Multi-Modal Reciprocal Spatiotemporal Framework for Predicting Usage Trend of Knowledge Services

Ruyu Yan, Haozhe Lin, Yushun Fan, Jia Zhang, *Senior Member, IEEE*, and Bing Bai

Abstract—As an emerging concept, Knowledge as a Service (KaaS) aims to provide on-demand content-based (data, information, knowledge) delivery to meet the needs of users. With the prosperity of knowledge services, the prediction of the usage tendency of knowledge services has become an important and timely research topic. This study focuses on speculating the possible popularity of knowledge services in the next period of time, which can assist other downstream service tasks such as service recommendations. The interactions among knowledge services and their rich information (such as historical usage observation and text information) provide grounding for predicting the usage trend of services. However, recent spatial-temporal prediction based on graph neural networks usually depends heavily on the quality of manually created graphs, which may be expensive for knowledge services. To tackle such a limitation, this article proposes a novel Multi-modal Reciprocal SpatioTemporal (MRST) framework, which can jointly mine spatial dependencies and model time patterns for spatiotemporal coupling prediction. Two types of Edge Inference Networks (called EIN-o and EIN-t) are designed to sufficiently discover the spatial dependencies among knowledge services based on the data of usage observation sequences and service descriptions, respectively, and generate multi-modal directed weighted knowledge service graphs. Based on these graphs, MRST integrates GCN-based spatiotemporal prediction models as backbones to make predictions. Particularly, MRST features a unique reciprocal framework. On the one hand, EINs infer and generate multi-modal graphs to serve GCNs; on the other hand, GCNs utilize such spatial dependencies to make predictions and then introduce feedback to optimize EINs. In the meantime, to facilitate reproducible research, we collect a new knowledge service dataset from *Wikipedia* called Wiki-EN dataset. Experiments on this real data set show that the proposed MRST framework significantly surpasses the baselines and can learn meaningful spatial dependencies outside the predefined graphic structure.

Index Terms—Graph convolutional networks, Knowledge as a Service, popularity prediction, spatiotemporal prediction

1 Introduction

Knowledge as a Service (KaaS)[1] is an emerging concept in recent years, which favors on-demand content-based (data, information, knowledge) delivery as utility services to meet the needs of users [1]. With the penetration of services computing technology in the past two decades, the types and quantity of knowledge-based services (short for knowledge services) published online have been gradually enriched. Various platforms offering knowledge services have emerged on the Internet, which has brought prosperity to KaaS. For example, *Wikipedia*[2] provides encyclopedia entries as knowledge services; knowledge-based Q&A platforms such as *Zhihu*[3] and *Quora*[4] provide answer lists as knowledge services; *Google Scholar*[5] provides retrieval and suggestions of papers and academic information as knowledge services.

With the prosperity of knowledge services, the topic of *predicting the popularity (or usage tendency) of knowledge services* has gained great momentum, because it helps to speculate on the possible popularity of knowledge services in the future, and may assist other downstream service tasks such as service recommendation tasks [18]. For example, by predicting the number of page views of Wikipedia entries, we are able to infer the topics that Internet users may care about in the next period of time. As another example, by predicting the number of citations of academic papers, we are able to identify potential popular papers from thousands of academic papers.

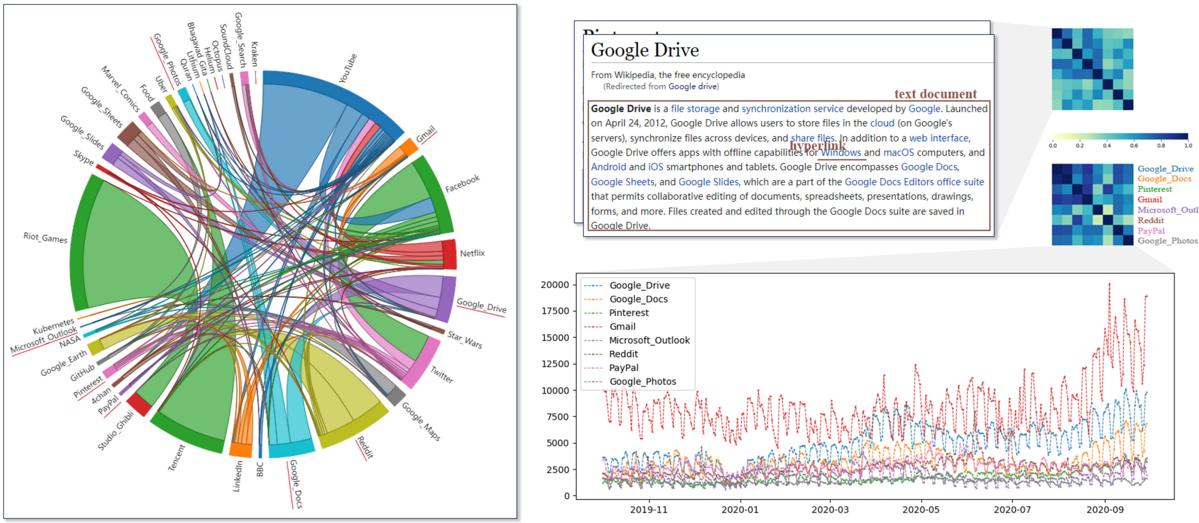
To accurately predict the usage tendency of knowledge services, various types of information related to knowledge services deserve careful considerations, such as their historical usage observations and associated text documents. How to

[1] https://en.wikipedia.org/wiki/Knowledge as a service

[2] https://www.wikipedia.org/
[3] https://www.zhihu.com/
[4] https://www.quora.com/
[5] https://scholar.google.com/

- *Ruyu Yan, Haozhe Lin, and Yushun Fan are with the Department of Automation, Tsinghua University, Beijing 100190, China, and also with the Beijing National Research Center for Information Science and Technology, Beijing 100190, China. E-mail: yanry18@mails.tsinghua.edu.cn, linhz@mail.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.*
- *Bing Bai is with the Cloud and Smart Industries Group, Tencent, Beijing 100085, China. E-mail: icebai@tencent.com.*

Manuscript received 19 April 2022; revised 11 November 2022; accepted 19 December 2022. Date of publication 22 December 2022; date of current version 8 August 2023.
This work was supported by the National Natural Science Foundation of China under Grant 62173199.
(Corresponding author: Yushun Fan.)
Recommended for acceptance by F. Liu.
Digital Object Identifier no. 10.1109/TSC.2022.3231382



Google Drive

From Wikipedia, the free encyclopedia
(Redirected from Google drive)

Google Drive is a file storage and synchronization service developed by Google. Launched on April 24, 2012, Google Drive allows users to store files in the cloud (on Google's servers), synchronize files across devices, and share files. In addition to a web interface, Google Drive offers apps with offline capabilities for Windows and macOS computers, and Android and iOS smartphones and tablets. Google Drive encompasses Google Docs, Google Sheets, and Google Slides, which are a part of the Google Docs Editors office suite that permits collaborative editing of documents, spreadsheets, presentations, drawings, forms, and more. Files created and edited through the Google Docs suite are saved in Google Drive.

Fig. 1. Spatial dependencies, usage observation sequences and text document information of real-world Wikipedia knowledge services. Hyperlinks among 38 random Wikipedia entries are shown on the left. The width of the line on the circumference represents the jumping frequency between knowledge services. Eight services underlined in red are selected to show further details. The middle upper part shows the text information of these services, and the right lower part shows their usage observations. The right upper part shows the spatial dependencies calculated from these two kinds of information.

synergistically utilize them remains a big challenge in three significant aspects. First, the historical usage observation sequences of knowledge services may carry a variety of complex factors, such as user behaviors or periodic events. As a result, it may be hard to precisely extract their time-domain features and measure the distance between two sequences, i.e., to detect correlation or causality. Second, although the service ecosystem can reveal whether or not there exists interactions between two knowledge services, it is difficult to quantify their dependencies. For example, service usage observation sequences and text information may imply different dependencies among knowledge services. Third, since the dependencies are not single-modal but multi-modal, it is difficult to integrate them into one framework and design a reasonable end-to-end training strategy. Here we take a knowledge service platform *Wikipedia* as an example. One type of interactions among Wikipedia entries is identified as hyperlinks, also called explicit spatial dependencies, each being a directed link from one entry to another. However, it may be difficult to further quantify the weights of these spatial dependencies, especially when the contents of Wikipedia entries are often modified. Furthermore, the implicit spatial dependencies among Wikipedia entries inferred by other service information, such as observation sequences and text information, are typically multi-modal. The right half of Fig. 1 shows the observation sequences and text information corresponding to eight selected knowledge services, as well as the correlation obtained from these information. Thus, it is not easy to construct an accurate quantitative graph in such situations, and human involvement may be sub-optimal.

The state-of-the-art methods cannot fully address the above challenges. With the development of graph convolution network (GCN) [8], [12], [16], many spatio-temporal prediction studies have been reported. Some research works [17], [31] utilize the interactions among multiple time series, in which different time series and their related relationships are modelled as vertices and edges in graphs. These spatiotemporal prediction methods significantly improve the service prediction effect but are highly dependent on the well-defined graph structure. In many practical service platforms, however, it is not always easy to obtain high-quality service relationship graphs.

