Knowledge and Information Systems (2024) 66:4527–4548
https://doi.org/10.1007/s10115-024-02093-8

REGULAR PAPER

Dynamic bipartite network model based on structure and preference features

Hehe Lv[1] · Guobing Zou[1] · Bofeng Zhang[2] · Shengxiang Hu[1] · Chenyang Zhou[1] · Liangrui Wu[1]

Received: 7 August 2023 / Revised: 23 December 2023 / Accepted: 28 February 2024 /
Published online: 6 April 2024

Abstract
Based on the complex network, the relationship in the real complex system can be modeled, and the bipartite network is a special complex network, which can describe the complex system containing two kinds of objects. Although existing bipartite networks can model complex systems, conventional methods are restricted to a couple of limitations. (1) The dynamic interaction between nodes cannot be described over time. (2) The implicit features of nodes in the network cannot be effectively mined. Based on these, this paper proposes a dynamic bipartite network model (DBN) to model the dynamic interaction between two types of objects in real complex systems, and mine the structure features and preference features of nodes in the network. First, the dynamic interaction between two types of objects in a complex system is modeled as a dynamic bipartite network, which can reflect the interaction between objects in each time slice. Then, the structure features and preference features of each time slice are mined based on the dynamic bipartite network, where the structure features reflect the dynamic structural changes of the nodes, and the preference features reflect the potential preferences of the nodes. Finally, the features of each time slice are fused and input into the gate recurrent unit model to predict the interaction between nodes. Extensive experiments

✉ Guobing Zou
gbzou@shu.edu.cn

Hehe Lv
hhlv@shu.edu.cn

Bofeng Zhang
bfzhang@sspu.edu.cn

Shengxiang Hu
shengxianghu@shu.edu.cn

Chenyang Zhou
chenyangzhou@shu.edu.cn

Liangrui Wu
lr_w@shu.edu.cn

1 School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China

2 School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai 201209, China

are performed on a large-scale real complex system. The results show that DBN significantly outperforms state-of-the-art prediction methods in terms of multiple evaluation metrics.

Keywords Complex system · Dynamic bipartite network · Structure feature · Preference feature

1 Introduction

In recent years, with the rapid development of the Internet and big data, the amount of data has grown exponentially. These data do not exist in isolation, and there is an intricate relationship between them. Accurately characterizing complex relationships in data and predicting their evolution is a major challenge in complex science. Currently, modeling relationships in data based on complex networks is an effective means of analyzing complex systems. The nodes of a complex network generally represent the objects in the complex system, and the edges represent the association or interaction between the objects. Based on the complex network model, the objective laws existing in the complex system can be explored, and the behavioral features and social relationship features of human beings can be better recognized and understood [1–3]. A bipartite network is a special complex network that contains two different types of nodes. There are only edge relationships between different types of nodes and no edge relationships between nodes of the same type [4, 5].

Many complex systems can be modeled as bipartite networks, such as user–item purchase networks, user service recommendation networks, and author–paper citation networks [6, 7]. With the rapid growth in the number and interactions of objects in complex systems, modeling these complex systems based on bipartite networks becomes a challenging task. At present, the unweighted bipartite network and the weighted bipartite network are the two main models of the bipartite network [8–13]. The unweighted bipartite network models do not consider the weights of interactions between objects when modeling complex systems. The weighted bipartite network model needs to consider the weight of the interaction between objects, which can better reflect the strength of the interaction relationship. Although the above-mentioned bipartite network models can model the interaction between two types of objects in complex systems, the edge relationships in the bipartite network are static and cannot reflect the dynamic changes of the interaction between objects in complex systems over time. The interactions between two types of objects in real complex systems will change dynamically over time, and these dynamic interactions will affect the possible interactions between objects. Therefore, the study of dynamic bipartite network models is of great significance for predicting the interaction between two types of objects in real complex systems.

Traditional temporal networks are capable of capturing dynamic relationships between objects. However, traditional temporal networks primarily involve modeling dynamic relationships between objects of a single type and cannot conduct a detailed investigation into the dynamic nature of relationships between two sets of different objects. Currently, many traditional methods have achieved temporal interaction prediction between two types of objects in complex systems, such as the methods based on improved collaborative filtering [14, 15], matrix factorization [16, 17], and latent factor analysis [18, 19]. Although time series is introduced in traditional methods to predict the interaction between two types of objects through historical information, it still cannot reach the satisfaction of researchers. The underlying reason is that these methods cannot effectively model the dynamic relationship between two types of objects in complex systems to achieve dynamic interaction prediction. Therefore, how to model the dynamic interactions between two types of objects in complex systems and predict future interactions is a challenging task. Based on this, this paper proposes a dynamic bipartite network (DBN) to model the dynamic relationship between two types of objects in real complex systems. Compared to temporal networks, DBN specifically emphasizes the temporal evolution of interactions between two types of objects, providing a more refined understanding of temporal dynamics when dealing with bipartite relationships. DBN excels in accurately characterizing the dynamic relationships between two sets of objects within complex systems, dynamically revealing the structure features of networks and the preference features of nodes, offering valuable insights that extend beyond the scope of traditional temporal network analyses. The main contributions of this paper are as follows:

(1) The dynamic bipartite network model is proposed to effectively model the dynamic interactions between two types of objects in real complex systems, and to achieve effective prediction of future interaction relationships between objects.
(2) The structure feature representation method and preference feature mining method based on dynamic bipartite network are proposed, which can not only dynamically represent the node structure, but also dynamically mine the node preference.
(3) Extensive experiments are conducted on a real complex system. The experimental results demonstrate that DBN receives superior performance, comparing with competing methods in MAE and RMSE.

The content of this paper is primarily divided into seven sections. The first section outlines the main objectives and contributions of this paper. The second section introduces four works relevant to our research, encompassing the complex network, bipartite network model, feature representation method, and attention mechanism. The third section formulates the problem definition and motivation, presenting the key symbols and their descriptions. The fourth section mainly introduces the proposed DBN model from three aspects: structure feature representation, preference feature mining, and dynamic interaction prediction. The fifth section details the experimental setup, including the dataset, evaluation metrics, and competing methods. The sixth section conducts an in-depth analysis and discussion of the experimental results. The seventh section summarizes the work presented in this paper and provides future recommendations.

2 Related work

In order to facilitate the understanding of our DBN method, this section introduces related research work on complex network, bipartite network model, feature representation method, and attention mechanism.

2.1 Complex network

Complex networks serve as mathematical and computational models to characterize the diverse interactions among entities or individuals within complex systems. These interactions, to a certain extent, can reflect the structure, functionality, and evolutionary patterns of the system. Based on complex network analysis, the information transmission mechanism of the system can be revealed, the information propagation pathways between entities can be predicted, and the relationship between the system structure and function can be explored. With the increasing availability of information, real-world complex networks often incorporate attribute information describing nodes and their interconnections. Such networks, known as attributed networks, acknowledge the comparable significance of both node attributes and topological structure. Various algorithms have emerged to integrate network structure and node attributes for attributed graph clustering. For instance, Berahmand introduced a novel method called depth attribute clustering with high-order proximity preserve (DAC-HPP), which aims to better capture cluster structures in attributed graphs [20]. The method integrates structural and attribute node information in a global, simultaneous, and integrated manner, providing a unique solution for attributed graph clustering and advancing the potential of research in this domain. Additionally, Berahmand extended the symmetric nonnegative matrix factorization (SNMF) technique by proposing the weighted symmetric NMF (WSNMF) method [21]. This approach constructs a similarity matrix based on attribute vectors and seamlessly integrates it into the objective function using the Hadamard operator. This method effectively blends topological and attribute information, leading to enhanced clustering outcomes.

