A Multi-source Information Graph-based Web Service Recommendation Framework for a Web Service Ecosystem

Zhixuan Jia[1], Yushun Fan[1,*], Jia Zhang[2], Xing Wu[1],
Chunyu Wei[1] and Ruyu Yan[1]

[1]*Beijing National Research Center for Information Science and Technology (BNRist), Department of Automation, Tsinghua University, Beijing, China*
[2]*Department of Computer Science, Southern Methodist University, TX, USA*
E-mail: jzx21@mails.tsinghua.edu.cn; fanyus@tsinghua.edu.cn; jiazhang@smu.edu; wuxing17@mails.tsinghua.edu.cn; cy-wei19@mails.tsinghua.edu.cn; yanry18@mails.tsinghua.edu.cn
[*]*Corresponding Author*

Received 16 May 2022; Accepted 11 January 2023;
Publication 13 March 2023

Abstract

Web service recommendation remains a highly demanding yet challenging task in the field of services computing. In recent years, researchers have started to employ side information comprised in a heterogeneous Web service ecosystem to address the issues of data sparsity and cold start in Web service recommendation. Some recent works have exploited the deep learning techniques to learn user/Web service representations accumulating information from multiplex sources. However, we argue that they still struggle to utilize multi-source information in a discriminating, unified and flexible manner. To tackle this problem, this paper presents a novel multi-source information graph-based Web service recommendation framework (MGASR), which can automatically and efficiently extract multifaceted knowledge from the heterogeneous Web service ecosystem. Specifically, different node-type and edge-type dependent parameters are designed to model corresponding types

Journal of Web Engineering, Vol. 21_8, 2287–2312.
doi: 10.13052/jwe1540-9589.2183

of objects (nodes) and relations (edges) in the Web service ecosystem. We then leverage graph neural networks (GNNs) with an attention mechanism to construct a multi-source information neural network (MIN) layer, for mining diverse significant dependencies among nodes. By stacking multiple MIN layers, each node can be characterized by a highly contextualized representation due to capturing high-order multi-source information. As such, MGASR can generate representations with rich semantic information toward supporting Web service recommendation tasks. Extensive experiments conducted over three real-world Web service datasets demonstrate the superior performance of our proposed MGASR as compared to various baseline methods.

Keywords: Web service recommendation, web service ecosystem, multi-source information, deep learning, graph neural networks, attention mechanism.

1 Introduction

Accelerated by phenomenal development, several newly emerged concepts such as Big Service [32], Internet of Services (IoS) [15], and increasingly more Web services have been published onto the Internet recently. Such Web services and various objects (e.g., Web service providers and users) related to them gradually form a knowledge network, i.e., a Web service ecosystem [11, 28], through a variety of complex correlations. Such a Web service ecosystem carries a large amount of multi-source information. As illustrated in Figure 1, through the invocation and recommendation relationship, published Web services are connected to users. On the left-hand side, Web services are typically linked to their providers, publication dates, descriptions, and labeled with categories. On the right-hand side, users are in turn linked to their preferences, social friendships, Web service invocation history, and their physical locations. Each type of information represents a unique data source.

Web service recommendation [12] has played an increasingly important role in the Web service ecosystem for meeting users' (or mashup developers') personalized demands and alleviating the issue of information overload. Existing Web service recommendation models are mostly based on the collaborative filtering (CF) [3, 37] paradigm. Their main focus is to learn user interests and to estimate user preference from historical interaction data (e.g., invocation or transaction). Common techniques range from matrix

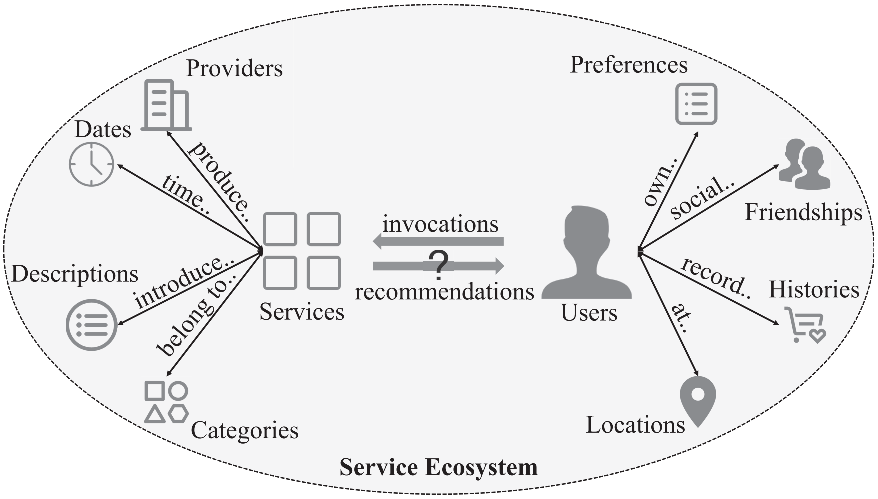


Figure 1 A multi-source information scenario of the Web service ecosystem.

factorization (MF) to deep learning [4, 5], such as state-of-the-art graph neural networks (GNNs) [29] and attention mechanisms [22]. Due to the severe data sparsity and cold-start problems in the real-world Web service ecosystem [8, 16], oftentimes these models can barely provide satisfactory Web service recommendations to users.

Therefore, some recent efforts consider introducing some side information [20] (e.g., geographical location, chronological sequence, and textual description) to enrich the profiling of users and Web services to enhance the performance of Web service recommendation [1, 14, 20, 30, 36]. Despite their effectiveness, these works suffer from two major limitations:

The semantics and strength of multi-source information are not sufficiently exploited. Each type of source may uniquely describe a relation or an attribute of a corresponding user/Web service in the Web service ecosystem. For example, *colleague* and *high-school-classmate* relationships over a user carry different semantics information to describe a social relationship. Moreover, one type of information may pose a distinct influence when a user selects and invokes the Web service, which is significant to Web service recommendation. Using the example above, the *colleague* relationship may impact a user more when selecting a professional software-based Web service, while the *high-school-classmate* relationship may impact more when selecting a social media-based Web service. However, most related methods simply assume that different types of information share the same feature space, or set up distinct non-sharing weights for every type of information alone [31, 35]. As illustrated by the simple example, without exploiting the semantics and strength of multi-source information, that is, without modeling multi-source information discriminately, may cause some valuable signals to get lost, which in turn may lead to a decline in Web service recommendation performance.

