MGMASR: Multi-Graph and Multi-Aspect Neural Network for Service Recommendation in Internet of Services

Zhixuan Jia, Yushun Fan, and Jia Zhang, *Senior Member, IEEE*

***Abstract*—With the flourishing development of Everything-as-a-Service (EaaS) and Internet of Everything (IoE), Internet of Services (IoS) has recently emerged as a new buzzword in the field of service computing. Providing accurate and personalized service recommendations to users is essential yet highly challenging in IoS, from a sea of services. Besides the severe sparsity of users' historical behavior data on services, little study has been reported in the literature on fully exploiting multiple relationship networks embedded in IoS. To fill this gap, we propose a novel Multi-Graph and Multi-Aspect neural network-powered method for Service Recommendation in IoS. Graph neural networks (GNNs) and attention mechanism are jointly employed to simultaneously extract information from a collection of heterogeneous knowledge graphs, constructed from historical data recorded in IoS including the user-service interaction graph, the user-user social graph, and the service-mashup graph. Based on the knowledge learned, user-service interactions are scrutinized from multiple aspects to better learn the multiple preferences of users and the multiple characteristics of services, in order to refine their profiles for future recommendation. The results of extensive experiments over the real-world datasets have demonstrated that MGMASR outperforms the baseline methods and can provide service recommendations more accurately for users in IoS.**

***Index Terms*—Internet of Services, service recommendation, deep learning, rating prediction.**

I. Introduction

With the booming of Service-oriented Computing (SOC) [1], Big Data [2] and Cloud Computing [3], recent years have witnessed a trend of Everything-as-a-Service (EaaS) [4] and Internet of Everything (IoE) [5]. As the trend gradually grows into its maturity, the concept of Servitization continues to grow in popularity and expansion, resulting in a new phenomenon called Internet of Services (IoS) [6]. IoS refers to an enormous and complex service network system, comprising a large number of cross-domain, cross-network, cross-border services, connected by various types of interconnection relationships [7]. Note that the services here are general-purpose services, including not only Web services and network services, but also real-world physical services such as daily life services, medical and pension services, and technology services [8]. IoS aims to provide a ubiquitous network and intelligent service foundation for the state, society, and industry.

Due to the information overload [9] caused by the massive amount of services, it becomes paramount to provide accurate and personalized service recommendations to users in IoS for three significant reasons. From the user's point of view, service recommendation can help users find the services they are interested in or need, which can also enhance the quality of experience (QoE). For example, if a user is a sports fan, proactively feeding him with some sports-related services will give him a better QoE than asking him to search on his own. From the perspective of services, service recommendation can increase the exposure of services, and save them from becoming long-tail services even from unnecessary service demise [10], [11]. From the perspective of the overall IoS system, service recommendation may constantly drive the creation of value-added activities, and promote healthy development of the ecosystem by through the formation of service chains [12].

In order to provide accurate personalized service recommendation, leveraging various types of relationships may be useful. Let us consider a real-world user request to recommend a hotel service in a new city. Instead of merely recommending a hotel service based on its description, we can first search the user's social friend circle (the user-user relationship shown in Fig. 1) and identify a good friend living in the city. Then we can search the friend's service usage records (the user-service relationship shown in Fig. 1) and find her or his favorite hotel service. Afterwards, we can further search the service-service relationship shown in Fig. 1, to examine whether there are restaurant services to which the people staying in that hotel usually go. As a result, we can not only recommend a trustworthy hotel service, but also reasonably recommend a restaurant service. This motivating example shows that utilization of various types of relationships can enhance service recommendation and provide users better service experiences.

As shown in this simple example illustrated in Fig. 1, IoS naturally exhibits complex network structural characteristics. The services and other subjects in IoS form a variety of relationship graphs, through various associative relationships.

Manuscript received 20 August 2022; revised 30 November 2022; accepted 23 January 2023. Date of publication 26 January 2023; date of current version 9 October 2023. This research has been partially supported by the National Natural Science Foundation of China (No.62173199). The associate editor coordinating the review of this article and approving it for publication was Y. Diao. *(Corresponding author: Yushun Fan.)*

Zhixuan Jia and Yushun Fan are with the Beijing National Research Center for Information Science and Technology, Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: jzx21@mails.tsinghua.edu.cn; fanyus@tsinghua.edu.cn).

Jia Zhang is with the Department of Computer Science, Southern Methodist University, Dallas, TX 75205 USA (e-mail: jiazhang@smu.edu).

Digital Object Identifier 10.1109/TNSM.2023.3239847

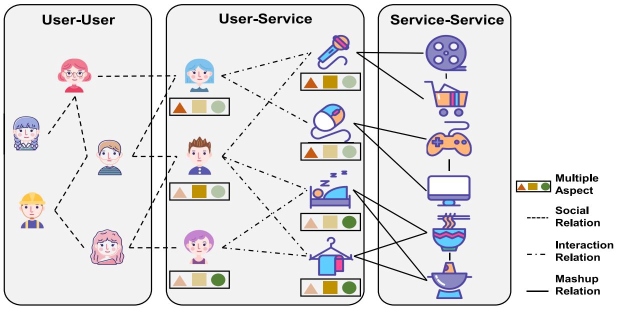


Fig. 1. An example of multiple relationship graphs and multiple aspects supporting service recommendation in IoS: a user-service interaction graph, a user-user social graph and a service-service mashup graph. Different evaluations reflect users' preferences (and services' characteristics) from multiple aspects implicitly.

To solve the motivating example, Fig. 1 shows an example of three relationship graphs carried in IoS. **(i) User-Service Interaction Graph**. The nodes in this graph represent users and services, and the edges represent the interaction relations (e.g., invocations or evaluations) between the users and the services. For example, when a user gives a rating to a hotel service he has used, an edge will be added between the user node and the service node, labeled by the rating record [13]. **(ii) User-User Social Graph**. The nodes in this graph denote the users, and the edges denote the social relations (e.g., trust or friendship) between the users. A common example is that when a user makes a friend with another user, an edge will be added between the two user nodes representing a mutual trust relationship with each other [14]. **(iii) Service-Service Mashup Graph**. To satisfy complex user demands, multiple services oftentimes collaborate in the format of a mashup (i.e., service composition) as a comprehensive service solution. As time goes by, some services gradually form a mashup relationship; that is, these services are more likely to be used together in a mashup [15]. Illustratively, the nodes in this graph indicate the services. An edge indicates that the two services at its ends were used in a mashup, which is shown as the label.

However, for service recommendation in IoS, it is not a trivial task to exploit its inherent relationships due to two significant challenges. The first challenge is the extremely sparse historical interaction data in IoS [16]. IoS is a big data environment, in which the average number of services used by users is much smaller than the existing number of services [12]. As a result, there is a great sparsity in users' historical behavior data. Coarsely using such sparse interaction data makes it difficult to fully understand user preferences and service characteristics, which in turn leads to a dramatic decrease in the accuracy of service recommendation [17]. The second challenge is how to reasonably utilize various relationship graph data together in IoS. The design of a service recommendation method for IoS shall consider and leverage diverse relationship graphs in IoS. Unlike regular Euclidean data such as sequence and image, the data of relationship graphs in IoS is non-Euclidean data. Traditional statistical learning methods, as well as classical deep learning techniques (e.g., convolutional neural networks and recurrent neural networks), have the limitation that they cannot be directly applied to handle large amount of different non-Euclidean data in IoS simultaneously [14], [18], [19].

To address the aforementioned two challenges, in this work, we propose **MGMASR**, a **M**ulti-**G**raph and **M**ulti-**A**spect neural network-powered method for **S**ervice **R**ecommendation in IoS.

Specifically, MGMASR tackles the first challenge of data sparsity in two steps. The first step is to use tailored strategies to explicitly enrich various relationship graphs in IoS. We first gather users' social information to enrich the User-User Social Graph. By bringing in side information like users' various kinds of social relationships (e.g., trust relationship or friend relationship), we can mitigate the adverse effects caused by the severely sparse users' historical behavior data [20]. We introduce a method to enrich the Service-Service Mashup Graph, by discovering and integrating multiple service pairs with potential mashup relationship based on the interaction data. Thus, we design a metric for calculating the **M**ashup **R**elevance (**MR**) between services. It can measure the likelihood of any two services falling into a service mashup, without collecting detailed statistics about mashups. The second step is to exploit the interaction data at a fine-grained level. As shown in Fig. 1, user preference is a composite of preferences from multiple aspects, such as response time, popularity, reliability, cost-effectiveness, and so on [21]. Different users may bear different preferences. For example, compared to the popularity of a service, one user may be more concerned about the reliability of the service, while some other users may do the opposite. Similarly, service characteristic can also be seen as a collection of characteristics from multiple aspects [22]. Different services exhibit different characteristics once they are released by service providers. For instance, some services may emphasize on cost-effectiveness, yet some other services may emphasize on quick response. To be concrete, we develop a novel multi-aspect analysis method for capturing fine-grained implicit user preference and service characteristic in user's historical evaluations over services from multiple aspects, by leveraging the fully connected layers [23] and a multi-aspect attention mechanism. Facing the second challenge of learning from comprehensive relationship graph data in IoS, we choose to apply a state-of-the-art deep learning technique, graph neural networks (GNNs) [24], which have demonstrated powerful processing capabilities for these different non-Euclidean data [25]. Furthermore, we apply the interaction/social/mashup attention mechanisms to distinguish the messages of neighbor nodes to jointly extract more valuable knowledge from different relationship graphs, respectively.