2.2 Bipartite network model

A bipartite network is a network composed of two types of nodes, and there are only edges between nodes of different types. Bipartite networks usually abstract objects in complex systems as nodes and the relationship between two different types of objects as edges [22–24]. Based on the bipartite network model, the objective laws existing in complex systems can be analyzed, and based on the analysis results, scientific basis for relevant decisions can be provided. For example, in a complex network based on the interaction between drug and target, the drug and target are abstracted as nodes in a bipartite network, and the interaction relationship is abstracted as edges. And based on network analysis methods such as link prediction and network evolution, new interactions between drugs and targets are predicted to provide a scientific basis for drug relocation [25, 26]. In recent years, the study of the information propagation law of complex systems based on the bipartite network model has attracted the attention. For example, link prediction based on the local structure of a bipartite network can help to study the law of information propagation in complex systems [27, 28]. Therefore, the bipartite network can effectively model the interaction between two types of objects in complex systems and conduct related application research.

2.3 Feature representation method

Feature representation can formally express and describe the features of things. In complex networks, the features of nodes are usually initialized based on one-hot encoding techniques. Taking the feature representation of a node as an example, only the value of the position indicated by the node ID is assigned as 1, whereas the values of remaining positions are set by 0. That is, the dimension of a node feature is equal to the number of all nodes of the same type. For example, the NDMF model initializes the features of users and services based on one-hot encoding, and then integrates user neighborhood selected by a collaborative way into an enhanced matrix factorization model via deep neural network (DNN), achieving excellent service recommendation performance [29]. The RNCF model firstly initializes the features of users and services based on the one-hot encoding technique, then transforms the latent features of users and services based on the embedding layer, and finally inputs the latent features into the gate recurrent unit (GRU) model to implement user service recommendation [30]. The feature representation based on the one-hot method is usually very sparse and cannot effectively express the features of nodes. Therefore, this paper represents node features based on the neighbors of nodes, which can not only represent the structure features of nodes, but also reflect the dynamic interaction over time.

2.4 Attention mechanism

The attention mechanism can help the model distinguish important features and pay attention to local important information to improve the performance of the model [31]. It is generally reflected in the form of weights and is widely used in many fields such as node embedding, natural language processing, and image analysis. For example, heterogeneous graph attention network (HAN) obtains important nodes based on node-level attentions and important meta-paths based on semantic-level attentions [32]. The node-level attention aims to learn the importance between a node and its meta-path, while the semantic-level attention can learn the importance of different meta-paths. The model can capture the complex structures and rich semantics behind heterogeneous graph. In recommender systems, hierarchical attention cooperative neural networks (HACN) model users and items separately based on the review texts, and then enrich the feature representations of users and items from the review texts based on two hierarchical attention mechanisms, respectively [33]. The model can adaptively enhance the feature representation of users and items, make full use of effective information, and reduce the interference of irrelevant information. Therefore, this paper distinguishes the importance of features in different time slices based on the attention mechanism.

The main motivation of this paper is to accurately model the dynamic relationships between two types of objects in complex systems and dynamically extract potential features from the network to achieve predictions of latent relationships among complex objects. Therefore, this paper proposes a dynamic bipartite network model, aiming to accurately model the dynamic relationship between two types of objects. The network can dynamically track the network structure and reflect node preferences in real time, thereby enhancing the dynamics and precision of feature representation, achieving accurate prediction of potential relationships between complex objects.

3 Problem formulation

In this section, we first focus on the understanding of dynamic bipartite network model by a set of formal definitions, which are explained by concrete examples. The main symbols and descriptions of this paper are detailed in Table 1.

Definition 1 (*Complex Network*) The complex network can be defined as $G = (V, E)$, where V represents the set of nodes and E denotes the collection of edges between nodes. Complex networks can effectively characterize various complex systems, including transportation networks, financial systems, brain neural networks, and more. Complex networks provide a powerful framework for studying the interactions and relationships between different components in various systems, helping to better understand and optimize the structure and behavior of complex systems.

Definition 2 (*Unipartite Network*) A unipartite network can be defined as $G_{\mathrm{un}} = (V, E)$, where V represents the set of nodes of the same type and E represents the set of edges between nodes. This network primarily characterizes complex systems composed of a single type of object. For instance, social connections between individuals can be depicted by a

Table 1 Symbols and their descriptions

Symbol	Describe	Symbol	Describe
u and v	Represents two types of nodes	x	Initialization feature
t	Time slices	l	The type of edge
e	The weight of edge	q	Structure features
W and U	Parameter matrix	b	Bias
Q	The structure feature of node pair	S	Similarity feature
p	Preference feature	P	The preference feature of node pair
T	Total feature	w	Attention weight
f	Temporal feature	r	Forget gates
h	Output of the GRU	z	Update gate
$\hat{y}$	Prediction weight	J	Loss value
α	Regularization term coefficient	N	The number of node pairs
$\oplus$	Concatenation operation	$\odot$	Multiplication operation by element

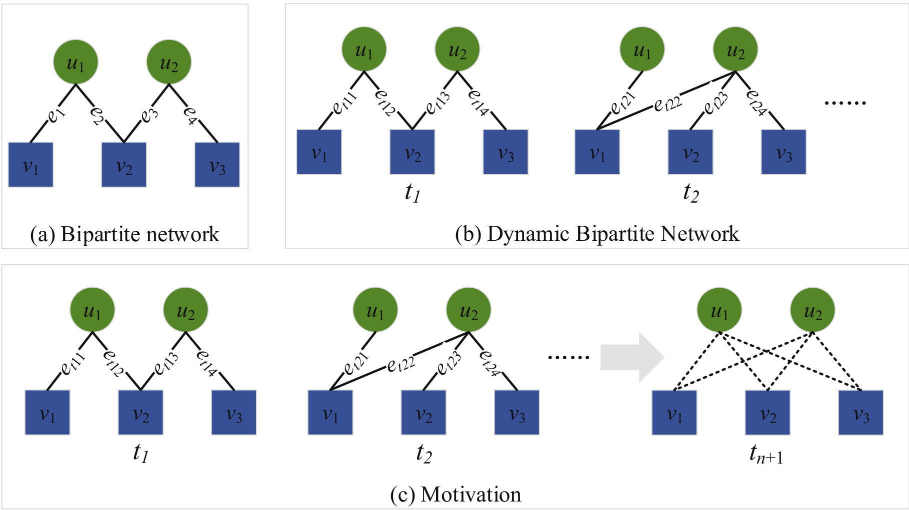


Fig. 1 Examples of bipartite network, dynamic bipartite network, and motivation. In this context, u and v represent nodes of different types, e denotes the weight of edge, t signifies time slices, and l indicates the type of edge connecting nodes

unipartite network, where individuals are abstracted as nodes in the network, and social relationships are abstracted as edges.

Definition 3 (*Bipartite Network*) A bipartite network is defined as $G_{\text{bi}}=(U, V, E)$, where $U = \{u_1, u_2, \ldots\}$ denotes one type of node, $V = \{v_1, v_2, \ldots\}$ denotes another type of node, and $E = \{e_1, e_2, \ldots\}$ represents the edge between nodes in set U and set V. Complex systems in the real world, composed of two types of objects, can be effectively described by bipartite networks. For example, the purchase relationship between users and items is described as a bipartite network, in which users and items are abstracted into different types of nodes in the network, and the purchase relationship is abstracted into edges. An example of a bipartite network is shown in Fig. 1a, where $U = \{u_1, u_2\}$, $V = \{v_1, v_2, v_3\}$, and $E = \{e_1, e_2, e_3, e_4\}$.

