature representation of user-service interactions, and all user devices collaborate and share the global feature representations.

For all the trainable parameters of the RPBs in the user-service residual ladder layer of RLN, we represent them as global feature parameters and denote them by Θ_G. In such case, the process of feeding the output $X_{u,s}$ from the user-service embedding layer into the residual ladder layer of RLN can be rewritten as a nonlinear transformation, formally expressed as

$$x_f, x_b \leftarrow \sigma\left(\Theta_G^T X_{u,s} + b\right) \tag{17}$$

Conversely, since the input, embedding and output layers work directly with the user's personalized local QoS invocation records, each client keeps them private and does not share with any other clients. Here, we denote the trainable parameters of the output layer by $\hat{\Theta}_{P,u}$, the QoS prediction head is rewritten as

$$\hat{r}_{u,s} = I\left(\hat{\Theta}_{P,u}\begin{bmatrix} x_f \\ x_b \end{bmatrix} + b\right) \tag{18}$$

To train and optimize the model parameters, we take Mean Absolute Error (MAE) as the loss function, and the loss to be optimized for a specific user u is expressed as

$$L_u(\Theta_{G,u}, \hat{\Theta}_{P,u}) = \frac{\sum_{s\in S}\left|f(X_{u,s};\Theta_{G,u};\hat{\Theta}_{P,u}) - r_{u,s}\right|}{n} + \lambda\|\Theta_u\|_2^2 \tag{19}$$

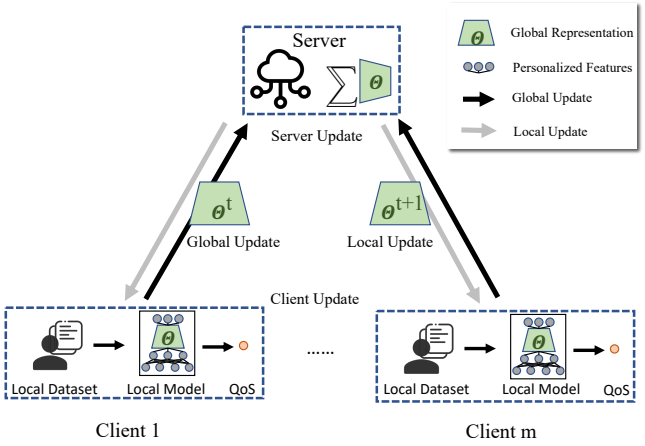


Fig. 4. The proposed scheme of server federated aggregation: All clients share a common global representation of diverse user-service QoS records, along with their unique local features tailored to individual preferences. The global update moves from several clients to a central server for collective learning, while a local update reverses this flow for back propagation.

Where S represents the service set and $|S| = n$, Θ_u is all the trainable parameters for each client's local QoS prediction model, and λ is the regularization parameter for controlling model overfitting.

By accumulating and averaging all the users from U, the global loss function is expressed as

$$min\ L(\Theta) = \min_{u_1,\ldots,u_m\in U} \sum_{u\in U} \frac{T_u}{T} L_u(\Theta_{G,u}, \hat{\Theta}_{P,u}) \tag{20}$$

Where T denotes the total number of samples and T_u refers to the local samples owned by participant u. The task of minimizing the global loss function is equivalent to optimizing each loss function owned by each client.

3.3.3 Algorithms of Client and Server Updates

To minimize the loss function described above, the user client and server perform updates alternately. In general, a client u takes local historical QoS invocation records to train the personalized QoS prediction model. Concurrently, the global representations extracted by RLN across multiple clients are transmitted to the central server that takes the global representation by a federated aggregation process, sharing the updated representation with other clients.

The pseudo codes of client and server updates are shown in Algorithms 1 and 2, respectively. For each round $t = 1, 2.., k$, clients are involved to perform client update based on a constant fraction $C \in (0, 1]$. In the client update, each client u receives the current global representation from the central server, and makes E epochs of gradient-based updates, such as Stochastic Gradient Descent (SGD), to optimize both global and personalized parameters. For $i = 1, ..., E$, client $u \in M$, it executes client updates as

$$(\Theta_{G,u}^t, \hat{\Theta}_{P,u}^t) \leftarrow SGD_i\left(\Theta_{G,u}^{t-1}; \hat{\Theta}_{P,u}^{t-1}; b; \eta\right) \tag{21}$$

Where η denotes learning rate and b indicates the batch size over the local historical QoS invocation records. We stipulate that $b = -1$ means all local user-service invocation

Algorithm 1: Client Update

Initialize $\Theta_{P_u}^{(0)}$
for *each round* $t = 1, 2...$ **do**
  $k \leftarrow \max(C \cdot |U|, 1)$
  $M \leftarrow$ randomly select k clients
  for *each client* $u \in M$ **do**
    Receive $\Theta_{G,u}^{t-1}$ from Server
    for *local epoch i from 1 to E* **do**
      $(\Theta_{G,u}^{t}, \hat{\Theta}_{P,u}^{t}) \leftarrow SGD_i\left(\Theta_{G,u}^{t-1}; \hat{\Theta}_{P,u}^{t-1}; b; \eta\right)$
    end
    Send $\Theta_{G,u}^{t}$ to Server
  end
end

Algorithm 2: Server Update

Initialize $\Theta_{G}^{(0)}$
for *each round* $t = 1, 2...$ **do**
  Receive $\Theta_{G,u}^{t}$ from each client $u \in M$
  Aggregate $\Theta_{G}^{t} \leftarrow \sum_{u \in M} \frac{T_u}{T} \Theta_{G,u}^{t}$
  Propagate Θ_{G}^{t} to each client
end

training samples are treated as a single minibatch. SGD optimizes parameters from $(\Theta_{G,u}^{t-1}; \hat{\Theta}_{P,u}^{t-1})$ to $(\Theta_{G,u}^{t}, \hat{\Theta}_{P,u}^{t})$ with (22)

$$\Theta_u = \Theta_u - \eta \cdot \nabla_{\Theta_u} L(\Theta_u) \tag{22}$$

Where Θ_u represents the parameters $(\Theta_{G,u}^{t-1}; \hat{\Theta}_{P,u}^{t-1})$, η represents the learning rate, which is a factor that controls the step size of parameter update, $L(\Theta_u)$ represents the loss function, and $\nabla_{\Theta_u} L(\Theta_u)$ represents the gradient of the loss function with respect to the parameters.