Comparing the representative and competitive baselines, extensive experiments over two real-world IoS datasets CIS and EPS have proved the effectiveness of MGMASR. MGMASR improves over the strongest baseline w.r.t MAE (Mean Absolute Error) by 0.83%, 1.22%; RMSE (Root Mean Squared Error) by 1.26%, 3.57% on CIS and EPS. We also perform the detailed main components study, the hyperparameter analysis and the computation cost experiment. Experiments have shown that MGMASR can be employed in IoS to provide accurate service recommendation. It can potentially serve as one of the technologies that drive the service economy of IoS.

The major contributions of our work can be summarized in three-fold.

- We present MGMASR, a multi-graph and multi-aspect neural network-powered service recommendation method, which can provide personalized service recommendation to users in IoS, by accurately predicting how users may rate services.
- We develop a technique to jointly learn multiple relationship graphs in IoS based on user's historical evaluations and social information, by leveraging GNNs and the interaction/social/mashup attention mechanisms.
- We take into account user preferences and service characteristics from multiple aspects at a fine-grained level based on user's historical evaluations, by deploying a multi-aspect attention mechanism and fully connected layers.

The remainder of the article is organized as follows. Section II compares with related work. Section III formally defines the problem. Section IV introduces our proposed MGMASR. Section V explains in detail how to train and learn relevant parameters. Section VI presents the experimental results together with analysis. Finally, Section VII concludes the article.

II. Related Work

Our work mainly touches on three research areas: Internet of Services, service recommendation, and deep learning for recommender system. In this section, we review representative related work in the three areas and distinguish our work from them.

A. Internet of Services

Existing research related to IoS is mainly focused on three topics: network architecture, optimal design of service system, and operational optimization and dynamic reconfiguration of service system.

Regarding the network architecture, researchers' main focus is on modeling, analysis, and evolution. In terms of basic theory, the openness, dynamicity and self-adaptability of service system have been studied, and the behavioral characteristics and dynamic evolutionary properties of service network have been analyzed [26], [27], [28]. In terms of architecture, researchers have examined how service individuals aggregate to form complex networks, and the impact of individual behaviors on the overall service network [12], [29], [30]. In terms of key technology, researchers have focused on service aggregation methods, service resource combination and scheduling [31], [32], [33]. As for the optimization design, the model-driven service methodology, the domain analysis-based service methodology, and the semantics-driven service methodology have been proposed. For example, for the purpose of service composition/aggregation and "Demand-Service" matching, researchers have tried to leverage domain prior knowledge, supplemented by intelligent algorithms [8], [34], [35]. For the operational optimization and dynamic reconfiguration of IoS, researchers have studied dynamic selection, self-adaptation and self-evolution of services in a static or mobile environment [36], [37].

However, few studies have examined how to better address the problem of information overload that exists in IoS. In contrast, our work aims to design a service recommendation method in the application scenario of IoS to tackle this problem.

B. Service Recommendations

A common service recommendation task takes the form of predicting whether a service will be invoked by a service developer/user, or predicting a future quality state of a service.

Solution techniques generally include matrix factorization, factorization machine, deep learning and so on. Fletcher [38] uses regularized matrix factorization to make personalized Web API recommendations. It considers both explicit and implicit user preferences. Cao et al. [39] employ a factorization machine based on the attention mechanism to solve the recommendation problem of Web APIs. Taking consideration of dynamic characteristics of both users and quality of service, Wu et al. [40] use recurrent neural network paired with matrix decomposition for time-aware service recommendation. They aim to recommend services based on QoS values, which is an attempt to recommend services with the highest QoS values for users. Wei et al. [14] use GNNs to capture high-order social information about users. So the model can refine the learning of representations of users and make more accurate service recommendation under a personalized ranking task. Yan et al. [41] employ the descriptions of services and GNNs to learn the bilateral information toward service recommendation. They use a domain-level attention unit to learn the similarity between user needs and related service domains. Besides, They design a graph convolution network on the service-composition graph to import the structured information into service embeddings.

Unlike existing studies, our work is based on a rating prediction task to provide accurate and desirable service recommendations to users in IoS. For the solution techniques, we mainly leverage GNNs and the attention mechanism to learn the representations of users and services via multiple relationship graphs in IoS and multiple aspects of user-service interactions.

C. Deep Learning for Recommender System

With the advancement of big data and artificial intelligence techniques, increasingly more deep learning methods have been applied in the scenario of recommender systems [42].

Here are some examples. NCF [43] is a representative neural network-based collaborative filtering framework. It can replace the traditional inner product operation with a multilayer perceptron. Its emergence has successfully demonstrated that models applying deep learning techniques can yield better recommendation performance. In recent years, GNNs have made great achievements in deep learning based recommender systems. For example, LGNN [44] simplifies feature transformation and nonlinear activation in GNNs based recommendation model for the task of personalized ranking. It advocates learning user and item embeddings by propagating them linearly on the user-item interaction graph. This makes it computationally efficient along with improved performance. Simultaneously, more and more models are considering the introduction of side information to enhance performance. In terms of introducing users' social information, DGNN [45] investigates user-to-user relationship, item-to-item relationship and user-to-item relationship for deep latent representation. It also considers social effects to be dynamic and multifaceted. Further, MGNN [46] uses the GNNs and the attention mechanism to learn the representations of users and items with users' opinions. It further employs cosine-based similarity to complete the construction of an item-item graph and extracts knowledge from three different graphs. However, both of them neglect to mine the most critical user interaction data at a fine-grained level from multiple aspects.

In contrast, our work not only considers the use of multiple relationship graphs, but also provides a fine-grained multi-aspect analysis of user interaction data. Therefore, our work can also be well-generalized to item recommendation.

III. Definitions And Problem Formulation

In this section, we will introduce the definitions, and describe the problem that we plan to solve in our article.

Definition 1 (Internet of Services, IoS): In this article, IoS is represented as a 5-tuple set $IoS = \{\mathcal{U}, \mathcal{S}, \mathbf{SC}, \mathbf{SM}, \mathbf{R}\}$ in the context of service recommendation. $\mathcal{U} = \{u_1, u_2, u_3, \ldots, u_n\}$ is the set of users, where n is the number of users in IoS. $\mathcal{S} = \{s_1, s_2, s_3, \ldots, s_m\}$ is the set of services, where m is the number of services in IoS. $\mathbf{SC} \in \mathbb{R}^{n\times n}$ denotes a user-user social graph, where $sc_{ij} = 1$ if user u_i has a directly connected social relationship with user u_j, and $sc_{ij} = 0$ if otherwise. $\mathbf{SM} \in \mathbb{R}^{m\times m}$ represents a service-service mashup graph, where $sm_{ij} = 1$ if service s_i has a direct relation to service s_j (e.g., they collaborate in a mashup), and 0 if otherwise. $\mathbf{R}$ is a $n \times m$ service usage rating matrix, also regarded as a user-service interaction graph, where $r_{ij} \in \{1, 2, 3, 4, 5\}$ when user u_i rates service s_j based on his/her experience, and $r_{ij} = 0$ means unknown service rating between user u_i and service s_j. The higher the rating, the more satisfied user u_i is with service s_j.

Definition 2 (Mashup Relevance, MR): $\mathrm{MR}(\cdot,\cdot)$ can calculate the mashup relevance of any two services in IoS to join a mashup.

$$N(s,s') = \{s, s' \in \mathcal{S} \mid \exists u \in \mathcal{U} : r_{us} = r_{us'} \geq r_t\} \tag{1}$$

$$\mathrm{MR}(s,s') = \frac{P(s,s')}{P(s)P(s')} \approx \frac{N(s,s')/N(\mathcal{S})}{(N(s)/N(\mathcal{S}))\cdot(N(s')/N(\mathcal{S}))} \tag{2}$$

where $s \neq s'$, r_t denotes rating threshold. $N(s,s')$ indicates the number of times users in IoS rate both service s and service s' with a high rating. $N(s) = \sum_{s'} N(s,s')$ and $N(s') = \sum_{s} N(s,s')$. $N(\mathcal{S})$ is the total number of all high-rated service pairs in the IoS. $P(s,s')$ represents the ratio of the number of times the service s and service s' are jointly high-rated to the number of all high-rated service pairs in the IoS. $\mathrm{MR}(s,s')$ represents the possibility of linkage between a pair of services s and s' in IoS to join a mashup.
def:

Based on the above definitions, we can define our target problem.

Problem (Service recommendation in IoS): Given a user-service interaction graph $\mathbf{R}$ and a user-user social graph $\mathbf{ST}$ in IoS, the problem is to predict the unknown links (ratings) in graph $\mathbf{R}$. Services with higher predicted ratings are more likely to be recommended to users by MGMASR.

More details will be provided in the following sections. The mathematical notations used in this article are summarized in Table I.

IV. MGMASR

In this section, we will elaborate on MGMASR. We will first describe the overall architecture of MGMASR, and then explain in detail its three integral learning units.