The available multi-source information is not thoroughly leveraged in a unified and flexible manner. Taking the state-of-the-art GNN-based methods as an example, most of them are designed for a homogeneous environment. Thus, for adopting GNNs to deal with multiple types of information in the Web service ecosystem, some existing methods build different graphs in terms of different types of data separately, and they utilize GNNs to learn representations over separate graphs [6], which adds to inefficient data processing efforts. Other methods construct different pre-defined meta-paths for different scenarios, which requires specific professional domain knowledge [7] and leads to trouble in generalizing. In a nutshell, how to model the Web service ecosystem as one integrated Web service network with multi-source information, and automatically extract more useful signals from the network for Web service recommendation, deserves further research.

In order to tackle the aforementioned limitations, based on **M**ulti-source information in the Web service ecosystem, inspired by the advances of **G**NNs and **A**ttention mechanisms, we propose a novel unified **S**ervice **R**ecommendation framework (**MGASR**). Firstly, the multi-source information scenario of the Web service ecosystem is modeled as a large-scale, heterogeneous graph. In this graph, users, Web services, and their attributes are treated as nodes, and the relations between them as edges. Secondly, on top of the graph, we construct a multi-source information neural network (MIN) layer for Web service recommendation. Different semantics of the nodes and edges are revealed by deploying type-specific projection parameters. To clarify the strength of multi-source information, we devise a node- and edge-type aware multi-head attention mechanism. Finally, for being scalable to any multi-source information scenarios, we develop an inductive multi-source message-passing GNN to aggregate and propagate multi-source messages among the connected nodes in different types on this graph. Note that MGASR can incorporate multi-source information from high-order neighbors by stacking multiple MIN layers with feedforward neural network and residual connections. Through these designs, our proposed MGASR can effectively address the aforementioned limitations, and help to exploit multi-source information in the Web service ecosystem for more accurate Web service recommendation, in a discriminating, unified and flexible manner.

To summarize, the main contributions of our work are three-fold:

- We construct a heterogeneous graph for representing the Web service ecosystem with multi-source information, where users, Web services, and their attributes are represented as multiple types of nodes, and different types of relations among them are represented as multiple types of edges.
- We develop a novel Web service recommendation framework MGASR on top of the constructed graph. Specifically, we devise a type-aware multi-head attention mechanism with type-specific parameters for reflecting the various semantics and strengths of multi-source information. Moreover, we design a multi-source message-passing GNN to capture the multi-source information flexibly and in unison.
- We conduct extensive empirical studies on three real-world Web service datasets. Experimental results show that MGASR can effectively improve the Web service recommendation performance, comparing to the baselines.

The remainder of this paper is organized as follows. We introduce relevant definitions and formulate our problem in Section 2. Section 3 details our proposed framework MGASR. The experimental results and analysis are presented in Section 4. We review the related work in Section 5. Finally, Section 6 concludes our work.

2 Preliminaries

In this section, we first introduce the definitions of the Web service ecosystem and message-passing GNNs, then we give a formulation of our target problem, i.e., Web service recommendation based on multi-source information.

2.1 Definitions

Definition 1: Web service ecosystem. To model a real-world Web service ecosystem, we define a Web service ecosystem containing multi-source information in the form of a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P}, \mathcal{Q})$. Each node $v \in \mathcal{V}$ and each edge $e \in \mathcal{E}$ (complex relations between them) in this graph can be discriminated into the types to which they belong by $\mathcal{V} \overset{\vartheta(v)}{\to} \mathcal{P}$ and $\mathcal{E} \overset{\varphi(e)}{\to} \mathcal{Q}$, respectively.

For example, nodes can be classified as users (or mashups) $\mathcal{U}$, Web services $\mathcal{S}$, and their attributes $\mathcal{A}$, while edges can be interpreted as different types of relations $\mathcal{R}$ between the nodes such as invocation, production, and composition. For instance, consider a triplet (t, e, n) comprising a node t, one of t's neighbor nodes n, and an edge e between them. Also we can define a type-specific relation about them as $\langle \vartheta(t), \varphi(e), \vartheta(n) \rangle$, which can reflect the intention of this relation.

Definition 2: Message-passing GNN. Generally, a GNN based on the message-passing paradigm can be regarded as updating the embedding (i.e., representation) $\mathbf{X}_{l+1}[t]$ of the target node t at the $(l+1)$th layer. It can be done by aggregating the messages $\mathbf{X}_l[n]$ passed from each of its neighbor nodes $n \in N(t)$, and finally combined with its current state $\mathbf{X}_l[t]$ at the lth layer:

$$\mathbf{X}_{l+1}[t] \leftarrow \underset{\substack{\forall n \in N(t) \\ \forall e \in E(t,n)}}{\mathrm{Cmb}} \left(\mathrm{Agg}(\mathbf{X}_l[n], e), \mathbf{X}_l[t] \right) \tag{1}$$

where $N(t)$ refers to all the neighboring nodes of t, and $E(t, n)$ denotes the edge(s) between t and n. $\mathrm{Agg}(\cdot)$ and $\mathrm{Cmb}(\cdot)$ represent the neighborhood aggregation operator and the message combination operator, respectively.

The attention mechanism is further added into message-passing GNNs to distinguish the importance of different neighbor messages:

$$\mathbf{X}_{l+1}[t] \leftarrow \underset{\substack{\forall n \in N(t) \\ \forall e \in E(t,n)}}{\mathrm{Cmb}} \left(\mathrm{Att}(t, e, n) \cdot \mathrm{Msg}(t, e, n), \mathbf{X}_l[t] \right) \tag{2}$$

where $\mathrm{Att}(\cdot)$ denotes the application of an attention mechanism, and $\mathrm{Msg}(\cdot)$ denotes the neighbor message.