Once the client update is finished, the server continues to receive the local-updated global representations, and takes a weighted average on them, until it minimizes the global loss function or reaches the convergence condition.

4 Experiments

4.1 Experimental Setup and Datasets

All experiments are conducted on a workstation with an NVIDIA GeForce 1080Ti GPU and an Intel Xeon Gold 6132 CPU at 2.60 GHz. Our approach's components were implemented in Python 3.7.0 and PyTorch 1.1.0.

To validate the performance of the proposed FRLN for data-protected QoS prediction by personalized federating learning, we conduct extensive experiments using the benchmark dataset WS-DREAM dataset [22]. It is a large-scale real-world user-service invocation QoS dataset, which has been widely used for QoS prediction. WS-DREAM contains 1,974,675 historical QoS invocation records from 339 users and 5,825 services. It has two kinds of user-service QoS invocations, including response time (RT) and throughput (TP). Each can be represented as a user-service invocation matrix, where a row has a set of QoS entries indicating a corresponding user who invokes all of the Web services, and a column includes a group of QoS entries indicating a corresponding Web service that is invoked by all of the users. Moreover, WS-DREAM provides user and service identifiers as well as location contextual information, such as region and autonomous system, which are used as model inputs for extracting user-service invocation deep features in FRLN. The detailed statistics of the experimental dataset are shown in Table 1, where an example of a data sample before and after pre-processing is shown in Table 2.

TABLE 1
The statistics of WS-DREAM experimental dataset

Item	Value
User ID	339
User regions	31
User AS	137
Service ID	5,825
Service regions	74
Service AS	992
QoS invocation records	1,974,675

TABLE 2
WS-DREAM data sample before and after pre-processing

	ID	RE (Before/After)	AS (Before/After)
User	111	United States/12	AS378 ILAN./4
Item	525	United States/12	AS271 BCnet/7
RT		0.285	
TP		7.782	

In real service-oriented application scenarios, a single user only invokes a limited number of services, which results in extreme sparsity of the user-service QoS invocation matrix. In our experiments, four different low densities of 5%, 10%, 15%, and 20% QoS dataset are generated on RT and TP as model training data, while the remaining 95%, 90%, 85% and 80% QoS invocation samples are used as model testing data set to compare the performance between FRLN and competing baselines. Note that under the consideration of federated setting, randomly generated training data samples are respectively distributed to different clients, where each client corresponds to an individual user and exclusively accesses their own user-service QoS invocation records within the training dataset.

4.2 Competing Methods and Evaluation Metrics

To evaluate the performance of FRLN, we compare it with eleven widely-used representative competing approaches, including two memory-based and one matrix factorization model-based approaches, two deep learning based approaches, and four federated learning based approaches, and two our self-developed variants of FRLN. They are described as below.

- **UPCC** [4]: It is a user-based collaborative filtering QoS prediction algorithm. It uses PCC to find the neighbor set of a target user and combines the deviation migration of the neighbor users, and the average of all QoS values, which are from the target user invoking Web services.

- **LACF** [7]: It is memory-based and state-of-the-art location-aware collaborative filtering QoS prediction approach, which incorporates location contextual information to better calculate the similarity among users and services.
- **PMF** [23]: It is probabilistic matrix factorization approach, which optimizes the traditional matrix factorization utilizing probabilistic model. PMF is used as a typical model-based collaborative filtering algorithm for vacant QoS prediction.
- **NCF** [18]: It is a deep learning-based collaborative filtering algorithm utilizing Generalized Matrix Factorization (GMF) and a Multi-Layer Perceptron to learn complex nonlinear user-service interactions, aimed at solving regression problems like QoS prediction.
- **LDCF** [20]: It is a deep learning location-aware collaborative filtering algorithm, which builds a bridge between deep learning and CF through similarity adaptive corrector. It embeds location information of users and services into feature representations, and utilizes MLP to learn high-dimensional nonlinear relationships between users and services.
- **EFMF** [24] It is a federated learning based matrix factorization approach that enhances prediction performance and protects user data privacy by predicting missing QoS values without central aggregation of user-service records, serving as the federated-oriented baseline for FRLN.
- **NCSF-GMF** [25] It is a federated learning based approach where users collaboratively upload perturbed updates during server aggregation to enhance privacy without affecting the global model's accuracy.
- **FedNCF** [18] It is a federated learning algorithm that is based on NCF. It uses the traditional FedAvg approach for parameter aggregation, whereby all NCF parameters are uploaded to a central server during the parameter aggregation phase and subsequently aggregated.
- **FedLDCF** [20] It is a federated learning algorithm based on LDCF, in which all parameters of the LDCF model are uploaded to the central server and aggregated using the FedAvg approach, offering a promising solution to privacy and security challenges in predicting missing QoS values.
- **FRLN-Avg** It is our self-developed non-personalized variant of FRLN, applying the FedAvg algorithm [12] to perform server federated aggregation for those parameters of the RLN model uploaded to the cloud center.
- **FRLN-(ϵ, δ)-DP**: It is our self-developed differential-privacy variant of FRLN, which utilizes the DP-SGD algorithm [26]. We adjust two kinds of parameters for different levels of privacy protection, including target privacy budget ϵ and acceptable privacy risk δ. Especially, a smaller ϵ raises more noise, indicating stronger privacy protection. By taking with different privacy budgets ϵ balance data protection and model performance of QoS prediction.

In our experiments, we use both Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to evaluate our QoS prediction models. They are defined as follows:

$$MAE = \frac{\sum_{u,s} |r_{u,s} - \hat{r}_{u,s}|}{N} \tag{23}$$

$$RMSE = \sqrt{\frac{\sum_{u,s} (r_{u,s} - \hat{r}_{u,s})^2}{N}} \tag{24}$$

Where u and s represent a given target user and target service, respectively. $r_{u,s}$ and $\hat{r}_{u,s}$ represent observed and predicted QoS values, respectively. N denotes the number of the test QoS samples in the experiments.

MAE mainly evaluates the overall accuracy of QoS prediction by calculating the averaged absolute deviation while RMSE is used to more sensitively assess the performance of QoS prediction model on outliers using a relatively higher weighting to large errors on predicted QoS values. The reason we choose MAE and RMSE as evaluation metrics is that they are instrumental in providing a clear picture of the model's overall QoS predictive accuracy and its effectiveness in handling outliers, which has been widely used for validating the performance of QoS prediction [10], [20], [22], [27].