A. Overview of MGMASR

As shown in Fig. 2, MGMASR comprises three main learning units: user representation, service representation, and rating prediction. In the upper portion of Fig. 2, the user representation unit aims to learn users' embedding vectors from the user-service interaction graph and the user-user social graph. The learned vectors will serve as the users' representations to reflect their preferences. In more detail, this unit contains two sub-units, a user interaction representation sub-unit and a user social representation sub-unit. The former sub-unit learns user interaction representations from the user-service interaction graph at multiple aspects; and the latter sub-unit learns user social representations from the user-user social graph. The final users' representations are obtained by concatenating the above two representations. In lower portion of Fig. 2, the service representation unit intends to learn the services' embedding vectors from the user-service interaction graph and the service-service mashup graph. The obtained embedding vectors will be used as service representations to express the services' characteristics. Two sub-units collaborate in this unit to portray the characteristics of the services from multiple aspects: a service interaction representation sub-unit and a service mashup representation sub-unit. The final service representations are derived by integrating the two types of information extracted by the two sub-units. In the right portion of Fig. 2, the rating prediction unit converts the representations of a user and a service into a rating, based on which the rating prediction task of service recommendation can be accomplished.

Without losing generality, we base the ID numbers of the users and the services to obtain their initial embedding vectors via the user/service embedding layer. The users' initial embedding vectors are denoted as $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n] \in \mathbb{R}^{d\times n}$, and the initial embedding vectors for the services are denoted as $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_m] \in \mathbb{R}^{d\times m}$ by the user/service embedding layer, where d is the size of the embedding vector. For $\mathbf{FC}(\cdot)$, they all have a similar two-layer fully connected neural

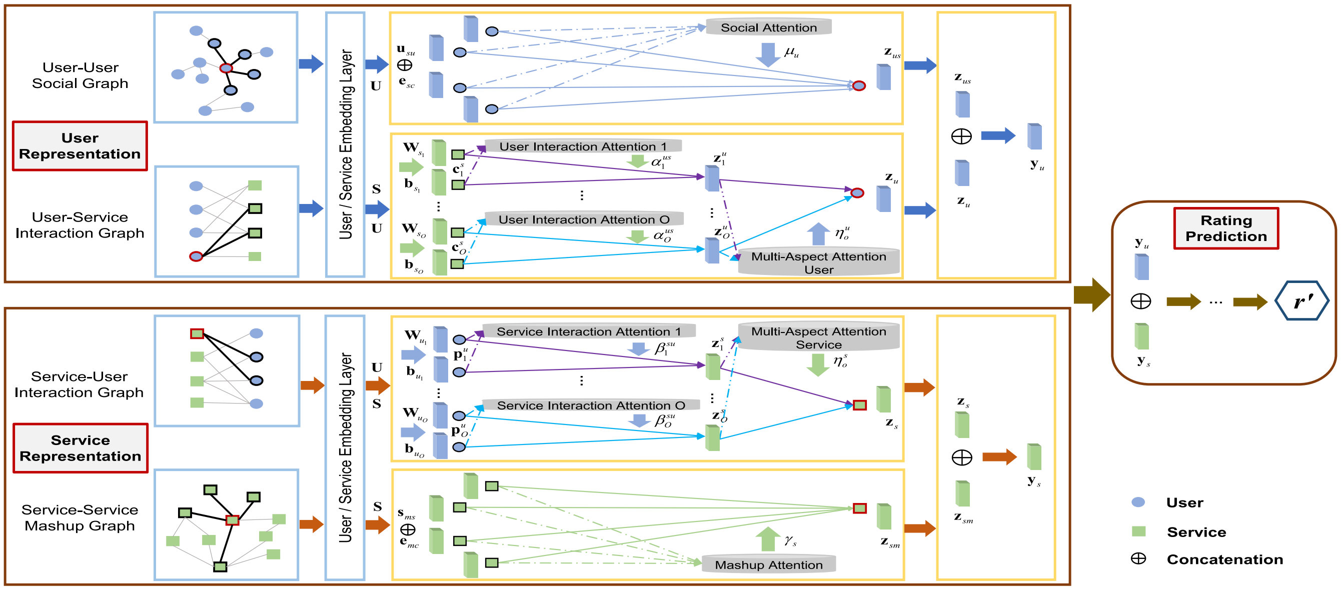

User-User Social Graph
User Representation
User-Service Interaction Graph
User / Service Embedding Layer
Social Attention
User Interaction Attention 1
User Interaction Attention O
Multi-Aspect Attention User
Service-User Interaction Graph
Service Representation
Service-Service Mashup Graph
Service Interaction Attention 1
Service Interaction Attention O
Multi-Aspect Attention Service
Mashup Attention
Rating Prediction
User
Service
Concatenation

Fig. 2. The architecture of MGMASR. It comprises three major learning units: user representation, service representation, and rating prediction.

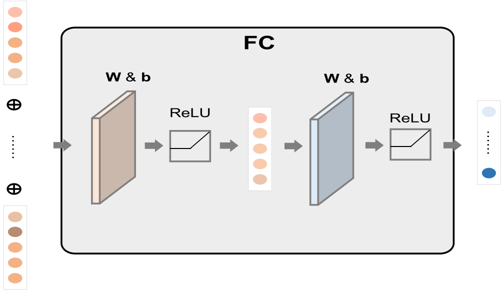


Fig. 3. A visualization example of $\mathbf{FC}(\cdot)$. $\mathbf{W}$ and $\mathbf{b}$ are different trainable weight and bias for each $\mathbf{FC}(\cdot)$. The linear rectification function (ReLU) is set as the nonlinear activation function in $\mathbf{FC}(\cdot)$ and the internal dimension of $\mathbf{FC}(\cdot)$ is d.

network structure but different initialization parameters [47]. A visualization example of $\mathbf{FC}(\cdot)$ can be seen in Fig. 3.

B. User Representation

The goal of the user representation unit is to learn an embedding vector for each user, which can reflect the user's preference. Unlike typical single graph-based learning tasks, here we aim to learn user representation from different types of relationship graphs where users are involved. As shown in Fig. 2, two types of relationship graphs, i.e., the user-service interaction graph and the user-user social graph, may contribute information regarding users. Our strategy is to learn user representation from each relationship graph separately and then to integrate them. That is to obtain the user interaction representation and the user social representation, respectively.

1) User Interaction Representation: The user-service interaction graph is mined to learn the representation of user interaction information. Based on the reality that a user's preferences usually exhibit as one manifestation of multiple aspects, we design a neural network framework to conduct the learning. First, for a target user $u \in \mathcal{U}$ and a service $s \in \mathcal{S}$, we make use of two distinct single-layer perceptrons with trainable parameters $\mathbf{W}_{u_o}$, $\mathbf{W}_{s_o}$, $\mathbf{b}_{u_o}$ and $\mathbf{b}_{s_o}$ to analyze the multi-aspect preferences of the user and the multi-aspect characteristics of the service:

$$\mathbf{p}_o^u = \mathbf{W}_{u_o}\mathbf{u}_u + \mathbf{b}_{u_o}, \tag{3}$$

$$\mathbf{c}_o^s = \mathbf{W}_{s_o}\mathbf{s}_s + \mathbf{b}_{s_o}, \tag{4}$$

where o refers to one of the multiple aspects, $\mathbf{p}_o^u$ is user u's interaction representation under aspect o. Correspondingly, $\mathbf{c}_o^s$ is the service s's interaction representation under aspect o.

We then refine the target user u's interaction representation by recursively aggregating the interaction messages passed by the services, with which the user has rating interactions in the user-service interaction graph. At the same time, we consider the need to differentiate the importance of the interaction messages passed by different neighboring services. Therefore, we adopt GNNs and the user interaction attention mechanism to design the learning process. Based on the current aspect o, considering the impact of the user u's evaluation to the service s, the update rules for user u's interaction representation can be formulated as follows:

Specifically, we calculate the weight coefficients α_o^{us} of the user interaction attention mechanism under aspect o in the message-passing process as follows:

$$\alpha_o^{us} = \frac{\exp(\mathbf{FC}_1(\mathbf{p}_o^u \| \mathbf{c}_o^s))}{\sum_{s \in S_u} \exp(\mathbf{FC}_1(\mathbf{p}_o^u \| \mathbf{c}_o^s))}, \tag{5}$$

where $\exp(\cdot)$ is the exponential function. S_u represents the sampled services that have interacted with user u and $\|$ denotes the concatenate operation.

