Privacy-Enhanced Federated Expanded Graph Learning for Secure QoS Prediction

Guobing Zou, Zhi Yan, Shengxiang Hu, Yanglan Gan, Bofeng Zhang, and Yixin Chen, *Fellow, IEEE*

***Abstract*—Current state-of-the-art QoS prediction methods face two main limitations. First, most existing QoS prediction approaches are centralized, gathering all user-service invocation QoS records for training and optimization, which causes privacy breaches. While some federated learning-based methods consider user privacy in a distributed way, they either directly upload local trained parameters or use simple encryption for global aggregation at the central server, thus failing to truly protect user privacy. Second, existing federated learning-based methods neglect distributed user-service topology and latent behavior-attribute correlations, compromising QoS prediction accuracy. To address these limitations, we propose a novel framework named Privacy-Enhanced Federated Expanded Graph Learning (PE-FGL) for secure QoS prediction. It first conducts user-service expansion on the invocation graph with advanced privacy-preserving techniques, upgrading first-order local QoS invocations to high-order interaction relationships. Then, it extracts hybrid features from the expanded invocation graph via deep learning and graph residual learning. Finally, a two-layer secure mechanism of federated parameters aggregation is designed to enable collaborative learning among users through local parameter segmentation and global aggregation, achieving effective and secure QoS prediction. Extensive experiments on WS-DREAM demonstrate effective QoS prediction across multiple metrics while preserving privacy in user-service invocations.**

***Index Terms*—Web service, secure QoS prediction, expanded invocation graph, hybrid feature extraction, federated parameter aggregation.**

I. Introduction

In recent years, the proliferation of web services has made them an important driving force for developing integrated applications through Service-Oriented Architectures (SOA) and the Internet of Services (IoS). Service providers are rapidly introducing new web services, leading to an abundance of similar options and making it increasingly challenging for consumers to choose the most suitable service. Quality of Service (QoS)—which encompasses diverse metrics such as response time, throughput, availability, and cost—emerges as a vital discriminant for comparable services by assessing their non-functional attributes. However, given the impracticality of monitoring every web service invocation to collect QoS data, researchers are confronted with the task of effectively predicting QoS based on sparse user-service historical invocations. It is of paramount importance for service-oriented applications, including service discovery, selection, recommendation, and mashup creation [1], [2], [3].

Collaborative Filtering (CF) is fundamental to QoS prediction, with approaches categorized into memory-based and model-based ones. Memory-based CF approaches predict unobserved QoS values using historical data and similarity metrics such as Pearson Correlation Coefficient (PCC) [4] and Ratio-Based Similarity (RBS). However, they are susceptible to performance degradation due to the sparsity of user-service interaction records, which negatively impacts prediction accuracy. Model-based CF approaches, including clustering, matrix factorization [5], [6], and machine learning algorithms [7], [8], have been developed to deduce user and service feature representations, reveal underlying linear or nonlinear associations to enhance QoS prediction accuracy. Furthermore, researchers have integrated contextual information of users and services, such as geographic location and temporal invocation series, to refine QoS prediction accuracy [9], [10]. Recently, deep learning models have been employed to extract complex nonlinear feature representations, further improving QoS prediction performance [1], [3], [11].

Despite advancements in QoS prediction approaches, they still fail to ensure user privacy and achieve optimal performance in service-oriented applications. First, traditional methods rely on centralized data processing, which collects users' historical QoS data and exposes privacy during similarity calculation or model training. Although some federated learning approaches, such as those in [12], [13], upload local parameters to a cloud server or use basic encryption, they may not sufficiently protect model parameters, due to suffering from the reconstruction risk of original data by attackers. Second, recent federated learning-based QoS prediction approaches, like those in [14], [15], partition historical QoS data by users, neglecting complex user-service topological structures, which hinders the extraction of deep features and reduces QoS prediction performance.

To address the two limitations, we propose a novel framework named Privacy-Enhanced Federated expanded Graph Learning for secure QoS prediction (PE-FGL). Specifically, we design a privacy-preserving graph expansion method that generates

Received 25 September 2024; revised 22 January 2025; accepted 20 March 2025. Date of publication 10 April 2025; date of current version 12 June 2025. This work was supported by National Natural Science Foundation of China under Grant 62272290 and Grant 62172088. *(Corresponding authors: Yanglan Gan; Bofeng Zhang.)*

Guobing Zou, Zhi Yan, and Shengxiang Hu are with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China (e-mail: gbzou@shu.edu.cn; cmengyz@shu.edu.cn; shengxianghu@shu.edu.cn).

Yanglan Gan is with the School of Computer Science and Technology, Donghua University, Shanghai 201620, China (e-mail: ylgan@dhu.edu.cn).

Bofeng Zhang is with the School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai 201209, China (e-mail: bfzhang@sspu.edu.cn).

Yixin Chen is with the Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA (e-mail: chen@cse.wustl.edu).

Digital Object Identifier 10.1109/TSC.2025.3559613

high-order user-service invocation relationships by expanding one-order local QoS records in a distributed manner. Then, we present a Hybrid Feature Extraction (HFE) network based on deep learning and graph residual learning to learn sophisticated, high-dimensional feature representations and complex interaction patterns, thereby improving the deep feature extraction of users and services. Finally, we design a multi-granularity secure federated parameter aggregation strategy that employs cryptographic measures [16], [17] to ensure secure collaborative learning through the partitioning of local parameters, resulting in effective and secure QoS prediction.

To ascertain the performance of our proposed PE-FGL, we carry out a comprehensive suite of experiments on the WS-DREAM dataset [18]. It encompasses QoS invocation records from 5,825 web services and 339 users, spanning 74 diverse geographic locales, totaling 1,974,675 user-service QoS invocation records. They are segregated into distinct user-service QoS invocation clusters based on service users, thereby maintaining stringent user privacy standards. The results demonstrate that PE-FGL outperforms state-of-the-art baselines across multiple evaluation metrics, particularly in privacy-preserving QoS prediction, thereby validating its superior performance in QoS prediction.

The main contributions are summarized as follows:

- We propose a novel framework called Privacy-Enhanced Federated Expanded Graph Learning for secure QoS prediction (PE-FGL). It incorporates Federated Learning [13], [19] as the foundation to facilitate collaborative training of a comprehensive QoS prediction model across multiple service users maintaining the decentralization of datasets, and applies privacy computing techniques to enhance secure QoS prediction by partitioned local parameter upload transmission and back propagation. Moreover, PE-FGL employs Graph Neural Networks (GNNs) [20] to analyze the extended user-service invocation graph for each entity, capturing complex interactions and extracting high-order latent features, which enhances both the security and accuracy of QoS prediction.
- We propose a suite of secure QoS prediction approach in federated learning, where it includes creating user-service expanded invocation graph by the encryption and decryption of users' and services' interactive information, extracting hybrid features of users and services by deep learning and graph residual learning, and two-layer local parameters segmentation and aggregation by intermediate computing units and federated aggregation server.
- We conduct extensive experiments on the WSDREAM dataset. The results show that, while guaranteeing to protect user privacy, PE-FGL can still achieve superior QoS prediction accuracy compared to state-of-the-art privacy-preserving competing approaches.

The remainder of this paper is organized as follows. Section II reviews related work. Section III formulates the problem of privacy-preserving QoS prediction. Section IV illustrates the proposed framework of PE-FGL. Section V elaborates the approach of privacy-preserving QoS prediction. Section VI displays and analyzes the experimental results. Section VII discusses the practical significance and challenges of PE-FGL. Section VIII concludes the paper and discusses future work.

II. Related Work

A. Collaborative Filtering and Deep Learning Based QoS Prediction

QoS prediction approaches based on Collaborative Filtering (CF) comprise memory-based and model-based methods. For memory-based approaches, Shao et al. [21] utilized PCC for prediction but encountered data sparsity issues. Zheng et al. [7] combined user and service neighborhoods for better accuracy, yet showed limited neighborhood feature modeling. Tang et al. [10] proposed LACF with location information to address sparsity, though raising privacy concerns. In model-based methods, Devi et al. applied Non-negative Matrix Factorization (NMF) [5], but faced limitations due to computational complexity and non-negativity constraints. Mnih et al. developed Probabilistic Matrix Factorization (PMF) [6], which struggled with non-linear relationship modeling. Tang et al. [22] and Xu et al. [23] introduced NAMF with user neighborhood data and enhanced it with reputation and geographical information, yet confronted challenges in data compatibility and feature integration.

The integration of deep learning into QoS prediction has brought innovative approaches. He et al. [24] introduced NCF, utilizing MLP to capture complex nonlinear interactions, which significantly improved prediction accuracy but neglected contextual information of users and services. Zhang et al. [25] developed LDCF, integrating MLP with a similarity-adaptive calibrator for enhanced QoS prediction accuracy, yet it faces challenges in computational complexity and privacy protection. Xia et al. [11] proposed JDNMFL, employing CNN to analyze multi-source data for improved accuracy, though lacking adaptability analysis in dynamic environments. Zou et al. [26] introduced NCRL, a twin-tower deep residual network incorporating neighborhood information, but failed to consider the structural information between users and services. Lu et al. [27] addressed noise and label imbalance issues through a supervised feedforward neural network modeling Gaussian feature distributions, but showed limitations in data distribution assumptions.

B. Privacy-Preserving QoS Prediction

Traditional centralized approaches for predicting QoS typically involve collecting and analyzing user-service QoS invocation records, which poses a risk of sensitive user privacy information leakage.