Additionally, it is still difficult to recognize whether a machine learning framework can be used for effective QoS prediction by purely applying MAE and RMSE. The reason is that these two evaluation metrics cannot well reflect the gap in absolute terms between the prediction error range and the ground truth QoS value. Thus, Normalized Mean Absolute Error (NMAE) [28] [29] is further taken to measure the QoS prediction accuracy. NMAE is defined as follows:

$$NMAE = \frac{MAE}{\sum_{u,s} r_{u,s}/N} \tag{25}$$

Where $r_{u,s}$ represents observed QoS values. N is the number of test QoS samples in the experiments. NAME can be used to calculate the ratio between MAE and the average real QoS value, reflecting the absolute gap of prediction error range and ground truth value for effective decision-making in application scenarios.

TABLE 3
Parameter settings of FRLN

Parameter	Value	Description
b	-1 or 8	Batch size of each client
d	16	Dimensionality
C	0.1	Client selection rate
E	1 or 5	Epoch of each client
η	0.001	Learning rate
$depth$	$[256, 128, 64]$	Depth of RLN model
$dropout$	0.15	Dropout rate
ϵ_1, ϵ_2	10, 5	Target privacy budget
δ	0.0001	Acceptable privacy risk
$Clip$	0.5	Maximum gradient norm
FL_Round	1000	Total rounds of FL

4.3 Experiment Results and Analyses

4.3.1 Experiment Results in Federated Settings

Comparisons of Federated Settings. To verify the effectiveness of our proposed FRLN, we first compare it with existing state-of-art QoS prediction approaches in a federated

TABLE 4
Experimental results of federated QoS prediction under multiple densities on RT dataset. The best results are marked in bold and the second-best results are highlighted in gray background. The gains are calculated between the best and the second-best predicted QoS values

Methods	Density 5%			Density 10%			Density 15%			Density 20%		
	MAE	RMSE	NMAE	MAE	RMSE	NMAE	MAE	RMSE	NMAE	MAE	RMSE	NMAE
EFMF	0.622	1.526	0.683	0.528	1.326	0.581	0.488	1.237	0.538	0.470	1.200	0.518
NCSF-GMF	0.569	1.557	0.625	0.481	1.456	0.530	0.453	1.340	0.498	0.400	1.286	0.441
FedNCF	0.492	1.478	0.543	0.431	1.374	0.474	0.417	1.361	0.458	0.403	1.326	0.444
FedLDCF	0.491	1.433	0.541	0.451	1.356	0.496	0.433	1.364	0.475	0.410	1.387	0.451
FRLN	**0.379**	**1.306**	**0.418**	**0.344**	**1.238**	**0.378**	**0.322**	**1.201**	**0.355**	**0.309**	**1.175**	**0.340**
FRLN-Avg	0.397	1.347	0.437	0.385	1.329	0.423	0.380	1.312	0.418	0.365	1.285	0.401
FRLN-(ϵ_1, δ)-DP	0.390	1.355	0.429	0.345	1.260	0.380	0.324	1.218	0.357	0.312	1.197	0.344
FRLN-(ϵ_2, δ)-DP	0.417	1.380	0.458	0.370	1.301	0.407	0.329	1.220	0.361	0.322	1.206	0.356
Gains	22.81%	8.86%	22.73%	20.19%	6.64%	20.25%	22.78%	2.91%	22.48%	23.32%	2.08%	22.90%

TABLE 5
Experimental results of federated QoS prediction under multiple densities on TP dataset

Methods	Density 5%			Density 10%			Density 15%			Density 20%		
	MAE	RMSE	NMAE	MAE	RMSE	NMAE	MAE	RMSE	NMAE	MAE	RMSE	NMAE
EFMF	26.92	68.86	0.567	19.62	53.37	0.411	16.91	48.69	0.354	15.34	43.87	0.323
NCSF-GMF	18.08	63.88	0.379	21.99	55.08	0.462	17.08	49.05	0.359	14.92	47.16	0.313
FedNCF	18.52	57.27	0.389	16.02	51.40	0.337	15.24	50.09	0.322	15.17	49.20	0.318
FedLDCF	16.48	54.07	0.346	14.77	48.43	0.311	13.20	45.83	0.276	12.96	45.23	0.272
FRLN	**14.94**	**51.62**	**0.314**	**12.57**	**44.67**	**0.264**	**11.37**	**40.87**	**0.239**	**11.06**	**39.60**	**0.232**
FRLN-Avg	15.02	52.49	0.314	13.81	45.12	0.290	13.48	44.37	0.284	13.47	42.82	0.282
FRLN-(ϵ_1, δ)-DP	17.08	54.02	0.359	14.84	47.39	0.311	13.60	44.27	0.285	13.36	43.53	0.281
FRLN-(ϵ_2, δ)-DP	17.85	56.75	0.377	14.86	46.98	0.312	14.00	44.92	0.295	13.86	44.32	0.291
Gains	9.34%	4.53%	9.45%	14.90%	7.76%	15.19%	13.86%	10.82%	13.40%	14.66%	9.73%	17.2%

learning environment, where the parameter settings used in our FRLN are shown in Table 3. FRLN-Avg uses the same parameter settings as FRLN. Parameter $\epsilon, \delta, Clip$ are specifically utilized in the FRLN-DP variant. In this variant, the per-sample gradients of the client models are clipped according to C, followed by the addition of gaussian noise to these gradients before executing the gradient descent. Considering the efficiency of the DP-SGD algorithm, we set b=8 and E=1. To achieve the best for the rest of competing approaches, we set the optimal model parameters directly, as recommended in their experiments. A larger number of total FL_round is set to ensure convergence to optimal performance for all competing baselines.

Table 4 and Table 5 show the experimental results of federated QoS prediction under multiple densities on RT and TP, respectively. The best results of FRLN and its variants are marked in bold and the best results of other competing approaches are highlighted in gray background. From the results, we demonstrate that all of the competing approaches have a gradual decrease of MAE, RMSE and NMAE on both RT and TP datasets as the density increases from 5% to 20%. The reason is that more available user-service QoS invocation records lead to sufficient training of different QoS prediction models, resulting in better performance on MAE, RMSE and NMAE. More specifically, FRLN outperforms other federated competing approaches, showing a significant improvement of 23.32% on MAE, up to 8.86% on RMSE and 22.90% on NMAE across multiple matrix densities on RT dataset. Likewise, on the TP dataset, FRLN achieved an impressive improvement of 14.9% on MAE, 10.82% on RMSE and 17.2% on NMAE, respectively.