Afterwards, we aggregate the messages passed by all the services in the collection S_u to participate in the computational process. In this way, we can learn user u's interaction

TABLE I
NOTATIONS

Symbols	Definitions/Descriptions
$\mathcal{U}$	the user set
$\mathcal{S}$	the service set
SC	the user-user social graph
SM	the service-service mashup graph
R	the user-service interaction graph
n	the number of users
m	the number of services
r	the rating ∈ {1,2,3,4,5}
sc	the social relationship
sm	the mashup relationship
su	the social user
ms	the service in a same mashup
U	the users' initial embedding vectors
S	the services' initial embedding vectors
O	the number of multiple aspects
$\mathbf{p}_o^u$	the user u's interaction representation under the aspect o
$\mathbf{c}_o^s$	the service s's interaction representation under the aspect o
$\mathbf{W}$, $\mathbf{b}$	the trainable parameters
α_o^{us}	the weight coefficients in learning the user u's interaction representation
S_u	the sampled services that have interacted with the user u
$\mathbf{z}_o^u$	the user u's interaction representation under aspect o
$\mathbf{g}_O^u$	the concatenation of the user u's interaction representation under multiple aspects
η_o^u	the attention weight coefficient of the user u's interaction representation under the aspect o
$\mathbf{z}_u$	the user u's interaction representation
$\mathbf{e}_{sc}$	the initial social situation representation of the user u
U_u	the set of sampled users who have direct social relationships with user u
μ_u	the attention weight of social information delivered by U_u to the user u
$\mathbf{z}_{us}$	the user u's social representation
$\mathbf{y}_u$	the user u's representation
β_o^{su}	the weight coefficients in learning the service s's interaction representation
U_s	the sampled users that have interacted with the service s
$\mathbf{z}_o^s$	the service s's interaction representation under aspect o
$\mathbf{g}_O^s$	the concatenation of the service s's interaction representation under multiple aspects
η_o^s	the attention weight coefficient of the service s's interaction representation under the aspect o
$\mathbf{z}_s$	the service s's interaction representation
$\mathbf{e}_{mc}$	the initial mashup situation representation of the service s
S_s	the set of sampled services who have direct mashup relationships with service s
γ_s	the attention weight of mashup information delivered by S_s to the service s
$\mathbf{z}_{sm}$	the service s's mashup representation
$\mathbf{y}_s$	the service s's representation

representation $\mathbf{z}_o^u$ under aspect o as follows:

$$\mathbf{z}_o^u = \mathbf{FC}_2\left[\left(\sum_{s\in S_u} \alpha_o^{us} \cdot \mathbf{c}_o^s\right) \| \mathbf{p}_o^u\right]. \tag{6}$$

Bearing in mind that the aspects are multiple with a number of O, we perform the above computational process separately to obtain user u's corresponding interaction representations under different aspects. We then concatenate $\{\mathbf{z}_1^u, \mathbf{z}_2^u, \ldots, \mathbf{z}_O^u\}$ and pass them through fully connected layers as:

$$\mathbf{g}_O^u = \mathbf{FC}_3(\mathbf{z}_1^u \| \mathbf{z}_2^u \| \cdots \| \mathbf{z}_O^u). \tag{7}$$

Considering that different aspects have different strengths in the final user interaction representation for user u, we normalize $\mathbf{g}_O^u$ for the attention weight coefficient of the aspect o in the user multi-aspect attention mechanism:

$$\eta_o^u = \frac{\exp(g_o^u)}{\sum_{i=1}^{O} \exp(g_i^u)}, \tag{8}$$

where η_o^u is the attention weight coefficient of the aspect o in this attention mechanism.

Finally, with the learned parameters, we can obtain the final interaction representation $\mathbf{z}_u$ for user u:

$$\mathbf{z}_u = \sum_{o=1}^{O} \eta_o^u \mathbf{z}_o^u. \tag{9}$$

2) User Social Representation: In order to learn users' social information representations, we mine over the user-user social graph. First, for the target user u, we concatenate the representation of his social user $su \in \mathcal{U}$, who is adjacent to u in the user-user social graph, with the social situation initial representation $\mathbf{e}_{sc}$ as below. Note that according to ***Definition 1***, the social situation here is a binary parameter, either 0 (without direct social relation) or 1 (with direct social relation).

$$\mathbf{x}_{usc} = \mathbf{FC}_4(\mathbf{u}_{su} \| \mathbf{e}_{sc}), \tag{10}$$

where $\mathbf{u}_{su}$ is the initial representation of the social user su, $\mathbf{e}_{sc}$ is the initial social situation representation of user u.

The obtained result is further combined with the representation of the target user u as:

$$\mu_u^* = \mathbf{FC}_5(\mathbf{x}_{usc} \| \mathbf{u}_u). \tag{11}$$

Generally speaking, there are differences in the social influences of different social users on the target user u. Thus, we introduce a social attention mechanism to differentiate and aggregate social information as:

$$\mu_u = \frac{\exp(\mu_u^*)}{\sum_{u\in U_u} \exp(\mu_u^*)}, \tag{12}$$

where U_u is the set of sampled users who have direct social relationships with user u. μ_u represents the attention weight of social information delivered by each user with direct social connections to the target user u.

Finally, we can obtain the social information representation $\mathbf{z}_{us}$ of the target user u:

$$\mathbf{z}_{us} = \mathbf{W}_s\left(\sum_{u\in U_u} \mu_u \cdot \mathbf{x}_{usc}\right) + \mathbf{b}_s, \tag{13}$$

where $\mathbf{W}_s$ and $\mathbf{b}_s$ are the learnable weight and bias.

3) User Final Representation: In order to obtain a user representation that can well represent the preferences of the target user u, we integrate her/his interaction representation with her/his social representation. Here we use $\mathbf{FC}_6(\cdot)$ to convert the results into a standard user embedding vector as the final user representation $\mathbf{y}_u$:

$$\mathbf{y}_u = \mathbf{FC}_6(\mathbf{z}_u \| \mathbf{z}_{us}). \tag{14}$$

C. Service Representation

The goal of the service representation unit is to learn an embedding vector for each service, to represent its comprehensive characteristics. As described in earlier section and also shown in Fig. 2, services not only have interactive connectivity with users, but also have combined connectivity with other services in a mashup. Therefore, we design a method to learn such service representation from two types of relationship graphs, i.e., the user-service interaction graph and the service-service mashup graph. Afterwards, separately learned embeddings will be integrated to generate the final embeddings for the corresponding services.

1) Service Interaction Representation: Similar to the learning process for the user interaction representation, we adopt the same approach to examine the characteristics of services under different aspects. Given a target service *s*, we introduce a service interaction attention mechanism to aggregate the interaction messages passed by neighboring users, from the service-user interaction graph. At first, the impact of the aspect *o* for the target service *s* which is given by user *u* can be defined as:

$$\beta_o^{su} = \frac{\exp(\mathbf{FC}_7(\mathbf{c}_o^s \| \mathbf{p}_o^u))}{\sum_{u \in U_s} \exp(\mathbf{FC}_7(\mathbf{c}_o^s \| \mathbf{p}_o^u))}. \tag{15}$$

where $\mathbf{c}_o^s$ and $\mathbf{p}_o^u$ represent service *s*'s and user *u*'s interaction representation under aspect *o* based on E.q. (3) and E.q. (4), respectively.

Then we learn the service interaction representation $\mathbf{z}_o^s$ of the target service *s* under the current aspect *o* as:

$$\mathbf{z}_o^s = \mathbf{FC}_8 \left[\left(\sum_{u \in U_s} \beta_o^{su} \cdot \mathbf{p}_o^u \right) \| \mathbf{c}_o^s \right], \tag{16}$$

where U_s represents the set of sampled users who have interaction with the target service *s*.

We further integrate the service interaction representations of service *s* obtained, by performing calculations under multiple aspects:

$$\mathbf{g}_O^s = \mathbf{FC}_9(\mathbf{z}_1^s \| \mathbf{z}_2^s \| \cdots \| \mathbf{z}_O^s), \tag{17}$$

where $\mathbf{g}_O^s$ is the result obtained by a concatenation operation.

Likewise, considering that the weights of different service characteristics in the service are different, we deploy a service multi-aspect attention mechanism as:

$$\eta_o^s = \frac{\exp(g_o^s)}{\sum_{i=1}^{O} \exp(g_i^s)}. \tag{18}$$

Finally, we can derive the service interaction representation $\mathbf{z}_s$ of the target service *s*:

$$\mathbf{z}_s = \sum_{o=1}^{O} \eta_o^s \mathbf{z}_o^s. \tag{19}$$

2) Service Mashup Representation: As explained in the introduction section, services in IoS are typically not independent from each other; instead, they rather have a composition relationship when they are used in a mashup. Based on ***Definition 2***, we intend to use the devised $\mathrm{MR}(\cdot,\cdot)$ to measure the mashup relevance between any two services. In particular, in order to avoid some unconvincing situation regarding a low sample size, for example only one user gives high scores to two services that co-exist in a mashup only once, we establish a co-occurrence threshold *H*. We then arrange the results of $\mathrm{MR}(\cdot,\cdot)$ in descending order according to their values, and select the top-K service pairs to compose the service-service mashup graph. In the first step, for the target service s, the services adjacent to it on the service mashup graph are combined with the mashup situation:

$$\mathbf{x}_{smc} = \mathbf{FC}_{10}(\mathbf{s}_{ms} \| \mathbf{e}_{mc}), \tag{20}$$

where $\mathbf{s}_{ms}$ is the initialized representation of service $ms \in \mathcal{S}$ connected to service *s* directly in the service-service mashup graph, and $\mathbf{e}_{mc}$ is the initialized mashup situation representation. According to ***Definition 1***, there are two mashup situations here, which are either presence (1) or absence (0).