Definition 4 (*Dynamic Bipartite Network*) A dynamic bipartite network is defined as $G_T = (U, V, T, E)$, where $U = \{u_1, u_2, \ldots\}$ denotes a type of node and $V = \{v_1, v_2, \ldots\}$ denotes another type of node, $T = \{t_1, t_2, \ldots\}$ is a set of time slices and $E_t = \{e_{t1}, e_{t2}, \ldots\}$ represents the edge between U node and V node in different slices. In the real world, complex systems composed of two types of objects typically evolve dynamically over time. The traditional bipartite networks are no longer effective in modeling the dynamic relationships between two types of objects. Therefore, we propose a dynamic bipartite network model to capture this dynamism. For instance, the dynamic relationships between users and items can be abstracted into a dynamic bipartite network, where users and items are represented as nodes in the network, and the dynamic relationships between users and items are abstracted as dynamic edges. An example of a dynamic bipartite network is shown in Fig. 1b, where $U = \{u_1, u_2\}$, $V = \{v_1, v_2, v_3\}$, and $E_{t1} = \{e_{t11}, e_{t12}, e_{t13}, e_{t14}\}$ in time slice t_1. The dynamic bipartite network can reflect the dynamic change of the edge relationship between nodes in different time slices.

Definition 5 (*Motivation*) In a dynamic bipartite network G_T, the motivation of this paper is to extract the features of target node pairs u and v in time slices $t_1, t_2, \ldots t_n$, and predict the edge relationship between u and v in time slice t_{n+1} based on these features. As shown in Fig. 1c, this paper aims to predict the interaction between nodes u and v in time slice t_{n+1} based on the bipartite network of time slice $t_1, t_2, \ldots t_n$.

4 DBN model

The overall framework of DBN is illustrated in Fig. 2. The motivation of DBN is to predict the interaction between nodes based on historical features. DBN consists of three independent components, including structure feature representation, preference feature mining, and dynamic interaction prediction. In structure feature representation, the structure features of nodes are characterized based on the neighbors of each time slice, which can reflect the dynamic structural information of nodes. In preference feature mining, the potential interaction preference of nodes is mined based on similarity, which can represent the potential preference features of nodes. In dynamic interaction prediction, the structure features and preference features are firstly concatenated, then the concatenated features are multiplied by the attention weight of each time slice, and finally, these features are fed into the gate recurrent unit (GRU) model to predict the interaction of node pairs in the next time slice.

4.1 Structure feature representation

The most intuitive feature of a node in a bipartite network is its connected neighbors. Therefore, the initialization feature of the node in this paper is represented by the neighbors. Taking the initialization feature representation of the u node as an example, the values of the positions where the node has neighbors are assigned as 1, whereas the values of remaining positions are set by 0. That is, the dimension of the u node feature is equal to the number of all v nodes. Table 2 shows the initialization feature representation of u_1 node in time slice t_1, where the number of v nodes is 6 and the feature dimension of u is 6. The u_1 node is connected to v_1, v_2, and v_6, respectively, then the first position, the second position, and the sixth position of the u_1 feature are assigned as 1, and the other positions are 0. Based on this method, the initialization feature x_{tu} of node u and the initialization feature x_{tv} of node v are obtained

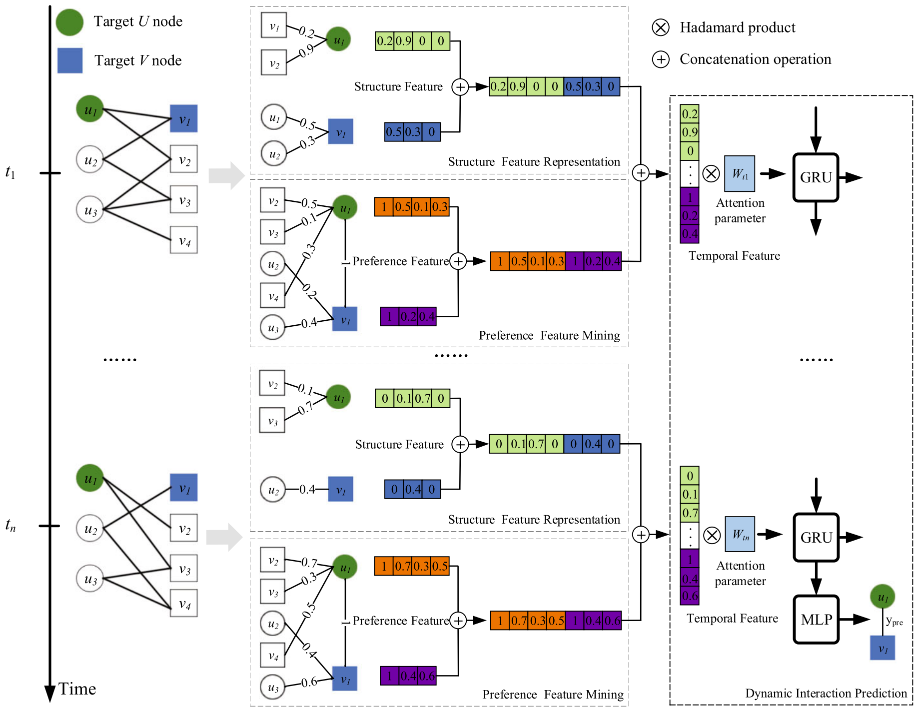


Fig. 2 Overall framework of DBN. First, a DBN is constructed based on different time slices. Then, structure feature representation and preference feature mining are performed. Finally, dynamic interaction prediction is performed based on temporal features and GRU model

Table 2 Feature representation of nodes

U	V	Temporal slice	Initialization feature (x_u)	Weight (e_u)	Structure features (q_u)
u_1	v_1, v_2, v_6	t_1	110,001	0.2,0.8,0,0,0,0.5	0.2 0.8 0 0 0 0.5
u_1	v_1, v_3, v_6	t_2	101,001	0.1,0,0.2,0,0,0.1	0.1 0 0.2 0 0 0.1
u_2	v_4, v_5	t_1	000,110	0,0,0,0.2,0.1,0	0 0 0 0.2 0.1 0
u_2	v_1, v_3, v_6	t_2	101,001	0.5,0,0.3,0,0,0.6	0.5 0 0.3 0 0 0.6

in time slice t. This method can not only represent the interaction information of nodes in a certain time slice, but also reflect the dynamic interaction changes of nodes over time.

To further represent the structure features of nodes, this paper introduces the weight feature e_t based on the initialization feature. Specifically, e_{tu} represents the weight of the edge between node u and its neighbors in time slice t, and the weight of no edge is 0. Based on the product of the initialization feature x_{tu} and the weight e_{tu}, the structure feature q_{tu} of u is obtained in the time slice t. Similarly, the structure feature q_{tv} of v can be obtained. The specific formulas are shown in (1)–(2).

$$q_{tu} = x_{tu} \odot e_{tu} \tag{1}$$

$$q_{tv} = x_{tv} \odot e_{tv} \tag{2}$$

where $\odot$ means that each corresponding entry in the two feature vectors is multiplied. As shown in Table 2, there are edges between u_1 and v_1, v_2, v_6 in the time slice t_1, and the corresponding weights are 0.2, 0.8, 0, 0, 0, 0.5, respectively. Based on the product, the structure feature q_{t1u1} of u_1 is obtained.

Since nodes in real bipartite networks have fewer neighbors, the structure features of nodes are relatively sparse. To transform a high-dimensional and sparse structure feature vector into a densely low-dimensional one, a fully connected network is used to perform dimensionality reduction, which is formalized as (3)–(4).

$$q'_{tu} = \sigma\left(W^q_{tu} q_{tu} + b^q_{tu}\right) \tag{3}$$

$$q'_{tv} = \sigma\left(W^q_{tv} q_{tv} + b^q_{tv}\right) \tag{4}$$

where σ is the activation function based on Relu, W^q is the learning parameters, and b^q is the bias. After performing the dimensionality reduction, the features of u node and v node are concatenated to obtain the structure feature Q_{tuv} of the node pair u-v in time slice t. Specifically, it is formalized as (5).

$$Q_{tuv} = q'_{tu} \oplus q'_{tv} \tag{5}$$

where $\oplus$ represents the concatenation operation of two feature vectors. The structure features can not only represent the structural information of nodes, but also reflect the dynamic structural changes of nodes in different times.