FRLN outperforms competing baselines in both RT and TP datasets on NMAE, potentially indicating its applicability in real service-oriented scenarios. Specifically, taking the RT dataset as an example, it is observed from NMAE and MAE values that invoking a Web service with an average response time close to 1 second, FRLN's average predicted QoS ranges from 0.6 to 1.4 seconds. The results demonstrate that FRLN can predict missing QoS values within an acceptable margin of prediction error, thus highlighting its potential possibility in practical applications. However, further enhancing machine learning frameworks for QoS prediction or expanding larger QoS datasets is of crucial importance towards seamless applications in various real-world distributed computing environments.

FRLN achieves superior performance in QoS prediction accuracy for two key reasons. First, we employ a specially designed RLN network for each client's QoS prediction model. It leverages two-way residual-aware blocks to extract latent user-service nonlinear interaction features, while also considering low and high dimensional feature fusion for better QoS prediction performance. Second, we mitigate the issue of client-drift caused by non-IID QoS user-service invocation records on a single client, which is evidenced by the comparison between FRLN and FRLN-Avg. Specifically, FRLN demonstrates superior performance over FRLN-Avg in both RT and TP datasets. The underlying mechanism of FRLN's enhanced performance can be attributed to its

TABLE 6
Experimental results of centralized QoS prediction under multiple densities on RT dataset

Methods	Density 5%			Density 10%			Density 15%			Density 20%		
	MAE	RMSE	NMAE	MAE	RMSE	NMAE	MAE	RMSE	NMAE	MAE	RMSE	NMAE
UPCC	0.698	1.665	0.7697	0.559	1.466	0.6147	0.496	1.349	0.5463	0.464	1.274	0.5098
LACF	0.631	1.439	0.6965	0.562	1.338	0.6169	0.513	1.269	0.5624	0.477	1.222	0.5251
PMF	0.623	1.532	0.6847	0.528	1.329	0.5818	0.488	1.238	0.5381	0.469	1.202	0.5174
NCF	0.472	1.438	0.5186	0.386	1.314	0.4241	0.362	1.303	0.3991	0.352	1.274	0.3883
LDCF	0.403	**1.277**	0.4427	0.364	1.233	0.4005	0.345	1.169	0.3797	0.331	**1.138**	0.363
RLN	**0.373**	1.303	**0.4108**	**0.325**	**1.227**	**0.3572**	**0.302**	**1.168**	**0.332**	**0.289**	1.146	**0.3181**
Gains	7.44%	-2.00%	7.21%	10.71%	0.49%	10.81%	12.46%	0.09%	12.56%	12.69%	-0.70%	12.37%

TABLE 7
Experimental results of centralized QoS prediction under multiple densities on TP dataset

Methods	Density 5%			Density 10%			Density 15%			Density 20%		
	MAE	RMSE	NMAE	MAE	RMSE	NMAE	MAE	RMSE	NMAE	MAE	RMSE	NMAE
UPCC	31.43	77.08	0.661	24.70	64.18	0.519	22.35	58.95	0.470	21.21	56.16	0.444
LACF	22.97	55.78	0.481	19.44	52.92	0.409	17.58	49.56	0.369	16.45	47.41	0.345
PMF	26.47	67.46	0.556	19.83	50.64	0.416	16.84	47.48	0.354	15.32	43.86	0.321
NCF	18.68	54.65	0.392	14.40	46.22	0.303	13.30	45.35	0.279	12.84	44.95	0.270
LDCF	13.84	**47.35**	0.291	12.38	43.48	0.259	11.27	39.81	0.236	10.84	38.99	0.228
RLN	**13.81**	47.90	**0.290**	**11.25**	**40.82**	**0.237**	**10.42**	**38.01**	**0.219**	**10.13**	**37.13**	**0.213**
Gains	0.22%	-1.16%	0.3%	9.13%	6.12%	8.49%	7.54%	4.52%	7.20%	6.55%	4.77%	6.57%

unique strategy of utilizing the residual ladder layer parameters within each client's QoS prediction model as shared parameters for server federated aggregation. Thus, it treats the remaining parameters as private features, ensuring that they are not shared with other clients. That approach enables each client to leverage local data to train their individualized models by both collaborative learning of shared parameters and keep interior training of personalized features, effectively countering the challenges posed by client-drift in non-IID scenarios of user-service QoS invocation records. Compared to our proposed FRLN, the competing baselines EFMF, FedNCF, FedLDCF and NCSF-GMF partially failed to address the issue of accurately capturing the complex nonlinear interaction relationships between users and services with the limited amount of historical QoS invocation records on each user client. Moreover, FedNCF and FedLDCF achieve better overall QoS prediction performance than EFMF, as they take advantage of the MLPs to capture non-linear user-services interaction features.

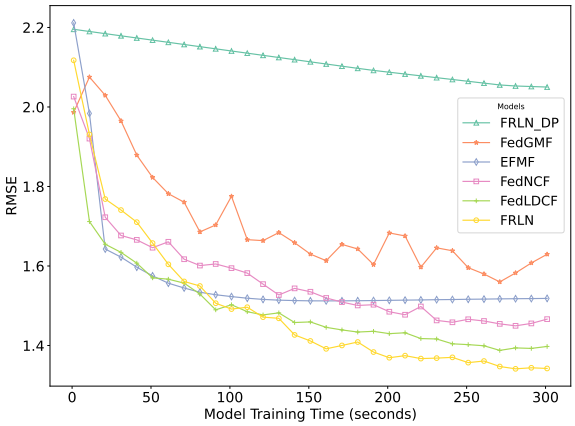


Fig. 5. A comparative analysis of various federated learning models over fixed training time under 5% density of RT dataset. The x-axis denotes the model training time in seconds, while the y-axis represents each model's QoS prediction performance on RMSE.

Analysis on Data-Protection. FRLN has implemented several strategies to protect user-service QoS invocation records. First, leveraging federated learning framework, FRLN uploads the learned model parameters instead of real QoS invocation data, ensuring users' originally local data is not accessed by the central server. Second, although some investigations [30], [31], [32], [33], [34] have raised concerns about potential privacy breaches through the analysis of neural network parameters, FRLN only updates the latent features identified by the layer of residual ladder network for server federated aggregation, while keeping those user-interactive parameters within local QoS prediction model. It aligns with the data processing inequality [12], which can further strength the capability of data protection that these selectively uploaded parameters contain less private information. Third, FRLN is also enhanced with advanced differential privacy (DP) encryption technique DP-SGD algorithm, by locally adding gaussian noise to user gradient updates for perturbing the model parameters before uploading. The experiments reveal a necessary balance between model accuracy and user privacy, emphasizing different levels of data protection in federated learning.