Since the strengths of the correlation between different services to become mashups are different, we apply a mashup attention mechanism in the aggregation process of learning the service mashup representation. Afterwards, we combine $\mathbf{x}_{smc}$ with the initial representation of the target service *s* itself:

$$\gamma_s^* = \mathbf{FC}_{11}(\mathbf{x}_{smc} \| \mathbf{s}_s). \tag{21}$$

In the same way, we again use the softmax function for normalization as:

$$\gamma_s = \frac{\exp(\gamma_s^*)}{\sum_{s \in S_s} \exp(\gamma_s^*)}. \tag{22}$$

At last, we can obtain the service mashup information representation $\mathbf{z}_{sm}$ of the target service *s* as:

$$\mathbf{z}_{sm} = \mathbf{W}_m \left(\sum_{s \in S_s} \gamma_s \cdot \mathbf{x}_{smc} \right) + \mathbf{b}_m, \tag{23}$$

where S_s is the service set where each comprising service has a service mashup relationship with the target service *s*. $\mathbf{W}_m$ and $\mathbf{b}_m$ are the weight and bias of a fully connected layer, respectively.

3) Service Final Representation: After gaining the above service interaction representation $\mathbf{z}_s$ and the service mashup representation $\mathbf{z}_{sm}$, we employ fully connected layers to process the final integration. Therefore, the final representation $\mathbf{y}_s$ of service *s* is defined as:

$$\mathbf{y}_s = \mathbf{FC}_{12}(\mathbf{z}_s \| \mathbf{z}_{sm}). \tag{24}$$

D. Rating Prediction

Based on the learned representations of user *u* and service *s*, the rating prediction unit aims to recommend services to users by predicting user *u*' rating over service *s*. Given the above obtained representations $\mathbf{y}_u$ and $\mathbf{y}_s$ of user *u* and service *s*, the final rating is predicted as:

$$\mathbf{y} = \mathbf{FC}_{13}(\mathbf{y}_u \| \mathbf{y}_s), \tag{25}$$

$$r' = \mathbf{W}_y \mathbf{y} + \mathbf{b}_y, \tag{26}$$

where r' is the final rating predicted by MGMASR given to the service *s* by the user *u*. $\mathbf{W}_y$ and $\mathbf{b}_y$ are the weight and bias of a last prediction fully connected layer.

V. Parameters Learning

In this section, we discuss how to learn parameters for MGMASR, which includes the definition of loss function and our design of sampling strategy.

A. Loss Function

The main objective of the task of rating prediction is to make the predicted results as consistent or similar as possible to the ground truth. To prevent the phenomenon of overfitting, in particular, we add an L2 regularization term. So we design the loss function as:

$$\mathcal{L} = \frac{\sum_{(u,s)\in\Theta}\left(r - r'\right)^2}{2|\Theta|} + \lambda\|\Phi\|_2, \qquad (27)$$

where Θ is the set of observed ratings, and r is the true rating of service s given by user u. Φ is the set of all trainable parameters in MGMASR, and λ is a hyperparameter for controlling the strength of L2 regularization.

B. Sampling Strategy

When computing the information aggregation for a target node in a relationship graph, it is usually impractical to consider all the neighboring nodes of this target node. This is because, on the one hand, it may cause the overfitting phenomenon; and on the other hand, it will impose an unnecessary computational burden in a large-scale graph.

Hence, we design different sampling strategies for different relationship graphs mined by MGMASR. In the user-user social graph and the service-service mashup graph, we adopt a random sampling strategy to sample neighboring nodes of the target user/service. The sampling threshold is set as the average degree of the nodes in each of these two graphs, respectively. While in the user-service interaction graph, we construct a sampling strategy with non-uniform weights. Intuitively, our hypothesis is that the higher the rating of a service by a user, the more the service indicates the user's preference to a certain extent. Thus, we first rank the users/services that have interactions with services/users in descending order of their ratings. Then we utilize the softmax function to calculate the probability that a node with each rating as a connection will be selected to participate in the information aggregation. For example, a user/service with a rating of 4 has a 23.41% probability of being sampled as follows. Similarly, we set the sampling threshold to the average number of users/services connected to the service/user, respectively.

$$P_r = \mathrm{Softmax}(r_{us}), \qquad (28)$$

where $\mathrm{Softmax}(\cdot)$ is the softmax function, P_r is the probability of a service/user being sampled for the target user/service with a rating of r_{us}.

VI. Experiments

In this section, we explicate a series of experiments conducted over MGMASR. Below we discuss experimental settings and present result analysis.

TABLE II
Statistics of Datasets

Dataset	CIS	EPS
# Users	7,368	21,651
# Services	105,110	291,777
# Reviews (Ratings)	283,057	901,793
# Social Relations (Trust)	111,741	341,319
Values of Reviews (Ratings)	{1,2,3,4,5}	{1,2,3,4,5}
Density of Reviews (Ratings)	0.037%	0.014%
Density of Social Relations (Trust)	0.206%	0.073%

A. Experimental Settings

1) Datasets: Both Ciao[1] and Epinions[2] are online social service platforms that allow users to write reviews and rate services they have used, and browse reviews of other services. They also offer an ability for users to follow users they trust, which forms a large number of social relations between users. Services covered on these sites include, but are not limited to, computer hardware and software services, personal finance services, newspapers and magazines services, family child-care services, travel and lodging services, health care services, education services, restaurant services, photofinishing services, and pet care services. Based on this heterogeneous nature, we can assume that these systems can be viewed as IoS.

We collected our datasets from the two above service platforms and named them CIS and EPS. For these datasets, we applied a similar preprocessing method mentioned in [48] to ensure that at least five service rating records exist for each user.

The statistics of our datasets after preprocessing are shown in Table II.

2) Baseline Methods: To validate the effectiveness of MGMASR, we compared the performance of MGMASR with the following eight representative and competitive baseline methods.

- *PMF [49]:* It is an optimization method based on the regularized matrix factorization (MF). It assumes that the feature matrices of both user and item obey Gaussian distribution.
- *SReg [50]:* It is designed with the concept of constraining the users to have different social influences from their respective friends.
- *SRec [51]:* It is a social recommendation algorithm based on feature sharing. In which, the user latent factors are learned by decomposing both the rating matrix and the social relationship matrix.
- *SMF [52]:* Its idea is to constrain that the behavioral characteristics of a user should be as similar as possible to the average preferences of the social neighbors to which this user is connected.

[1] https://www.ciao.co.uk/

[2] https://www.epinions.com/

- *NCF [43]:* It employs deep neural networks (DNNs) to model the features of users and items, and enables the model to have a nonlinear representation capability.
- *LGNN [44]:* It is a state-of-the-art recommendation model based on GNNs, which is simple yet powerful only by neighborhood aggregation and multi-layer propagation.
- *DGNN [45]:* It utilizes dual graph attention neural networks to improve the interpretability of the model for social influence, by taking into account the reality that the influence of a user's social friends on the user is dynamic.
- *MGNN [46]:* This model uses the GNNs and the attention mechanism on the user-user graph, the user-item graph, and the item-item graph based on the similarity between items, also considers incorporating the opinions to learn the representations of users and items.

3) Evaluation Metrics: In order to illustrate the performance of MGMASR and the baseline methods, we adopted two widely-used evaluation metrics **M**ean **A**bsolute **E**rror (**MAE**) and **R**oot **M**ean **S**quared **E**rror (**RMSE**) for the rating prediction task based on these datasets, which is a common practice in recommendation system [46], [53].

4) Hyperparameter Settings: In this section, we will detail the settings of hyperparameters used in MGMASR. For the datasets, when conducting each experiment, we randomly selected 65% of each dataset as the training set for learning the parameters in MGMASR, 15% as the validation set to tune the hyperparameters, and the remaining 20% as the test set for measuring the performance of MGMASR. For convincing results, we repeated each experiment 7 times independently and took the average as the final reported result.

Following the related work reported in the literature [44], [46], [53], with the help of grid search, the hyperparameter λ for the L2 regularization was searched in {0.00001, 0.0001, 0.001, 0.01, 0.1}. For the batch size, we tuned it in {128, 256, 512, 1024}. For the learning rate, we investigated it in {0.0001, 0.001, 0.005, 0.01, 0.05}. For the size of the embedding vector *d*, we tested six different values in {8, 16, 32, 64, 128, 256}. We also tested five different values in {1, 2, 3, 4, 5} as the number of multiple aspects *O* in our experiments. In particular, when constructing the service-service mashup graph, we set the rating threshold r_t for $\mathrm{MR}(\cdot,\cdot)$ as 4, the co-occurrence number *H* in the description of service mashup representation as 15, the value of *K* in the top-*K* service pairs was tested in the set of {15, 30, 50, 75, 100, 130}.

For all trainable parameters in MGMASR, we utilized a Gaussian distribution with a mean of 0 and a standard deviation of 0.1 for the random initialization of all these parameters. $\mathbf{FC}(\cdot)$ denotes a two-layer fully connected layers with the linear rectification function (ReLU) as the nonlinear activation function. For optimizing our designed loss function, we used the Adam optimization algorithm as an optimizer during these experiments. The reason why we selected it against other optimization algorithms is its simplicity and effectiveness [54]. Moreover, the dropout strategy [55] and the early stopping strategy [56] were also applied in the training process of MGMASR. The dropout rate was set as 0.5. We stopped a training if the RMSE on the validation set did not decrease for 20 consecutive epochs, and reported the results on the test set.