4.2 Preference feature mining

In a bipartite network, the similarity between nodes of the same type may have an impact on the interactions between nodes of different types. For example, there is an interaction between node u_1 and node v_1, while node v_1 and node v_2 are similar, and there may be potential interactions between node u_1 and node v_2. Based on this, this paper proposes a preference feature mining method based on the similarity of nodes. First, the similarity between nodes of the same type is measured based on structure features. Then, the preference relationship between nodes is established based on similarity, where the weights correspond to the similarity coefficients. Finally, based on the structure feature representation method, the preference features P_{tu} and P_{tv} of node u and node v in time slice t are represented.

An example of preference feature mining in time slice t is shown in Fig. 3. First, the similarity feature S_{tu1} of u_1 is obtained based on the feature similarity between u_1, u_2, and u_3. Then, the similarity feature S_{tv1} of v_1 is obtained based on the feature similarity between v_1, v_2, v_3, and v_4. Due to the interaction between u_1 and v_1 in the original data, there may be preference relationships between u_1 and v_2, v_3, v_4 (v_2, v_3, v_4 are similar nodes of v_1), and there may be preference relationships between v_1 and u_2, u_3 (u_2, u_3 are similar nodes of u_1). The weights of the preference relationships in this paper correspond to the elements of the similarity feature. The similarity between u_1 and u_2, u_3 is 0.4, 0.2, respectively, and the weight of preference relationship between v_1 and u_2, u_3 is 0.4, 0.2. The similarity between v_1 and v_2, v_3, v_4 is 0.5, 0.1, 03, and the weight of the preference relationship between u_1 and v_2, v_3, v_4 is 0.5, 0.1, 03. There is an interaction between u_1 and v_1 in the original data. Therefore, the weight of the preference relationship between them is 1. Finally, based on the structural feature representation method, the preference feature P_{tu1} of node u_1 and the preference feature P_{tv1} of node v_1 are represented in time slice t, where P_{tu1} is equal to S_{tv1}, and P_{tv1} is equal to S_{tu1}.

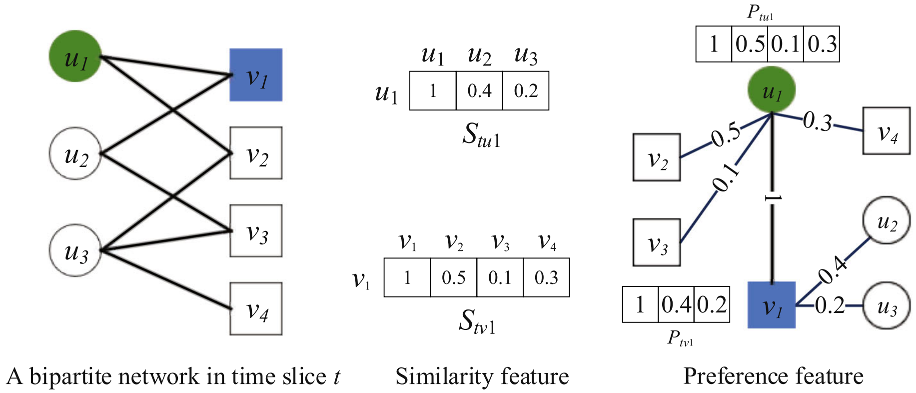


Fig. 3 An example of preference feature mining in time slice t. In this context, u and v denote nodes of distinct types; S represents similarity features, P represents preference features, and t signifies the time slice

The similarity between nodes of the same type is calculated based on structure features and Pearson correlation coefficient (PCC), as shown in formula (6).

$$\mathrm{sim}(x, y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x}) (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x}) \sum_{i=1}^{n} (y_i - \bar{y})}} \tag{6}$$

where x and y represent two nodes of the same type, i represent the position of the feature, and $\bar{x}$ and $\bar{y}$ represent the feature mean. Based on similarity calculation, the similarity feature S_{tu} of u node in time slice t and the similarity feature S_{tv} of v node in time slice t are obtained. Based on the similarity feature S_{tv}, the preference feature p_{tu} of the node u in the time slice t is represented, as shown in formula (7). Similarly, the preference feature p_{tv} of node v in time slice t can be obtained, as shown in formula (8).

$$p_{tu} = S_{tv} \tag{7}$$

$$p_{tv} = S_{tu} \tag{8}$$

Due to the large number of nodes in a real bipartite network, the preference feature vector is high-dimensional. To transform a high-dimensional feature vector of a preference feature into a densely low-dimensional one, a fully connected network is used for dimensionality reduction, as shown in formula (9)–(10). The preference features of u and v are concatenated to obtain the preference feature P_{tuv} for node pair u-v in time slice t, as shown in formula (11).

$$p'_{tu} = \sigma \left(W^{p}_{tu} p_{tu} + b^{p}_{tu} \right) \tag{9}$$

$$p'_{tv} = \sigma \left(W^{p}_{tv} p_{tv} + b^{p}{}_{tv} \right) \tag{10}$$

$$P_{tuv} = p'_{tu} \oplus p'_{tv} \tag{11}$$

where σ is the activation function based on Relu, W^P is the learning parameters, b^P is the bias, and $\oplus$ represents the concatenation operation of two feature vectors. The preference features can reflect the interaction preferences of nodes and provide personalized information for potential interactions.

4.3 Dynamic interaction prediction

Based on the structure feature Q_{tuv} and the preference feature P_{tuv} of the node pair u–v in the time slice t, the total feature T_{tuv} of the time slice t is obtained by concatenation, as shown in formula (12).

$$T_{tuv} = Q_{tuv} \oplus P_{tuv} \tag{12}$$

This paper predicts the interaction based on the features of node pairs in different time slices. The features of different time slices have different effects on the interaction prediction. Therefore, an attention mechanism is introduced to learn the importance of features in different time slices. Based on the adaptation of the neural network, the attention weights of features in different time slice are automatically learned. The attention weight w_t in the time slice t is obtained by learning, and the temporal feature f_{tuv} is obtained based on the attention weight w_t and the total feature T_{tuv}, as shown in formula (13).

$$f_{tuv} = w_t T_{tuv} \tag{13}$$

In this paper, the temporal feature f_{tuv} is fed into the gate recurrent unit (GRU) model to mine implicit information. Given a time slice t, the extraction process of implicit information is expressed as formula (14).

$$r_t = \sigma \left(W_r f_{tuv} + U_r h_{t-1}\right) \tag{14}$$

where f_{tuv} represents the temporal feature of the time slice t, σ is the activation function, and the forget gates r_t are calculated by the current input f_{tuv}, the weight coefficients W_r and U_r, and the output h_{t-1} of the previous time slice. At the starting time slice t_0, h_{t-1} is a randomly initialized feature vector. Based on the r_t in the above formula, $\tilde{h}_t$ of the time slice t can be calculated, and the formula is shown in formula (15).

$$\tilde{h}_t = \tanh \left(W f_{tuv} + U \left(r_t \odot h_{t-1}\right)\right) \tag{15}$$

where $\odot$ represents multiplication operation by element. When r_t is close to 0, the information of the current state is the main; when r_t is close to 1, the information in the historical data needs to be retained, and the input information at the current state is ignored. A weighting factor z_t is learned based on $\tilde{h}_t$ of the above state, and the features of the current output are updated according to the weight, as shown in formula (16) and (17).

$$z_t = \sigma \left(W_z f_{tuv} + U_z h_{t-1}\right) \tag{16}$$

$$h_t = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{17}$$

where z_t is the updating weight to be learned in model training and h_t is the output result of the GRU in time slice t. The final feature vector h_k is output based on a set of GRUs. Then the edge weight $\hat{y}$ between node u and node v is obtained based on the fully connected network, as shown in formula (18).

$$\hat{y} = \mathrm{Relu} \left(W h_k + b\right) \tag{18}$$

where W and b are the learned parameters from model training, respectively.