Comparisons on Uniform Training Time. To test the model performance under equivalent training time con-

TABLE 8
Performance changes of QoS prediction approaches under different training settings on both RT and TP datasets

Methods	RT		TP	
	MAE	RMSE	MAE	RMSE
LDCF →FedLDCF	-23.86%	-13.09%	-19.55%	-16.00%
NCF ⟶FedNCF	-14.48%	-4.08%	-18.14%	-9.45%
PMF ⟶EFMF	-0.21 %	0.16 %	-0.13 %	0.02%
RLN ⟶FRLN	-6.92 %	-2.61 %	-9.18%	-6.65%

straints, experiments have been conducted and the results are shown in Figure 5 that although all models exhibit a general reduction on RMSE over time, FRLN stands out with its significantly lower RMSE, achieving superior convergence and its efficacy in latent feature extraction. Specifically, despite an additional complexity due to the incorporation of RLN, FRLN receives its optimal performance within a reasonably acceptable timeframe. It demonstrates certain suitability by personalized QoS prediction model training for data-protected real-world applications, where the accuracy of model prediction is of the utmost importance. As for FRLN-DP, more training time is required for model convergence, since it has the highest data-protected level with the consideration of integrating differential privacy and training personalized parameters. In practical applications, it can regulate the critical equilibrium among QoS prediction performance, user privacy, and model training efficiency in FRLN by applying different data-protected levels.

4.3.2 Experiment Results in Centralized Settings

To further validate the performance of our proposed method, we conducted additional experiments using a centralized training model and compared it with representative centralized QoS prediction competing baselines. Although FRLN is designed for the federated learning paradigm, local QoS prediction model RLN also exhibits outstanding performance in the centralized training model. Tables 6 and 7 show the experimental results of centralized QoS prediction under multiple densities on RT and TP datasets, respectively. Specifically, RLN enhances the MAE, RMSE, and NMAE on the RT dataset by up to 12.69%, 0.49%, and 12.37%, respectively. On the TP dataset, the improvements on MAE, RMSE, and NMAE are up to 9.13%, 6.12%, and 8.49%, respectively.

The reason for the poor performance of traditional CF-based centralized QoS prediction methods is their inability to effectively capture low-dimensional nonlinear user-service interactions, and the sparse historical QoS data limiting their learning of latent user and service features. However, NCF, which utilizes a multi-layer perceptron to exploit nonlinear interactions between users and services, shows a significant improvement in QoS prediction performance by effectively compensating for QoS sparsity limitations. LDCF further enhances the performance by adding contextual information for better feature representation, such as geographical locations of users and services. Overall, our RLN model surpasses LDCF in most QoS density scenarios on the RT and TP datasets. This is due to its effective deep feature extraction, utilizing a bi-directional residual structure for both low and high-dimensional space analysis.

TABLE 9
Impact of BN and Dropout

	RT		TP	
	MAE	RMSE	MAE	RMSE
Neither	0.392	1.310	15.22	51.82
BN	0.423	1.377	18.53	57.91
Dropout	0.379	1.306	14.94	51.62

Thus, RLN allows for the exploration of more latent relationships between users and services, ultimately improving the performance of centralized QoS prediction.

4.3.3 Performance Variations in Two Training Settings

We compare the decline variations of QoS prediction accuracy among four competing approaches under the same density of 20%, by switching training modes from centralized to federated way. Table 8 shows the results of performance changes on RT and TP, respectively. It was observed that, in addition to EFMF, each QoS prediction model experienced a decrease in prediction performance of both MAE and RMSE to different degrees. As for EFMF showing little performance degradation, it can be attributed to the fact that EFMF is considered a federated version of PMF and undergoes a distributed computation of matrix factorization. However, the QoS prediction accuracy of EFMF is inferior when compared to the other three deep learning-based QoS prediction competing approaches.

The superior performance of NCF and LDCF observed in centralized training is not replicated in the federated QoS prediction. In RT dataset, they experienced a decrease on MAE of 14% and 23% for NCF and LDCF; a decrease on RMSE of 4% and 13%, respectively. Also, they have a similar performance drops on TP dataset. The possibility of leading to the phenomenon is that FedNCF and FedLDCF are implemented using conventional federated learning by averaging all of the parameters in NCF and LDCF. However, the heterogeneity of user-service QoS invocation records across different clients is ignored, and it affects the performance of corresponding federated QoS prediction models. In contrast, our proposed FRLN experienced a relatively more moderate performance degradation in the federated training, since it considers the non-IID QoS distributions by training a bunch of personalized local QoS prediction models among multiple clients.

4.4 The Absence of Batch Normalization

Batch Normalization (BN) standardizes input data per mini-batch, enabling faster and more stable network training [35]. However, we observe that BN is typically applied in centralized QoS prediction models, which assume that the accumulated historical QoS records follow the same distribution as the entire dataset. In the federated paradigm, on the other hand, distributed QoS invocation records held by each user are personalized and highly heterogeneous. Additionally, the number of QoS invocations is often small, making it difficult to support large batch sizes. As a result, individual nodes can differ remarkably across multiple clients, leading to a significant degradation in QoS prediction performance.

To validate our hypothesis, experiments were conducted with FRLN under 5% QoS density on RT and TP datasets,

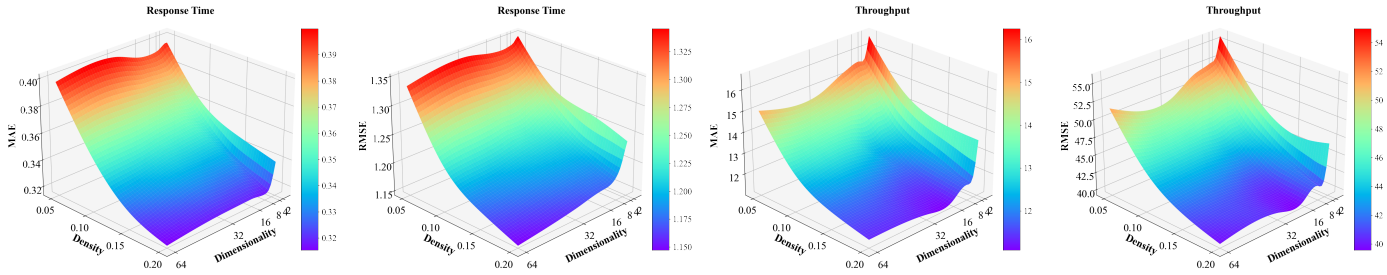


Fig. 6. Performance impact of dimensionality and density.