For all baselines, we carefully reproduced them based on their original papers and set their specific hyperparameters according to the reported optimal hyperparameters. And then we carefully tuned them to achieve optimal performance on CIS and EPS. Note that for NCF and LGNN, we adapted them for the rating prediction task. To be specific, for NCF and LGNN, we stitched the learned user representation and service representation together and then passed them through a $\mathbf{FC}(\cdot)$ to obtain the final prediction rating. And we replaced the loss function of them with the same loss function as MGMASR to complete the model training under the rating prediction task.

B. Results Analysis

The prediction performance results of MGMASR are shown in this section for comparison and analysis. Here we will first compare and analyze the performance of MGMASR with that of the baseline methods. Then, we will analyze the validity of each component of MGMASR. To illustrate the need for reasonable hyperparameters, we will discuss the impacts of hyperparameters in MGMASR.

1) Performance Comparison: In order to compare the performance of MGMASR with that of the baseline methods fairly, we chose to conduct experiments using the same datasets and the same division of the training set, the validation set and the test set. After the training and testing of these methods, their performance results are summarized in Table III. The underlined numbers are the best performance achieved by the baseline methods. The bolded numbers are the performance results of MGMASR. *Improvement* refers to the percentage improvement in the performance of MGMASR compared to the best performance of the baseline methods. By carefully comparing and analyzing the reported results of each of the above methods, we observed a total of four findings. Note that in practice, small improvement in RMSE or MAE terms can have a significant impact on the quality of the top-few recommendations [46], [57].

- The performance of classical baseline method PMF is always inferior to that of social recommendation baseline methods, like SReg, SRec, and SMF. Although these methods are all based on the concept of MF for construction, the SReg, SRec, and SMF consider the introduction of user social information, which further indicates that the utilization of user social information is useful for the study of recommendation in the context of rating prediction.
- The performance of deep learning baseline method NCF and LGNN is better than that of traditional baseline method PMF, which means that the application of deep learning can improve the performance of recommendation method to a certain extent, even without introducing side information like social information. The outperformance of LGNN over NCF reflects a more powerful representation capability of GNNs.
- The baseline methods of DGNN and MGNN, which consider the use of social information, outperform SReg,

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT METHODS

Method	CIS		EPS	
	MAE	RMSE	MAE	RMSE
PMF	0.9038±0.0028	1.2617±0.0024	0.9685±0.0023	1.5155±0.0022
SReg	0.8916±0.0026	1.1028±0.0022	0.9397±0.0029	1.2479±0.0011
SRec	0.8452±0.0019	1.0864±0.0017	0.9174±0.0026	1.2298±0.0020
SMF	0.8347±0.0015	1.0713±0.0020	0.9004±0.0021	1.2130±0.0027
NCF	0.7653±0.0017	1.0311±0.0011	0.8512±0.0015	1.1612±0.0012
LGNN	0.7468±0.0012	1.0010±0.0017	0.8367±0.0010	1.1133±0.0006
DGNN	0.7432±0.0010	1.0027±0.0009	0.8359±0.0013	1.1117±0.0010
MGNN	0.7427±0.0014	0.9941±0.0009	0.8361±0.0012	1.1034±0.0008
MGMASR	**0.7365**±0.0009	**0.9816**±0.0007	**0.8257**±0.0008	**1.0640**±0.0005
Improvement(%)	0.83%	1.26%	1.22%	3.57%

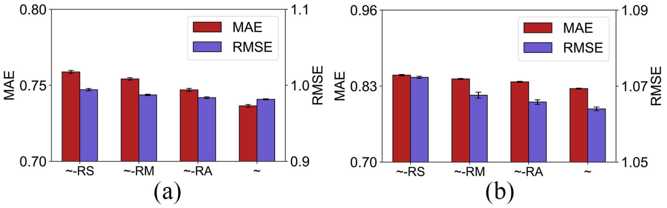
SRec, SMF, NCF and LGNN. They are also based on the concept of graph and deep learning (GNNs). It means that the performance of the recommendation method can be significantly improved by adopting deep learning, while processing social information between users.

- MGMASR outperforms all the baseline methods aforementioned. Compared to SRec, SReg, and SMF, MGMASR leverages the state-of-the-art deep learning techniques such as GNNs and the attention mechansism. Compared to NCF and LGNN, MGMASR considers the use of user social information. Compared with DGNN and MGNN, MGMASR not only constructs and mines the relationship information contained in multiple relationship graphs, but also considers the user preferences and service characteristics from multiple aspects reflected by the user's rating of the service. Meanwhile, the error deviation results illustrate that MGMASR owns an effective improvement and good stability. In short, the superior performance of MGMASR can be attributed to the ability to learn more robust and accurate representations of users and services.

In summary, the performance comparison experiments show that the side information (i.e., social information) can help to improve the performance of a service recommendation method, and deep learning techniques like GNNs help to learn representation capabilities. MGMASR can achieve the best performance compared with the baseline methods.

2) Impact of Main Components: In this section, we will analyze the impact of the main components on our proposed MGMASR. MGMASR consists of three main components: introducing user social information, mining service mashup information, and multi-aspect analysis of interaction. In order to further evaluate our proposed MGMASR, three variants of MGMASR were designed for each of its main components, and their differences are described below. The performance comparison between them is shown in Fig. 4.

- *MGMASR-RS:* This variant indicates that the learning part of the user's social representation is removed from the MGMASR. That is, the user's social information is not considered in the construction of MGMASR.

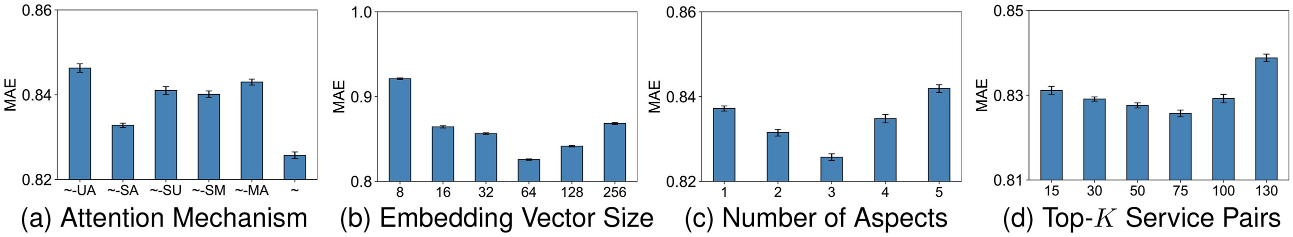


(a) (b)

Fig. 4. Performance comparison after removing the different main components of MGMASR.

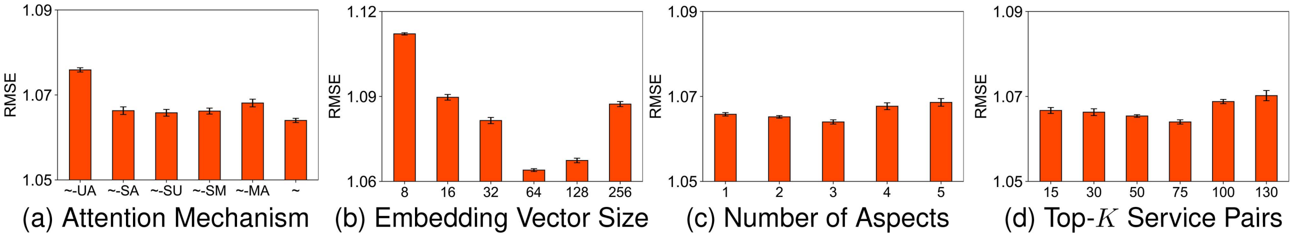
- *MGMASR-RM:* This variant represents that the learning part of the service mashup representation is removed from the MGMASR. That is, the service mashup information of the services is not considered in the construction of MGMASR.
- *MGMASR-RA:* This variant represents the deletion of the multi-aspect analysis about the user-service interaction in MGMASR, which also means that only a single aspect of user-service interaction is analyzed in MGMASR.

For simplicity, we denote MGMASR in Fig. 4 as $\sim$. As shown in Fig. 4, the performance of the MGMASR-RS variant illustrates that the user social information is helpful for the recommendation accuracy improvement of the service recommendation method. It also implies that a service recommendation method will be able to learn the user representation better, if the user social information is introduced. The performance of the MGMASR-RM variant illustrates that the service mashup information is useful for optimizing the service recommendation method. Similarly, it also illustrates that introducing the service mashup information can help to learn the service representation better. The performance of the MGMASR-RA variant is also inferior to that of MGMASR, which proves that the user-service interaction needs to be examined from various aspects. Therefore, it is meaningful to consider multiple aspects to analyze the user-service interaction. Besides, by examining the error bars, the error variance does not overlap in extreme between compared values, and we can consider that the improvement from introducing these types of information is valid.



(a) Attention Mechanism (b) Embedding Vector Size (c) Number of Aspects (d) Top-K Service Pairs

Fig. 5. Performance comparison for different hyperparameters on EPS dataset (MAE).



(a) Attention Mechanism (b) Embedding Vector Size (c) Number of Aspects (d) Top-K Service Pairs

Fig. 6. Performance comparison for different hyperparameters on EPS dataset (RMSE).