Since the weight prediction between node u and node v is a regression problem, mean squared error (MSE) is taken as the optimization objective, and its objective function is as formula (19).

$$J=\alpha^{*}\frac{1}{N}\sum_{i=1}^{N}\left(y_i-\hat{y}_i\right)^2+(1-\alpha)^{*}\sum_{j}w_j^2 \tag{19}$$

where $\hat{y}_i$ is the model prediction weight between node u and node v, y_i is the real weight between node pairs, N is the number of node pairs, w_j is a parameter value in the model; $\sum_j w_j^2$ is the regularization term of the model, which is used to avoid overfitting in model training; α is used to balance the importance of the regularization term and is generally set to a value approximately close to 1 after iterative validation in experiments.

5 Experimental setup

This paper performs experiments on a workstation equipped with Intel Xeon Gold 6132 CPU, NVIDIA GeForce GTX 1080Ti GPU, and 192 GB RAM. The DBN module is implemented by Python 3.8.3 with PyTorch 1.11.0.

5.1 Dataset

This paper conducts experiments based on the Rtdata dataset in WS-Dream [34], which is a real-world service invocation dataset including 27,392,643 invocation records for 142 users and 4,500 web services. The invocation records between users and services are divided into 64 different time slices. The divided training datasets span from 5 to 20% with a density interval of 5%. Because the invocations between users and services in the real world are very sparse, this kind of partitioning on dataset can simulate the realistic application situation as much as possible.

5.2 Evaluation metrics

The weight prediction of edges between nodes in this paper is a regression problem. Mean absolute error (MAE) and root mean square error (RMSE) were used as evaluation metrics to measure the accuracy of predictions in the experiments. MAE is defined as formula (20).

$$\mathrm{MAE}=\frac{\sum_{uv}\left|y_{uv}-\hat{y}_{uv}\right|}{N} \tag{20}$$

where y_{uv} represents the true weight of edge between node u and node v, $\hat{y}_{uv}$ is the predicted value, and N is the number of samples. The MAE is linear to the deviation of prediction value, and all individual differences are weighted equally in the average. Therefore, MAE cannot well reveal outliers with large deviations between the predicted value and the true value. To this end, outliers with large deviations between the predicted and true values are measured based on RMSE. RMSE is defined as formula (21).

$$\mathrm{RMSE}=\sqrt{\frac{\sum_{uv}\left(y_{uv}-\hat{y}_{uv}\right)^2}{N}} \tag{21}$$

In the experiments, MAE reflects the overall accuracy of prediction, which averages absolute deviations to the original values. Compared with MAE, RMSE is more sensitive to individual outliers by representing a relatively higher weighting to large errors on predicted values.

5.3 Competing methods

To evaluate the effectiveness of DBN, eight state-of-the-art methods are used, including CLUS, TMF, PLMF, RTF, K-SLOPE, NRCF, TUIPCC, and DeepTSQP. They are described as below.

CLUS [35]: CLUS excels in predicting the reliability of atomic web services, serving as a dedicated reliability prediction model for such services. It estimates the reliability of ongoing service invocations based on data collected from previous invocations. In comparison with other methods, CLUS strength lies in addressing unique challenges related to the reliability of individual web services by combining the user, service, and environment specific parameters of the invocation context.

TMF [36]: TMF, based on the integration of QoS time series, offers a two-phase QoS prediction for cloud service recommendations. This approach is a time-aware matrix factorization model that takes service invocation time as a dynamic factor in the model, and then predicts missing QoS values based on adaptive matrix factorization, thereby recommending high-quality services to target users. In comparison with other methods, TMF integrates time awareness into QoS prediction for cloud services, enhancing the accuracy of recommendations in dynamic cloud environments.

PLMF [37]: PLMF is a matrix factorization method based on personalized LSTM, designed for predicting online service recommendations, particularly suitable for real-time and personalized prediction scenarios. This approach can capture the dynamic latent feature representations of multiple users and services, and prediction model can be updated in time to handle emerging data. In comparison with other methods, PLMF dynamically updates latent features based on current observations, limited historical data, and some long-term retained information, enabling personalized and customized online QoS prediction services.

RTF [38]: RTF is a novel time-aware recommendation method based on deep learning, capable of capturing both long-term and short-term dependency patterns between users and services. Its primary focus is on addressing challenges related to the processing of new data and the difficulty in capturing dynamic long-term dependency patterns. The approach is based on personalized gated recurrent unit (PGRU) and generalized tensor factorization (GTF) to memorize long-term and short-term dependency patterns between users and services, and through comprehensive analysis predict unknown invocation. In comparison with other methods, RTF can integrate tensor factorization with deep learning to deliver time-aware service recommendations, effectively memorizing the dynamic temporal behavior of users and significantly alleviating the problem of data sparsity in real world.

K-SLOPE [39]: K-SLOPE is a time-aware web service recommendation system designed to address the impact of time effects on user choices. This method combines the k-means clustering algorithm with Slope-One collaborative filtering prediction technology to effectively provide service recommendations for the target user. In comparison with other methods, K-SLOPE excels in integrating time-aware user clustering with multi-valued QoS prediction for web service recommendations, thereby enhancing the quality of recommendations.

NRCF [30]: RNCF is a recurrent neural network based collaborative filtering for QoS prediction in the Internet of Vehicles (IoV). It primarily addresses the dynamic variations in QoS values within the context of the Internet of Vehicles (IoV), stemming from objective factors like changes in the physical environment and network conditions. This method adds a multilayer GRU structure to the framework of neural collaborative filtering, which can model the dynamic state of the physical environment or network conditions. At the same time, this method can share invocation records in different time slices. In comparison with other methods, RNCF excels in mitigating data sparsity issues by leveraging historical invocation records from different time slices. It also demonstrates robust feature mining capabilities in capturing the latent representations of multiple users and services.

TUIPCC [15]: TUIPCC is an improved collaborative filter based missing QoS prediction approach, designed to cope with the timeliness characteristics of QoS values. This method can not only filter out historical QoS values with good timeliness to accurately represent the current network environment, but also consider valuable information from time historical QoS values. This method achieves accurate missing QoS prediction based on a time-aware collaborative filtering mechanism. In comparison with other methods, TUIPCC minimizes the impact of dynamic application environments to the greatest extent possible and selects genuinely similar users (or services).

DeepTSQP [40]: DeepTSQP is a QoS prediction method based on deep learning, designed to address the issue of temporal-aware service QoS prediction by dynamic feature representations of users and services. This method can reflect the dynamic temporal feature of a user and a service along with the variations of interactive invocations over time. Moreover, GRU is applied to mine temporal aggregated features across multiple time slices, which can more effectively capture the implicit nonlinear relationship between users and services, thereby improving the performance of service QoS prediction. In comparison with other methods, DeepTSQP excels in precisely represent the features of users and services at each time slice, enabling comprehensive temporal-aware service QoS prediction based on deep neural networks.

6 Results and discussion

In this section, we verify the prediction performance of our proposed method. Compared with existing prediction methods, our proposed method has better performance.