In this article, we propose a novel Multi-modal Reciprocal Spatio-temporal (MRST) framework to synergistically address the aforementioned challenges and predict the usage tendency of knowledge services. Our MRST framework consists of two main modules: a spatial dependency inference module and a spatiotemporal prediction module. For the former module, we have designed two types of Edge Inference Networks (called EIN-t and EIN-o) to mine the spatial dependencies among knowledge services based on the information of usage observation sequences and text, respectively. In the latter module, we seamlessly integrate GCN-based spatiotemporal prediction modules (such as DCRNN [17] and Graph WaveNet [31]) as backbones for making spatiotemporal predictions. The two modules synergistically collaborate to support each other. On the one hand, from the inference side to the prediction side, EINs infer and generate multi-modal directed weighted knowledge service graphs to serve GCNs. On the other hand, from the prediction side to the inference side, GCNs utilize these spatial dependencies to make predictions and then introduce feedback to help EINs better learn distance measurement. Therefore, this iterative learning cycle between the two modules within the MRST framework can gradually enhance spatiotemporal coupling prediction over time, resulting in a promising ”reciprocity.”

Our main contributions can be summarized in three aspects:

- We have designed two types of Edge Inference Networks to efficiently infer multi-modal, directed and weighted spatial dependencies.
- We have developed a novel multi-modal reciprocity spatiotemporal framework, which can integrate different spatiotemporal prediction modules and enable ”reciprocity” between its comprising inference side and prediction side.

- Our experiments over a real-world knowledge service platform demonstrate that our MRST is superior to the most advanced spatiotemporal prediction algorithms in prediction accuracy.

The remainder of this paper is organized as follows. Section 2 gives the symbols and mathematically restates the prediction problem of the usage tendency of knowledge services. Section 3 describes the proposed model and Section 4 describes the training details. Section 5 reports our experimental results. Section 6 reviews the related work. Finally, Section 7 draws a conclusion.

2 Preliminaries

In this section, we give the mathematical definitions of important notations and the targeted problem.

2.1 Notation Definition

Definition 1. *(Usage observations of knowledge services). The usage observations of knowledge services refer to their usage time series in the past p time steps. We denote usage observations of knowledge services by time series $\boldsymbol{x}_i$, where $\boldsymbol{x}_i$ can be decomposed into $\boldsymbol{x}_i = \{x_i^0, x_i^1, \ldots, x_i^p\}$. In this paper, we take the day as the time unit. Therefore, x_i^t represents the number of times service i has been used in the past t-th day. For the whole N services, we use $\boldsymbol{X} = \{\boldsymbol{x}_1; \boldsymbol{x}_2; \ldots; \boldsymbol{x}_N\}$ to represent the usage observation collection of these services.*

Definition 2. *(Usage trend of knowledge services). The usage trend of knowledge services reflects the usage of knowledge services in the future. We denote the usage trend by $\hat{\boldsymbol{y}}_i = \{\hat{y}_i^{p+1}, \hat{y}_i^{p+2}, \ldots, \hat{y}_i^{p+q}\}$, where $\hat{y}_i^t (t \in (p+1, p+q))$ represents the predicted value of knowledge service i calls in the next t-th day. Similarly, $\hat{\boldsymbol{Y}} = \{\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \ldots, \hat{\boldsymbol{y}}_N\}$ represents a collection of N knowledge service usage trends.*

Definition 3. *(Text documents of knowledge services). A knowledge service document contains all the words in the textual description of the knowledge service. We denote knowledge service i's document by $\boldsymbol{w}_i = [w_i^1, w_i^2, \ldots, w_i^l]$, where l refers to the number of words of which the document consists. For the whole N services, we write $\boldsymbol{D} = \{\boldsymbol{w}_1; \boldsymbol{w}_2; \ldots; \boldsymbol{w}_N\}$ to denote all the N documents.*

Definition 4. *(Spatial dependencies among knowledge services). We construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ to represent the spatial dependencies among knowledge services, where vertices $\mathcal{V}$ refers to services set and $|\mathcal{V}| = N$, edges $\mathcal{E}$ refers to dependencies set, and $\mathcal{W}$ refers to the corresponding weights for each dependency. The graph $\mathcal{G}$ starts from a handcraft graph $\mathcal{G}^0$, which is generated by the initial 0-1 interactions among knowledge services. Then we consider inferring spatial dependencies $\mathcal{W}$ by multi-modal information (usage observations and text of knowledge services). As a result, we obtain multi-modal, weighted and directed knowledge service graphs $\mathcal{G}^o$ and $\mathcal{G}^t$, which corresponds to spatial dependencies $\boldsymbol{w}^o$ and $\boldsymbol{w}^t$, respectively. The former weight w_{ij}^o refers to the spatial dependency from usage observation of service i to j, and the latter weight w_{ij}^t refers to the spatial dependency from text information of service i to j.*

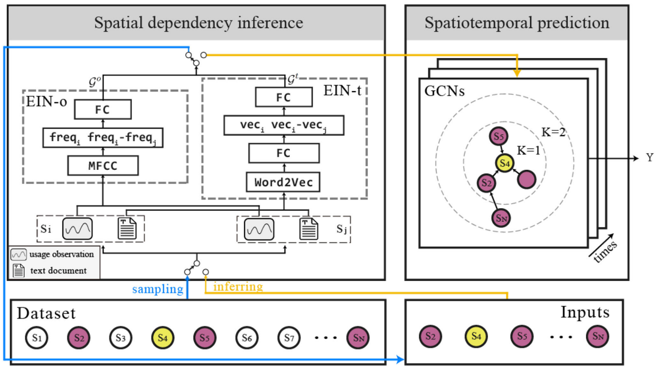


Fig. 2. The overview of our MRST framework.

2.2 Problem Restatement

Problem (Predicting the usage tendency of knowledge services). Given the usage observations $\boldsymbol{X}$ and text documents $\boldsymbol{D}$ of N knowledge services, our goal is to predict the usage tendency $\hat{Y}$ of knowledge services, as well as to infer their underline spatial dependencies $\mathcal{G}$. The mapping relation is represented as follows:

$$\left[\boldsymbol{X}^1, \boldsymbol{X}^2, \ldots, \boldsymbol{X}^p, \boldsymbol{D}, \mathcal{G}^0\right] \longrightarrow \left[\hat{\boldsymbol{Y}}^{p+1}, \ldots, \hat{\boldsymbol{Y}}^{p+q}, \mathcal{G}^o, \mathcal{G}^t\right] \tag{1}$$

3 Model Architecture

Fig. 2 depicts the overall structure of our MRST model, which is mainly composed of two components: a spatial dependency inference module and a spatiotemporal prediction module. We develop two types of Edge Inference Networks (EINs) to sufficiently discover the dependencies among knowledge services and construct multi-modal knowledge service graphs. Based on these graphs, we integrate a GCN-based spatiotemporal prediction module as backbones to make predictions.

The blue and orange lines in Fig. 2 represent the sampling and inferring processes, respectively. To deal with large-scale knowledge services, we develop a localized mini-batch training scheme. In particular, within a mini-batch, we sample potential neighbors (i.e., purple circles) for the central vertex (i.e., the yellow circle) according to the spatial dependencies inferred by EINs. These sampling vertices are fed into the model through the orange line as input to a mini-batch for training. EIN-o and EIN-t first infer distances for pairwise knowledge services through the frequency features of usage observation sequences and text document features, respectively, and then construct multi-modal graphs $\mathcal{G}^o$ and $\mathcal{G}^t$ that provide spatial dependencies for the GCNs. GCNs propagate feedback to EINs for spatial distance learning and thus the reciprocity is developed. Specifically, knowledge service s_4 (yellow circle) in Fig. 2 is selected as the central node in this mini-batch. The K-layer neighbors (purple circles) of s_4 are obtained by the weighted sampling of the graph inferred from EINs. These nodes are sent to the subsequent modules as inputs. EIN-o and EIN-t use the knowledge service s_i and s_j's usage observations and text documents to infer multi-modal spatial dependencies between s_i and s_j, respectively. The obtained $\mathcal{G}^o$ and $\mathcal{G}^t$ are sent to the spatiotemporal prediction module to predict the usage trend of the central node s_4. The subsections will introduce each of the comprising components in detail.

3.1 Spatial Dependency Inference

As mentioned in Section 1, the different kinds of relevant information of knowledge services contain different relationships among services, which all can be used to help infer the spatial dependencies. In our paper, without losing generality, we consider two modalities of information, usage observation and text, and design two Edge Inference Networks, EIN-o and EIN-t, to discover and quantify spatial dependencies, respectively.

EIN-o. The Edge Inference Network for usage observation (short for EIN-o) infers spatial dependencies among usage observations of knowledge services. The usage observation sequences can be viewed as signals in the time domain. As mentioned in Section 1, the historical usage observation sequences of knowledge services are usually composed of a variety of complex factors (like user behaviors, periodic events), which makes it difficult to extract their time-domain features and accurately measure the distance between two sequences. Inspired by the experience of audio compression and speech recognition, we consider transforming the signals in the time domain into the frequency domain and adopting the Mel-Frequency Cepstrum Coefficients (MFCCs) [7], [23] to capture their frequency domain features. In more detail, MFCCs employ the Fourier transform and Mel filter to extract features and generate low dimensional dense representation. The formula of MFCCs is as follows:

$$
\begin{aligned}
X[k] &= \mathrm{fft}(x[n]) \\
Y[c] &= \log\left(\sum_{k=f_{c-1}}^{f_{c+1}} |X[k]|^2 B_c[k]\right) \\
c_x[n] &= \frac{1}{C}\sum_{c=1}^{C} Y[c]\cos\left(\frac{\pi n\left(c-\frac{1}{2}\right)}{C}\right)
\end{aligned}
\tag{2}
$$

where $x[n]$ refers to the usage observations of knowledge services; $\mathrm{fft}(\cdot)$ refers to fast Fourier transform; $B_c[k]$ refers to filter banks; C refers to the number of MFCCs to retain; and $c_x[n]$, also denoted by $\boldsymbol{c}_x$, refers to MFCCs of usage observation $\boldsymbol{x}$. MFCCs could generally be smoother features for neural networks than original time series, since they present the envelope of the frequency feature from Fourier transform.