3) Impact of Attention Mechanism: As discussed in details in Section IV, five parts in MGMASR involve the use of the attention mechanism:

- User Interaction Attention: An attention mechanism in the interaction information aggregation part of the learning process for the user interaction representation on the user-service interaction graph.
- Service Interaction Attention: An attention mechanism in the interaction information aggregation part of the learning process for the service interaction representation on the user-service interaction graph.
- Social Attention: An attention mechanism in the social information aggregation part of the learning process for the user social representation on the user-user social graph.
- Mashup Attention: An attention mechanism in the mashup information aggregation part of the learning process for the service mashup representation on the service-service mashup graph.
- Multi-Aspect Attention: An attention mechanism in the multiple aspects analysis for aggregating multiple user preferences and multiple service characteristics.

In order to better understand the utility and impact of the attention mechanism applied in MGMASR, five variants of MGMASR were designed with respect to the five parts of the attention mechanism applied within MGMASR. Their differences are described below:

- *MGMASR-UA:* This variant represents the removal of the user interaction attention, and replaces it with a uniform weight method.
- *MGMASR-SA:* This variant represents the removal of the service interaction attention, and replaces it with a uniform weight method.
- *MGMASR-SU:* This variant represents the removal of social attention, and replaces it with a uniform weight method.
- *MGMASR-SM:* This variant represents the removal of mashup attention, and replaces it with a uniform weight method.
- *MGMASR-MA:* This variant represents the removal of the multi-aspect attention, and replaces it with a uniform weight method.

For simplicity, we also abbreviate MGMASR in Fig. 5(a) and Fig. 6(a) as $\sim$. The results of these tests on the EPS dataset are shown in Fig. 5(a) and Fig. 6(a). After the experimental study and examination of the results, we concluded with the following two findings. First, if the attention mechanism used in each part of MGMASR is removed, it will make the recommendation accuracy of MGMASR decrease. Second, it shows that it is necessary to distinguish the different importance of neighboring messages on the target nodes in the process of learning the representations of users and services, using various types of relationship graphs. Beyond that, by examining the error variation terms, the true improvement brought by employing different attention mechanisms in MGMASR can be verified.

4) Impact of Embedding Vector Size: We also conducted experiments and analyzed the size of the embedding vector in MGMASR. The size of the embedding vector was set to six different values of 8, 16, 32, 64, 128, and 256, respectively, as described in the hyperparameter settings. After conducting the experiments separately, the results on the EPS dataset are shown in Fig. 5(b) and Fig. 6(b).

It can be found that, in the initial stage, the performance of MGMASR increases with the increase of the embedding vector size, which indicates that the increase of the embedding vector size can better describe the information the embedding vector contains. Moreover, MGMASR can achieve the best performance when the size of the embedding vector is set to 64. However, the performance of MGMASR decreases as the size of the embedding vector continues to increase, and further increase leads to a further decrease. It means that an oversized embedding vector does not improve the accuracy of MGMASR, but decreases its performance instead. An oversized embedding vector also will additionally occupy the related computational resources. Therefore, it is meaningful to search and set the size of the embedding vector reasonably.

5) Impact of the Number of Aspects: Moreover, we watched in detail the impact of the number of aspects in MGMASR on its performance, and performed experiments on the EPS dataset and conducted analysis. As described in hyperparameter settings, the number of aspects was assumed to be five different values of 1, 2, 3, 4, and 5, respectively. In particular, the experimental result for the number of aspects set to 1 is the same as the MGMASR-RA in Fig. 4. The test results are shown in Fig. 5(c) and Fig. 6(c).

Therefore, we can find that the performance of MGMASR increases as the number of aspects increases, and the best performance of MGMASR is achieved when the number of aspects is set to 3. However, it decreases as the number of aspects increases further. This implies that the number of aspects needs to be reasonably studied and set. An excessive number of multi-aspect will not only increase the complexity of MGMASR, but also may cause the overfitting problem and reduce the service recommendation accuracy of MGMASR.

6) Impact of the Top-K Service Pairs: Besides, we conducted experiments and analyzed the number of service pairs of mashup involved in MGMASR. We set the number of service pairs of mashup to six different values as 15, 30, 50, 75, 100 or 130 for the test, respectively. The obtained results on the EPS dataset are shown in Fig. 5(d) and Fig. 6(d). It is shown that the performance of MGMASR increases as the number of service pairs of mashup employed by MGMASR increases. The best performance of MGMASR is achieved when the number of service pairs is set to 75.

However, it again decreases as the number of service pairs of mashup employed by MGMASR increases further. This implies that the selection of the number of service pairs for MGMASR needs to be studied and set reasonably. It will be difficult to optimize MGMASR by introducing mashup information with a too small number of service pairs. However, a too large number of service pairs may introduce some unnecessary noise into MGMASR, which may affect the service recommendation performance of MGMASR.

7) Computational Complexity Analysis: We analyzed the computational complexity of our proposed MGMASR. The time cost of MGMASR mainly lies in the utilization of GNNs and attention mechanisms on different relationship graphs. Given *n* users and *m* services, we assume that under the background of $\tilde{a}_a$ aspects, each user connects with $\tilde{a}_{us}$ services in the user-service interaction graph and $\tilde{a}_{uu}$ users in the user-user social graph on average, and that each service connects with $\tilde{a}_{su}$ users in the user-service interaction graph and $\tilde{a}_{ss}$ services in the service-service mashup graph on average.

Take the user representation as an example, for the user interaction representation, the computational time cost mainly lies in the aggregations of GNNs and the attention mechanism at different aspects with a two-layer multi-layer perceptron, which we can roughly express it as $\mathrm{O}(\tilde{a}_a n \tilde{a}_{us} d^2)$. For the user social representation, similarly, we can roughly denote it as $\mathrm{O}(n \tilde{a}_{uu} d^2)$. Further combined with the service representation part, the overall computational complexity is about $\mathrm{O}((\tilde{a}_a \tilde{a}_{us} + \tilde{a}_{uu}) n d^2 + (\tilde{a}_a \tilde{a}_{su} + \tilde{a}_{ss}) m d^2)$. For the best baseline MGNN, its computational complexity is about $\mathrm{O}((\tilde{a}_{us} + \tilde{a}_{uu}) n d + (\tilde{a}_{su} + \tilde{a}_{ss}) m d)$. Compared to it, the complexity of MGMASR is obviously increased, but its complexity remains acceptable for the operation, because $\tilde{a}_a$, $\tilde{a}_{us}$, $\tilde{a}_{uu}$, $\tilde{a}_{su}$ and $\tilde{a}_{ss}$ are much smaller than any one of n and m. This shows that MGMASR can achieve performance improvement on service recommendation in IoS, while the amount of computation resource is effectively controlled.

VII. Conclusion

With the further expansion of IoS, how to make accurate service recommendations for users in IoS has become a highly demanding issue at present. However, little research has focused on this problem in IoS, and it is intractable to achieve satisfactory results with the current traditional service recommendation methods.

In this article, we tackle this problem by presenting a novel multi-graph and multi-aspect neural network-powered service recommendation method called MGMASR. To solve the data sparsity problem in IoS, we introduce users' social information to increase the understanding of users' preferences. We not only further apply GNNs and the attention mechanism to mine multiple relationship graph data in IoS, but also reveal user preferences and service characteristics underlying the user interaction data from multiple aspects at a fine-grained level. Extensive experiments over two real-world IoS datasets have demonstrated that MGMASR can outperform the baseline methods.

In our future research, we plan to follow three directions. First, we will try to consider more subjects in IoS, such as service providers, service mashup developers, and service platform managers to compose more relationship graphs for building the service recommendation method. Second, we will study how to better solve the cold-start problem in service recommendations of IoS. Third, we plan to design a more efficient sampling strategy for GNNs used in MGMASR.

References

[1] S. Yangui, A. Goscinski, K. Drira, Z. Tari, and D. Benslimane, "Future generation of service-oriented computing systems," *Future Gener. Comput. Syst.*, vol. 118, pp. 252–256, May 2021.

[2] J. Wang, C. Xu, J. Zhang, and R. Zhong, "Big data analytics for intelligent manufacturing systems: A review," *J. Manuf. Syst.*, vol. 62, pp. 738–752, Jan. 2022.

[3] Q. Liu, Y. Peng, J. Wu, T. Wang, and G. Wang, "Secure multi-keyword fuzzy searches with enhanced service quality in cloud computing," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 2046–2062, Jun. 2021.

[4] J. Guerreiro, L. Rodrigues, and N. Correia, "Allocation of resources in SAaaS clouds managing thing mashups," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 3, pp. 1597–1609, Sep. 2020.

[5] H. Sami, H. Otrok, J. Bentahar, and A. Mourad, "AI-based resource provisioning of IoE services in 6G: A deep reinforcement learning approach," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 3527–3540, Sep. 2021.

[6] H. Shi, H. Xu, X. Xu, and Z. Wang, "How big service and Internet of Services drive business innovation and transformation," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, 2022, pp. 517–532.