To enhance privacy protection, Zhu et al. [28] introduced a service recommendation framework utilizing data obfuscation techniques to protect user information, though it may introduce noise affecting QoS prediction accuracy. Liu et al. [29] subsequently developed a collaborative framework for QoS prediction incorporating differential privacy to protect individual data while maintaining prediction accuracy, yet its effectiveness may vary depending on data characteristics and service properties. Extended from [29], Zhang et al. [30] employed fine-grained differential privacy strategies and collaborative filtering to dynamically mask QoS data, while leveraging server geographical location information to access and utilize records from users with similar profiles for more accurate QoS prediction.

With the widespread application of federated learning in service computing, Jin et al. [31] proposed a security-aware

QoS prediction method for mobile edge computing environments. While it protected distributed data security, it encountered practical implementation challenges, including substantial communication overhead and potential security vulnerabilities. To address these limitations, Zhang et al. [14] introduced a federated learning QoS prediction method based on data reduction techniques. Although it achieved data privacy protection, the model remained susceptible to security risks in communication channels and parameter aggregation processes. Recently, Zou et al. [15] designed a comprehensive joint QoS prediction framework that significantly enhanced prediction accuracy by integrating hierarchical clustering algorithms, fine-grained user partitioning strategies, and context-aware deep neural networks. However, it failed to adequately consider the high-order structural dependencies in user-service interactions, thereby limiting the model's expressive capability as well as privacy protection of user-service QoS invocations.

III. Problem Formulation

In this section, we investigate secure QoS prediction with a special emphasis on issues pertaining to privacy protection of user-service invocations. By presenting formal definitions, we formulate the problem of securely predicting unknown QoS values.

Definition 1 (Service User): A service user is an entity that has utilized at least one web service and is characterized by a unique identifier and associated location attributes. Specifically, a user u in the set $U = \{u_1, u_2, \ldots, u_m\}$ is represented by a tuple $u =< ID, RG, AS >$, where ID denotes the user's unique identifier, and RG, AS represent the user's multi-granularity geographical positioning.

Here, the location information of a service user primarily comprises two parts: the Region (RG) and the Autonomous System (AS).

Definition 2 (Web Service): For the QoS prediction problem, we focus on the non-functional attributes of web services, which are represented as tuples comprising an identifier and location information. Specifically, within the collection $S = \{s_1, s_2, \ldots, s_n\}$, a web service is denoted by a tuple $s =< ID, RG, AS >$, where ID is the service's unique identifier. In the same way, RG and AS refer to the service's Region and Autonomous System, respectively.

Given a service user u, there are a set of corresponding web services invoked by the user with privacy-preserving QoS invocation records. It is defined as below.

Definition 3 (Privacy-Preserving QoS Invocation Record): A user-service invocation record is defined as a triplet $r =< u, s, r_{u,s} >$, where u is a user from U, s is a web service from S, and $r_{u,s}$ is the resulting QoS value from invoking s by u.

Matrix R captures user-service QoS interactions, with rows representing users and columns representing services. Each entry $R_{u,s}$ denotes the QoS value for user u invoking service s. An interaction is represented by the triplet $< u_i, s_j, r_{u_i,s_j} >$, where r_{u_i,s_j} is the corresponding QoS value. The absence of the triplet in R indicates that u has not invoked s.

Therefore, how to effectively predict missing QoS value under the condition of protecting the privacy of user-service invocation records has become a challenging research to be solve. It is defined as below.

Definition 4 (Secure QoS Prediction Problem): To address the challenge of QoS prediction within a privacy-preserving context, we define a secure QoS prediction problem as $\Omega =< U, S, R', u, s >$, where it consists of a set of users U, a set of services S, and a privacy-preserving QoS invocation records set R' that omits any user-service pair $< u, s >$ for preserve the privacy of user-service invocations.

In contrast to traditional centralized QoS prediction problem, it employs a collection of disjoint submatrices $R' = \{R_1, R_2, \ldots, R_m\}$ rather than the entire matrix R, where each user-service QoS invocation record is transfered to a privacy-preserving representation. Thus, it ensures user privacy while performing the prediction of $< u, s, \hat{r}_{u,s} >$, by secured collaborative way among multiple service users.

IV. Framework

To achieve secure and effective QoS prediction, the proposed framework PE-FGL, illustrated in Fig. 1, consists of three interrelated components: privacy-preserving user-service graph expansion, hybrid feature extraction, and secure federated parameters aggregation. The processes of these components are described below.

- The privacy preserving user-service graph expansion module starts with users generating anonymized vectors to conceal identities and transmit encrypted invocation records with associated privacy metadata to the server. Subsequently, the server establishes user-service invocation relationships using their identifiers and adaptively distributes the necessary expansion associations. Each user then decrypts the expanded information to construct a high-order user-service invocation graph.
- Based on the privacy-preserving user-service expanded invocation graph, hybrid feature extraction embeds the initial features of users and services using their descriptive information. Initially, the features are input into a multi-layer perceptron to extract deep semantic features. Subsequently, graph residual learning is employed to refine these features by aggregating information from neighboring nodes in the user-service expanded invocation graph, capturing high-order interactive features. These two kinds of extracted features are ultimately combined at the predictive layer to predict unknown QoS within each service user.
- To facilitate global learning among service users while training their own local QoS prediction models, the multi-granularity federated parameter aggregation adopts a two-level structure for segmenting and encrypting parameters. Parameters are segmented and encrypted pre-transmission, and intermediate calculation nodes bridge users and the central aggregation server, amalgamating parameters in subsets. The combined security measures of parameter segmentation and homomorphic encryption guarantee the secure aggregation of parameters, while improving the effectiveness of QoS prediction by distributed collaboration among service users.

V. Approach

A. Privacy-Preserving User-Service Invocation Graph Expansion

In the context of federated learning, user local datasets mainly consist of service invocation QoS records, which are

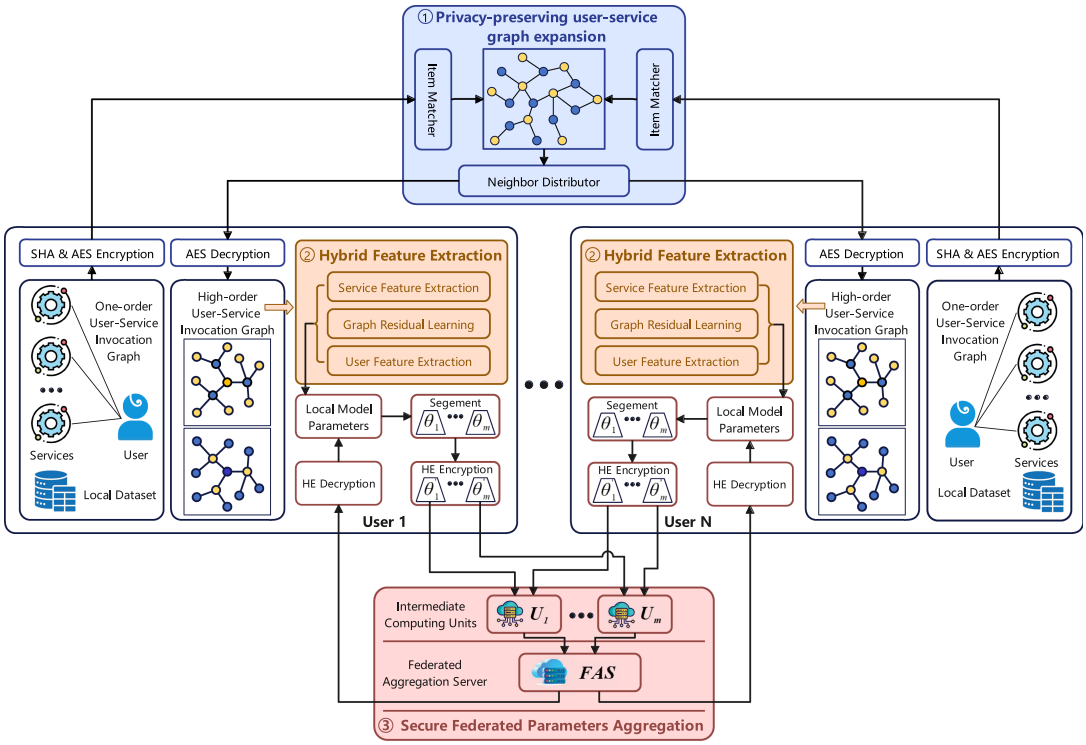


Fig. 1. The framework of privacy-enhanced federated expanded graph learning for secure QoS prediction.

instrumental in creating first-order user-service invocation graph. However, research by Zhou et al. [32] suggests that GNNs are more effective when integrating second or third-order neighborhood data. Our goal is to enhance user local data to construct high-order user-service invocation graphs, while rigorously maintaining user privacy.

First, we employ a key distribution strategy using a Key Distribution Center (KDC) to securely distribute a global key to users through asymmetric encryption. We then present a privacy-aware graph expansion mechanism, as shown in Fig. 2, enabling users with the global key to securely expand their local QoS from one-order to high-order user-service invocation relationships via an expansion server, thereby ensuring sensitive data confidentiality. Before graph expansion, the local dataset $\mathcal{D}_i$ of user u_i comprises user-service interaction information, as depicted by:

$$\mathcal{D}_i = \{(u_i, s_j, m_{i,j}) \mid \forall s_j \in \mathcal{S}\} \tag{1}$$

Where u_i and s_j are the identifiers for the i-th user and j-th service, respectively. S represents the aggregation of services, and $m_{i,j}$ aggregates all relevant details pertaining to the usage of service s_j by user u_i.

The overall graph expansion process is shown in Algorithm 1. In the user upload phase, the algorithm secures data transmission by encrypting user information through hash functions and AES encryption. The server aggregation phase performs global integration of encrypted data and k-step graph expansion. In the user download phase, users decrypt the expanded subgraph structures to complete local higher-order graph construction. Based on Algorithm 1, we can further formalize the key operations, the detailed procedure is as follows.