After obtaining the MFCC features, EIN-o infers the spatial dependency between two usage observation sequences through a fully connected layer. We use $\boldsymbol{c}_i \in \mathbb{R}^C$ to refer service i's MFCC feature $c_x[n]$. We then concatenate $\boldsymbol{c}_i$ with $(\boldsymbol{c}_i - \boldsymbol{c}_j)$ to model the directed spatial dependency between service i and service j. The equation is as follows:

$$
a_{ij}^o = \mathrm{ReLU}\big(\boldsymbol{W}^o \cdot \mathrm{CONCAT}\big(\boldsymbol{c}_i, \boldsymbol{c}_i - \boldsymbol{c}_j\big) + b^o\big) \tag{3}
$$

where $\boldsymbol{a}_{ij}^o$ refers to inferred asymmetric distance from usage observations of knowledge service i to j; $\boldsymbol{W}^o \in \mathbb{R}^{2C}$ and b^o refer to learnable parameters for usage observation sequences distance inference. Note that, here we consider modeling the directed spatial dependencies. In order to consider undirected (or symmetric) relationship between usage observation sequences i and j, $\boldsymbol{c}_i$ should be concatenated with $\boldsymbol{c}_j$, instead.

EIN-t. The Edge Inference Network for text document (short for EIN-t) infers spatial dependencies among the text documents of knowledge services. As the main part of knowledge service, text information is meaningful and can be used to mine their potential features. To exploit the semantics of text information, we adopt Word2Vec that is an approach widely used in many Natural Language Processing (NLP) applications [10], [13], which inquires each word embedding representation from a look-up table. Given a knowledge service text document $\boldsymbol{w}_i = [w_i^1, w_i^2, \ldots, w_i^l]$, we first project each word to its embedding representation: $\boldsymbol{E}_i = [e_1, e_2, \ldots, e_l], e_k \in \mathbb{R}^{dim}$, where l is the document length and dim is the word embedding dimension. In order to capture the context information in the document, we first transform representation matrix $\boldsymbol{E}_i$ into a one-dimension vector $\boldsymbol{V}_i \in \mathbb{R}^L$, then perform a fully connected layer with an ReLU activation function. Here $L = l \times dim$, the resultant feature vector is $\boldsymbol{v}_i \in \mathbb{R}^d$. It is worth noting that the fully connected layer here can be replaced by convolution operation or other more complex structures to contain more context semantics. The algorithm is as follows:

Algorithm 1. Edge Inference Network for Text Document

```
Input : text information $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ for knowledge service $i$ and $j$
Output: spatial dependency $\boldsymbol{a}_{ij}^t$ between service $i$ and $j$
$\boldsymbol{E}_i \leftarrow \mathrm{Word2vec}(\boldsymbol{w}_i)$
$\boldsymbol{V}_i \leftarrow \mathrm{One-dim}(\boldsymbol{E}_i)$
$\boldsymbol{v}_i \leftarrow \mathrm{ReLU}(\boldsymbol{W}^v \cdot \boldsymbol{V}_i + \boldsymbol{b}^v)$
$a_{ij}^t \leftarrow \mathrm{ReLU}(\boldsymbol{W}^t \cdot \mathrm{CONCAT}(\boldsymbol{v}_i, \boldsymbol{v}_i - \boldsymbol{v}_j) + b^t)$
```

where $\boldsymbol{a}_{ij}^t$ refers to inferred asymmetric distance from text documents of knowledge services i to j; $\boldsymbol{W}^v \in \mathbb{R}^{d\times L}$ and $\boldsymbol{b}^v \in \mathbb{R}^d$ refer to the parameters for capturing the context information in the document; $\boldsymbol{W}^t \in \mathbb{R}^{2d}$ and b^t refer to learnable parameters for text documents distance inference.

Based on their ability to infer spatial dependencies among information of knowledge services, EIN-o and EIN-t play two important roles in our MRST framework: sampling and inferring. Fig. 2 illustrates the two functions initiated by blue line and orange line, respectively. First, at the stage of data preparation, the inferred spatial structure of EINs can help to sample possible adjacent candidates (i.e., purple vertices) for the central vertices (i.e., yellow vertices). Afterwards, these sampling vertices are fed into the model, and EINs will in turn infer and quantify the multi-modal spatial dependencies for GCNs. In this section, we have shown how EINs generate spatial dependencies for GCNs, and we will discuss in Section 3.3 how they learn from the GCNs for optimization.

3.2 Spatiotemporal Prediction

Based on the spatial dependencies of knowledge services inferred by EIN-o and EIN-t modules, our MRST framework then integrates a GCN-based spatiotemporal structure to predict the usage tendency of knowledge services. The spatiotemporal structure aggregates both spatial and temporal information. In more detail, it first aggregates the spatial influence of the relevant neighbor services through the graph convolution operator, then adopts RNN or CNN structure to capture temporal dependencies.

Aggregate Spatial Influence. As mentioned in Section 1, the trend of a knowledge service is related not only to its past records, but also to the observations of its neighbors. In the spatial dependency inference subsection, we speculate on the neighbor network of the services. Using GCN-based methods to aggregate the information from neighbor nodes on this graph can aggregate the spatial influence to improve the prediction accuracy. Researchers have reported a variety of methods to aggregate spatial dependencies through different graph convolution operators, e.g., Chebyshev convolution [12] and diffusion convolution [2]. Here we take the diffusion convolution as an example to show the workflow of our MRST framework. Considering only one knowledge service network, the diffusion process is characterized by a random walk on this graph. The transition matrix of the diffusion process is $\boldsymbol{L} = \boldsymbol{D}^{-1}\boldsymbol{A}$, where $\boldsymbol{D} = diag(\boldsymbol{A}1)$ is the out-degree diagonal matrix, and $1 \in \mathbb{R}^N$ denotes the all one vector. we use a finite K-step truncation of the diffusion process and assign a trainable weight to each step. Furthermore, we adopt a bidirectional diffusion process to capture the impact from both the upstream and the downstream flows, which can provide greater flexibility for the model. The bidirectional diffusion convolution can be formulated in Equation (4):

$$\boldsymbol{Z}_{:,d_i} \star_{\mathcal{G}} g_\theta \approx \sum_{k=0}^{K-1} \left(\theta_{k,0}\boldsymbol{L}_I^k + \theta_{k,1}\boldsymbol{L}_O^k\right)\boldsymbol{Z}_{:,d_i}, \tag{4}$$

where $\boldsymbol{Z} \in \mathbb{R}^{N\times d_I}$ refers to the inputs of graph convolution filter and $d_i \in \{1, \ldots, d_I\}$; g_θ refers to diffusion convolution filter with $\theta \in \mathbb{R}^{K\times 2}$ as trainable parameters; $\boldsymbol{L}_I$ and $\boldsymbol{L}_O$ refer to input and output Laplacian matrix, respectively. Empirically, the diffusion convolution can often be truncated by not more than 3 ($K \leq 3$) [14]. Due to the sparsity of most graphs, the complexity of the recursively computed Equation (4) is $\mathcal{O}(K|\mathcal{E}|) < < \mathcal{O}(N^2)$.

Based on the previous discussions, we have inferred the multi-modality spatial dependencies. Therefore, we extend the diffusion convolution to obtain the following formula:

$$\boldsymbol{h}_s = \mathrm{ReLU}\left(\boldsymbol{Z}\star_{\mathcal{G}^o} g_{\Theta_o} + \boldsymbol{Z}\star_{\mathcal{G}^t} g_{\Theta_t}\right) \tag{5}$$

where $\boldsymbol{h}_s \in \mathbb{R}^{N\times d_O}$ refers to spatial hidden states, namely output of diffusion convolution operators; $\mathcal{G}^o$ and $\mathcal{G}^t$ refer to modality predefined graphs inferred by EIN-o and EIN-t, respectively; then $\Theta_o, \Theta_t \in \mathbb{R}^{d_O\times d_I\times K\times 2} = [\theta]_{q,p}$, where $\Theta_{d_o,d_i,:,:} \in \mathbb{R}^{K\times 2}$ parameterizes the convolutional filter for the d_i-th input and the d_o-th output. Note that here we use the direct additivity to aggregate the outputs obtained from the two graphs, which could be replaced by other aggregation methods.

Temporal Dependency. After aggregating spatial influence through an enhanced diffusion convolution filter, we consider capturing temporal dependencies. Existing methods typically employ RNNs (e.g., DCRNN [17]) or CNNs (e.g., Graph WaveNet [31]) to capture the temporal dependency of the time series. In this paper, we adopt DCRNN as the backbone to show how our MRST framework merges temporal dependencies to generate predictions. The model applies a module named DCGRUs to capture the temporal dependencies by replacing the multiplication of GRU with diffusion convolution. The formula is as follows:

$$\begin{aligned}
\boldsymbol{r}^t &= \sigma\left(f_r\star_{\mathcal{G}^m}\left[\boldsymbol{X}^t, \boldsymbol{H}^{t-1}\right] + \boldsymbol{b}_r\right)\\
\boldsymbol{u}^t &= \sigma\left(f_u\star_{\mathcal{G}^m}\left[\boldsymbol{X}^t, \boldsymbol{H}^{t-1}\right] + \boldsymbol{b}_u\right)\\
\boldsymbol{C}^t &= \tanh\left(f_C\star_{\mathcal{G}^m}\left[\boldsymbol{X}^t, \left(\boldsymbol{r}^t \odot \boldsymbol{H}^{t-1}\right)\right] + \boldsymbol{b}_C\right)\\
\boldsymbol{H}^t &= \boldsymbol{u}^t \odot \boldsymbol{H}^{t-1} + \left(1 - u^t\right) \odot \boldsymbol{C}^t
\end{aligned} \tag{6}$$

where $\boldsymbol{X}^t \in \mathbb{R}^N$ refer to the observations of all included knowledge services; $\boldsymbol{H}^{t-1}$ refer to temporal hidden states generated by last DCGRUs; $\star_{\mathcal{G}^m}$ refers to diffusion convolution operator given graph $\mathcal{G}^m$, here including $\mathcal{G}^o$ and $\mathcal{G}^t$. $\boldsymbol{r}^t$ and $\boldsymbol{u}^t$ refer to the output of reset and update gates at time t; f_r, f_u, and f_C refer to graph convolutional filters with different trainable parameters; finally, $\boldsymbol{H}^t$ refer to temporal hidden states. In DCRNNs, DCGRUs are stacked to construct encoders and decoders.