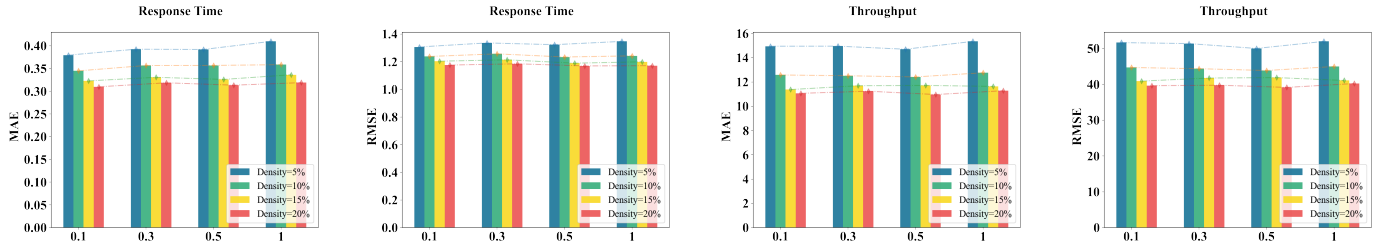


Fig. 7. Performance impact of client selection rates.

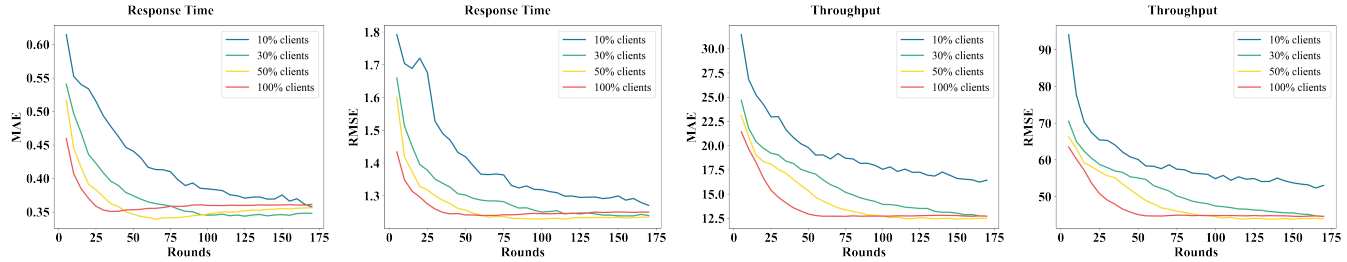


Fig. 8. Convergence efficiency of QoS prediction model training under 10% density with different client selection rate.

with results in Table 9 confirming our assumptions. Specifically, we found that FRLN with BN suffered from obvious performance loss on both MAE and RMSE, especially for TP dataset. However, the use of dropout can effectively improve the QoS prediction accuracy. Therefore, BN is excluded from our designed residual perceptron block of RLN where dropout is used to prevent model overfitting for better prediction performance in FRLN.

4.5 Performance Impact of Parameters

4.5.1 Impact of Dimensionality and Density

To test the performance impact of feature vector dimensionality d, we set six different dimension values of 2, 4, 8, 16, 32, and 64 at 5%, 10%, 15%, and 20% densities on the RT and TP datasets, respectively. Under the above settings, we generate 3D visualizations on MAE and RMSE, as illustrated in Figure 6. The results show that for low density QoS matrices, there emerges an initially decreasing and then increasing variation tendency on MAE and RMSE as the dimensionality of the feature vectors gradually ascends among RT and TP, respectively. The underlying reason is that there is limited information in sparse QoS matrices, the prediction performance is better as the dimensionality increases. However, if the dimensionality continues to increase, the model can dilute useful information about the user-service interactions, making the QoS prediction model less effective due to worsening model overfitting.

Conversely, for denser QoS matrices, the higher dimensionality of the feature vectors facilitates the QoS prediction model in mining more information representations among user and service interactions. As seen in Figure 6, the improvement in QoS prediction performance by increasing the dimensionality of the feature vectors from 2 to 16 is more significant than that by increasing it from 16 to 64. As a result, setting the dimensionality of the feature vectors to 16 can achieve the optimal QoS prediction performance across multiple QoS densities.

4.5.2 Impact of Client Selection Rate

To assess the impact of the client selection rate C that determines the number of users participating in a particular training round, we conducted experiments with four different values of 0.1, 0.3, 0.5, and 1 under four QoS densities, including 5%, 10%, 15% and 20%, respectively. Figure 7 demonstrates that for low density QoS matrices, a higher client selection rate C results in decreased QoS prediction performance on MAE and RMSE overall. This is due to

each client having fewer QoS invocation records at a low density, resulting in inadequate model training. An increase in client selection rate C introduces undertrained clients, which further reduces the model performance. Therefore, by selecting a smaller client selection rate, we can improve the model's generalization ability. However, with high density QoS matrices, more local data is included in the training process, enabling every client to receive sufficient training and become fully trained. In such cases, an increase in the client selection rate C does not significantly reduce QoS prediction performance.

Furthermore, we evaluate the convergence efficiency of QoS prediction model training under a fixed 10% QoS density by adjusting C. As shown in Figure 8, decreasing C increases the required training rounds for model convergence. For instance, in the RT dataset, C=1 converges in 50 rounds, while C=0.1 remains unconverged after 170 rounds. It indicates that more clients participating in a single round of model training increase the time required, while fewer clients participating reduce the time per round but increase the total number of communication rounds.