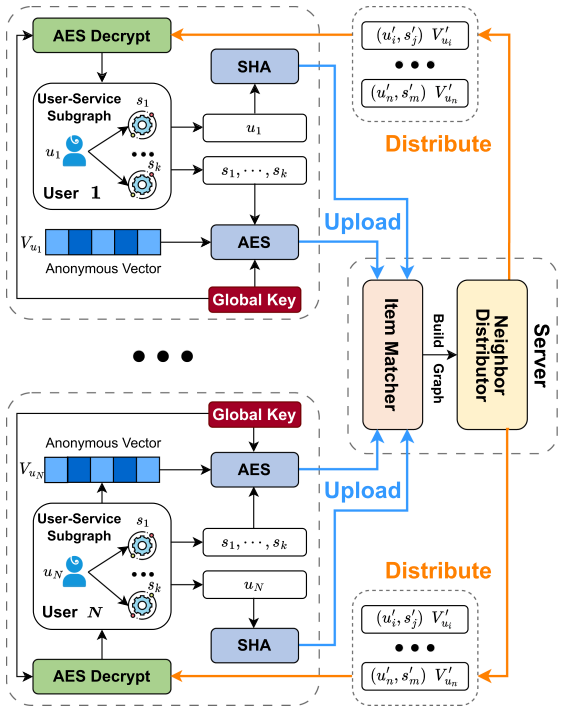


Fig. 2. Privacy-preserving user-service invocation graph expansion.

1) One-Order User Service Invocation Graph Encryption: To protect user privacy during graph expansion and ensure node anonymization, user u_i uses the anonymous vector V_{u_i}. We introduce a random vector generator to guarantees each user's

vector is unique, unpredictable, and uncorrelated, as expressed by the formula:

$$V_{u_i} = \text{Gen}(u_i, \text{seed}, \text{params}) \tag{2}$$

The function Gen(*) generates anonymous vectors V_{u_i} using the user identifier u_i and a cryptographically secure seed. It ensures unique vector generation, distinguishes user information, and hides real identities to prevent identity leakage. The secure seed guarantees randomness, preventing attackers from inferring user information. The params parameter optimizes vector characteristics, balancing complexity and enhancing the model's ability to capture user-service relationships and improve QoS prediction accuracy.

To enhance privacy, u_i is encrypted as u_i' by SHA-512 hash algorithm [33]. As shown in (3), $\text{H}_{\text{SHA}}(*)$ is the hash function.

$$u_i' = \text{H}_{\text{SHA}}(u_i) \tag{3}$$

For user privacy protection, we finally use the Advanced Encryption Standard (AES) [16] symmetric encryption algorithm, users encrypt s_j and V_{u_i} with the global key as the AES cipher key. The process is detailed as follows:

$$\begin{aligned} s_j' &= \text{Enc}_{\text{AES}}(key, s_j) \\ V_{u_i}' &= \text{Enc}_{\text{AES}}(key, V_{u_i}) \end{aligned} \tag{4}$$

Here, key denotes the global key, and s_j', V_{u_i}' are the AES-encrypted ciphertexts of s_j and V_{u_i}, respectively. In the graph expansion module, SHA-512 hashes user identifiers, ensuring privacy and data integrity, while AES encrypts services and anonymous vectors to maintain confidentiality. User IDs are SHA-512 hashed, and services and anonymous vectors are AES-encrypted before being uploaded to the server. The server builds a global encrypted graph, ensuring user ID integrity with SHA-512 and ensuring service data confidentiality with AES. During graph distribution, the user decrypts service nodes using AES and combines SHA-512-processed ID information to expand the local invocation graph for QoS prediction. This synergy enables secure data handling, supporting the stable and private operation of PE-FGL.

The ciphertext dataset $\mathcal{D}_i'$ that needs to be uploaded to the server for neighbor user expansion should be the ciphertext invocation relation $\mathcal{I}_i$ with the anonymization vector V_{u_i}' as in the following equation:

$$\begin{aligned} \mathcal{I}_i' &= \{(u_i', s_j') \mid \forall s_j \in S\} \\ \mathcal{D}_i' &= \{\mathcal{I}_i', V_{u_i}'\} \end{aligned} \tag{5}$$

2) Global User Service Invocation Graph Construction: The graph aggregation server merges the ciphertext dataset $\mathcal{D}_i'$ from users into the global ciphertext dataset $\mathcal{D}'$:

$$\mathcal{D}' = \bigcup_{i=1}^{n} \mathcal{D}_i' \tag{6}$$

Using u_i' and s_j' as indices, the server can assemble a global encrypted user-service invocation graph from $\mathcal{D}'$ using a specified alignment method, all without data decryption. The definition of this graph is as follows:

$$\mathcal{G}' = (\mathcal{U}', \mathcal{S}', \mathcal{E}, \mathcal{A}') \tag{7}$$

Where $\mathcal{U}'$ and $\mathcal{S}'$ respectively denote the sets of encrypted user and service. The inclusion of a tuple (u_i', s_j') in the dataset $\mathcal{D}'$ signifies an edge $e_{u_i', s_j'}$ within the encrypted graph $\mathcal{G}'$. Consequently, the edge set E' can be defined by the following relation:

$$\mathcal{E} = \{e_{u_i', s_j'} \mid (u_i', s_j') \in \mathcal{D}'\} \tag{8}$$

Moreover, an attribute function $\mathcal{A}$ is established to bind anonymization vector V_{u_i}' to its respective user node.

$$\mathcal{A} : V_{u_i}' \to u_i' \tag{9}$$

3) Adaptive Extended Graph Distribution: For the target user u_i', the server adaptively constructs an expanded graph. Specifically, we initiate a k-$step$ graph expansion operation from u_i' on the graph $\mathcal{G}'$, which results in the construction of the expanded graph $\mathcal{G}_k'(u_i')$ for u_i':

$$\begin{aligned} \mathcal{G}_k'(u_i') &= (\mathcal{U}_k', \mathcal{S}_k', \mathcal{E}_k, \mathcal{A}) \\ \mathcal{U}_k'(u_i') &= \{u' \in \mathcal{U}' \mid \text{dist}(u_i', u') \le K+1\} \\ \mathcal{S}_k'(u_i') &= \{s' \in \mathcal{S}' \mid \text{dist}(u_i', s') \le K+1\} \\ \mathcal{E}_k(u_i') &= \{e_{u',s'} \in \mathcal{E} \mid u' \in \mathcal{U}_k' \wedge s' \in \mathcal{S}_k'\} \end{aligned} \tag{10}$$

Among them, $\text{dist}(*,*)$ calculates the length of the shortest path between two nodes in graph $\mathcal{G}'$. $\mathcal{U}_k'(u_i')$, $\mathcal{S}_k'(u_i')$, $\mathcal{E}_k(u_i')$, and $\mathcal{A}$ respectively represent the set of encrypted user nodes,

Algorithm 1: Privacy-Preserving User-Service Graph Expansion.

Input:
- N: The number of users;
- $\{u_i\}_{i=1}^N$: The set of all users;
- $\{s_j\}$: The set of services related to each user;
- key: The encryption key for AES encryption;

Output: The set of expanded high-order graphs $\{\mathcal{G}_k(u_i)\}_{i=1}^N$ for each user u_i;

I. User Side Upload:
foreach *user* u_i *in* $\{1, \cdots, N\}$ **do**
  Get global key; Generate anonymous vector V_{u_i};
  $u_i' \leftarrow \text{H}_{\text{SHA}}(u_i)$;
  $V_{u_i}' \leftarrow \text{Enc}_{\text{AES}}(key, V_{u_i})$;
  foreach s_j *in services of* u_i **do**
    $s_j' \leftarrow \text{Enc}_{\text{AES}}(key, s_j)$;
  end
  Build ciphertext dataset $\mathcal{D}_i'$ and upload to server;
end
II. Server Side Aggregation:
Merge the dataset $\mathcal{D}'$ and build global graph $\mathcal{G}'$;
foreach *user* u_i' *in* $\{1, \cdots, N\}$ **do**
  $\mathcal{G}_k'(u_i') \leftarrow$Expand graph from u_i' in k-$step$;
  Distributes extended graph $\mathcal{G}_k'(u_i')$ to u_i';
end
III. User Side Download:
foreach *user* u_i *in* $\{1, \cdots, N\}$ **do**
  $s_m', V_{u_n}' \leftarrow$ Analyze $\mathcal{G}_k'(u_i')$;
  $s_m \leftarrow \text{Dec}_{\text{AES}}(key, s_m')$;
  $V_{u_n} \leftarrow \text{Dec}_{\text{AES}}(key, V_{u_n}')$;
  Expand to local high-order graph $\mathcal{G}_k(u_i)$.
end

encrypted service nodes, edge set, and attribute function in the expanded graph $\mathcal{G}'_k(u'_i)$.

Upon receiving the server-distributed expanded graph $\mathcal{G}'_k(u'_i)$, user u_i decrypts it for a usable model training graph. If the graph includes a neighboring user u'_n who utilized service s'_m, decryption of s'_m and V'_{u_n} proceeds via AES with the global key.

$$s_m = \text{Dec}_{\text{AES}}(key, s'_m)$$
$$V_{u_n} = \text{Dec}_{\text{AES}}(key, V'_{u_n}) \tag{11}$$

Given that u'_n is encrypted using the SHA-512 hashing algorithm, which is non-invertible, the user-side expanded graph can be represented as:

$$\mathcal{G}_k(u_i) = (\mathcal{U}'_k(u_i), \mathcal{S}_k(u_i), \mathcal{E}_k(u_i), \mathcal{A}) \tag{12}$$

Similarly, if we specify a target service s_j, after the above process, we can obtain an expanded graph $\mathcal{G}_k(s_j)$ centered on the service s_j.

$$\mathcal{G}_k(s_j) = (\mathcal{U}'_k(s_j), \mathcal{S}_k(s_j), \mathcal{E}_k(s_j), \mathcal{A}) \tag{13}$$

Since u'_n is anonymized and cannot be decrypted in reverse, and the corresponding vertex representation V_{u_n} contains no information that could directly identify the neighboring user, this design ensures that the privacy of neighboring nodes is effectively protected, even during the data expansion process.