6.1 Comparison of model performance

After the model training is completed, the test samples are input into the model to obtain the prediction results, and then compared with the state-of-the-art method in terms of MAE and RMSE, where MAE represents the prediction accuracy and RMSE represents the prediction stability. Table 3 shows the experimental results compared with state-of-the-art methods in terms of MAE and RMSE. At a specific density of 0.1, TUIPCC achieves excellent performance on both MAE and RMSE, but does not hold well at the remaining densities. Overall, among all state-of-the-art competing methods, DeepTSQP performs the best at different densities in terms of MAE and RMSE. However, it can be observed that our proposed DBN method outperforms the most effective one DeepTSQP, which has the highest prediction accuracy among all competing methods. The main reason can be explained by the following two aspects. First, structure features and preference features can reflect the changes of interaction and the preference of interaction. The latent features of nodes can be effectively mined based on structural features and preference features. Second, an attention mechanism is introduced to distinguish feature importance in different time slices, providing important features for the model.

Table 3 Experimental results compared to competing methods

Methods	Density = 5%		Density = 10%		Density = 15%		Density = 20%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CLUS	0.7842	1.8921	0.7542	1.903	0.735	1.9046	0.7185	1.8948
TMF	0.7801	1.7698	0.5802	1.4079	0.8315	2.1847	0.7132	1.6593
PLMF	0.7267	1.7059	0.6786	1.6126	0.6582	1.5749	0.6444	1.5525
RTF	0.7896	1.8613	0.5772	1.2218	0.8253	2.0935	0.6917	1.6419
K-SLOPE	0.8127	1.9363	0.6082	1.3394	0.8581	2.3516	0.7536	1.7115
NRCF	1.048	1.616	1.010	1.546	0.974	1.503	0.958	1.470
TUIPCC	0.7814	1.7761	0.5767	1.2076	0.8196	2.0595	0.6970	1.6358
DeepTSQP	0.6980	1.5937	0.5794	1.4572	0.5202	1.3366	0.4526	1.2140
DBN	0.5483	0.7407	0.4845	0.7621	0.4161	0.6794	0.4222	0.7316

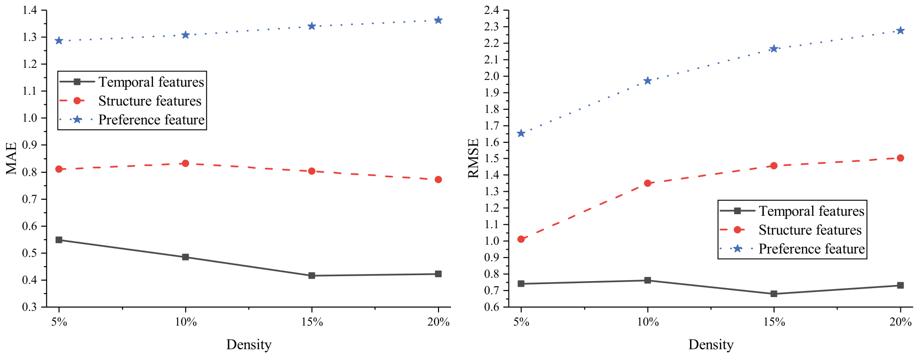


Fig. 4 Importance analysis of structure feature and preference feature. The three lines, respectively, represent the performance of preference features, structure features, and temporal features as input under different density conditions. The results show that temporal features based on the concatenation of structure features and preference features can achieve the best prediction performance

6.2 Analysis of structure feature and preference feature

In this paper, based on the concatenation of structure feature and preference feature as temporal feature, we analyze the impact of structure feature and preference feature on the prediction performance of DBN. There are mainly three situations: (1) based on structure feature as temporal feature; (2) based on preference feature as temporal feature; and (3) the concatenation of structure feature and preference feature proposed in this paper is used as temporal feature. The results are shown in Fig. 4. The temporal feature based on the concatenation of structure feature and preference feature can achieve the best prediction performance. The main reason is that the method can not only extract the structure feature of node pair in each time slice, but also mine the preference feature of node pair. Compared with the preference feature, the structure feature can effectively mine the change law of the interaction between nodes in a bipartite network. The preference feature only considers the similarity between nodes and cannot capture the dynamic changes of interactions. Therefore, the structure features and preference features are concatenated as temporal features.

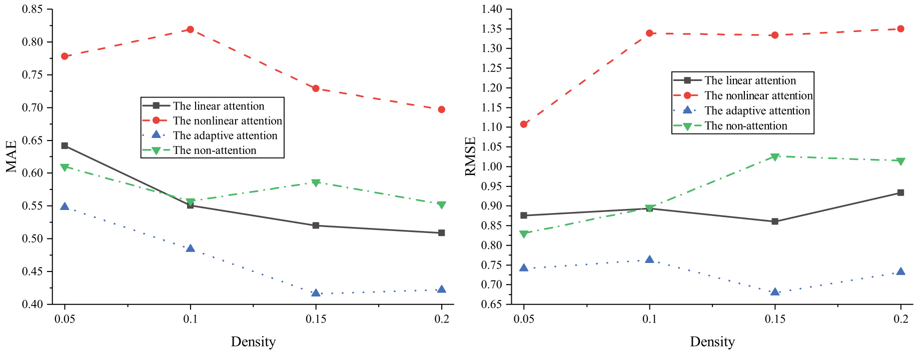


Fig. 5 Impact of attention on performance. The four lines, respectively, represent the performance of adaptive attention, linear attention, nonlinear attention, and non-attention in weighting temporal features under different density conditions. The results show that adaptive attention weighting can achieve the best prediction performance

6.3 Analysis of attention

In this paper, we analyze the impact of attention on prediction performance by comparing non-attention and three kinds of attention, where the three kinds of attention are: (1) the adaptive attention based on model learning; (2) the linear attention $\frac{1}{t}$ based on manual definition; and (3) the nonlinear attention $\frac{1}{e^t}$ based on manual definition, where t represents the time slice. As shown in Fig. 5, the adaptive attention can achieve the best prediction performance compared to the other two kinds of attention and non-attention. The main reason is that the adaptive attention can automatically learn the importance of features in different time slices. The self-defined attention mechanism measures the importance of features based on the distance of historical time slices and cannot automatically learn the importance of features in each time slice. The DBN model without attention mechanism shows the worst prediction performance, which shows that the introduction of attention in this paper is reasonable.

6.4 Performance impact of time slices and density

To analyze the performance impact of the proposed method DBN on MAE and RMSE, a set of experiments are carried out by sampling the density and time slices of the dataset. The data density spans from 5 to 20%, and the interval is set by 5%. Meanwhile, the size of the time slice is set from 1 to 64, where the interval is set by 8. The performance impact of time slice and density on MAE and RMSE is shown in Fig. 6. It is observed from the results that DBN can achieve excellent prediction performance in the case of high data density and short time slice as well as in the case of low data density and long time slice. The main reason can be explained by the following two aspects. On the one hand, in case of low sample data density, prediction cannot be effectively performed by the current time slice due to the insufficient provision of invocation records between users and services. Therefore, the influence of a larger time slice is beneficial to positively provide more feature information for mining the implicit potential relationship between users and services, yielding to better prediction accuracy. On the other hand, when the sample data density is high enough, there is adequate information in the current time slice for accurate prediction, where the incorporation of temporal features may conversely reduce the prediction accuracy.