5 Threats to Validity

Threats to application validity. In our proposed data-protected FRLN, each client maintains its local user-service QoS invocation records in a distributed manner, posing significant challenges to traditional centralized QoS prediction methods. Specifically, FRLN in real-world distributed computing scenarios, like edge computing networks, where dynamic user mobility significantly affects network connectivity and interactions between edge users and servers, may introduce additional complexities to the QoS prediction task. Additionally, the limited computing capacity of edge servers and the restricted node coverage in edge computing networks pose further challenges to FRLN's applicability. Therefore, future efforts should focus on refining FRLN to enhance QoS prediction in various downstream distributed computing paradigms, like mobile edge computing networks, addressing both theoretical and practical challenges to confirm FRLN's applicability. Furthermore, according to experimental results on NMAE, current machine learning techniques are still not sufficiently sophisticated for high-precision QoS prediction in practical applications. Thus, it is expected to pose more efforts on exploring more advanced machine learning frameworks and incorporating larger datasets to further improve QoS prediction accuracy, better satisfying the application demands in real service-oriented scenarios similar to edge computing environments by providing low response latency services.

Threats to dataset validity. In the experiments, we employed WS-DREAM dataset for the evaluation of the proposed FRLN. It comprises real-world user-service QoS invocation records and is widely-used for QoS prediction task, which includes a comprehensive range of characteristics essential for effective QoS prediction. Under this experimental circumstances, FRLN integrates deeply with WS-DREAM dataset, utilizing its rich contextual elements such as multi-granularity geographical information to enhance the feature learning and representation of both users and services. Despite the superior results achieved with the WS-DREAM dataset, the threats of FRLN generalizability need to be validated by adapting to other experimental datasets with diverse contextual information. To this end, it is expected to involve extending FRLN's application to a broader range of datasets, with a specific focus on recommender systems. It can improve the adaptability, applicability, and robustness of FRLN, thereby enlarging its long-term utility and relevance across various application scenarios.

6 Related Work

6.1 Collaborative Filtering based QoS Prediction

The basic idea of memory-based CF approaches for QoS prediction lies in using similarity calculation to obtain similar neighborhoods, and predicting unknown QoS value. Shao et al. [4] first introduced using the Pearson correlation coefficient (PCC) to determine user similarity and predict the QoS for a target user and service. Zheng et al. [22] proposed WSRec, integrating similar user and service considerations for recommendations. To enhance similarity calculation accuracy, researchers began incorporating contextual factors like reliability, time, and location. Chen et al. [6] considered the reputation value as well as location information before similarity calculation. Zou et al. [5] proposed an reinforced collaborative filtering algorithm by filtering noisy services with low similarity to the target service, which significantly improves the accuracy of QoS prediction. However, in real application scenarios, the prediction accuracy of memory-based collaborative filtering is remarkably reduced due to the sparse QoS invocation records.

Model-based CF methods, such as matrix factorization and gradient descent, tackle memory-based sparsity by using historical QoS data to predict unknown values. PMF [23] uses a probabilistic model to optimize matrix decomposition for better recommendations. Wang et al. [36] developed a multi-dimensional CF approach for QoS prediction HDOP.

6.2 Deep Learning based QoS Prediction

Traditional CF-based QoS prediction approaches are dedicated to learning low-dimensional, linear user-service invocation relationships. When faced with the sparsity of historical QoS invocation records, it is inadequate for learning more latent features. Recently, deep learning techniques have been initially applied to extract user-service complex nonlinear invocation relationships, boosting the QoS prediction performance. He et al. [18] first applied deep learning techniques to recommender systems, thus solving the problem that matrix factorization fails to express high-dimensional nonlinear features. Xu et al. [19] proposed SDNN with lateral connections and further improved NCF by learning relations in both high-dimensional and low-dimensional spaces simulatneously, which is the state-of-art in recommender system.

Wu et al. [37] proposed a deep neural model DNM for multi-attribute QoS prediction, which captures the higher-order interaction features of users and services through the interaction and perception layers of the DNM. Zhang et al. [20] introduced LDCF, a location-aware deep neural network for collaborative filtering, incorporating geographical data into the model. LDCF connects deep learning with collaborative filtering using a similarity adaptive corrector, integrating user and service context into its features and

learning complex user-service relationships. However, these methods are part of centralized QoS prediction, aggregating historical user-service QoS data for training, making data-protected QoS prediction challenging.

6.3 Data-Protected QoS Prediction

Zhu et al. [38] first focused on privacy protection for service recommendation and proposed a simple and efficient data-protected framework with the help of data obfuscation techniques, under which two typical data-protected QoS prediction approaches were developed, including, P-UIPCC and P-PMF. Liu et al. [39] proposed a data-protected collaborative QoS prediction framework by introducing differential privacy techniques, which can protect users' private data, while maintaining accurate QoS prediction capability. In recent years, some researchers have started to consider applying the federated learning framework to QoS prediction. Zhang et al. [24] proposed a data-protected QoS prediction approach based on the federated learning matrix decomposition technique, and further improved the prediction efficiency by reduction strategy. However, existing data-protected approaches ignore the significant impact on local QoS prediction model performance and personalized model training by considering diverse data distribution across different user clients.

Recent research has identified new security and privacy threats in federated learning. Huang et al. [33] explore gradient inversion attacks in federated learning and present effective mitigation with little impact on data utility. Fraboni et al. [34] discuss "free-rider attacks" in federated learning, offering strategies for detection and prevention in environments with limited data and important models.

7 Conclusion

In this paper, we propose a novel framework of data-protected QoS prediction based on federated residual ladder network, named FRLN. First, we design a residual ladder network as a feature extraction model called RLN, which captures the complex nonlinear invocation relationships between users and services from both low and high dimensional spaces with forward and backward residual-aware blocks, which is more beneficial to learn latent features of users and services for better QoS prediction accuracy. Second, we propose a personalized federated model training strategy to overcome the QoS prediction performance loss due to data heterogeneity across multiple user clients, where only the global expressions learned from RLN model are uploaded to the cloud server for aggregation. It further improves the accuracy of QoS prediction, while protecting users' data privacy of QoS invocations. We conduct extensive experiments on two large-scale QoS datasets, and the results demonstrate that FRLN achieves the best performance for data-protected QoS prediction. While the proposed FRLN framework marks a significant step forward for QoS prediction in machine learning, ongoing advancements in ML techniques and expanded datasets are essential for further enhancing its precision and applicability in real-world scenarios.

Acknowledgments

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

References

[1] S. H. Ghafouri, S. M. Hashemi, and P. C. K. Hung, "A Survey on Web Service QoS Prediction Methods," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 2439–2454, 2022.

[2] K. Kritikos and D. Plexousakis, "Requirements for QoS-Based Web Service Description and Discovery," *IEEE Transactions on Services Computing*, vol. 2, no. 4, pp. 320–337, 2009.