2.2 Problem Formulation

Problem: Web service recommendation based on multi-source information. Given a $\mathcal{G}$, the goal is to recommend Web services (e.g., $s \in \mathcal{S}$) to users (e.g., $u \in \mathcal{U}$) who are of potential interest to them based on multi-source information carried in $\mathcal{G}$.

3 Methodology

In this section, we present our MGASR framework. We detail the three main segments of MGASR, which are *multi-source message*, *type-aware*

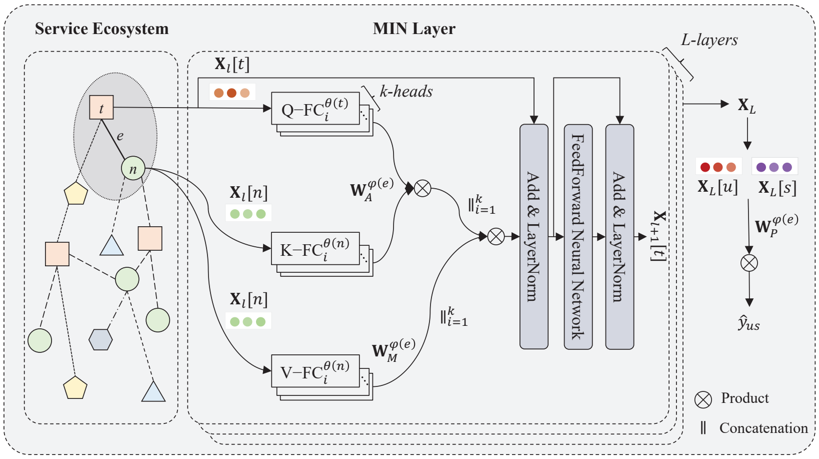


Figure 2 The architecture of MGASR. We build the heterogeneous Web service ecosystem as an integrated graph, where different shapes represent different types of nodes and relation edges. The MIN layer is constructed through *multi-source message* and *type-aware multi-head attention*. After *multi-layer propagation*, we can obtain high-quality representations of users/Web services.

multi-head attention, and *multi-layer propagation*. The architecture of MGASR is illustrated in Figure 2.

3.1 Multi-source Message

In order to learn the representation of a node in a Web service ecosystem $\mathcal{G}$, it is important to consider the heterogeneity of its neighbor nodes and connected edges. Therefore, for a triplet (t, e, n) highlighted in Figure 2 in a grey oval, we design an operator of multi-source message $\mathrm{M}(\cdot)$, corresponding to the multi-head attention mechanism as follows:

$$\mathrm{M}(t, e, n) = \|_{i=1}^{k} \, M\text{-}head_i(t, e, n) \tag{3}$$

$$M\text{-}head_i(t, e, n) = \text{V-FC}_i^{\vartheta(n)}(\mathbf{X}_l[n])\mathbf{W}_M^{\varphi(e)} \tag{4}$$

where $M\text{-}head_i(t, e, n)$ is the ith multi-source message head. k is the number of heads. $\|$ denotes the concatenation operation.

For each type-specific relation $\langle \vartheta(t), \varphi(e), \vartheta(n) \rangle$, we deploy a distinct projection matrix $\mathbf{W}_M^{\varphi(e)} \in \mathbb{R}^{\frac{d}{k} \times \frac{d}{k}}$ for capturing different semantics of relations, where d is the embedding size. $\text{V-FC}_i^{\vartheta(n)}(\cdot)$ is presented as a single fully connected layer for a value vector according to the ith multi-source message head and the neighbor node type $\vartheta(n)$, as shown in Figure 2 in the lower portion.

3.2 Type-aware Multi-head Attention

In general, not all neighbor messages contribute equally to the state update of the target node. For example, for Web service recommendation, information about a user's invocation history of a Web service is often more important than other information. Thus, motivated by the multi-head attention mechanism, we propose a type-aware multi-head attention mechanism $\mathrm{A}(\cdot)$ for the diverse nature of multi-source information.

Depending on target node type $\vartheta(t)$, neighbor node type $\vartheta(n)$ and edge type $\varphi(e)$, according to the ith multi-source message head, we project the embedding of the target node (t) into a query vector $\mathbf{Q}_i(t)$, and the embedding of the neighbor node (n) into a key vector $\mathbf{K}_i(n)$. The operation of $\mathrm{A}(\cdot)$ relies on the help of $\text{Q-FC}_i^{\vartheta(t)}(\cdot)$ and $\text{K-FC}_i^{\vartheta(n)}(\cdot)$, which both resemble $\text{V-FC}_i^{\vartheta(n)}(\cdot)$, as shown in Figure 2. The similarity between $\mathbf{Q}_i(t)$ and $\mathbf{K}_i(n)$ is taken as the type-aware attention coefficient, which is calculated as follows:

$$\mathrm{A}(t,e,n) = \underset{\forall n \in N(t)}{\mathrm{Softmax}}(\|_{i=1}^{k}\ A\text{-}head_i(t,e,n)) \tag{5}$$

$$A\text{-}head_i(t,e,n) = \frac{\mathbf{Q}_i(t)\mathbf{W}_A^{\varphi(e)}\mathbf{K}_i(n)^{\mathsf{T}}}{\sqrt{d}} \tag{6}$$

$$\mathbf{Q}_i(t) = \text{Q-FC}_i^{\vartheta(t)}(\mathbf{X}_l[t]) \tag{7}$$

$$\mathbf{K}_i(n) = \text{K-FC}_i^{\vartheta(n)}(\mathbf{X}_l[n]) \tag{8}$$

where $\sqrt{d}$ is the scaling factor, and a measure matrix $\mathbf{W}_A^{\varphi(e)} \in \mathbb{R}^{\frac{d}{k}\times\frac{d}{k}}$ is also set for each edge type. Then, for each target node t, $\sum_{\forall n \in N(t)} \mathrm{A}(t,e,n) = \mathbf{1}_k$ can be derived by a softmax function $\mathrm{Softmax}(\cdot)$.