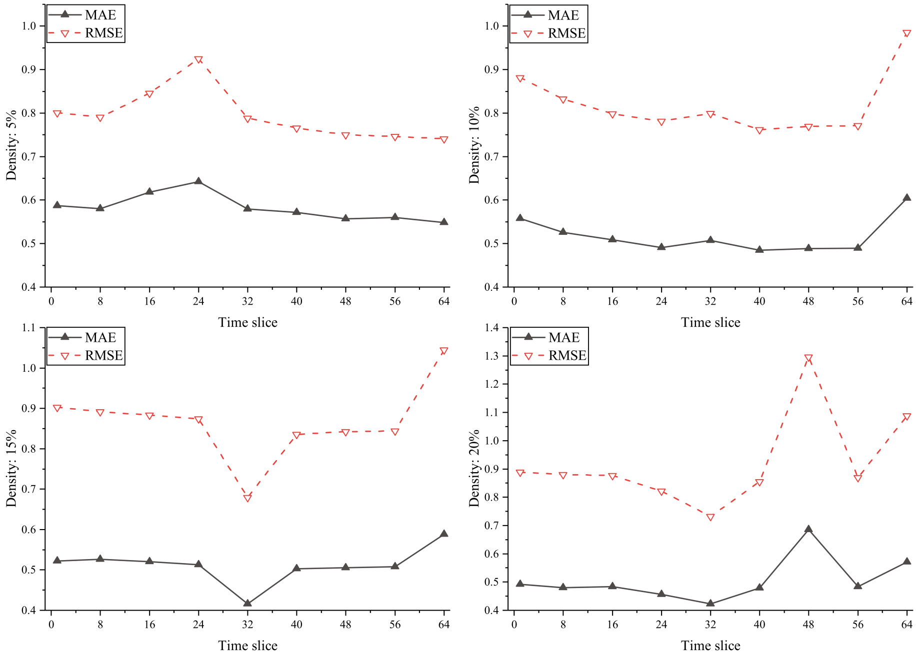


Fig. 6 Performance impact of time slice and density in terms of MAE and RMSE. This experiment analyzes the trend of model performance as the density increases from 5 to 20% over time slices. The two curves represent two evaluation metrics: MAE and RMSE. The results show that DBN consistently achieves outstanding prediction performance, whether in scenarios with high data density and short time slices or in situations with low data density and long time slices

7 Conclusion

In this paper, we propose a dynamic bipartite network to model the dynamic interaction between two types of objects in a complex system, and predict the interaction based on the structure features and preference features of nodes in different time slices. This method mainly has the following innovations:

(1) The dynamic bipartite network model (DBN) is proposed to describe the dynamic interaction between two types of objects, and to realize the dynamic modeling of the interaction over time.
(2) The structural feature representation method and preference feature mining method are proposed, which can represent implicit features of nodes.
(3) The importance of features in different time slices is distinguished based on the attention mechanism, which provides important feature information for model prediction.

Specifically, the model can describe the dynamic interaction between two types of objects in complex systems. At the same time, structure features can reflect the dynamic structural changes of nodes, and preference features can reflect the potential interaction preferences of nodes. The concatenation of structure features and preference features can effectively represent the implicit features of nodes in bipartite networks. In addition, the importance of features in different time slices can be distinguished based on attention, and the nonlinear relationship implicit in the features can be captured based on the GRU model to achieve effective prediction of node interactions. Extensive experiments have been conducted on a real complex system consisting of users and services. The results show that DBN can receive superior accuracy of prediction compared with state-of-the-art benchmarking methods. Based on the research in this paper, the following future suggestions are proposed: 1. A dynamic bipartite network modeling method that more accurately characterizes complex systems will be proposed to effectively capture and model the complexity and dynamics of the system. 2. Data augmentation and model pre-training methods that combine specialized domain knowledge will be explored to develop a more general and powerful prediction framework. 3. Personalized prediction models will be studied based on user behavior, personal preferences and contextual information to improve the overall service recommendation experience for users. 4. Real-time adaptation strategies of the model in response to dynamic changes will be explored to construct a more resilient and responsive service recommendation prediction system.

Acknowledgements Not applicable.

Author Contributions HL proposed the research plan, coding, and writing of this paper. GZ and BZ supervised the projects. SH, CZ, and LW were in charge of data collection and analysis, translation, and discussion. All authors contributed to the article and approved the submitted version.

Funding The author acknowledges the support provided by the Natural Science Foundation of Xinjiang Uygur Autonomous Region (NoN2022D01A236).

Data availability The dataset and code for this study can be found at https://github.com/lvhehe/DBN.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval Not applicable as this study did not involve human participants.

References

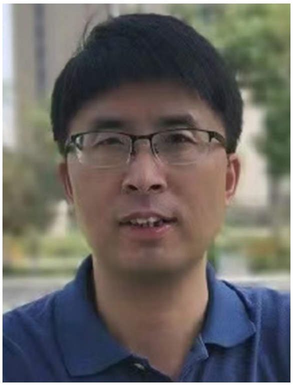
1. Klishin AA, Bassett DS (2022) Exposure theory for learning complex networks with random walks. J Complex Netw 486:75–174
2. Ghosh S, Miyoshi N, Shirai T (2022) Disordered complex networks: energy optimal lattices and persistent homology. IEEE Trans Inf Theory 68:5513–5534
3. Zhao J, Deng Y (2021) Complex network modeling of evidence theory. IEEE Trans Fuzzy Syst 29:3470–3480
4. Arthur R (2020) Modularity and projection of bipartite networks. Phys A Stat Mech Appl 549:124341
5. Guillaume J, Latapy M (2006) Bipartite graphs as models of complex networks. Phys A Stat Mech Appl 371(2):795–813
6. Li X, Zhang M, Wu S, Liu Z, Wang L, Philip SY (2020) Dynamic graph collaborative filtering. In: 2020 IEEE international conference on data mining (ICDM). IEEE, pp 322–331
7. Jiang S, Koch B, Sun Y (2021) Hints: citation time series prediction for new publications via dynamic heterogeneous information network embedding. In: Proceedings of the web conference 2021, pp 3158–3167
8. Fu H, Huang F, Liu X, Qiu Y, Zhang W (2022) MVGCN: data integration through multi-view graph convolutional network for predicting links in biomedical bipartite networks. Bioinformatics 38(2):426–434
9. Zhou C, Zhang J, Gao K, Li Q, Hu D, Sheng VS (2022) Bipartite network embedding with symmetric neighborhood convolution. Expert Syst Appl 198:116757
10. Gao M, He X, Chen L, Liu T, Zhang J, Zhou A (2020) Learning vertex representations for bipartite networks. IEEE Trans Knowl Data Eng 34:379–393