[3] Y. Syu, J. Kuo, and Y. Fanjiang, "Time series forecasting for dynamic quality of Web services: An empirical study," *Journal of Systems and Software*, vol. 134, pp. 279–303, 2017.

[4] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS Prediction for Web Services via Collaborative Filtering," in *IEEE International Conference on Web Services*, 2007, pp. 439–446.

[5] 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.

[6] K. Chen, H. Mao, X. Shi, Y. Xu, and A. Liu, "Trust-aware and Location-based Collaborative Filtering for Web Service QoS Prediction," in *IEEE Annual Computer Software and Applications Conference (COMPSAC)*, 2017, pp. 143–148.

[7] 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.

[8] Z. Chen, L. Shen, and F. Li, "Exploiting Web service geographical neighborhood for collaborative QoS prediction," *Future Generation Computer Systems*, vol. 68, pp. 248–259, 2017.

[9] Y. Yang, Z. Zheng, X. Niu, M. Tang, Y. Lu, and X. Liao, "A Location-Based Factorization Machine Model for Web Service QoS Prediction," *IEEE Transactions on Services Computing*, vol. 14, no. 5, pp. 1264–1277, 2021.

[10] G. Zou, S. Wu, S. Hu, C. Cao, Y. Gan, B. Zhang, and Y. Chen, "NCRL: Neighborhood-based Collaborative Residual Learning for Adaptive QoS Prediction," *IEEE Transactions on Services Computing*, vol. 16, no. 3, pp. 2030–2043, 2023.

[11] H. Gao, Y. Xu, Y. Yin, W. Zhang, R. Li, and X. Wang, "Context-aware QoS Prediction with Neural Collaborative Filtering for Internet-of-Things Services," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4532–4542, 2020.

[12] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273–1282.

[13] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[14] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *International Conference on Machine Learning (ICML)*, 2021, pp. 2089–2099.

[15] H. Li, X. Sun, and Z. Zheng, "Learning to attack federated learning: A model-based reinforcement learning attack framework," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35 007–35 020, 2022.

[16] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning," in *International Conference on Machine Learning (ICML)*, 2020, pp. 5132–5143.

[17] J. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," 1998.

[18] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering." International World Wide Web Conferences Steering Committee (WWW), 2017, pp. 173—-182.

[19] R. Xu, J. Li, and G. e. a. Li, "SDNN: Symmetric deep neural networks with lateral connections for recommender systems," *Information Sciences*, vol. 595, pp. 217–230, 2022.

[20] Y. Zhang, C. Yin, Q. Wu, Q. He, and H. Zhu, "Location-aware Deep Collaborative Filtering for Service Recommendation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3796–3807, 2021.

[21] H. Touvron, P. Bojanowski, M. Caron *et al.*, "ResMLP: Feedforward Networks for Image Classification with Data-Efficient Training," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 5314–5321, 2023.

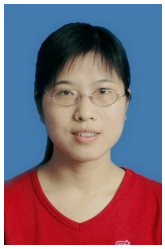
[22] Zibin Zheng, Hao Ma, M. R. Lyu, and I. King, "QoS-Aware Web Service Recommendation by Collaborative Filtering," *IEEE Transactions on Services Computing*, vol. 4, no. 2, pp. 140–152, 2011.

[23] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in Neural Information Processing Systems (NIPS)*, 2007, pp. 1–8.
[24] 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.
[25] 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.
[26] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.
[27] 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, 2013.
[28] J. Liu, M. Tang, Z. Zheng, X. Liu, and S. Lyu, "Location-aware and personalized collaborative filtering for web service recommendation," *IEEE Transactions on Services Computing*, vol. 9, no. 5, pp. 686–699, 2015.
[29] 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, 2021.
[30] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*, 2019, pp. 739–753.
[31] M. Vero, M. Balunović, D. I. Dimitrov, and M. Vechev, "TabLeak: Tabular data leakage in federated learning," in *Proceedings of the 40th International Conference on Machine Learning (ICML)*, vol. 202, 2023, pp. 35 051–35 083.
[32] P. Kairouz, H. B. McMahan, B. Avent *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.
[33] Y. Huang, S. Gupta, Z. Song, K. Li, and S. Arora, "Evaluating gradient inversion attacks and defenses in federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7232–7241, 2021.
[34] Y. Fraboni, R. Vidal, and M. Lorenzi, "Free-rider attacks on model aggregation in federated learning," in *International Conference on Artificial Intelligence and Statistics*, 2021, pp. 1846–1854.
[35] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015.
[36] S. Wang, Y. Ma, B. Cheng, F. Yang, and R. N. Chang, "Multi-Dimensional QoS Prediction for Service Recommendations," *IEEE Transactions on Services Computing*, vol. 12, no. 1, pp. 47–57, 2019.
[37] H. Wu, Z. Zhang, J. Luo, and et al., "Multiple Attributes QoS Prediction via Deep Neural Model with Contexts," *IEEE Transactions on Services Computing*, vol. 14, no. 4, pp. 1084–1096, 2021.
[38] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "A Privacy-Preserving QoS Prediction Framework for Web Service Recommendation," in *IEEE International Conference on Web Services(ICWS)*, 2015, pp. 241–248.
[39] A. Liu, X. Shen, Z. Li, G. Liu, J. Xu, L. Zhao, K. Zheng, and S. Shang, "Differential private collaborative Web services QoS prediction," *World Wide Web*, vol. 22, no. 6, pp. 2697–2720, 2019.

Guobing Zou is a full professor and vice dean of the School of Computer Science, 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 110 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, Information Sciences, Expert Systems with Applications, Knowledge-Based Systems, etc.

Wenzhuo Yu is currently a master student in the School of Computer Engineering and Science, Shanghai University, China. Before that, he received a Bachelor degree in Computer Science and Technology at Hangzhou Dianzi University, China, 2021. His research interests include QoS prediction, 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 preprocess and analyze large-scale real-world diagnosis and treatment data, and apply the results to medical assisted diagnosis and treatment services.

Shengxiang Hu is currently a PhD candidate in the School of Computer Engineering and Science, Shanghai University, China. Before that, he received a Bachelor degree in 2018 and Master degree in 2021 both 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 more than five papers on IEEE Transactions on Services Computing, International Conference on Service-Oriented Computing (ICSOC), Knowledge-Based Systems, International Conference on Parallel Problem Solving from Nature (PPSN), etc.



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 human-computer 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.