Predict Trend. After obtaining the spatiotemporal hidden states $\boldsymbol{H}^t$, a fully connected layer then generates predictions, which is shown in Equation (7):

$$\hat{\boldsymbol{Y}}^{t+1} = \boldsymbol{W} \cdot \boldsymbol{H}^t + \boldsymbol{b} \tag{7}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are trainable parameters.

3.3 Reciprocity

As shown in Fig. 2, the spatial dependency inference and spatiotemporal prediction parts of our MRST framework interact in the process of iterative learning. From the inference side to the prediction side, spatial dependency inference generates multi-modal directed weighted knowledge service graphs through EINs structure, and promotes GCNs to make more accurate predictions in the process of forward propagation. From the prediction side to the inference side, EINs are optimized through the temporal labels of GCNs in back propagation, so as to better learn distance measurement. Therefore, the reciprocity of our MRST framework is developed between the two sides.

However, due to the complexity of the model, EINs and GCNs may interfere with each other, making MRST unable to learn in the expected direction, especially in the initialization phase. We will discuss how to solve the issue of parameter initialization in Section 4.2 and give the discussion and verification of the results in the experimental part.

4 Learning Details

In this section, we discuss parameter tuning and optimization for our MRST framework.

4.1 Loss Function

We choose the mean absolute error (MAE) as the loss function to supervise the training process under our MRST framework, which is formulated by Equation:

$$\mathcal{L} = \frac{1}{n}\sum_{i,t}\left|y_i^t - \hat{y}_i^t\right| \tag{8}$$

where n refers to the number of observations of all time series in a batch; $y_{i,t}$ and $\hat{y}_{i,t}$ refer to the ground truth and predictions of time series i at time t, respectively. Note that for applications where the orders of magnitude of time series significantly differ from each other, we evaluate loss under the logarithmic scale.

4.2 Phased Heuristics

We have discussed in Section 3.3 that EINs and GCNs modules can interact to achieve reciprocity. However, at the beginning of the training process, because both sides are initialized with random states, it may be difficult for the training to move in the right direction. Furthermore, EINs and GCNs may interfere with each other, making training even more difficult. In order to avoid such situation, we have developed a phased heuristic strategy. With the progress of the training process, the reciprocity process gradually works. This heuristic strategy can be divided into three stages.

In the first stage, we first use the limited service space tags, such as the service collaboration relationship (here, hyperlinks), to learn the parameters of graph convolution. The way of the scheduled sampling strategy [4] is adopted in this stage. We set a probability ϵ to control the ratio between the ground truth and the previous prediction. At the beginning of training, ϵ is set very large (close to 1), that is, it relies on the ground truth for training. With the increase of training steps, ϵ gradually decreases until it completely depends on the previous prediction for training. Specifically, we choose inverse S-shaped decay to gradually reduce epsilon:

$$\epsilon = \frac{k}{k + \exp\left(\frac{i}{k}\right)}, \tag{9}$$

where k represents the hyper-parameter of sampling probability attenuation rate, which is generally adjusted according to the size of the dataset and batch size. i refers to the current number of training steps. In this stage, the graph convolution network first takes the ground truth as the model input for one-step prediction, then gradually switches to the autoregressive mode, and the prediction task transitions from one-step prediction to multi-step prediction.

When ϵ decays to 0.1, the second stage begins. EINs begin to use a variety of information to learn how to measure the distance between knowledge service nodes, and build multi-modal spatial correlation for GCN. Similarly, we then set another attenuation factor γ to match ϵ. The same speed is reduced from 1 to 0. This stage uses limited tags to activate the multi-modal topology estimator. However, in this stage, unknown potential links are not introduced, and the quantization ability of the multi-mode topology estimator is mainly studied.

When γ decays to 0.1, the third stage begins, and the multi-mode EINs begin to explore the possible connections between all paired service nodes and use the new graph prediction to introduce the diversity of service structure for graph convolution.

When exploring small graphs, we can start all three stages. However, when the number of nodes is huge, the third stage often needs to occupy a large amount of memory. Under such a situation, we may achieve better results by only executing the first two stages instead of all three stages. How to explore more potential edges in a huge graph remains to be a further study in the future.

Through the phased heuristic algorithm, different parts of our MRST framework can be trained sequentially, making it easier for the MRST framework to achieve local optimization. In the follow-up experiments, we will discuss the effect of this phased heuristic strategy.

5 Experiments

We have conducted experiments to evaluate the effectiveness and efficiency of our proposed MRST framework. In this section, we first introduce our experimental settings, and then analyze experimental results in detail.

5.1 Experimental Settings

5.1.1 Dataset

Wikipedia is a well-known online encyclopedia that provides knowledge services. It is created, maintained, edited and modified by hundreds of millions of Internet users, and has been widely studied as a common dataset in public competitions and the literature. The temporal observations of wiki entries (i.e., the usage observations of knowledge services) exhibit periodicity, nonlinearity and (non) stationarity. As shown in Fig. 3, the usage observation tendency of different knowledge services presents different features. Due to the editability of Wikipedia entries, the hyperlink relationships among entries may change over time. Such features reflect the complexities of knowledge services in spatial and temporal aspects. Furthermore, the existing Wikipedia data sets (such as Wiki-EN-small) are not sufficient and comprehensive. Taking Wiki-EN-small as an example, it lacks text information of entries. Moreover, it only contains 4,118 entries and 11,198 edges, which means the graph structure is relatively simple. To get closer to the real scene, we collected an English Wikipedia dataset called Wiki-EN, which contains 12,508 entries and 376,700 edges, together with the text information of each entry.

Specifically, we grabbed the page views of Wikipedia entries from WikiStat[6], which records the page views of thousands of Wikipedia entries on an hourly frequency. We preprocessed the hourly page views into daily observation sequences. We observed that some pages do not have access records at some times due to a lack of data or excessive popularity. To ensure the reliability of the data, we only kept the knowledge services with complete records, and screened 12,508 Wikipedia entries with clear usage observations from October 1, 2019 to September 30, 2021. At the same time, according to the Wikipedia hyperlink jump record in September 2019, we obtained a graph of 376,700 hyperlink relationships (i.e., edges), and regarded this graph as the initial knowledge service network. In addition, we collected the text information of each entry. Because the content of the entry is very different, the length of the text information is also very different. The detail of our dataset is summarized in Table 1.

We validated the baselines and our proposed MRST on the Wiki-EN dataset for three reasons. First, for the spatial dependencies, the Wikipedia entries interact with each other by hyperlinks. Thus, the initial knowledge service network could be constructed through this relationship.

[6] https://dumps.wikimedia.org

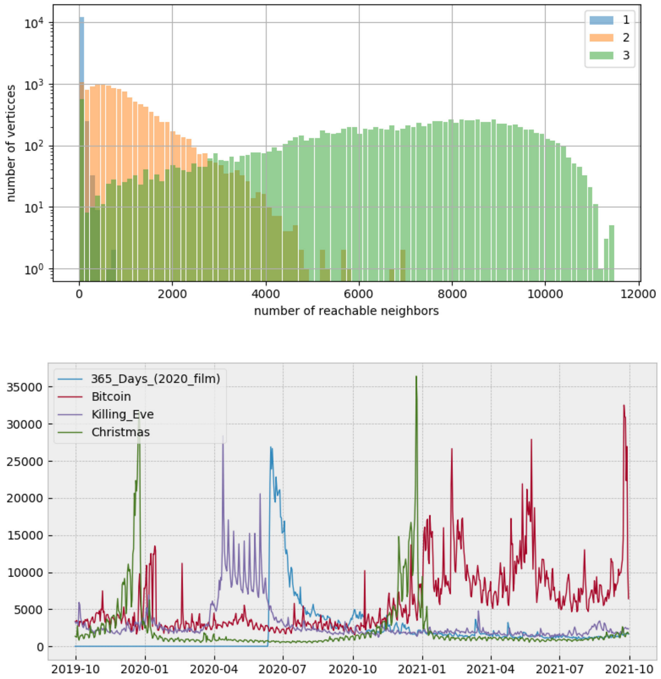


Fig. 3. a) Histogram of Wiki-EN dataset. Within 12,508 knowledge services, the blue, yellow and green ones represent the histogram of 1, 2 and 3-step reachable neighbors, respectively. When we consider 3-step reachable neighbors, most of the services' vertices are connected, which proves the correlations among knowledge services are complex. b) Service Christmas shows an annual peak on Christmas Day, and Killing Eve shows a weekly peak since it was updated once a week. However, movie 365_Days' peak is reached from the time of release, and the usage observations gradually decrease thereafter. Service Bitcoin continues to receive attention throughout the year. This shows that different knowledge services have different temporal features.