4) Time Complexity Analysis: From the perspective of a single user, we analyze the complexity of each part involved in Algorithm 1 as below.

User Side Upload: For a single user, operations such as obtaining the global key and generating an anonymous vector have a time complexity of approximately $O(1)$. The AES encryption of m services and the anonymous vector, with each encryption being $O(1)$, results in the time complexity of $O(m)$.

Server Side Aggregation: The server's aggregation operation comprises two main steps. First, constructing a global graph from N users each connected to an average of m services, with a complexity of $O(Nm)$. Second, graph expansion on the user-service bipartite graph. Given that each user invokes m services on average, and each service is invoked by n users on average, with N total users and k expansion steps, the complexity after k steps is approximately $O(Nm^{1+k/2}n^{k/2})$.

User Side Download: The expanded graph received by the user contains $m^{1+k/2} + n^{k/2}$ nodes. Decrypting the service information and neighboring users' anonymous vectors, with each decryption being $O(1)$, results in the time complexity of $O(m^{1+k/2} + n^{k/2})$.

Overall Complexity: Combining the three parts, the total time complexity of the Privacy-Preserving User-Service graph expansion algorithm is $O(m + Nm + Nm^{1+k/2}n^{k/2} + m^{1+k/2} + n^{k/2})$. For the highest complexity consumption $O(Nm^{1+k/2}n^{k/2})$, it could be still very efficient while performing user-service invocation graph expansion, since k is normally set as 2 or 3 and the number of users n and services m are also strictly constrained with relatively small settings in federate learning scenarios. As a result, the proposed algorithm of expanding the graph user-service multi-order invocations ensures the highly and efficiently computational loads, preventing the overall complexity from becoming too large and maintaining high efficiency for real application demands.

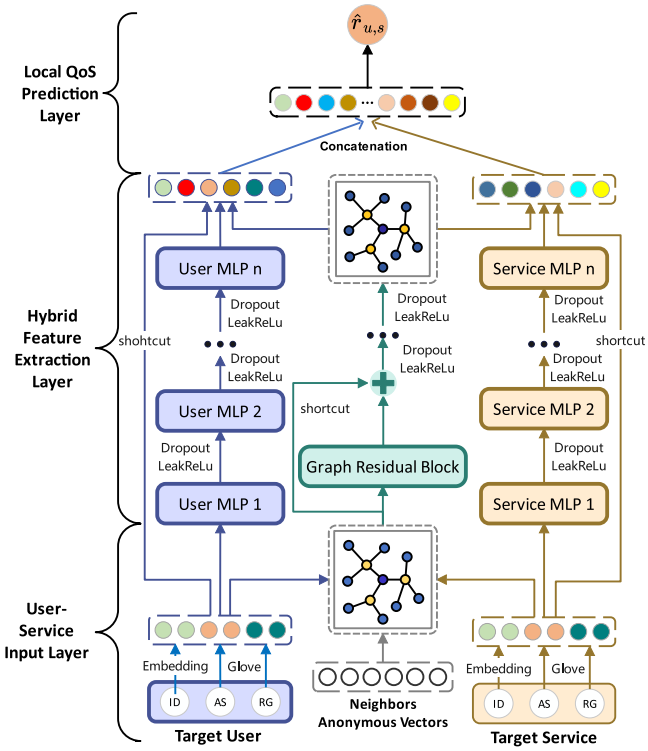


Fig. 3. Multi-layer hybrid feature extraction for users and services.

B. Hybrid Feature Extraction

Fig. 3 outlines multi-layer hybrid feature extraction network, engineered to distill profound feature representations of users and services. Starting with the interactive data pertaining to users and services, it embeds the initial feature representation and then extract hybrid features of users and services by multi-layer perceptron and graph residual learning, which are fed to predict locally missing QoS values.

1) User and Service Embedding: To enhance QoS prediction accuracy, we incorporate user and service identifiers, along with their location information (AS and RG), and renumber these identifiers before mapping them to dense vector representations using the neural network's embedding layer.

$$ID_u = M_{\text{emb}}(u_{ID}) \tag{14}$$

For AS and RG, since they may contain multiple users/services, in order to further explore the associated information, we utilize Glove word vectors for processing.

$$AS_u = \frac{1}{n}\sum_{i=0}^{n} \text{Glove}(w_{i,AS})$$
$$RG_u = \frac{1}{m}\sum_{i=0}^{m} \text{Glove}(w_{i,RG}) \tag{15}$$

Where $w_{i,AS}$ and $w_{i,RG}$ represent AS and RG name words, with n and m being their word counts, and Glove$(*)$ converts words into vectors. AS_u and RG_u are the user embeddings for AS and RG. Then, we combine the vectors of ID, AS, and RG to obtain the embedding feature vectors of users and services. It is represented by the following formula where Φ is the feature

concatenation operation.

$$E_u = \Phi\left(ID_u, AS_u, RG_u\right) = [ID_u, AS_u, RG_u]$$
$$E_s = \Phi\left(ID_s, AS_s, RG_s\right) = [ID_s, AS_s, RG_s] \quad (16)$$

In the graph expansion process, we prioritize user privacy by ensuring that only locally used feature vectors are kept private and never shared with neighboring users. Instead, anonymous vectors are shared, preventing the leakage of sensitive information and reducing privacy risks. These anonymous vectors enable continuous analysis and user representation learning, supporting hybrid feature extraction and QoS prediction while maintaining strict privacy protection.

After graph expansion, the locally expanded graph $\mathcal{G}_k = (U'_k, S_k, E_k, A)$ obtained by the user u_i can be combined with the embedding vectors to construct an initial graph representation that can be used for representation learning. Specifically, it is only necessary to update the attribute function A as in (17), where u_i is the current user, s_j is the service included in the extended graph, and u_n is the neighbor user of the current user in the graph.

$$A = \begin{cases} E_u \to node & \text{if } node = u_i \\ E_s \to node & \text{if } node = s_j \\ V'_{u_n} \to node & \text{if } node = u'_n \end{cases} \quad (17)$$

2) Semantic Feature Extraction: Our research design a deep feature extraction architecture to probe the nonlinear dynamics between users and services, using their embedding representations as input. To reveal intricate interaction features, we applied MLP to process these embeddings.

We feed the embedding vectors of users and services into two separate MLPs. In each MLP, every layer comprises neurons that apply a linear transformation to the input data and then apply the LeakyReLU activation function. The output of each layer not only provides a high-order representation of the input data but also serves as input for the next layer, allowing the model to learn complex patterns and deep features within the data. Through this layering approach, we can gradually abstract and refine deep high-dimensional features of users and services, which may relate to user preferences, service quality, or potential associations between the two. This process can be represented as:

$$M_{aff}(X, \sigma) = \sigma(W^T X + b)$$
$$S_u = M_{aff}(\cdots(M_{aff}(E_u, \sigma)\cdots), \sigma)$$
$$S_s = M_{aff}(\cdots(M_{aff}(E_s, \sigma)\cdots), \sigma) \quad (18)$$

M_{aff} represents the affine layer, σ represents the LeakyReLU activation function, E_u and E_s are the embedding vectors of users and services, and S_u and S_s are the semantic features vectors of users and services extracted by the MLPs.

LeakyReLU activation function is applied in the hybrid feature extraction. Different from the traditional ReLU that suffers from vanishing gradients when the input is less than 0 and thus hampers model training, it ensures the continuous flow of gradients during backpropagation by having a small negative slope for negative inputs, significantly accelerating the model's convergence. Meanwhile, when dealing with large-scale data and complex models in PE-FGL, it reduces resource consumption and speeds up training and feature extraction with efficient computation. Moreover, its nonlinear property enables capturing the complex relationships between user-service features effectively, enhancing semantic feature extraction to improve the accuracy of QoS prediction.

3) High-Order Interaction Feature Extraction: The user-service invocation graph $\mathcal{G}$ effectively delineates user interaction patterns. Upon a user invoking a service, $\mathcal{G}$ adds an edge to represent this user-service relation, facilitating quick identification of similar neighboring user behaviors. Leveraging these similarities, we introduce a LightGCN-based [34] approach for efficient feature propagation and aggregation within $\mathcal{G}$'s topology.

To more effectively capture the intricate high-order interactions between users and services, we have integrated residual learning, as proposed by He et al. [35], with nonlinear activation functions into the LightGCN, thus constructing Graph Residual Blocks (GRB). The improved graph convolution operation is defined as follows:

$$e_u^{(k)} = \sigma\left(e_u^{(k-1)} + \sum_{s\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_s|}} e_s^{(k-1)}\right)$$
$$e_s^{(k)} = \sigma\left(e_s^{(k-1)} + \sum_{u\in\mathcal{N}_s} \frac{1}{\sqrt{|\mathcal{N}_s|}\sqrt{|\mathcal{N}_u|}} e_u^{(k-1)}\right) \quad (19)$$

Where $e_u^{(k)}$ and $e_s^{(k)}$ represent the k-layer embeddings of user and service nodes, with $\mathcal{N}_u$ and $\mathcal{N}_s$ as their respective neighboring node sets and σ represents the LeakyReLU activation function.

Through iterative message propagation, the output of each layer is added to the output of the preceding layer as input for the next layer. This model can capture complex interactions between user nodes and service nodes, and further extract high-order latent interaction features G_u and G_s. These high-order features not only contain personalized preferences of users but also reflect potential associations among services, thereby providing richer and more accurate information support for user behavior prediction and service recommendation.