3.3 Multi-layer Propagation

In view of the effectiveness of residual connections and layer normalization, we can obtain the representation $X_{l+1}[t]$ of the target node t at the $(l+1)$th layer by a combination operation as:

$$\mathbf{X}_{l+1}[t] = \mathrm{MIN}(\mathbf{X}_l[t]) \tag{9}$$

$$\mathbf{X}_{l+1}[t] = \mathrm{LN}(\mathrm{FFN}(\widetilde{\mathbf{X}}_l[t]) + \widetilde{\mathbf{X}}_l[t]) \tag{10}$$

$$\widetilde{\mathbf{X}}_l[t] = \mathrm{LN}\left(\sum_{\forall n \in N(t)} \mathrm{A}(t,e,n) \cdot \mathrm{M}(t,e,n) + \mathbf{X}_l[t]\right) \tag{11}$$

where $\mathrm{MIN}(\cdot)$ is our developed MIN layer, $\mathrm{LN}(\cdot)$ denotes the layer normalization for stabilizing the training process. $\mathrm{FFN}(\cdot)$ represents a two-layer fully-connected feed-forward neural network module, as shown in Figure 2 on the right-hand side of a MIN layer.

Take node h as an example, $\mathrm{FFN}(\cdot)$ can be represented as:

$$\mathrm{FFN}(\mathbf{X}[h]) = (\sigma(\mathbf{X}[h]\mathbf{W}_1 + \mathbf{b}_1))\mathbf{W}_2 + \mathbf{b}_2 \tag{12}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d\times d}$, $\mathbf{W}_2 \in \mathbb{R}^{d\times d}$, $\mathbf{b}_1 \in \mathbb{R}^{1\times d}$ and $\mathbf{b}_2 \in \mathbb{R}^{1\times d}$ are the trainable parameters of this module, and σ is the activation function.

Since incorporating high-order information can benefit the representation learning of nodes in a graph, and in order to acquire the high-quality representation $X_L[t]$ of the target node t, we can realize a multi-layer propagation by stacking L MIN layers.

Through this design, the target node t can reach L-hop neighbors and capture high-order multi-source information to learn a good-quality representation.

3.4 Prediction and Optimization

For our formalized problem defined in Section 2.2, we now set the target node t in the above content as either a user u, or a Web service s. Thus, we can obtain their learned representation $\mathbf{X}_L[u]$, or $\mathbf{X}_L[s]$, respectively. Then, we can calculate the possibility that user u will invoke Web service s by:

$$\widehat{y}_{us} = \mathrm{Sigmoid}(\mathbf{X}_L[u]\mathbf{W}_P^{\varphi(e)}\mathbf{X}_L[s]^{\mathsf{T}}) \tag{13}$$

where $\mathbf{W}_P^{\varphi(e)} \in \mathbb{R}^{d\times d}$ is a trainable weight matrix for measuring the potential recommendation relation between Web service s and user u. $\widehat{y}_{us}$ is the predicted possibility derived by a sigmoid function $\mathrm{Sigmoid}(\cdot)$.

Finally, we optimize the following cross-entropy loss to estimate the parameters Θ of MGASR:

$$\mathcal{L} = -\sum_{(u,s)\in\Psi} (y_{us}\mathrm{log}(\widehat{y}_{us}) + (1-y_{us})\mathrm{log}(1-\widehat{y}_{us})) + \lambda\|\Theta\|_2 \tag{14}$$

where y_{us} is the ground truth, Ψ is the training set, and λ is the L2-regularization parameter for reducing overfitting.

4 Experiments

In this section, we present our experiments conducted and analyze the experimental results.

To evaluate our proposed MGASR framework, we designed and conducted a series of experiments on three real-world Web service datasets. We aimed to answer the following three research questions through the experiments.

RQ1: How does our Web service recommendation framework MGASR perform compared with the baselines?

RQ2: Does introducing multi-source information improve MGASR's Web service recommendation performance?

RQ3: How do the hyper-parameters (the embedding size, the number of heads in type-aware multi-head attention, and the number of MIN layers) of MGASR affect the Web service recommendation performance?

4.1 Datasets and Metrics

To evaluate the performance of MGASR, we conducted experiments on three real-world Web service datasets: PGW,[1] Amazon[2] and Douban.[3] The statistics of these three Web service datasets are summarized in Table 1.

- **PGW:** A world-largest Web API management platform for publishing, searching, and invoking RESTful Web services. We crawled the data related to mashups and Web services, which includes Web service invocation information and their category information.
- **Amazon:** A well-known e-commerce platform that allows users to rate its product-based Web services. We crawled related information about 2747 Web services, which includes the categories views, and brands of the Web services. In addition, we treated a rating greater than 3 as a Web service invocation actually happening.
- **Douban:** A famous social platform that records user ratings and reviews of book-based Web services. For 3000 Web services, we crawled

[1] https://www.programmableweb.com/

[2] https://www.amazon.cn/

[3] https://book.douban.com/

Table 1 The statistics of datasets

Dataset	Item	Value
PGW	# Invocations	16,274
	# Mashups	7,814
	# Apis	1,720
	# Mashup categories	422
	# Api Ccategories	159
	# Invocation sparsity	99.879%
Amazon	# Invocations	46,908
	# Users	6,164
	# Web services	2,747
	# Web service categories	22
	# Web service views	3,844
	# Web service brands	332
	# Invocation sparsity	99.723%
Douban	# Invocations	106,171
	# Users	12,450
	# Web services	3,000
	# User locations	440
	# Web service providers	1,479
	# Web service production dates (year)	48
	# Invocation sparsity	99.716%

the location information of the users who interacted with these Web services, as well as the provider information and production date information of the Web services. Similarly, we also viewed a rating greater than 3 as a Web service invocation actually happening.

We adopted two widely used metrics called MRR (mean reciprocal rank) and NDCG (normalized discounted cumulative gain) in our study, which are defined as follows:

- **MRR@K:** It can compute the reciprocal rank of the true relevant items predicted by a recommendation framework in the top-K ranking list.