TABLE 1
Numerical Properties of Datasets

Item	Value
Number of Entries	12,508
Number of edges	376,700
Number of observation samples	9,143,348
Shortest number of text words	28
Maximum number of text words	28,858
Average number of text words	4,447

Second, since the manually created hyperlink relationships are difficult to fully reflect the accurate spatial dependencies among knowledge services, it is necessary to rely on models to learn more accurate knowledge service networks. The text information and usage observation sequences of Wikipedia entries can both be used to train the knowledge service networks. Third, for the temporal dependencies, different page views apparently present different and complicated temporal characteristics. Based on these features in all three aspects, we believe it is reasonable to use the Wiki-EN dataset to test and verify our model.

We used Z-score normalization and logarithm to the inputs to eliminate the potential hazard caused by the vast difference in the order of magnitude. During the experiments, we chronologically split the dataset, with the first 70% as the training set, the following 10% as validating set, and the final 20% as the testing set.

5.1.2 Evaluation Schemes

We evaluated our MRST framework and baselines by the mean absolute error (MAE), root mean squared logarithmic error (RMSLE), and symmetric mean absolute percentage error (SMAPE) over the Wiki-EN dataset.

The first indicator MAE evaluates the average value of absolute error, which can better reflect the actual situation of predicted value error. The lower the MAE, the higher the prediction accuracy. The mean absolute error is defined as:

$$\mathrm{MAE} = \frac{1}{\Omega}\sum_{i\in\Omega}|y_i - \hat{y}_i| \tag{10}$$

The second indicator RMSLE evaluates the model by shrinking the prediction result to a logarithmic scale, which alleviates the impact caused by order of magnitude. RMSLE can be formulated by Equation (10), and a lower RMSLE represents a higher prediction accuracy.

$$\mathrm{RMSLE} = \sqrt{\frac{1}{n}\sum_{i,t}\left[\log\left(y_i^t + 1\right) - \log\left(\hat{y}_i^t + 1\right)\right]^2} \tag{11}$$

The third indicator SMAPE reflects a relative residual of prediction results, which can also solve the problem of the order of magnitude. Besides, an under-forecasting prediction gets a higher value than an over-forecasting one. It is quite suitable for our problem, because a slight redundancy is essential for any kind of role in the service ecosystem, as explained in earlier sections. SMAPE can be formulated by Equation (11), and a lower SMAPE represents a higher prediction accuracy

$$\mathrm{SMAPE} = \frac{2}{n}\sum_{i,t}\frac{\left|\hat{y}_i^t - y_i^t\right|}{\left(\hat{y}_i^t + y_i^t\right)}. \tag{12}$$

5.1.3 Baselines

In our experiments, we mainly chose two GCN-based spatio-temporal structures as the backbones of our MRST framework, DCRNNs and GraphWaveNet, and compared our MRST with seven representative baselines described as follows.

- *ARIMA* [6]: AutoRegressive Integrated Moving Average (ARIMA) is the most classical time series prediction model used in many industries. ARIMA needs to model each sequence separately and can easily capture the linearity of one time series. We used the open source statsmodel Python package[7] to implement this benchmark method.
- *VAR* [22]: Vector AutoRegressive (VAR) is a vectorized high-dimensional extension of ARIMA, which further considers simple dependencies between sequences. In the experiments, we set one sequence with all its one-order neighbors as a group, and trained different VAR models for each individual sequence.
- *SVR* [25]: By setting different kernel functions, Support Vector Regression(SVR) model has achieved good performance in multiple complex time series prediction tasks and has been widely used. We implemented this method through the sklearn python package[8].

[7] https://www.statsmodels.org
[8] https://pypi.org/project/scikit-learn

- *FC-LSTM* [26]: RNN with fully connected LSTM hidden units uses the long-term and short-term memory unit as the basic unit of the cyclic neural network, and has a good ability to model sequence nonlinearity, periodicity and long-term dependence.
- *WaveNet* [24]: WaveNet models the long-term dependencies of sequences by expansion convolution. It was originally designed for speech synthesis task, and later expanded to other domains because of its good timing modelling performance.
- *DCRNNs* [17]: Diffusion Convolutional Recurrent Neural Networks (DCRNNs) is the state-of-the-art model for spatiotemporal prediction, which exploits diffuse convolution to extract the spatial dependence between time series and combines this spatial modelling ability with the gated cyclic unit. DCRNNs utilize the GRUs to capture the temporal dependencies, and then make predictions.
- *Graph WaveNet* [31]: Graph WaveNet is the state-of-the-art model for spatiotemporal prediction, which exploits the first-order approximate spectral convolution to extract the spatial dependence between time series, and combines this spatial modelling ability with causal expansion convolution.

We carefully selected the above seven baseline methods to cover various aspects. Among these baseline approaches, ARIMA and SVR are designed for individual time series; VAR considers the interactions among multiple time series; FC-LSTM and WaveNet are RNN-based and CNN-based deep learning models, which show the significant capability of modelling nonlinear and long-term dependencies for individual time series; DCRNNs and Graph WaveNet introduce graph convolution filter to exploit spatial dependencies among time series, which are the state-of-the-art in this domain. Note that Graph WaveNet can also infer and quantify the spatial dependencies from time series.

5.1.4 *Hyper-Parameters and Other Settings*

All of our experiments were conducted on an Ubuntu server [CPU: Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz, GPU: NVIDIA GTX 1080 Ti]. To make the comparison fair, the hyper-parameters were tuned for different models to achieve their best performance. In particular, we considered one-step adjacent vertices for VAR to predict the trend of central vertices. We set 128 temporal hidden states for each recurrent units, for FC-LSTM, DCRNNs, and MRST. In MRST, we empirically set $C = 13$ for the number of MFCCs to retain; set $d = 13$ for the dimension of the text feature vector; and set 128 spatial hidden states for graph convolution filters. Due to the great difference in the lengths of knowledge service documents, we removed stop words and truncated the text into sentences of the same length $l = 50$. As for the most sensitive hyper-parameters, i.e., predefined graph convolution depth K, we conducted several experiments to carefully study them and will discuss them in detail in Section 5.2.2.

5.2 Experimental Results and Analysis

5.2.1 *Main Results*

In order to compare the overall prediction accuracy of our MRST framework with those of baseline models, we conducted repeated experiments eight times under different initializations. Table 2 records the average values of MAE, RMSLE and SMAPE of different methods in Wiki-EN's 3-day, 7-day and 14-day prediction. In particular, this paper selects two spatiotemporal models, DCRNNs and Graph WaveNet, as the GCNs backbone of the MRST framework. By examining the results of the dataset, we noticed four consistent phenomena.

First, all neural network-based models, including our MRST, performed significantly better than the previous models. This is because RNN and TCN structures have a strong ability to model nonlinearity and long-term time dependence.

Second, by comparing the accuracy of ARIMA and VAR, we observed that although VAR considers the spatial dependencies among time series, its errors are not less than ARIMA. This shows that manual spatial dependencies may carry a lot of noise, and simple matrix operations are not enough to understand the intricate relationship among knowledge services.

Third, compared with other deep learning models without considering spatial features, the method based on spectral convolution has made significant progress. DCRNN and Graph WaveNet, which use spectral convolution to extract the spatial correlations among knowledge service

TABLE 2
Performance Comparison of MRST Framework and Other Baselines

Models	3 days			7 days			14 days		
	MAE	RMSLE	SMAPE	MAE	RMSLE	SMAPE	MAE	RMSLE	SMAPE
ARIMA	301.6	0.3329	15.48%	391.2	0.4341	22.24%	549.9	0.4223	23.91%
VAR	402.0	0.3709	19.89%	603.2	0.5288	30.04%	783.2	0.6456	35.48%
SVR	260.8	0.2502	14.05%	339.6	0.3230	20.01%	381.9	0.3575	21.98%
FC-LSTM	226.2	0.2245	11.95%	240.0	0.2528	13.71%	272.4	0.2850	16.72%
WaveNet	220.9	0.2261	12.06%	235.5	0.2534	13.79%	269.1	0.2861	16.80%
DCRNN	222.4	0.2229	12.02%	237.1	0.2511	13.85%	266.8	0.2833	15.73%
MRST(DCRNN)	218.3	0.2208	11.95%	234.2	0.2496	13.78%	265.1	0.2826	15.71%
Graph WaveNet	216.7	0.2231	11.90%	231.7	0.2510	13.67%	263.7	0.2826	15.76%
MRST(GraphWN)	214.0	0.2202	11.79%	230.2	0.2489	13.66%	261.4	0.2816	15.76%

Here 3, 7, 14 days refer to 3, 7, 14 steps predictions. The lower value reflects the higher prediction accuracy. The bold font highlights the best performance.