11. Ramalingeswara Rao T, Ghosh SK, Goswami A (2021) Mining user–user communities for a weighted bipartite network using spark GraphFrames and Flink Gelly. J Supercomput 77(6):5984–6035
12. Xia J, Wu F, Xiong Z, Qiu M, Xie C (2017) Modeling recommender systems via weighted bipartite network. Concurr. Comput. Pract. Exp. 29(14):3895
13. Pan L, Shao H, Mesbahi M, Xi Y, Li D (2018) Bipartite consensus on matrix-valued weighted networks. IEEE Trans. Circuits Syst. II Express Briefs 66(8):1441–1445
14. Noulapeu Ngaffo A, El Ayeb W, Choukair Z (2021) A time-aware service recommendation based on implicit trust relationships and enhanced user similarities. J Ambient Intell Humaniz Comput 12(2):3017–3035
15. Tong E, Niu W, Liu J (2021) A missing QoS prediction approach via time-aware collaborative filtering. IEEE Trans Serv Comput 15:3115–3128
16. Wu X, Fan Y, Zhang J, Lin H, Zhang J (2019) QF-RNN: QI-matrix factorization based RNN for time-aware service recommendation. In: 2019 IEEE international conference on services computing (SCC). IEEE, pp 202–209
17. Zhu J, He P, Xie Q, Zheng Z, Lyu MR (2017) Carp: context-aware reliability prediction of black-box web services. In: 2017 IEEE international conference on web services (ICWS). IEEE, pp 17–24
18. Meng S, Zhou Z, Huang T, Li D, Wang S, Fei F, Wang W, Dou W (2016) A temporal-aware hybrid collaborative recommendation method for cloud service. In: 2016 IEEE international conference on web services (ICWS). IEEE, pp 252–259
19. Luo X, Wu H, Yuan H, Zhou M (2019) Temporal pattern-aware QoS prediction via biased non-negative latent factorization of tensors. IEEE Trans Cybern 50(5):1798–1809
20. Berahmand K, Li Y, Xu Y (2023) DAC-HPP: deep attributed clustering with high-order proximity preserve. Neural Comput Appl 35(34):24493–24511
21. Berahmand K, Mohammadi M, Sheikhpour R, Li Y, Xu Y (2024) WSNMF: weighted symmetric nonnegative matrix factorization for attributed graph clustering. Neurocomputing 566:127041
22. Zhang J, Gao M, Yu J, Yang L, Wang Z, Xiong Q (2021) Path-based reasoning over heterogeneous networks for recommendation via bidirectional modeling. Neurocomputing 461:438–449
23. Yu Z, Huang F, Zhao X, Xiao W, Zhang W (2021) Predicting drug-disease associations through layer attention graph convolutional network. Brief Bioinform 22(4):243
24. Bai X, Zhang F, Li J, Xu Z, Patoli Z, Lee I (2021) Quantifying scientific collaboration impact by exploiting collaboration-citation network. Scientometrics 126(9):7993–8008
25. Wang W, Lv H, Zhao Y (2020) Predicting DNA binding protein-drug interactions based on network similarity. BMC Bioinform 21(1):1–13
26. Wang W, Lv H, Zhao Y, Liu D, Wang Y, Zhang Y (2020) DLS: a link prediction method based on network local structure for predicting drug–protein interactions. Front Bioeng Biotechnol 8:330
27. Lv H, Zhang B, Hu S, Xu Z (2022) Deep link-prediction based on the local structure of bipartite networks. Entropy 24(5):610
28. Xu S, Yang C, Shi C, Fang Y, Guo Y, Yang T, Zhang L, Hu M (2021) Topic-aware heterogeneous graph neural network for link prediction. In: Proceedings of the 30th ACM international conference on information and knowledge management, pp 2261–2270
29. Zou G, Chen J, He Q, Li K, Zhang B, Gan Y (2020) NDMF: neighborhood-integrated deep matrix factorization for service QoS prediction. IEEE Trans Network Serv Manag 17(4):2717–2730
30. Liang T, Chen M, Yin Y, Zhou L, Ying H (2021) Recurrent neural network based collaborative filtering for QoS prediction in IoV. IEEE Trans Intell Transp Syst 23(3):2400–2410
31. Yao K, Liang J, Liang J, Li M, Cao F (2022) Multi-view graph convolutional networks with attention mechanism. Artif Intell 307:103708
32. Wang X, Ji H, Shi C, Wang B, Ye Y, Cui P, Yu PS (2019) Heterogeneous graph attention network. In: The world wide web conference, pp 2022–2032
33. Du Y, Wang L, Peng Z, Guo W (2021) Review-based hierarchical attention cooperative neural networks for recommendation. Neurocomputing 447:38–47
34. Zheng Z, Zhang Y, Lyu MR (2010) Distributed QoS evaluation for real-world web services. In: 2010 IEEE international conference on web services. IEEE, pp 83–90
35. Silic M, Delac G, Srbljic S (2014) Prediction of atomic web services reliability for QoS-aware recommendation. IEEE Trans Serv Comput 8(3):425–438
36. Li S, Wen J, Luo F, Ranzi G (2018) Time-aware QoS prediction for cloud service recommendation based on matrix factorization. IEEE Access 6:77716–77724
37. Xiong R, Wang J, Li Z, Li B, Hung PC (2018) Personalized LSTM based matrix factorization for online QoS prediction. In: 2018 IEEE international conference on web services (ICWS). IEEE, pp 34–41
38. Zhang Y, Yin C, Lu Z, Yan D, Qiu M, Tang Q (2019) Recurrent tensor factorization for time-aware service recommendation. Appl Soft Comput 85:105762

39. Fayala M, Mezni H (2020) Web service recommendation based on time-aware users clustering and multi-valued QoS prediction. Concurr Comput Pract Exp 32(9):5603
40. Zou G, Li T, Jiang M, Hu S, Cao C, Zhang B, Gan Y, Chen Y (2022) Deeptsqp: temporal-aware service QoS prediction via deep neural network and feature integration. Knowl-Based Syst 241:108062

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Hehe Lv is currently a doctoral student at the School of Computer Engineering and Science, Shanghai University, China. Prior to this, he obtained a Master's degree in Computer Applied Technology from Henan Normal University in 2020. His research interests encompass complex networks, recommendation systems, and graph neural networks. He has authored or co-authored multiple papers in relevant international journals.

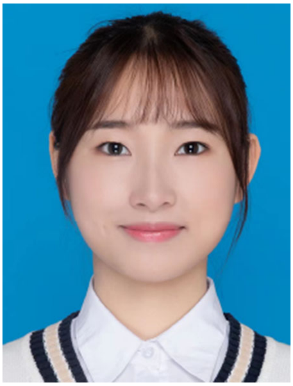


Guobing Zou is a full professor and vice dean of the School of the Computer Science, Shanghai University, China. He received his PhD degree in Computer Science from Tongji University, Shanghai, China, 2012. He has worked as a visiting scholar in the Department of Computer Science and Engineering at Washington University in St. Louis from 2009 to 2011, USA. His current research interests mainly focus on services computing, edge computing, data mining and intelligent algorithms, and recommender systems. He has published more than 110 papers on premier international journals and conferences, including IEEE transactions on services computing, IEEE transactions on network and service management, IEEE ICWS, ICSOC, IEEE SCC, AAAI, information sciences, expert systems with applications, knowledge-based systems, etc.

Bofeng Zhang is a full professor and dean of the School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai, China. He received his PhD degree from the Northwestern Polytechnic University (NPU) in 1997, China. He experienced a Postdoctoral Research at Zhejiang University from 1997 to 1999, China. He worked as a visiting professor at the University of Aizu from 2006 to 2007, Japan. He worked as a visiting scholar at Purdue University from 2013 to 2014, USA. His research interests include personalized service recommendation, intelligent human -computer interaction, and data mining. He has published more than 200 papers on international journals and conferences. He worked as the program chair for UUMA and ICSS. He also served as a program committee member for multiple international conferences.

Shengxiang Hu is currently a PhD candidate in the School of Computer Engineering and Science at Shanghai University, China. Prior to his ongoing PhD work, he successfully completed his Master's degree in Computer Science and Technology from the same university in 2021. His primary areas of research encompass Quality of Service (QoS) prediction, graph neural networks, and natural language processing. Over the course of his academic career, Hu has contributed significantly to the field through his authorship and co-authorship of 15 scholarly papers. These papers have been published in esteemed international journals and presented at prestigious conferences, such as knowledge-based systems, IEEE transactions on service computing, ICSOC, and PPSN.

Chenyang Zhou is currently a master student at the School of Computer Engineering and Science, Shanghai University, China. His research interests include recommendation systems, group recommendation, social networks, and graph neural network. He has authored or co-authored about three papers in relevant international journals and conferences.



Liangrui Wu is currently a master student in the School of Computer Engineering and Science, Shanghai University, China. Before that, she received a Bachelor's degree in Computer Science and Technology from Shanghai Normal University, 2021. Her research interests include recommendation systems, social network, and imbalanced learning. She has co-authored several papers in a relevant international journal.