- **NDCG@K:** A measure of ranking quality, where positions are discounted logarithmically. It assigns higher scores to the hits at a higher position in the top-K ranking list.

We reported the results on these two metrics MRR@K and NDCG@K at K = 5, 10, and 20 in our experiments, respectively.

4.2 Experimental Setup

Our MGASR framework was implemented in Pytorch.[4] Unless otherwise specified, all experiments were conducted with GPU acceleration.

For each dataset used, we randomly selected 70% of the Web service invocations in a whole dataset as the training set, 10% as the validation set, and the remaining 20% as the test set. We removed the mashups and users that do not appear in the training set during the testing phase. The learning rate was set as $1e^{-3}$, L2-regularization parameter λ was $1e^{-4}$, and the dropout rate was 0.1. The Linear rectification function (ReLU) was used as the activation function. For the embedding size, we tested it in the set of $\{$16, 32, 64, 128, 256$\}$. For the number of heads in type-aware multi-head attention, we tuned it in the set of $\{2, 4, 8, 16\}$. For the number of MIN layers, we investigated it in the set of $\{1, 2, 3, 4\}$. We also optimized all the trainable parameters of MGASR with an AdamW optimizer, by a batch size of 1024 for the PGW dataset, and 2048 for the Amazon and Douban datasets. For each positive sample (u, s), five negative samples from unobserved invocation history of user u were chosen to train along with it. We adopted an early stopping strategy to stop the training process if NDCG@20 on the validation set does not increase for 50 epochs. For each test case, our evaluation protocol ranked the test Web service with all the Web services except the remaining positive ones invoked by the user. We randomly initialized these methods and ran each of them five times. Finally, we reported the average results.

4.3 Baselines

To verify the effectiveness of our MGASR, we compared it with the following seven representative and competitive baseline methods.

- **MF:** [18] A widely used baseline that optimizes pairwise loss based on the idea of bayesian personalized ranking and matrix factorization.

[4] https://pytorch.org/

- **DNN:** [10] A neural CF method based on deep neural networks, which can capture user-item collaborative signals.
- **GCN:** [9] A state-of-the-art recommender model based on graph convolutional networks (GCNs), which is simple yet powerful only by neighborhood aggregation and multi-layer propagation.
- **HGNN:** [35] A heterogeneous graph neural network model which jointly considers node heterogeneous contents encoding, type-based neighbors aggregation, and heterogeneous types combination.
- **HGAT:** [25] A model constructed based on the attention mechanism for heterogeneous graph, including node-level attention and semantic-level attention.
- **MHGNN:** [7] A meta-path aggregated GNNs, which can perform inter-meta path aggregation by attention to combine messages from multiple meta-paths.

For the baseline methods HGNN, HGAT, and MHGNN, we borrowed their main embedding approaches and carefully reconstructed them to tailor to the Web service recommendation task.

4.4 Performance Comparison (RQ1)

To answer RQ1, we compared the Web service recommendation performance of our MGASR with the aforementioned baselines. The performance comparison results are presented in Table 2. In Table 2, the bold scores are the best in the method group. *Improv.* represents that improvements over baselines are statistically significant with p-value < 0.05. Note that the *Improv.* column is the performance of MGASR relative to the best baseline method.

From the reported results in Table 2, three observations can be made:

First, we can find that the models based on deep neural networks perform better than those on MF, which can be explained by the excellent representation capability of deep neural networks. In addition, the models based on GNNs achieve the best performance. The reason may be their powerful ability to handle graph-structured data, which empowers the effectiveness of representation learning.

Second, since multi-source information contains more useful knowledge, the models that leverage multi-source information outperform those that do not. Effectively extracting and mining the meta-paths of multi-source information has been proven to facilitate recommendation enhancement.

Third, our MGASR, which can automatically and efficiently recognize valuable multi-source information within an entire complex system, outperforms all baselines substantially on all MRR@K and NDCG@K metrics. The average improvement of our framework MGASR to the best baseline is 1.63% and 2.61% for MRR and NDCG on the PGW dataset, 1.81% and 1.83% on the Amazon dataset, meanwhile, 1.94% and 1.84% on the Douban dataset. Such a finding justifies the effectiveness of MGASR.

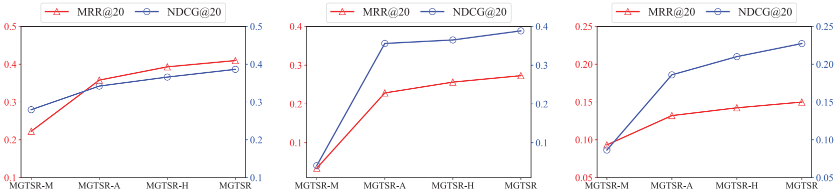
4.5 Ablation Study (RQ2)

To verify the effectiveness of the main components in MGASR and answer RQ2, we compared the performance of our MGASR with its following three variants on MRR@20 and NDCG@20. The experimental results are recorded in Figure 3.

- **MGASR-M:** A variant of MGASR that only considers the single source information (the users' invocation of the Web service).
- **MGASR-A:** A variant of MGASR that utilizes uniform weights in the designed type-aware multi-headed attention mechanism.