4) Hybrid Feature-Based Local QoS Prediction: The extracted semantic features and high-order interaction features are concatenated together for users and services. We then synthesized the user's initial embedding E_u, semantic feature S_u, and high-order interaction feature G_u into an integrated representation X_u. For services, we analogously derived their final representation X_s. Concatenating X_u and X_s, we input this composite vector into a neural network to predict the QoS value. The process is encapsulated by:

$$X_u = \Phi\left(E_u, S_u, G_u\right) = \left[E_u\ S_u\ G_u\right]$$
$$X_s = \Phi\left(E_s, S_s, G_s\right) = \left[E_s\ S_s\ G_s\right]$$
$$\hat{r}_{u,s} = \sigma\left(W^T\left[X_u\ X_s\right] + b\right) \quad (20)$$

Here, σ is the nonlinear activation function, W represents the weight matrix, b is the bias vector, and $\hat{r}_{u,s}$ is the predicted QoS value.

C. Secure Federated Parameters Aggregation

1) Secured Segmentation and Aggregation of Federated Parameters: Federated Learning (FL) keeps users' historical data

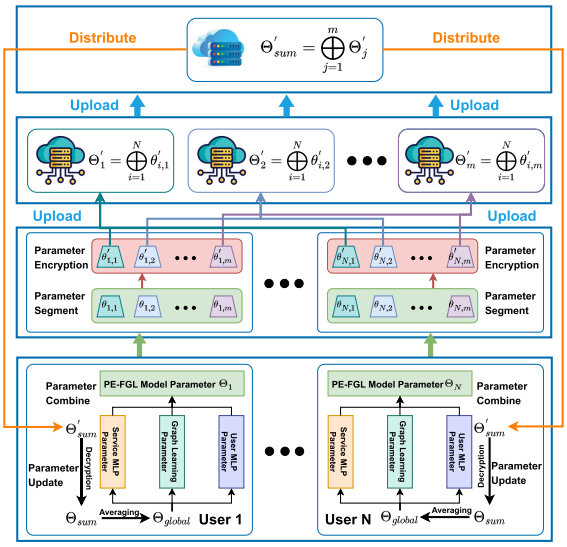


Fig. 4. The process of secure federated parameters aggregation.

on their personal devices, ensuring privacy and reducing risks associated with centralized data storage. However, research investigations such as [36] have shown that parameter sharing in FL can lead to privacy leaks, as attackers may infer sensitive information from shared model parameters. To mitigate the potential risk, we propose a multi-layer secure federated parameters segmentation and aggregation strategy, as shown in Fig. 4, which enhances privacy protection during the collaborative learning of federated parameters.

The secure parameter aggregation process is shown in Algorithm 2. First, users partition and encrypt local parameters and upload them. Then, intermediate computation units perform homomorphic addition on parameters of the same partition. Next, the federated aggregation server aggregates all partitions. Finally, global parameters are distributed to users for local model updates. We further formalize these key steps, and the detailed process is as follows.

Specifically, we devised a parameter segmentation scheme enhanced by homomorphic encryption [17], which allows computations on encrypted data to yield outcomes equivalent to those on the original data. The local model parameters Θ_u of client u can be conceptualized as a vector in a high-dimensional space, that is, $\Theta_u \in \mathbb{R}^n$. Each element within the vector Θ_u corresponds to a specific parameter in the model. To achieve weighted aggregation, it is first necessary to multiply Θ_u by the scalar T_u, thereby obtaining the weighted parameters $\Theta_{u,T}$.

$$\Theta_{u,T} = T_u \Theta_u \tag{21}$$

For parameters segmentation and encryption, we partition $\Theta_{u,T}$ into m segments on an element-wise basis, with $\theta_{u,j} \in \mathbb{R}^n$ denoting the parameters of the j-th segment. Consequently, we have:

$$\Theta_{u,T} = \sum_{i=1}^{m} \theta_{u,i} = \theta_{u,1} + \theta_{u,2} + \cdots + \theta_{u,m} \tag{22}$$

The aforementioned expression indicates that $\Theta_{u,T}$ can be reconstructed by aggregating the m parameter components $\theta_{u,j}$ at the element level. Subsequently, each component is subjected

Algorithm 2: Secure Federated Parameters Segmentation and Aggregation.

Input:
- N: The number of users;
- $\{\Theta_i\}_{i=1}^{N}$: The local parameters of each user u_i;
- pk: The public key of Homomorphic Encryption;
- sk: The private key of Homomorphic Encryption;
- m: The number of parameter segmentation segments;

Output: Each user's updated local models relying on the aggregated global parameter Θ_{global};

I. Local Parameter Segmentation and Encryption:
foreach *user* u_i *in* $\{1, \cdots, N\}$ **do**
  Divide Θ_i into m segments, $\theta_{i,j} (j = 1, \cdots, m)$;
  foreach *segment* $\theta_{i,j}$ **do**
    $\theta'_{i,j} \leftarrow \text{Enc}_{\text{HE}}(pk, \theta_{i,j})$;
    Upload $\theta'_{i,j}$ to the corresponding CU_j;
  end
end
II. Intermediate Computing Unit Operation:
foreach *intermediate computing unit* CU_j **do**
  $\Theta'_j \leftarrow$Aggregate $\theta'_{i,j}$ by homomorphic addition;
  Upload Θ'_j to federated aggregation server;
end
III. Federated Aggregation Server Operation:
$\Theta'_{sum} \leftarrow$Aggregate Θ'_j by homomorphic addition;
IV. Distribution and Local Model Update:
foreach *user* u_i *in* $\{1, \cdots, N\}$ **do**
  $\Theta_{global} \leftarrow \frac{1}{N} \text{Dec}_{\text{HE}}(sk, \Theta'_{sum})$;
  Update the local model with Θ_{global}.
end

to homomorphic encryption as follows:

$$\theta'_{u,j} = \text{Enc}_{\text{HE}}(pk, \theta_{u,j}) \tag{23}$$

Where $\text{Enc}_{\text{HE}}(pk, *)$ signifies encryption with the public key pk, the pair pk and sk are the public and private keys distributed to all users by the key center.

For parameters segmentation and encryption, we establish m intermediate computing units. Clients transmit their j-th segment's ciphertext to the respective node, which executes additive operations on the ciphertexts directly. Given N clients, the aggregate ciphertext for the j-th segment, Θ'_j, is:

$$\Theta'_j = \bigoplus_{i=1}^{N} \theta'_{i,j} = \theta'_{1,j} \oplus \theta'_{2,j} \oplus \ldots \oplus \theta'_{N,j} \tag{24}$$

Intermediate computing units send the ciphertext results Θ'^{t}_{j}, representing the global parameter sum for the j-th segment, to the federated aggregation server. It aggregates them to calculate the overall ciphertext global parameter sum, Θ'_{sum}.

$$\Theta'_{sum} = \bigoplus_{i=1}^{m} \Theta'_i = \Theta'_1 \oplus \Theta'_2 \oplus \cdots \oplus \Theta'_m \tag{25}$$

The aggregation server distributes Θ'_{sum} to all clients, where they decrypt it using the private key sk and subsequently divide by the global sample count T to obtain the global model

parameters Θ_{global}.

$$\Theta_{global} = \frac{1}{T}\mathrm{Dec}_{\mathrm{HE}}(sk, \Theta'_{sum}) \tag{26}$$

Where $\mathrm{Dec}_{\mathrm{HE}}(sk, *)$ indicates the homomorphic decryption operation with the private key sk.

In the multi-layer secure federated parameters aggregation, both the aggregation server and intermediate calculation nodes process only encrypted parameters, ensuring robust security against potential server data leakage. Moreover, by employing parameter segmentation, we prevent attackers from reconstructing individual client parameters, even in case of encryption compromise. That significantly enhances privacy protection in federated learning, particularly for Non-IID data.

2) Time Complexity Analysis: Algorithm 2 comprises four parts: parameter segmentation and encryption, intermediate computing unit operation, federated aggregation server operation, and distribution and local model update. The complexity of each part and the overall algorithm is analyzed as below.

Parameters Segmentation and Encryption: For user u_i, parameters are divided into m segments, with a time complexity of $O(m)$. Each segment is encrypted and uploaded, both operations being $O(1)$.

Intermediate Computing Unit Operation: Each intermediate computing unit CU_j performs homomorphic addition on segments from N users. Assuming homomorphic addition is $O(1)$, the complexity for each CU_j is $O(N)$.

Federated Aggregation Server Operation: The federated aggregation server gathers results from m intermediate units using homomorphic addition, assumed to be $O(1)$. Thus, the complexity is $O(m)$.

Distribution and Local Model Update: User u_i retrieves and decrypts the global parameter, which is $O(1)$, and updates the local model.

Overall Complexity: The total time complexity of the Algorithm 2 is $O(N + m)$, derived from summing the complexities of the above four individual parts: $O(m + N + m + 1)$, which is linear with the number of users and services for secure federated parameters segmentation and aggregation.

VI. Experiments

A. Experimental Setup and Datasets

All our experiments were conducted on a workstation equipped with an NVIDIA Geforce RTX 4090 GPU and an Intel Xeon Silver 4210 R CPU @ 2.40GHZ. The components of PE-FGL were implemented using Python 3.7.15 and PyTorch 1.13.1.

To comprehensively evaluate the performance of the proposed PE-FGL for secure QoS prediction, we conducted extensive experiments using the widely recognized dataset WS-DREAM [18]. It is a large-scale real-world dataset for validating the effectiveness of QoS prediction approaches and comprises 1,974,675 historical QoS invocation records, involving 339 users and 5,825 web services. These records include two critical QoS criteria: Response Time (RT) and Throughput (TP). In addition to QoS records, the dataset encompasses contextual information like identifiers and geographic locations for users and services. These contextual informations serves as input for the HFE.