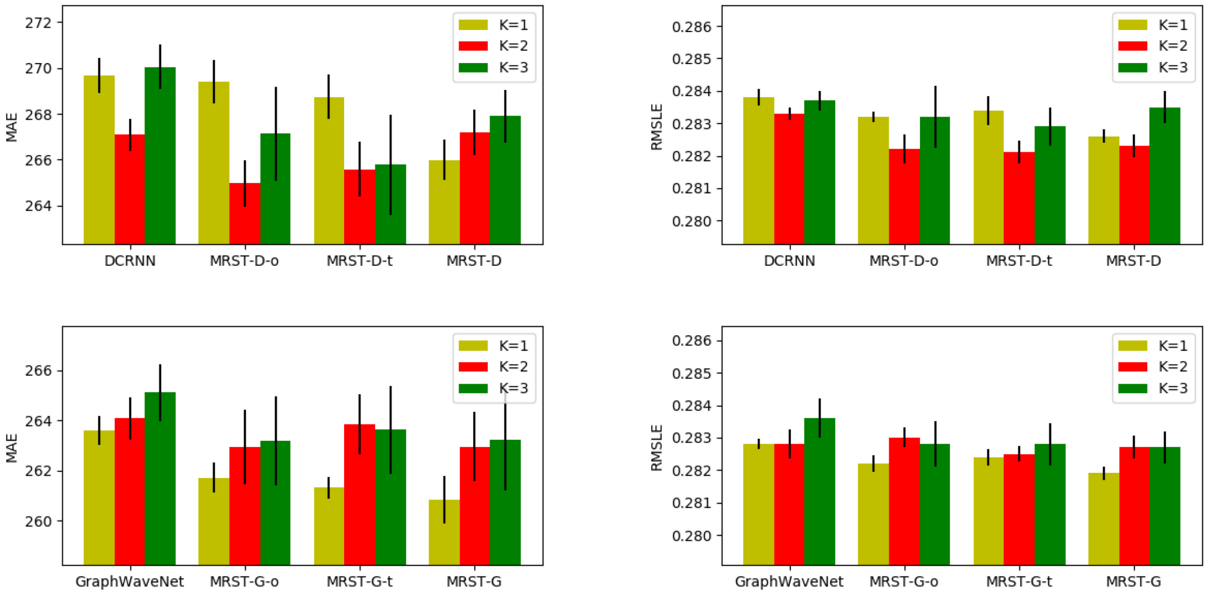


Fig. 4. Performance comparison for DCRNNs, Graph WaveNet and MRST framework with different graph convolution depth K and EIN components. In particular, MRST-t and MRST-o refer to EIN-t and EIN-o that are concerned.

nodes, have further reduced the prediction error compared with their basic models FC-LSTM and WaveNet, respectively. It is proved that the spectral convolution operator is effective in modelling complex service relationships. However, it is worth mentioning that in the process of training, Graph WaveNet has to embed learning for each time series. This makes the computational complexity $\mathcal{O}(n^2)$. On the English Wikipedia knowledge service data set, it seriously exceeded the memory of our experimental environment, so the self-adaptive adjacency matrix part of Graph WaveNet was omitted for the experiment.

Finally, MRST (DCRNN) and MRST (GraphWN) are improved compared with DCRNN and Graph WaveNet, respectively. This phenomenon shows the effectiveness of our MRST in improving service usage tendency prediction tasks, and EINs can infer better spatial dependencies than manual methods.

5.2.2 *Important Components and Hyper-Parameters Discussions*

In our MRST framework, the predefined number of graph convolution depth K of GCNs and two types of EIN components (EIN-o and EIN-t) are two significant concerns. In this section, we carefully compare the performance of the MRST framework with different K and EIN components to study how they couple with each other in our MRST framework. Specifically, in this part, we chose the spatio-temporal model DCRNNs and Graph WaveNet as backbones for the MRST framework. As introduced in Section 3, these two kinds of EINs use usage observation sequences and text information to infer the spatial dependence among knowledge services. We designed the following comparative experiment to investigate whether the two components work.

- DCRNN (or Graph WaveNet): does not contain any EIN component.
- MRST-D-o (or MRST-G-o): only includes EIN-o component. That is, only the usage observation sequences of the service are used to build spatial dependencies.
- MRST-D-t (or MRST-G-t): only includes EIN-t component. That is, only text information is used to build spatial dependencies.
- MRST-D (or MRST-G): contains both EIN-o and EIN-t components.

In addition, we studied the influence of $K = 1, 2, 3$. Fig. 4 reports the mean and standard deviation of MSE and RMSLE indexes under different settings. We observed that when DCRNN and Graph WaveNet are used as backbones, K has different effects on the model. The MSE and RMSLE of the DCRNN cluster all drop from 1 to 2, and then rebound. The increase of K increases the perceptual field of the model, so that more neighbor node information can be used to enrich the node representation. By fixing other settings and increasing K, the MAE and RMSLE of the Graph WaveNet cluster do not fall but rise. This is because Graph WaveNet itself has the problem that the effect of increasing K becomes worse on the Wiki-EN dataset. It can be observed that when $K = 2, 3$, the variance of the Graph Wavenet cluster on each index is large. This shows that under this setting, the convergence of learning results is not in place, resulting in more unstable results. We suspect the selection of K is related to the backbones in the framework. For the DCRNN cluster, K=2 best fits the Wiki-EN dataset. For the Graph WaveNet cluster, the best result is achieved when K=1.

We also observed that within the convolution depth K of each graph, the MRST framework with EIN components is better than DCRNN. In particular, the MRST-D framework is more accurate than both MRST-D-o and MRST-D-t prediction with only one EIN component. This shows that EIN can infer the high-quality spatial correlation of GCN. When more information is integrated to build spatial dependencies, the corresponding performance will be improved. By mining service multimodal information, EIN promotes model learning, richer spatial representation and GCN to make more accurate

TABLE 3
Phased Heuristics Ablation

MRST(DCRNN)	MAE	RMSLE	SMAPE
with heuristics	265.1 ± 2.9	0.2826 ± 0.0011	$15.71\% \pm 0.10$
w/o heuristics	273.2 ± 2.8	0.2855 ± 0.0009	$15.83\% \pm 0.10$
MRST(GraphWN)	**MAE**	**RMSLE**	**SMAPE**
with heuristics	261.4 ± 1.6	0.2816 ± 0.0007	$15.76\% \pm 0.08$
w/o heuristics	261.5 ± 1.9	0.2824 ± 0.0003	$15.72\% \pm 0.10$

predictions. This result can also be found in the Graph WaveNet cluster when $K = 1$. As mentioned earlier, Graph WaveNet's ability to use graph network information is degraded when $K = 2, 3$, which makes it difficult for the model to use this information to improve the effect even if we update the graph network structure. Therefore, the best result is still obtained when $K = 1$ in MRST-G.

5.2.3 Heuristics Ablations

To evaluate the effect of phased heuristics, we conducted eight ablation experiments for each configuration. Table 3 reports the experimental results when using DCRNN and Graph WaveNet (short for GraphWN) as the backbones of the MRST framework, and successively records MAE, RMSLE and SMAPE when using phased heuristics. By examining the mean and standard deviation of these indicators in Table 3, it can be observed that the phased heuristic driving shows different effects on MRST (DCRNN) and MRST (GraphWN). The performance of MRST (DCRNN) framework driven by phased heuristics is significantly better than that without it. MAE is improved by 3%, and RMSLE and SMAPE are improved by about 1% and 0.8%. Fig. 5 shows the loss change of phased heuristic driving in MRST (DCRNN) training. However, phased heuristic driving has little effect on MRST(GraphWN). Compared with those without heuristics, MAE, RMSLE and SMAPE of the framework with heuristics have changed by $+0.04\%$, $+0.3\%$ and -0.3%, respectively. Through the observation of the experimental results, we found that the MRST (GraphWN) framework without phased heuristic algorithm can also achieve the same MRST prediction error as the MRST (GraphWN) framework with it. Further analysis shows that this may be due to the different internal structures of DCRNN and Graph WaveNet.

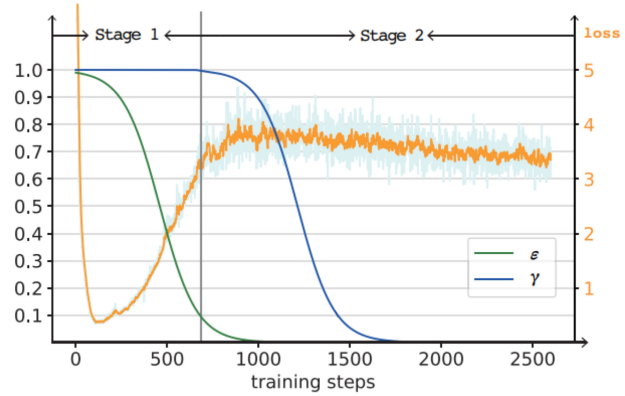


Fig. 5. Heuristics in MRST(DCRNN) framework.

TABLE 4
Time Cost and Model Size of Different Models

Model	Time	Parameters	Model	Time	Parameters
MRST-D-t	0.623	230,839	MRST-G-t	0.472	289,156
MRST-D-o	0.576	198,326	MRST-G-o	0.476	256,643
MRST-D	0.955	230,892	MRST-G	0.536	289,209

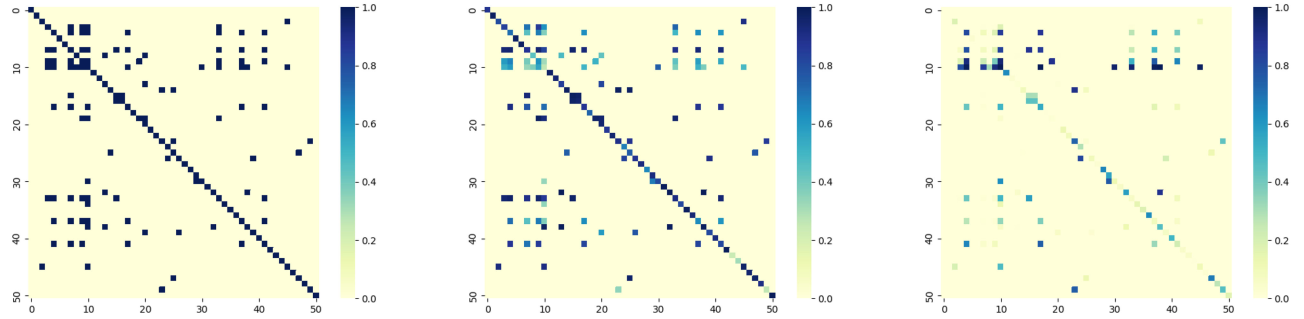
Fig. 5 shows the heuristics in MRST(DCRNN) framework, here we only start the first two stages. According to the inverse sigmoid decay, the first stage starts from the whole training process and ends when the decay factor ϵ drops to 0.1. In the first stage, the graph convolution network initially takes the ground truth as the model input for one-step prediction. In this case, since the prediction task becomes simple, the loss decreases rapidly. With the decay of sampling probability (the green curve in the figure), the graph convolution network gradually switches to the autoregessive mode, and the prediction task transitions from one-step prediction to multi-step prediction. Therefore, the value of the loss function gradually increases. In the second stage, loss is declining. GCN depends on the spatial dependence of EINs or prior knowledge, and stably depends more on the former. When γ reduce to 0.1, MRST start to count to early stop.