Table 2 Performance comparison of different methods

Dataset	Metric	MF	DNN	GCN	HGNN	HGAT	MHGNN	MGASR	*Improv.*
PGW	MRR@5	0.1250	0.1952	0.2072	0.3456	0.3548	0.3613	**0.3672**	1.63%
	MRR@10	0.1343	0.2052	0.2183	0.3671	0.3763	0.3844	**0.3909**	1.66%
	MRR@20	0.1403	0.2106	0.2242	0.3847	0.3929	0.4031	**0.4096**	1.61%
	NDCG@5	0.1391	0.2170	0.2361	0.2747	0.2811	0.2859	**0.2934**	2.62%
	NDCG@10	0.1614	0.2408	0.2630	0.3014	0.3085	0.3113	**0.3211**	3.15%
	NDCG@20	0.1827	0.2607	0.2844	0.3635	0.3704	0.3786	**0.3864**	2.06%
Amazon	MRR@5	0.0139	0.0165	0.0280	0.1826	0.1870	0.1910	**0.1947**	1.90%
	MRR@10	0.0162	0.0190	0.0324	0.2486	0.2524	0.2601	**0.2650**	1.88%
	MRR@20	0.0182	0.0207	0.0354	0.2559	0.2613	0.2685	**0.2729**	1.64%
	NDCG@5	0.0120	0.0137	0.0272	0.2310	0.2369	0.2402	**0.2449**	1.96%
	NDCG@10	0.0175	0.0180	0.0362	0.3397	0.3473	0.3552	**0.3620**	1.91%
	NDCG@20	0.0218	0.0242	0.0452	0.3628	0.3699	0.3794	**0.3855**	1.61%
Douban	MRR@5	0.0565	0.0634	0.0825	0.1195	0.1229	0.1246	**0.1275**	2.33%
	MRR@10	0.0621	0.0700	0.0901	0.1326	0.1354	0.1393	**0.1418**	1.79%
	MRR@20	0.0663	0.0743	0.0946	0.1405	0.1433	0.1474	**0.1499**	1.70%
	NDCG@5	0.0470	0.0442	0.0593	0.1549	0.1568	0.1617	**0.1649**	1.98%
	NDCG@10	0.0569	0.0574	0.0733	0.1767	0.1816	0.1859	**0.1893**	1.83%
	NDCG@20	0.0681	0.0693	0.0876	0.2129	0.2178	0.2235	**0.2273**	1.70%



(a) PGW (b) Amazon (c) Douban

Figure 3 Ablation study of MGASR.

- **MGASR-H:** A variant of MGASR that deploys only one MIN layer to realize the user and Web service representation learning.

Through careful observation and comparison of the content of Figure 3, we can draw the following three findings.

First, by comparing the performance of MGASR-M with that of MGASR, it illustrates that the use of multi-source information can better model users and Web services, thus significantly boosting Web service recommendation.

Second, according to the performance of MGASR-A, it proves that different multi-source messages have different strengths. This is because their diverse semantics can have inconsistent effects in different scenarios. Consequently, it is also necessary to design an appropriate attention mechanism for different types of information.

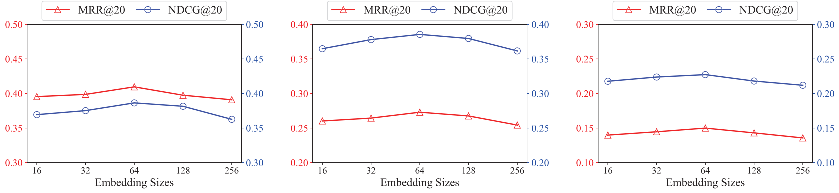
Third, the performance of MGASR-H demonstrates that incorporating higher-order multi-source information can enhance Web service recommendation.

4.6 Sensitivity Analysis (RQ3)

To understand how hyper-parameters influence the performance of MGASR and to answer RQ3, we performed a sensitivity analysis on some main hyper-parameters in MGASR.

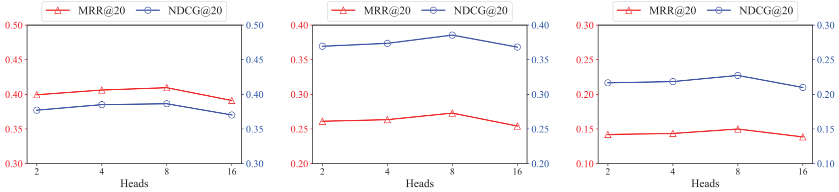
Firstly, we experimented with different embedding sizes on the three Web service datasets to check on their respective influences. Afterwards, we varied the head number of the designed type-aware multi-head attention. Finally, we tried to figure out whether MGASR can benefit from stacking multiple MIN layers. Figures 4, 5, and 6 summarize the experimental results on MRR@20 and NDCG@20 over the three hyper-parameters, respectively.

For the embedding size, we can find a satisfactory performance on all three Web service datasets when it is set to be 64. The MGASR performance



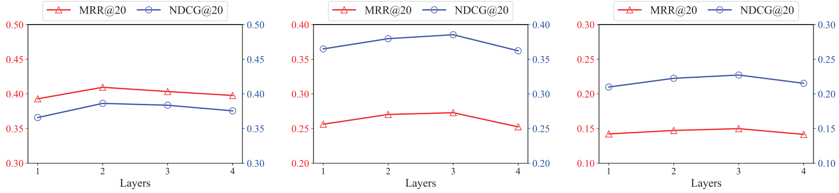
(a) PGW (b) Amazon (c) Douban

Figure 4 Impact of the different embedding sizes.



(a) PGW (b) Amazon (c) Douban

Figure 5 Impact of the different number of heads in type-aware multi-head attention.



(a) PGW (b) Amazon (c) Douban

Figure 6 Impact of the different number of MIN layers.

variation curves illustrate that a suitable embedding size can be found experimentally. Furthermore, we can infer that a smaller embedding size is difficult to contain valid information, while a larger one may lead to an overfitting dilemma.

When we carried out the designed type-aware multi-head attention mechanism, we examined the number of different attention heads. We found that our MGASR achieves the best performance on the PGW, Amazon, and Douban datasets when the number of heads was 8. Similarly, a smaller number of heads would make MGASR struggle to express the advantages of the multi-head attention mechanism, while a larger number would increase the complexity of computation.

We can see that, by increasing the number of MIN layers, the Web service recommendation results on the PGW, Amazon, and Douban datasets are improved in the beginning. However, after reaching their peak, their performance begins to decline. The reason might be that deeper MIN layers might introduce noise and lead to an over-smooth phenomenon. Specifically, the best performance is achieved on the PGW dataset when the number of MIN layers equals 2, and on both Amazon and Douban datasets when it is 3.

5 Related Work

In this section, we briefly review three lines of research closely related to ours, namely Web service recommendation, GNNs for a heterogeneous system, and GNNs/attention-based recommendation.