In practical SOA applications, users invoke few services, leading to a sparse QoS matrix. To simulate this and evaluate PE-FGL, we created QoS datasets with 5%, 10%, 15%, and 20% known QoS records for training, reserving the rest for testing. In a federated setting, each client represents an independent user and only possesses QoS invocation records between that user and corresponding invoked web services. It simulates the real-world data distribution, which is naturally distributed across multiple clients (users), with each user's QoS invocations being private.

B. Competing Methods and Evaluation Metrics

To assess the performance of PE-FGL, we compared it with nine widely-used competing baselines, including two centralized memory-based, two centralized model-based, and five federated learning based approaches. They are described as follows.

- *UIPCC [7]:* It is a memory-based collaborative filtering approach that combines user (UPCC) and service (IPCC) similarities with weighted refinement for better QoS prediction.
- *LACF [10]:* It is a location-aware collaborative filtering approach that improves QoS prediction by incorporating geographical proximities of users and services.
- *NMF [5]:* It is a QoS prediction approach that decomposes user-service matrix into latent non-negative matrices, leveraging similar users' information for enhanced prediction in non-negative data scenarios.
- *PMF [6]:* It is a model-based approach that employs probabilistic matrix factorization with Gaussian-distributed features to address data sparsity and improve QoS prediction accuracy.
- *EFMF [14]:* It is a federated matrix factorization approach that enhances QoS prediction via local updates and central aggregation, preserving privacy without sharing user-service QoS records.
- *FedNCF:* It is a federated learning approach based on NCF [24] that combines MLP and matrix factorization, utilizing FedAvg [13] to aggregate client models for distributed QoS prediction.
- *FedLDCF:* It is a federated learning approach based on LDCF [25] that incorporates location-aware feature correction and MLP for privacy-preserving QoS prediction with nonlinear user-service relationship modeling.
- *FedFSNet:* It is a federated learning approach based on FSNet [27] that employs feature distribution smoothing to fit Gaussian features and an improved W-Huber loss to address noise and label imbalance issues.
- *FHC-DQP [15]:* It is a privacy-preserving approach combines hierarchical clustering and distributed training with fine-grained partitioning and context-aware learning for distributed QoS prediction.

In our experiments, we employed two widely used evaluation metrics, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), to measure the deviation between ground-truth QoS values and the predicted ones. MAE and RMSE are defined as follows:

$$MAE = \frac{\sum_{u,s} |r_{u,s} - \hat{r}_{u,s}|}{N} \tag{27}$$

$$RMSE = \sqrt{\frac{\sum_{u,s} (r_{u,s} - \hat{r}_{u,s})^2}{N}} \tag{28}$$

TABLE I
PERFORMANCE COMPARISONS OF QOS PREDICTION AMONG COMPETING METHODS ON RT DATASET

Methods	Density=5%		Density=10%		Density=15%		Density=20%	
	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
EFMF	0.622	1.526	0.528	1.326	0.488	1.237	0.470	1.200
FedNCF	0.492	1.478	0.431	1.374	0.417	1.361	0.403	1.326
FedLDCF	0.491	1.433	0.451	1.356	0.433	1.364	0.410	1.387
FedFSNet	0.405	1.389	0.371	1.297	0.355	1.276	0.341	1.236
FHC-DQP	0.510	1.400	0.434	1.316	0.395	1.236	0.338	1.205
PE-FGL	**0.366**	**1.304**	**0.336**	**1.254**	**0.321**	1.232	**0.306**	1.181
Gains	9.63%	5.98%	9.43%	2.26%	9.58%	-2.50%	9.47%	-1.55%

TABLE II
PERFORMANCE COMPARISONS OF QOS PREDICTION AMONG COMPETING METHODS ON TP DATASET

Methods	Density=5%		Density=10%		Density=15%		Density=20%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UIPCC	26.75	60.79	22.37	54.45	20.21	50.70	18.92	48.29
LACF	22.97	58.78	19.44	52.92	17.58	49.56	16.45	47.41
NMF	21.88	60.53	16.57	49.82	15.85	46.80	14.59	42.44
PMF	19.07	57.88	15.99	48.08	15.08	46.05	13.92	42.16
EFMF	26.92	68.86	19.62	53.37	16.91	48.69	15.34	43.87
FedNCF	18.52	57.25	16.02	51.40	15.24	50.09	15.17	49.20
FedLDCF	16.48	54.07	14.77	48.43	13.20	45.83	12.96	45.23
FedFSNet	15.06	50.87	13.25	45.92	12.65	43.29	11.81	41.64
FHC-DQP	17.27	51.52	14.34	46.61	13.50	44.92	12.64	42.20
PE-FGL	**13.06**	**44.51**	**11.26**	**39.21**	**10.15**	**37.20**	**9.85**	**35.05**
Gains	13.28%	12.50%	15.02%	14.61%	19.76%	14.07%	16.60%	15.83%

where u and s represent the given target user and service, respectively. $r_{u,s}$ and $\hat{r}_{u,s}$ are the original and predicted QoS values, respectively. N is the number of test samples for predicting QoS values.

C. Experiment Results and Analyses

Tables I and II show comparative performance results of PE-FGL against centralized and federated baselines for QoS prediction on RT and TP datasets, where lower MAE and RMSE values indicate superior predictive accuracy. Bold values denote the best results, while gray highlights indicate second-best QoS prediction values. The empirical analysis demonstrates consistent improvement in QoS predictive accuracy as matrix density increases from 5% to 20%, with an interval step 5%.

In evaluating centralized QoS prediction models, UIPCC demonstrated inferior MAE and RMSE indices, chiefly due to its dependence on sparse QoS records, making it vulnerable to data deficiency. Conversely, LACF improved predictive accuracy by incorporating geographic information as a supplementary heuristic for neighborhood selection. Additionally, matrix factorization methods like NMF and PMF surpassed traditional memory-based models by uncovering more profound user-service correlations and interaction patterns, offering enhanced robustness and precision in QoS prediction.

In federated QoS prediction, distributed matrix factorization-based EFMF exhibits inferior performance compared to centralized approaches like NMF and PMF, corroborating existing findings [37] that centralized learning better captures intrinsic data patterns through global dataset access. Subsequent federated models have received progressive improvements. Specifically, FedNCF leverages MLP to model nonlinear user-service relationships, addressing data sparsity challenges. FedLDCF incorporates geographic contextual information to enhance feature representation. FHC-DQP employs federated hierarchical clustering for refined user segmentation. FedFSNet takes feature distribution smoothing and W-Huber loss into account to mitigate noise and label imbalance, culminating in superior QoS prediction performance.

Unlike the above competing approaches, PE-FGL integrates a graph expansion mechanism, context-aware deep neural network, and graph residual learning to capture complex nonlinear user-service interactions. From the results, it demonstrates superior performance, achieving MAE reductions of approximately 9% and 13% on RT and TP datasets, respectively. While PE-FGL outperforms competitors in terms of RMSE on TP dataset by 12%, it slightly falls short of centralized NMF and PMF at 15% and 20% matrix densities on RT dataset. This disparity likely stems from centralized models' enhanced capabilities of collaborative pattern detection in dense QoS matrices, highlighting

TABLE III
ABLATION STUDY RESULTS ON RT DATASET

Methods	Density=5%		Density=10%		Density=15%		Density=20%	
	MAE/Impro	RMSE/Impro	MAE/Impro	RMSE/Impro	MAE/Impro	RMSE/Impro	MAE/Impro	RMSE/Impro
PE-FGL-SG	0.527/30.55%	1.689/22.79%	0.498/32.53%	1.625/22.83%	0.466/31.12%	1.599/22.95%	0.452/32.30%	1.507/21.63%
PE-FGL-G	0.456/19.74%	1.578/17.36%	0.439/23.46%	1.519/17.45%	0.418/23.21%	1.486/17.09%	0.390/21.54%	1.389/14.97%
PE-FGL-S	0.435/15.86%	1.457/10.50%	0.417/19.42%	1.396/10.17%	0.375/14.40%	1.354/9.01%	0.347/11.82%	1.316/10.26%
PE-FGL	0.366	1.304	0.336	1.254	0.321	1.232	0.306	1.181

TABLE IV
ABLATION STUDY RESULTS ON TP DATASET

Methods	Density=5%		Density=10%		Density=15%		Density=20%	
	MAE/Impro	RMSE/Impro	MAE/Impro	RMSE/Impro	MAE/Impro	RMSE/Impro	MAE/Impro	RMSE/Impro
PE-FGL-SG	19.24/32.12%	70.36/36.74%	17.35/35.10%	63.59/38.34%	16.65/39.04%	60.21/38.22%	16.05/38.63%	58.55/40.14%
PE-FGL-S	17.89/27.00%	63.46/29.86%	15.72/28.37%	59.16/33.72%	15.14/32.96%	53.98/31.09%	14.59/32.49%	52.84/33.67%
PE-FGL-G	15.04/13.16%	51.91/14.26%	13.89/18.93%	47.03/16.63%	12.97/21.74%	44.74/16.85%	12.61/21.89%	44.09/20.50%
PE-FGL	13.06	44.51	11.26	39.21	10.15	37.20	9.85	35.05

an inherent disadvantage of federated approaches in predicting vacant QoS with dense user-service invocations.