[7] Y. Wang, Z. Tu, Y. Bai, H. Yuan, X. Xu, and Z. Wang, "A blockchain-based infrastructure for distributed Internet of Services," in *Proc. IEEE World Congr. Services (SERVICES)*, 2021, pp. 108–114.

[8] H. Xu et al., "Domain priori knowledge based integrated solution design for Internet of Services," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, 2020, pp. 446–453.

[9] W. Zhang, S. Zhang, and S. Guo, "A PageRank-based reputation model for personalised manufacturing service recommendation," *Enterprise Inf. Syst.*, vol. 11, no. 5, pp. 672–693, 2017.

[10] G. Zou, S. Yang, S. Duan, B. Zhang, Y. Gan, and Y. Chen, "DeepLTSC: Long-tail service classification via integrating category attentive deep neural network and feature augmentation," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 2, pp. 922–935, Jun. 2022.

[11] B. Bai, Y. Fan, W. Tan, and J. Zhang, "DLTSR: A deep learning framework for recommendations of long-tail Web services," *IEEE Trans. Services Comput.*, vol. 13, no. 1, pp. 73–85, Jan./Feb. 2017.

[12] X. Xu, Q. Z. Sheng, L.-J. Zhang, Y. Fan, and S. Dustdar, "From big data to big service," *Computer*, vol. 48, no. 7, pp. 80–83, Jul. 2015.

[13] W.-D. Xi, L. Huang, C.-D. Wang, Y.-Y. Zheng, and J.-H. Lai, "Deep rating and review neural network for item recommendation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6726–6736, Nov. 2022.

[14] C. Wei, Y. Fan, and J. Zhang, "High-order social graph neural network for service recommendation," *IEEE Trans. Netw. Service Manag.*, early access, Jun. 30, 2022, doi: 10.1109/TNSM.2022.3186396.

[15] G. Kang, J. Liu, Y. Xiao, B. Cao, Y. Xu, and M. Cao, "Neural and attentional factorization machine-based Web API recommendation for mashup development," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4183–4196, Dec. 2021.

[16] 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.

[17] Z. Yang, Z. Wang, L. Guo, W. Liu, and S. Sun, "Meta path-aware recommendation method based on non-negative matrix factorization in LBSN," *IEEE Trans. Netw. Service Manag.*, early access, Nov. 4, 2022, doi: 10.1109/TNSM.2022.3219456.

[18] V. S. Lalapura, J. Amudha, and H. S. Satheesh, "Recurrent neural networks for edge intelligence: A survey," *ACM Comput. Surveys*, vol. 54, no. 4, pp. 1–38, 2021.

[19] J. Gu et al., "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[20] Q. Guo, Z. Sun, and Y.-L. Theng, "Exploiting side information for recommendation," in *Proc. Int. Conf. Web Eng.*, 2019, pp. 569–573.

[21] X. Wang et al., "Multi-component graph convolutional collaborative filtering," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 6267–6274.

[22] C. Jaw, J.-Y. Lo, and Y.-H. Lin, "The determinants of new service development: Service characteristics, market orientation, and actualizing innovation effort," *Technovation*, vol. 30, no. 4, pp. 265–277, 2010.

[23] K. Huang, S. Li, W. Deng, Z. Yu, and L. Ma, "Structure inference of networked system with the synergy of deep residual network and fully connected layer network," *Neural Netw.*, vol. 145, pp. 288–299, Jan. 2022.

[24] N. A. Asif et al., "Graph neural network: A comprehensive review on non-Euclidean space," *IEEE Access*, vol. 9, pp. 60588–60606, 2021.

[25] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[26] X. Zhang and X. Liu, "A two-stage robust model for express service network design with surging demand," *Eur. J. Oper. Res.*, vol. 299, no. 1, pp. 154–167, 2022.

[27] W. Ren, K. Wu, Q. Gu, and Y. Hu, "Intelligent decision making for service providers selection in maintenance service network: An adaptive fuzzy-neuro approach," *Knowl.-Based Syst.*, vol. 190, Feb. 2020, Art. no. 105263.

[28] S. T. Arzo, R. Bassoli, F. Granelli, and F. H. Fitzek, "Multi-agent based autonomic network management architecture," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 3595–3618, Sep. 2021.

[29] X. Xue, Z. Chen, S. Wang, Z. Feng, Y. Duan, and Z. Zhou, "Value entropy: A systematic evaluation model of service ecosystem evolution," *IEEE Trans. Services Comput.*, vol. 15, no. 4, pp. 1760–1773, Jul./Aug. 2022.

[30] M. Liu, Z. Tu, H. Xu, X. Xu, and Z. Wang, "Community-based service ecosystem evolution analysis," *Service Oriented Comput. Appl.*, vol. 16, pp. 97–110, Apr. 2022.

[31] H. Lin, Y. Fan, J. Zhang, B. Bai, Z. Xu, and T. Lukasiewicz, "Toward knowledge as a service (KaaS): Predicting popularity of knowledge services leveraging graph neural networks," *IEEE Trans. Services Comput.*, early access, Jan. 25, 2022, doi: 10.1109/TSC.2022.3145019.

[32] G. Kang, J. Liu, Y. Xiao, Y. Cao, B. Cao, and M. Shi, "Web services clustering via exploring unified content and structural semantic representation," *IEEE Trans. Netw. Service Manag.*, early access, Aug. 10, 2022, doi: 10.1109/TNSM.2022.3197725.

[33] H. Li, S. Mi, Q. Li, X. Wen, D. Qiao, and G. Luo, "A scheduling optimization method for maintenance, repair and operations service resources of complex products," *J. Intell. Manuf.*, vol. 31, no. 7, pp. 1673–1691, 2020.

[34] B. Shahzaad and A. Bouguettaya, "Top-K dynamic service composition in skyway networks," in *Proc. Int. Conf. Service-Oriented Comput.*, 2021, pp. 479–495.

[35] H. Mezni, D. Benslimane, and L. Bellatreche, "Context-aware service recommendation based on knowledge graph embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5225–5238, Nov. 2022.

[36] N. T. T. Van et al., "Dynamic network service selection in intelligent reflecting surface-enabled wireless systems: Game theory approaches," *IEEE Trans. Wireless Commun.*, vol. 21, no. 8, pp. 5947–5961, Aug. 2022.

[37] X. He, Z. Tu, X. Xu, and Z. Wang, "Programming framework and infrastructure for self-adaptation and optimized evolution method for microservice systems in cloud–edge environments," *Future Gener. Comput. Syst.*, vol. 118, pp. 263–281, May 2021.

[38] K. Fletcher, "Regularizing matrix factorization with implicit user preference embeddings for Web API recommendation," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, 2019, pp. 1–8.

[39] Y. Cao, J. Liu, M. Shi, B. Cao, T. Chen, and Y. Wen, "Service recommendation based on attentional factorization machine," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, 2019, pp. 189–196.

[40] X. Wu, Y. Fan, J. Zhang, H. Lin, and J. Zhang, "QF-RNN: QI-matrix factorization based RNN for time-aware service recommendation," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, 2019, pp. 202–209.

[41] R. Yan, Y. Fan, J. Zhang, J. Zhang, and H. Lin, "Service recommendation for composition creation based on collaborative attention convolutional network," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2021, pp. 397–405.

[42] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surveys*, vol. 52, no. 1, pp. 1–38, 2019.

[43] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.

[44] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 639–648.

[45] Q. Wu et al., "Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems," in *Proc. World Wide Web Conf.*, 2019, pp. 2091–2102.

[46] W. Fan et al., "A graph neural network framework for social recommendations," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2033–2047, May 2022.

[47] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Stat.*, 2011, pp. 315–323.

[48] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Social attentional memory network: Modeling aspect-and friend-level differences in recommendation," in *Proc. 12th ACM Int. Conf. Web Search Data Min.*, 2019, pp. 177–185.

[49] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.

[50] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Min.*, 2011, pp. 287–296.

[51] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: Social recommendation using probabilistic matrix factorization," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, 2008, pp. 931–940.

[52] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 135–142.

[53] W. Fan et al., "Graph neural networks for social recommendation," in *Proc. World Wide Web Conf.*, 2019, pp. 417–426.

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

[55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[56] L. Prechelt, "Early stopping—But when?" in *Neural Networks: Tricks of the Trade*, 2nd ed. Heidelberg, Germany: Springer, 2012, pp. 53–67.

[57] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2008, pp. 426–434.

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

Yushun Fan received the 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, the Director of the System Integration Institute, and the Director of the Networking Manufacturing Laboratory, Tsinghua University. He is a member of IFAC TC 5.1 and TC 5.2 and the Vice Director of China Standardization Committee for Automation System and Integration. From September 1993 to 1995, he was a Visiting Scientist, supported by Alexander von Humboldt Stiftung, with the Fraunhofer Institute for Production System and Design Technology, Germany. He has authored ten 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. He is an Editorial Member of the *International Journal of Computer Integrated Manufacturing*.

Jia Zhang (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science from Nanjing University, China, and the 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, and a Professor with the Department of Computer Science, Southern Methodist University. She has published more than 180 refereed journal articles, book chapters, and conference papers. 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 interdisciplinary applications of all of these interests in the area of Earth science. She is currently an Associate Editor of the IEEE TRANSACTIONS ON SERVICES COMPUTING.