5.1 Web Service Recommendation

Most classical Web service recommendation models are mainly designed based on semantic techniques or matrix factorization techniques [17, 34]. In recent years, deep learning techniques have made great progress in the field of Web service recommendation. Among them, DLTSR [2] designs a deep learning framework for recommending long-tail web services. QF-RNN [27] employs MF with long short term memory (LSTM) model to handle the time-aware Web service recommendation. HINGAN [31] proposes a generative adversarial network-based Web service recommendation model with rich side information for mashup creation. A-HSG [26] mines high-order social similarity and difference by GNNs to produce accurate Web service recommendations. coACN [33] focuses on learning the bilateral information toward Web service recommendation with GNNs.

Despite great successes, how to model the Web service ecosystem more concretely, and enhance Web service recommendation based on multi-source data and state-of-the-art deep learning techniques deserve further investigation.

5.2 GNNs for a Heterogeneous System

Recently, a number of efforts have aimed at extending GNNs to study heterogeneous systems. RGCN [19] models relations by employing specialized transformation matrices for each type. HGNN [35] samples neighbors through random walk and uses Bi-LSTMs to generate heterogeneous node embeddings. HGAT [25] can distinguish the different importance of neighbors and multiple pre-defined meta-paths based on an attention mechanism. MHGNN [7] considers the intermediate nodes and aggregates the intra-meta-path and inter-meta-path information.

In contrast, our work dexterously exploits GNNs and an attention mechanism in a uniform and extensible manner, to efficiently and automatically mine semantics-sensitive knowledge from multiplex sources in the Web service ecosystem.

5.3 GNNs/Attention-based Recommendation

Lately, GNNs have been proved to have superior performance in processing graph-structured data. For the field of recommender systems, NGCF [24] learns the representations on a user-item bipartite graph, while displaying the collaborative filtering signals. LightGCN [9] utilizes neighborhood aggregation and multi-layer propagation to learn the representation of users and items. GraphRec [2] applies GNNs on both a user–user graph and user–item graph to tackle the social recommendation problem. KGAT [23] combines GNNs with an attention mechanism on a collaborative knowledge graph. SASRec [13] applies a self-attention-based encoder to learn item importance in sequences, which can characterize complex transition correlations. Further, BERT4Rec [21] proposes a bidirectional self-attention-based layer.

In particular, our MGASR is focused on the Web service recommendation task in a multi-source information scenario of the Web service ecosystem with our devised MIN layer based on GNNs and attention mechanism.

6 Conclusions

In this paper, we have presented a deep learning-powered Web service recommendation framework MGASR based on GNNs and amulti-head attention mechanism for a Web service ecosystem comprising multi-source information. MGASR can automatically and efficiently extract valuable multi-source information in a unified and flexible manner to realize more accurate Web service representation and recommendation. Through extensive experiments, MGASR has been confirmed to attain better performance compared to the baselines. For our future work, we plan to generalize MGASR to the large-scale Web service datasets. We also plan to introduce the chronological order information about Web service invocation to make MGASR time-aware.

Acknowledgements

This research has been partially supported by the National Natural Science Foundation of China (No.62173199).

References

[1] Hossein Arabi, Vimala Balakrishnan, and Nor Liyana Mohd Shuib. A context-aware personalized hybrid book recommender system. *Journal of Web Engineering*, pages 405–428, 2020.

[2] Bing Bai, Yushun Fan, Wei Tan, and Jia Zhang. DLTSR: A deep learning framework for recommendations of long-tail web services. *IEEE Transactions on Services Computing*, 13(1):73–85, 2017.

[3] Khalid Benabbes, Khalid Housni, Ali El Mezouary, and Ahmed Zellou. Recommendation system issues, approaches and challenges based on user reviews. *Journal of Web Engineering*, 21(4):1017–1054, 2022.

[4] Shuhui Chen, Yushun Fan, Wei Tan, Jia Zhang, Bing Bai, and Zhenfeng Gao. Service recommendation based on separated time-aware collaborative poisson factorization. *Journal of Web Engineering*, pages 595–618, 2017.

[5] Debashis Das, Laxman Sahoo, and Sujoy Datta. A survey on recommendation system. *International Journal of Computer Applications*, 160(7):6–10, 2017.

[6] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *Proceedings of The World Wide Web Conference*, pages 417–426, 2019.

[7] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The World Wide Web Conference*, pages 2331–2341, 2020.

[8] Zhenfeng Gao, Yushun Fan, Xiu Li, Liang Gu, Cheng Wu, and Jia Zhang. Discovery and analysis about the evolution of service composition patterns. *Journal of Web Engineering*, 18(7):579–626, 2019.

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of The 26th International Conference on World Wide Web*, pages 173–182, 2017.

[11] Keman Huang, Yushun Fan, and Wei Tan. Recommendation in an evolving service ecosystem based on network prediction. *IEEE Transactions on Automation Science and Engineering*, 11(3):906–920, 2014.

[12] R Kalpana, K Saruladha, and J Jayabharathy. Studying the performance of qos specific web service recommendation system using virtural regions. *J. Web Eng.*, 15(5&6):397–411, 2016.

[13] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *Proceedings of IEEE International Conference on Data Mining*, pages 197–206. IEEE, 2018.

[14] Tingting Liang, Liang Chen, Jian Wu, Hai Dong, and Athman Bouguettaya. Meta-path based service recommendation in heterogeneous information networks. In *Proceedings of International Conference on Service Oriented Computing*, pages 371–386. Springer, 2016.

[15] Rafael Moreno-Vozmediano, Rubén S Montero, and Ignacio M Llorente. Key challenges in cloud computing: enabling the future Internet of services. *IEEE Internet Computing*, 17(4):18–25, 2012.

[16] Senthilselvan Natarajan, Subramaniyaswamy Vairavasundaram, Sivaramakrishnan Natarajan, and Amir H Gandomi. Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. *Expert Systems with Applications*, 149(3):113248, 2020.

[17] Tian Qiu, Lei Li, and Pin Lin. Web service discovery with uddi based on semantic similarity of service properties. In *Proceedings of International Conference on Semantics, Knowledge and Grid*, pages 454–457. IEEE, 2007.