5.2.4 Training Efficiency

In this part, we discussed the training efficiency of MRST. We fixed the number of central vertices to 32 and the batch size to 32 and considered $K = 1$. Table 4 records the average training time and parameter number of each batch. The additional cost of MRST comes from the EIN module that infers spatial correlation. The fully connected layer and batch norm structure in EIN-o bring $2 \times C + 1$ and $2 \times C$ parameters to MRST-o, respectively. According to Algorithm 1, the EIN-t module increases the parameter amount of $L \times d + d$ when converting V_i to v_i. The fully connected layer and batch norm mechanism used to compare the correlation between v_i and v_j in pairs subsequently increase the parameter amount of $2 \times d + 1$ and $2 \times d$. In addition, since our MRST framework can integrate different spatiotemporal modules as the backbones, the number of parameters is also related to the selected modules.

5.2.5 Visualization

In order to intuitively understand the spatial dependencies among knowledge services inferred by our MRST framework, we took the top 51 knowledge services on the Wiki-EN dataset as an example to show the spatial correlation based on expert knowledge and the spatial correlation inferred by our model. The results are illustrated in Fig. 6. Based on the manual spatial dependency, the directed hyperlink relationship was used to define the initial spatial dependency between services, and the 0-1 value was used as weight. The final result is shown in Fig. 6a. EINs generate two spatial dependence patterns, namely Figs. 6b and 6c, where Fig. 6b is the spatial dependencies inferred by EIN-o



(a) handcraft graph

(b) EIN-o inferred

(c) EIN-t inferred

Fig. 6. Visualize the spatial dependencies of the first 51 knowledge services on the Wiki-EN dataset. The darker the color, the higher the correlation. Specifically, (a) is defined by hyperlinks; (b) is the spatial correlation inferred by EIN-o from the usage observation sequences. (c) is the spatial correlation inferred by EIN-t from text.

from the usage observation sequences. Fig. 6c is the spatial dependencies inferred by EIN-t from the text documents. Comparing Figs. 6b and 6c with 6a, it can be found that EINs are able to infer abundant spatial dependencies from multi-modal information of knowledge services.

We selected a representative knowledge service and visualized the prediction results to help better understand the model. Fig. 7 shows the ground truth and prediction results of the Wikipedia entry *COVID-19 pandemic*, which has 529 neighbors in the Wiki-EN dataset. In this typical case, the SMAPE of the DCRNNs, MRST-D, GraphWaveNet, and MRST-G are 9.21, 6.42, 8.80, and 7.44, respectively. We found that since the performance of GCN-based models greatly depends on the graph, they may be more likely to be misled by the not-so-important neighbors when facing the hot knowledge services with many neighbors. At the same time, the correlation between nodes may change over time, leading to the negative impact of outdated correlation between sequences on model results. Specifically, the sequences of *COVID-19 pandemic* and *COVID-19 pandemic in mainland China* in July and August 2020 strongly correlate. As time goes on, the concern of the English wiki community about the epidemic in Chinese Mainland has gradually weakened, and the correlation between the two knowledge services' sequences has also weakened. On the other hand, *2020 United States presidential election* began to show a strong correlation with *COVID-19 pandemic*. The successful vaccine development and reduced viral toxicity also led to a reverse correlation between *Severe acute respiratory syndrome*, *Remdisivir* and *COVID-19 pandemic*. Our MRST framework perceiving the dynamic correlation among sequences through multi-modal information such as text and time series can better utilize the graph and make more accurate predictions than other models.

6 Related Work

With the development of cloud computing, big data and Internet of Things, SOC (Service-Oriented Computing) has attracted more and more attentions from both industry and academia [29], [35], [36], [37]. The types of services are becoming increasingly abundant, covering software, data, storage, information and other aspects. Among them, Knowledge as a Service (KaaS) has become an emerging concept, which meets the needs of users by providing content-based delivery (data, information and knowledge). In recent years, a large number of services have been released into the service ecosystem, which then built complex service networks [5], [28], [33]. In this context, network services have brought many new challenges to the traditional services computing paradigm. For example, in terms of service tendency prediction, traditionally, people mainly focus on predicting individual time series[3], [38], [39], [40], while ignoring the interactions among services. In this paper, we discuss the trend predictions of knowledge services and propose a validated solution.

The prediction of usage observation series (also known as time series) has been a long-standing problem for decades. From the beginning, researchers have realized the prediction of a single time series by studying the linearity and nonlinearity of time series. Among them, the most classical methods include autoregressive integrated moving average(ARIMA) [6] and support vector regression (SVR) [22]. Work [19], [32] successfully predicted the usage trend of knowledge services (paper / patent citation count) using these methods. Some researchers also try to study the interaction between time series, among which the representative ones are vector autoregressive model (VAR) [25] and multiple-output SVR [27]. In recent years, with the development of deep learning, researchers begin to introduce deep learning model to improve the accuracy of time series prediction. Typical examples are recurrent neural network (RNN) and convolutional neural network (CNN). Recurrent neural networks (RNN) such as long short-term memory (LSTM) [15] and gated recursive unit (GRU) [9] can well capture the long-term dependence of time series. As an example related to our work, Wen et al. predicted the citation counts by appling RNNs [30]. Based on RNN with fully connected LSTM hidden units, Sutskever et al. proposed FC-LSTM [26], which is one of the classical methods to predict the trend of time series based on deep neural network model. Convolutional neural networks (CNN), such as dilated causal convolution [34] and gating mechanism [11] can also accumulate sequence information through receptive fields. WaveNet is considered a successful application [24].

With the development of graph neural networks(GNNs), researchers consider introducing spatial relationships to further improve prediction accuracy. Graph convolution network (GCN) is widely used in many tasks such as node

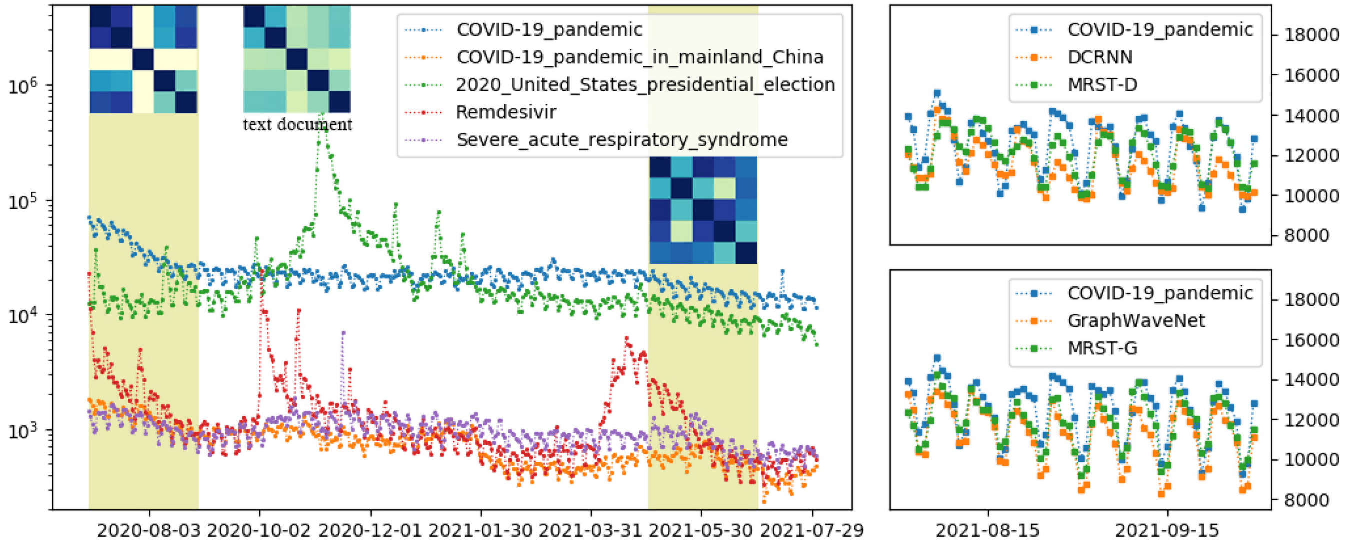


Fig. 7. The left figure shows the ground truth of the Wikipedia entry *COVID-19 pandemic* and its four neighbor knowledge service nodes from July 1, 2020, to July 31, 2021. The three matrices denote the correlation among the time series from July 1, 2020, to August 31, 2020, the correlation among the time series from May 1, 2021, to June 30, 2021, and the correlation among text documents, respectively; The right figure shows the predicted usage trends of *COVID-19 pandemic* from August 1, 2021, to September 30, 2021, generated by DCRNS, MRST-D, GraphWaveNet, and MRST-G.

classification, link prediction, and time series prediction because of its potential in aggregating graph structure information. Spectral-based Graph Convolutional Networks (GCNs) were first introduced by Bruna et al. [8], which incorporate spectral graph theory into deep learning models. After this work, many authors have proposed improvements, extensions and approximations of these spectral graph convolutions. Defferard et al. Proposed ChebNet [12], which uses Chebyshev polynomials to replace the convolution kernel in the spectral domain to reduce the complexity of parameter learning. Kipf and welling [16] made a local first-order approximation of spectral convolution, and expanded the receptive field by superimposing multi-layer linear GCN, which further improves the efficiency of GCN. Atwood and towsley [2] proposed a diffusion progressive neural network (DCNN), which defines rotation as a diffusion process across each node in the graph structure input to deal with the directed graph. In light of the large-scale graph and numerous nodes in the big data service scenario, researchers began to consider efficient and scalable methods to solve the model training challenges. Liu et al. [20] summarized the sampling methods for efficient training of graph convolutional networks. Luo and Wu et al. proposed latent tensor/factor analysis-based approaches [21], [33] that effectively perform representation learning on a dynamically weighted directed network.