In our research, we developed a hybrid feature extraction network that contribute significantly to QoS prediction. To rigorously dissect the significance of each constituent in HFE, we conducted a suite of ablation studies. HFE comprises two principal elements: Semantic Feature Extraction (SFE) component and Graph Residual Learning (GRL) component. Specifically, we constituted four experimental groups to evaluate the influence:

- *PE-FGL-SG group:* As a baseline, this group only includes the input and prediction layers of the PE-FGL method without integrating any parts of the HFE network.
- *PE-FGL-S group:* Only utilizes GNNs to explore the high-order potential latent interaction features between users and services, thus examining the contribution of the GRL module independently.
- *PE-FGL-G group:* Focus on exploiting deep learning techniques for semantic feature extraction of user-services to evaluate the effect of SFE.
- *PE-FGL group:* Represents the complete PE-FGL method, combining both the SFE and GRL to provide a comprehensive feature extraction network.

We conducted comprehensive ablation studies of PE-FGL utilizing datasets with varying matrix densities (5%, 10%, 15%, and 20%), with the remaining data allocated for testing. The experimental results, presented in Tables III and IV, demonstrate the superior performance of the PE-FGL model across both RT and TP datasets. It is observed that, PE-FGL exhibits significant improvements over the its variant PEFGL-SG, achieving 20%-32% enhancement on RT and 32%-40% on TP datasets, respectively.

Component-wise analysis reveals dataset-specific performance variations. On RT, PE-FGL demonstrated improvements of $13\%-23\%$ over PE-FGL-S and $8\%-19\%$ over PE-FGL-G. Similarly for TP, gains of $27\%-33\%$ and $13\%-21\%$ were observed compared to PE-FGL-S and PE-FGL-G, respectively. Notably, PE-FGL-G consistently outperformed PE-FGL-S across both datasets.

The outstanding QoS prediction performance of PE-FGL stems from the synergistic integration of SFE and GRL within the HFE framework. Here, SFE facilitates deep mining of nonlinear relationships and extraction of rich semantic features from user-service interactions. These semantic embeddings further enhance GRL's capacity to focus on key high-order interaction patterns, while minimizing irrelevant information interference.

The ablation results indicate that the absence of SFE (PE-FGL-S) or GRL (PE-FGL-G) significantly impacts model performance. Without SFE, the model exhibits increased susceptibility to noise during high-order feature extraction. Conversely, the absence of GRL limits the utilization of semantic guidance for interaction pattern identification. These findings emphasize that the integration of both modules is crucial for improving QoS prediction accuracy in practical applications.

D. Performance Impact of Parameters

1) Impact of Clients Selection Rate: In practical deployment of PE-FGL, due to the large number and variability of client environments, only a subset of clients is randomly selected to participate in each training iteration. The client selection rate F, defined as the percentage of clients selected for training in a given round, significantly influences QoS prediction performance. To investigate the impact of F on model performance, experiments were conducted using RT and TP datasets with matrix densities of 5%, 10%, 15%, and 20%, and six selection rates: 3%, 5%, 10%, 30%, 50%, and 100%. The number of clients participating in each training round is the product of the total number of clients and F; for example, $F = 100\%$ means all clients participate in that round.

Fig. 5 illustrates the variations in QoS prediction performance across different selection rates F. The model achieves the best MAE and RMSE at F=10%, likely due to a balance between data diversity and noise. That is, lower F values may reduce data diversity, hindering the generalization of PE-FGL. Oppositely, higher F values can introduce excessive noise, leading to model overfitting and reducing the accuracy of QoS prediction. At

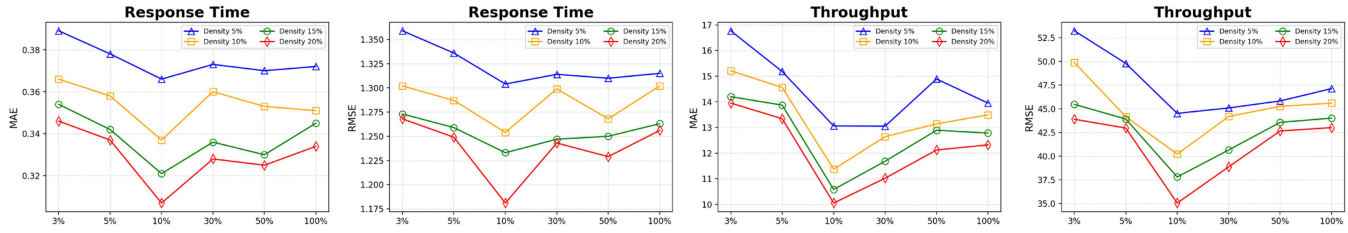


Fig. 5. Performance impact of clients selection rate.

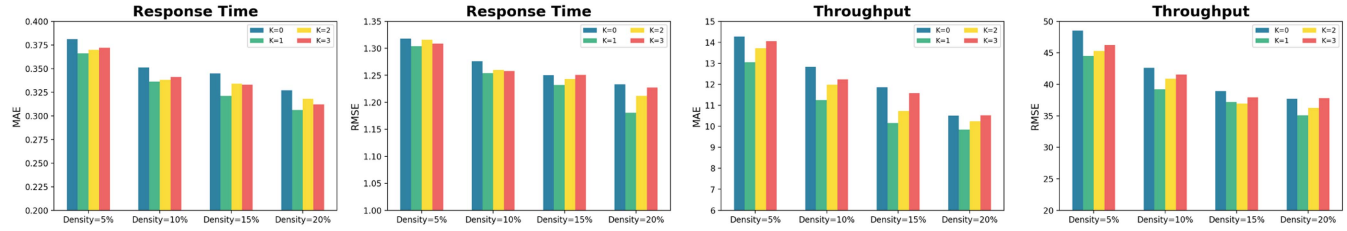


Fig. 6. Performance impact of graph expansion step.

F=10%, the model maintains sufficient diversity to generalize well without being overwhelmed by noise.

2) Impact of Graph Expansion Step: In the graph expansion module, local users gather neighboring users' invocation records to enhance their local prediction model training, improving QoS prediction accuracy. A user's service invocation graph is represented as a first-order subgraph centered on them in the global graph. The graph expansion step k is defined as the distance from a node in the global graph to the corresponding first-order subgraph. Experiments are conducted on RT and TP datasets with matrix densities of 5%, 10%, 15%, and 20%, using k values of 0, 1, 2, and 3, where k=0 indicates first-order subgraph without any graph expansion.

Fig. 6 reveals a clear correlation between prediction accuracy (MAE/RMSE) and the expansion factor k, with performance peaking at $k = 1$ through moderate graph expansion and declining progressively as k exceeds this threshold. The poorest results at $k = 0$ underscore the necessity of graph expansion, while over-expansion (e.g., $k = 4$) introduces excessive privacy-preserving noise from anonymized vectors, degrading accuracy below lower k values (e.g., $k = 3$). These findings highlight the critical trade-off between data integration depth and privacy-induced noise in maintaining optimal model performance.

VII. Discussion

A. The Practical Significance of PE-FGL

In real-world scenarios, users' QoS invocation records are naturally distributed across local devices, with each user's data being small in scale and highly personalized. To train a widely applicable and accurate QoS prediction model while safeguarding user privacy, we propose the privacy-conscious PE-FGL prediction model, which struggles to address key challenges for both improving QoS prediction accuracy and ensuring the guarantee of the privacy protection.

First, PE-FGL expands the user-service invocation graph using advanced privacy-preserving techniques. During the first-order graph encryption stage, user identities and services are anonymized and encrypted using secure hash algorithms [33] and AES encryption [16]. It ensures strict privacy protection during graph expansion, preventing data leakage during transmission and storage. The server assembles the global encrypted graph without decryption, further enhancing the privacy of users and services. In the adaptive graph distribution stage, the server adaptively constructs extended high-order invocation graphs, allowing users access only to necessary information while restricting access to feature vectors of adjacent nodes, thereby protecting tthe privacy of users and services and also improving the effectiveness of hybrid feature extraction for better QoS prediction accuracy.

Second, PE-FGL employs a secure federated parameters aggregation strategy based on homomorphic encryption [17]. Clients encrypt segmented local model parameters before uploading while both intermediate nodes and the server process only encrypted model parameters. It ensures the security of local model parameters during transmission and aggregation, preventing attackers from recovering the original parameters. Thus, PE-FGL further guarantees the privacy of model parameters during model training when multiple user clients collaboratively learn better local QoS prediction models.

From the above analyses on PE-FGL, existing QoS prediction approaches often fail to adequately protect user privacy. Traditional centralized approaches, such as NMF [5] and PMF [6], require collecting all users' QoS records, exposing sensitive information during similarity calculation and model training. Federated learning-based approaches like EFMF [14] and FHC-DQP [15] provide limited privacy protection by relying on simple encryption or plaintext parameter aggregation, which leaves shared parameters vulnerable to be attacked in real applications. Additionally, they overlook the high-order

topological relationships between users and services, reducing QoS prediction accuracy due to lack of training samples for local model parameters optimization. PE-FGL addresses these limitations by integrating advanced privacy-preserving techniques and user-service invocation graph expansion.

B. Optimizing Federated Parameter Aggregation

In federated QoS prediction, geographically distributed users face challenges when collaboratively learning QoS patterns. Network latency between distributed users not only hinders timely model updates but also affects the accuracy of QoS measurements. This leads to sparse user-service interaction data and degraded prediction performance. Moreover, the large volume of encrypted model parameters and user behavior data imposes substantial privacy protection overhead.

To address these issues, we first design multi-layer network architecture by introducing edge computing and deploying intermediate nodes near clients that can reduce transmission distances and network latency. For example, regional edge nodes can perform local QoS model parameters aggregation, minimizing communication with central servers. Then, adopting hierarchical parameter aggregation and decentralized collaboration can further enhance the efficiency of globally calculating those aggregated local model parameters. Specifically, regional nodes aggregate local parameters and generate compact results, and they collaborate via secure protocols to produce the final global model, reducing data volume and computational load for secure federated parameters aggregation.