[18] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of The 25th Conference on Uncertainty in Artificial Intelligence*, pages 452–461, 2009.

[19] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *Proceedings of European Semantic Web Conference*, pages 593–607. Springer, 2018.

[20] Sushmita Singh and Manvi Siwach. Handling heterogeneous data in knowledge graphs: A survey. *Journal of Web Engineering*, pages 1145–1186, 2022.

[21] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of The 28th ACM International Conference on Information and Knowledge Management*, pages 1441–1450, 2019.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:6000–6010, 2017.

[23] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 950–958, 2019.

[24] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 165–174, 2019.

[25] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *Proceedings of The World Wide Web Conference*, pages 2022–2032, 2019.

[26] Chunyu Wei, Yushun Fan, Jia Zhang, and Haozhe Lin. A-HSG: Neural attentive service recommendation based on high-order social graph. In *Proceedings of IEEE International Conference on Web Services*, pages 338–346. IEEE, 2020.

[27] Xing Wu, Yushun Fan, Jia Zhang, Haozhe Lin, and Junqi Zhang. QF-RNN: QI-matrix factorization based rnn for time-aware service recommendation. In *Proceedings of IEEE International Conference on Services Computing*, pages 202–209. IEEE, 2019.

[28] Xing Wu, Zhenfeng Gao, Yushun Fan, Xiu Li, Liang Gu, Jia Zhang, Chang Chen, Hao Zhang, and Qiang Wang. T-dses: A blockchain-powered trusted decentralized service eco-system. *Journal of Web Engineering*, pages 2199–2242, 2021.

[29] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.

[30] Fenfang Xie, Liang Chen, Yongjian Ye, Zibin Zheng, and Xiaola Lin. Factorization machine based service recommendation on heterogeneous

information networks. In *Proceedings of IEEE International Conference on Web Services*, pages 115–122. IEEE, 2018.

[31] Fenfang Xie, Shenghui Li, Liang Chen, Yangjun Xu, and Zibin Zheng. Generative adversarial network based service recommendation in heterogeneous information networks. In *Proceedings of IEEE International Conference on Web Services*, pages 265–272. IEEE, 2019.

[32] Xiaofei Xu, Quan Z Sheng, Liang-Jie Zhang, Yushun Fan, and Schahram Dustdar. From big data to big service. *Computer*, 48(07):80–83, 2015.

[33] Ruyu Yan, Yushun Fan, Jia Zhang, Junqi Zhang, and Haozhe Lin. Service recommendation for composition creation based on collaborative attention convolutional network. In *Proceedings of IEEE International Conference on Web Services*, pages 397–405. IEEE, 2021.

[34] Qi Yu, Zibin Zheng, and Hongbing Wang. Trace norm regularized matrix factorization for service recommendation. In *Proceedings of IEEE International Conference on Web Services*, pages 34–41. IEEE, 2013.

[35] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. Heterogeneous graph neural network. In *Proceedings of The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 793–803, 2019.

[36] Yiwen Zhang, Chunhui Yin, Qilin Wu, Qiang He, and Haibin Zhu. Location-aware deep collaborative filtering for service recommendation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(6):3796–3807, 2019.

[37] Zibin Zheng, Hao Ma, Michael R Lyu, and Irwin King. QoS-aware web service recommendation by collaborative filtering. *IEEE Transactions on Services Computing*, 4(2):140–152, 2010.

Biographies

Zhixuan Jia is currently pursuing a Ph.D. degree at the Department of Automation, Tsinghua University. His research interests include services computing, Web service recommendation and data mining.

Yushun Fan received his Ph.D. degree in control theory and application from Tsinghua University, China, in 1990. He is currently a tenured professor with the Department of Automation, Director of the System Integration Institute, and Director of the Networking Manufacturing Laboratory, Tsinghua University. He is a member of IFAC TC 5.1 and TC 5.2, Vice director of the China Standardization Committee for Automation System and Integration, and an editorial member of the International Journal of Computer Integrated Manufacturing. From September 1993 to 1995, he was a visiting scientist, supported by the Alexander von Humboldt Stiftung Foundation, with the Fraunhofer Institute for Production System and Design Technology (FHG/IPK), Germany. He has authored 10 books in enterprise modeling, workflow technology, intelligent agent, object-oriented complex system analysis, and computer integrated manufacturing. He has published more than 500 research papers in journals and conferences. His research interests include enterprise modeling methods and optimization analysis, business process re-engineering, workflow management, system integration, modern service science and technology, and petri nets modeling and analysis.

Jia Zhang received her Ph.D. degree in computer science from the University of Illinois at Chicago. She is currently the Cruse C. and Marjorie F. Calahan Centennial Chair in Engineering, Professor of Department of Computer Science at Southern Methodist University. Her research interests emphasize the application of machine learning and information retrieval methods to tackle data science infrastructure problems, with a recent focus on scientific workflows, provenance mining, software discovery, knowledge graph, and their interdisciplinary applications. Dr Zhang has co-authored one textbook "Services Computing" and has published over 170 refereed journal papers, book chapters, and conference papers. Dr Zhang has served as an associated editor of the IEEE TSC since 2008. She served as Program Committee Chair for IEEE SCC (2020), ICWS (2019), CLOUD (2018), and BigData Congress (2017). She is a senior member of the IEEE.

Xing Wu received the BS degree in control theory and application from Tsinghua University, China, in 2017. He is currently working toward a Ph.D. degree in the Department of Automation, Tsinghua University. His research interests include services computing, Web service recommendation, federated learning and blockchain.

Chunyu Wei received his B.Sc. degree in control theory and application from Tsinghua University, China, in 2019. He is currently working toward his Ph.D. degree in the Department of Automation, Tsinghua University. His research interests include services computing, Web service recommendation, and social computing.

Ruyu Yan received a B.Sc. degree from Tsinghua University, China, in 2018. She is currently a Ph.D. student in the Department of Automation at Tsinghua University, China. Her research interests include services computing, recommender systems and time series prediction.