Based on such efforts, some researchers consider introducing graph convolution network (GCN) to aggregate the spatial correlation between time series, so as to further improve the prediction accuracy. Diffusion convolution recurrent neural networks (DCRNNs) [17] and Graph WaveNet [31] are two representative works. DCRNNs uses diffusion convolution to extract the spatial correlation between time series, and combines this spatial modeling ability with gated cycle unit. Graph WaveNet uses the first-order approximate spectral convolution to extract the spatial correlation between time series, and combines this spatial modeling ability with extended causal convolution. On this basis, E-GCRNN [18] further considers the rapid evolution of knowledge service dependency. Although these models show attractive ability in using the spatial dependence between time series, the inferred results may be inaccurate because the vertices of time series are characterized by multi peaks and noises, and there is no obvious label to represent the similarity between vertices. So far, there are few studies on the spatial dependence of autonomous learning time prediction, while we believe that this method is very important to further improve the degree of freedom of spatiotemporal model. In addition to using time series to learn spatial dependencies, other relevant information of services (such as text information) can also be used to mine spatial dependencies between service nodes. Therefore, our MRST model considers the interaction between the information of multimodal knowledge service nodes, so as to effectively improve the prediction accuracy of the target knowledge service field.

7 Conclusion

In this paper, we have presented a new Multi-modal Reciprocal Spatio-Temporal (MRST) framework to predict the usage tendency of knowledge services. We designed two types of Edge Inference Networks (called EIN-o and EIN-t) to infer the spatial dependencies among knowledge services based on the information of usage observation sequences and text, respectively. Based on the inferred graphs, we integrated GCN-based spatiotemporal prediction models as backbones to make trend prediction. In such a reciprocal framework, EINs infer multi-modal directed weighted graphs to serve GCNs, and GCNs use these spatial correlations for prediction, and then introduces feedback to optimize EINs. Through the iterative joint learning process, the performance of EINs and GCNs both benefit from each other, and eventually lead to accurate spatio-temporal prediction. In addition, we also collected a new knowledge service data set Wiki-EN. A large number of experiments on this real data set have demonstrated that our proposed MRST framework significantly surpasses the baselines, and can learn meaningful spatial dependencies outside the predefined graphical structure.

In our future work, we plan to focus on the following three aspects: (1) to study how to provide more feedback to EINs to promote their exploration of possible links; (2) to study the function of EIN module when integrating different types of spatio-temporal prediction models, and further explore the efficiency of the model; (3) to study how our MRST can be applied to long series prediction.

References

[1] R. Abdullah, Z. D. Eri, and A. M. Talib, "A model of knowledge management system for facilitating knowledge as a service (KaaS) in cloud computing environment," in *Proc. Int. Conf. Res. Innov. Inf. Syst.*, 2011, pp. 1–4.

[2] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 1993–2001.

[3] M. Bano, D. Zowghi, N. Ikram, and M. Niazi, "What makes service oriented requirements engineering challenging? A qualitative study,," *IET Softw.*, vol. 8, no. 4, pp. 154–160, 2014.

[4] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 1171–1179.

[5] A. Bouguettaya et al., "A service computing manifesto: The next 10 years,," *Commun. ACM*, vol. 60, no. 4, pp. 64–72, 2017.

[6] G. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis Forecasting and Control*, 4th ed. Hoboken, NJ, USA: Wiley, 2008.

[7] J. S. Bridle and M. D. Brown, "An experimental automatic word recognition system," *JSRU Rep.*, vol. 1003, no. 5, 1974, Art. no. 33.

[8] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*.

[9] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[10] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, 2011.

[11] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 933–941.

[12] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[13] Y. Goldberg and O. Levy, "word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method," 2014, *arXiv:1402.3722*.

[14] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory,," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[17] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.

[18] 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.*, to be published, doi: 10.1109/TSC.2022.3145019.

[19] X. Liu, J. Yan, S. Xiao, X. Wang, H. Zha, and S. Chu, "On predictive patent valuation: Forecasting patent citations and their types," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1438–1444.

[20] X. Liu, M. Yan, L. Deng, G. Li, X. Ye, and D. Fan, "Sampling methods for efficient training of graph convolutional networks: A survey," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 2, pp. 205–234, Feb. 2022.

[21] X. Luo, H. Wu, Z. Wang, J. Wang, and D. Meng, "A novel approach to large-scale dynamically weighted directed network representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9756–9773, Dec. 2022.

[22] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*. Berlin, Germany: Springer, 2005.

[23] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental," *Pattern Recognit. Artif. Intell.*, vol. 116, pp. 374–388, 1976.

[24] A. V. D. Oord et al., "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.

[25] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression,," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[26] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[27] S. B. Taieb, A. Sorjamaa, and G. Bontempi, "Multiple-output modeling for multi-step-ahead time series forecasting,," *Neurocomputing*, vol. 73, no. 10–12, pp. 1950–1957, 2010.

[28] W. Tan, Y. Fan, A. Ghoneim, M. A. Hossain, and S. Dustdar, "From the service-oriented architecture to the web API economy," *IEEE Internet Comput.*, vol. 20, no. 4, pp. 64–68, Jul./Aug. 2016.

[29] Y. Wei and M. B. Blake, "Service-oriented computing and cloud computing: Challenges and opportunities," *IEEE Internet Comput.*, vol. 14, no. 6, pp. 72–75, Nov./Dec. 2010.

[30] J. Wen, L. Wu, and J. Chai, "Paper citation count prediction based on recurrent neural network with gated recurrent unit," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emerg. Commun.*, 2020, pp. 303–306.

[31] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," 2019, *arXiv:1906.00121*.

[32] S. Xiao et al., "On modeling and predicting individual paper citation count over time," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2676–2682.

[33] X. Xu, G. Motta, Z. Tu, H. Xu, Z. Wang, and X. Wang, "A new paradigm of software service engineering in big data and big service era,," *Computing*, vol. 100, no. 4, pp. 353–368, 2018.

[34] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.

[35] L.-J. Zhang, "EIC editorial: Introduction to the knowledge areas of services computing," *IEEE Trans. Services Comput.*, vol. 1, no. 2, pp. 62–74, Second Quarter 2008.

[36] L.-J. Zhang, J. Zhang, and H. Cai, *Services Computing*. Berlin, Germany: Springer, 2007.

[37] L.-J. Zhang, J. Zhang, J. Fiaidhi, and J. M. Chang, "Hot topics in cloud computing,," *IT Prof.*, vol. 12, no. 5, pp. 17–19, 2010.

[38] N. Zhang, J. Wang, and Y. Ma, "Mining domain knowledge on service goals from textual service descriptions," *IEEE Trans. Services Comput.*, vol. 13, no. 3, pp. 488–502, May/Jun. 2020.

[39] X. Zhang, J. Wu, X. Yang, H. Ou, and T. Lv, "A novel pattern extraction method for time series classification,," *Optim. Eng.*, vol. 10, no. 2, pp. 253–271, 2009.

[40] Y. Zhong, Y. Fan, K. Huang, W. Tan, and J. Zhang, "Time-aware service recommendation for mashup creation," *IEEE Trans. Services Comput.*, vol. 8, no. 3, pp. 356–368, May/Jun. 2015.

Ruyu Yan received the BS degree from Tsinghua University, China, in 2018. She is currently working toward the PhD degree with the Department of Automation, Tsinghua University, China. Her research interests include services computing, recommender systems, and time series prediction.

Haozhe Lin received the BS degree from Central South University, and the PhD degree from the Department of Automation, Tsinghua University, China. He is currently a postdoc with the Department of Automation, Tsinghua University, China. He received the Best Student Paper Award from the 2018 IEEE International Conference on Web Services. His research interests include services computing, time series prediction, and data mining.

Yushun Fan received the PhD degree in control theory and application from Tsinghua University, China, in 1990. He is currently a tenured professor with the Department of Automation, vice director of the National CIMS Engineering Research Center of China, director of the System Integration Institute, and director of the Networking Manufacturing Laboratory, Tsinghua University. He is a member of IFAC TC5.1 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 Alexander von Humboldt Stiftung, 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, system integration, modern service science, and technology.

Jia Zhang (Senior Member, IEEE) received the BS and MS degrees in computer science from Nanjing University, China, and the PhD degree in computer science from the University of Illinois at Chicago. She is currently Cruse C. and Marjorie F. Calahan Centennial chair in engineering, professor with the Department of Computer Science, Southern Methodist University. Her recent research interests center on data science infrastructure, with a focus on scientific workflows, software discovery, and knowledge graph. She has co-authored one textbook titled “Services Computing” and has published more than 180 refereed journal papers, book chapters, and conference papers. She is currently an associate editor of the *IEEE Transactions on Services Computing* (TSC).

Bing Bai received the BS and PhD degrees in control theory and application from Tsinghua University, China, in 2013 and 2018, respectively, and he is currently a senior researcher with the Cloud and Smart Industries Group, Tencent, Beijing, China. He received the Best Paper Award from the 14th IEEE International Conference on Services Computing (2017). His research interests include data mining and recommender systems.