VIII. Conclusion

In this paper, we propose a novel framework of Privacy-Enhanced Federated expanded Graph Learning(PE-FGL), which is dedicated to both protecting user privacy and improving the effectiveness of QoS prediction. Initially, we design a privacy-preserving graph expansion mechanism that is capable of augmenting each user's one-order local QoS invocations into an expanded user-service invocation graph, reflecting high-order invocation relationships. Based on the expanded graph, we then present hybrid feature extraction by leveraging deep learning and graph residual learning, capturing semantic features and high-order interaction features of users and services, respectively. Finally, we design a multi-granularity secure federated parameters aggregation, which takes local parameters segmentation encryption and multi-layer parameters aggregation for effectively secure QoS prediction. Experimental results on WSDREAM dataset demonstrate the superior QoS prediction performance, while ensuring its protection of user privacy.

In the future, we plan to explore more personalized training strategies with the aim of optimizing QoS prediction performance and further enhancing the capability of user privacy protection.

References

[1] W. Liang, Y. Li, and J. Xu, "QoS prediction and adversarial attack protection for distributed services under DLaaS," *IEEE Trans. Comput.*, vol. 73, no. 3, pp. 669–682, Mar. 2024.

[2] X. Wang, M. Xi, Y. Li, and X. Pan, "SEHGN: Semantic-enhanced heterogeneous graph network for web API recommendation," *IEEE Trans. Services Comput.*, vol. 17, no. 5, pp. 2836–2849, Sep./Oct. 2024.

[3] J. Wang, X. Zhang, Q. Wang, W. Zheng, and Y. Xiao, "QoS prediction method via multi-task learning for web service recommendation," in *Proc. IEEE Int. Conf. Web Serv.*, 2024, pp. 1353–1355.

[4] G. Li, A. Zhang, Q. Zhang, D. Wu, and C. Zhan, "Pearson correlation coefficient-based performance enhancement of broad learning system for stock price prediction," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 5, pp. 2413–2417, May 2022.

[5] J. Gan, T. Liu, L. Li, and J. Zhang, "Non-negative matrix factorization: A survey," *Comput. J.*, vol. 64, no. 7, pp. 1080–1092, 2021.

[6] J. Deng, X. Ran, Y. Wang, L. Y. Zhang, and J. Guo, "Probabilistic matrix factorization recommendation approach for integrating multiple information sources," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 53, no. 10, pp. 6220–6231, Oct. 2023.

[7] Z. Zheng, X. Li, M. Tang, F. Xie, and M. R. Lyu, "Web service QoS prediction via collaborative filtering: A survey," *IEEE Trans. Services Comput.*, vol. 15, no. 4, pp. 2455–2472, Jul./Aug. 2022.

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

[9] Q. Wang, C. Yan, and Wang, "Location-aware collaborative filtering and feature interaction learning for QoS prediction," in *Proc. IEEE Int. Conf. Web Serv.*, 2024, pp. 291–299.

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

[11] Y. Xia, D. Ding, Z. Chang, and F. Li, "Joint deep networks based multi-source feature learning for QoS prediction," *IEEE Trans. Services Comput.*, vol. 15, no. 4, pp. 2314–2327, Jul./Aug. 2022.

[12] Y. Zhang, X. Li, Q. Luo, Y. Wang, and Y. Shen, "Practical and efficient secure aggregation for privacy-preserving machine learning," in *Proc. Asia Conf. Artif. Intell. Mach. Learn. Robot.*, 2023, pp. 1:1–1:8.

[13] Y. Zhou, Q. Ye, and J. Lv, "Communication-efficient federated learning with compensated Overlap-FedAvg," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 1, pp. 192–205, Jan. 2022.

[14] Y. Zhang, P. Zhang, Y. Luo, and J. Luo, "Efficient and privacy-preserving federated QoS prediction for cloud services," in *Proc. IEEE Int. Conf. Web Serv.*, 2020, pp. 549–553.

[15] G. Zou et al., "FHC-DQP: Federated hierarchical clustering for distributed QoS prediction," *IEEE Trans. Services Comput.*, vol. 16, no. 6, pp. 4073–4086, Nov./Dec. 2023.

[16] J. Kaur, S. Lamba, and P. Saini, "Advanced encryption standard: Attacks and current research trends," in *Proc. Int. Conf. Adv. Comput. Innov. Technol. Eng.*, 2021, pp. 112–116.

[17] C. Marcolla, V. Sucasas, M. Manzano, R. Bassoli, and Fitzek, "Survey on fully homomorphic encryption, theory, and applications," in *Proc. IEEE*, vol. 110, no. 10, pp. 1572–1609, Oct. 2022.

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

[19] A. Akhtarshenas, M. A. Vahedifar, and Ayoobi, "Federated learning: A cutting-edge survey of the latest advancements and applications," *Comput. Commun.*, vol. 228, pp. 107964–107965, 2024.

[20] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 249–270, Jan. 2022.

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

[22] 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 Trans. Netw. Service Manag.*, vol. 13, no. 1, pp. 126–137, Mar. 2016.

[23] J. Xu, Z. Zheng, and M. R. Lyu, "Web service personalized quality of service prediction via reputation-based matrix factorization," *IEEE Trans. Rel.*, vol. 65, no. 1, pp. 28–37, Mar. 2016.

[24] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. Int. Conf. World Wide Web*, 2017, pp. 173–182.

[25] Y. Zhang, C. Yin, Q. Wu, Q. He, and H. Zhu, "Location-aware deep collaborative filtering for service recommendation," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 6, pp. 3796–3807, Jun. 2021.

[26] G. Zou et al., "NCRL: Neighborhood-based collaborative residual learning for adaptive QoS prediction," *IEEE Trans. Services Comput.*, vol. 16, no. 3, pp. 2030–2043, May/Jun. 2023.

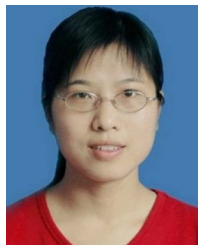
[27] T. Lu, X. Zhang, Z. Wang, and M. Yan, "A feature distribution smoothing network based on Gaussian distribution for QoS prediction," in *Proc. IEEE Int. Conf. Web Serv.*, 2023, pp. 687–694.

[28] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "A privacy-preserving QoS prediction framework for web service recommendation," in *Proc. IEEE Int. Conf. Web Serv.*, 2015, pp. 241–248.
[29] A. Liu et al., "Differential private collaborative web services QoS prediction," *World Wide Web*, vol. 22, no. 6, pp. 2697–2720, 2019.
[30] P. Zhang, H. Jin, H. Dong, W. Song, and A. Bouguettaya, "Privacy-preserving QoS forecasting in mobile edge environments," *IEEE Trans. Services Comput.*, vol. 15, no. 2, pp. 1103–1117, Mar./Apr. 2022.
[31] H. Jin, P. Zhang, and H. Dong, "Security-aware QoS forecasting in mobile edge computing based on federated learning," in *Proc. IEEE Int. Conf. Web Serv.*, 2020, pp. 302–309.
[32] J. Zhou et al., "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.
[33] P. J. F. Bemida, A. M. Sison, and R. P. Medina, "Modified SHA-512 algorithm for secured password hashing," in *Proc. Innov. Power Adv. Comput. Technol.*, 2021, pp. 1–9.
[34] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 639–648.
[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[36] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 691–706.
[37] K. Muhammad, Q. Wang, and O'Reilly-Morgan, "FedFast: Going beyond average for faster training of federated recommender systems," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 1234–1242.

Guobing Zou received the PhD degree in computer science from Tongji University, Shanghai, China, in 2012. He is a full professor and vice dean with the School of Computer Science, Shanghai University, China. He has worked as a visiting scholar with the Department of Computer Science and Engineering, 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 120 papers on premier international journals and conferences, including *IEEE Transactions on Services Computing*, *IEEE Transactions on Network and Service Management*, *ACM Transactions on Autonomous and Adaptive Systems*, IEEE ICWS, ICSOC, AAAI, *Information Sciences*, *Expert Systems with Applications*, *Knowledge-Based Systems*, etc.

Zhi Yan received the bachelor's degree in computer science and technology from Shanghai University, in 2022. He is currently working toward the master degree with the School of Computer Engineering and Science, Shanghai University, China. 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 received the master's degree in computer science and technology from Shanghai University, China, in 2021. He is currently working toward the PhD degree with the School of Computer Engineering and Science, Shanghai University, China. His primary areas of research encompass Quality of Service (QoS) prediction, graph neural networks, and natural language processing. Over the course of his academic career, he has contributed significantly to the field through his authorship and co-authorship of 17 scholarly papers. These papers have been published in esteemed international journals and presented at prestigious conferences, such as the *Knowledge-Based Systems*, *IEEE Transactions on Service Computing*, ICSOC, PPSN, etc.



Yanglan Gan received the PhD degree in computer science from Tongji University, Shanghai, China, in 2012. She is a full professor with the School of Computer Science and Technology, Donghua University, Shanghai, China. Her research interests include bioinformatics, service computing, and data mining. She has published more than 50 papers on premier international journals and conferences, including the *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 received the PhD degree from Northwestern Polytechnic University (NPU), China, in 1997. He is a full professor and dean with the School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai, China. He experienced a postdoctoral research with Zhejiang University from 1997 to 1999, China. He worked as a visiting professor with the University of Aizu from 2006 to 2007, Japan. He worked as a visiting scholar with 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 (Fellow, IEEE) 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 with 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 the *Artificial Intelligence*, *Journal of Artificial Intelligence Research*, *IEEE Transactions on Services Computing*, *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Industrial Informatics*, 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 of 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 AAAI and AAIA.