Knowledge-Based Systems 203 (2020) 106168

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Supervised link prediction in multiplex networks

Na Shan [a], Longjie Li [a,*], Yakun Zhang [a], Shenshen Bai [a,b], Xiaoyun Chen [a]

[a] *School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China*
[b] *Department of Electronic and Information Engineering, Lanzhou Vocational Technical College, Lanzhou 730070, China*

ARTICLE INFO

Article history:
Received 27 January 2020
Received in revised form 11 May 2020
Accepted 18 June 2020
Available online 23 June 2020

Keywords:
Link prediction
Multiplex networks
Complex networks
Supervised learning
Feature extraction

ABSTRACT

In recent years, multiplex networks have been introduced to describe real complex systems, where the same group of entities make different types of interaction. In a multiplex network, each layer expresses one distinct type of interaction. Link prediction is a research hotspot in complex network analysis. A large number of link prediction methods have been proposed, but only a few were designed for multiplex networks. In this paper, we focus on the link prediction problem in multiplex networks. In our opinion, an approach in which link prediction is performed by simultaneously considering the information from all layers is advisable, because the formation of links in one layer can be affected by links of the same node pairs in other layers. A supervised method is proposed in this study to implement link prediction in multiplex networks, which regards link prediction as a binary classification problem. In the proposed method, a classification model is fed by a set of elaborate structural features of node pairs that are extracted from all layers. Extensive experiments are conducted on six networks to analyze the effectiveness of the proposed method. The results demonstrate that the proposed method outperforms the compared methods significantly.

1. Introduction

In real world, complex networks can be used to model various complex systems like society, biology, information and technology systems, in which nodes represent individuals or entities, and links or edges describe relationships or interactions between nodes [1–3]. As a basic problem in complex network analysis, link prediction has a wide range of applications in many fields. It estimates the connection likelihood of unconnected node pairs based on the observed network structures [4–6]. In some networks, such as biological networks [7,8], electric power grid [9,10] and air transportation networks [8,11], how to find out which entities may generate new links in the near future is a very important and challenging problem. This problem can be defined as the so-called *link prediction* problem.

To address the problem, a wealth of link prediction methods or models have been put forward by researchers from diverse disciplines [4,12,13], which are mainly divided into similarity-based methods and supervised learning-based methods [5,14,15]. Similarity-based methods hold the view that two nodes with higher similarity score are more likely to have a missing link [14,16,17]. To compute the similarity of two target nodes, Common Neighbors (CN) index [18] counts the number of shared neighbors between them. The Jaccard index [12], Hub Promoted index [19], Hub Depressed index [20] and local Leicht–Holme–Newman index [21] normalize the result of CN by using the degrees of endpoints from different perspectives. Both Adamic–Adar (AA) index [22] and Resource Allocation (RA) [20] index consider that common neighbors with different degrees have different contribution to the similarity. The Preferential Attachment (PA) index [23], which is motivated by the mechanism of preferential attachment in generating evolving scale-free networks, estimates the similarity of two target nodes by calculating the product of their degrees. On the other hand, supervised learning-based methods determine the connection likelihood of node pairs by using some machine learning algorithms. A host of studies have treated link prediction as a binary classification problem [5,14,24], where a prediction model is trained based on the observed network information. To do that, a training set composed of instances, each of which is a node pair with features and a label, is generated. The feature set of a node pair is generally constituted of a series of topological features extracted from the network, and the label is determined by whether there is a link between them. Ahmed et al. [25] proposed a supervised learning framework that uses eight similarity indexes representing connectivity, community, interaction and trust in social networks, and nine classifiers like Decision Trees to predict missing links in Twitter. Fire et al. [26] proposed a set of simple and easy-to-compute structural features and employed one Decision Trees algorithm, i.e., J48, and two ensemble learning, namely Bagging and RandomForest, to predict the missing links in social networks.

* Corresponding author.
E-mail address: ljli@lzu.edu.cn (L. Li).

https://doi.org/10.1016/j.knosys.2020.106168
0950-7051/© 2020 Elsevier B.V. All rights reserved.

Most of the existing methods focus on the link prediction problem in single-layer networks. However, in recent studies [11, 27,28], it has been found that there may exist different kinds of connection between entities in real life. For example, students in the same class may also follow each other on Twitter and Facebook. These different kinds of connection can be expressed via multiplex networks. In a multiplex network, nodes in each layer are the same, and different layers of the network represent different types of relationship between the same group of entities [29]. A number of studies have shown that the topological characteristics of different layers in a multiplex network are indeed interrelated [27,28,30]. That is to say, ignoring the topological features in other layers may lead to the loss of important information for link prediction in target layer.

In the past few years, some researchers have paid attention to link prediction problem in multiplex networks (see details in Section 2). For instance, Yao et al. [31] proposed the NSILR index that predicts missing links in multiplex networks based on interlayer relevance. Mandal et al. [32] solved the link prediction problem in a two-layer network composed of Twitter and Foursquare using several supervised learning methods. Sharma et al. [33] proposed a method to uncover missing links in target layer by considering the information from other layers. Although these methods can address the problem of link prediction in multiplex networks to a certain extent, there are still some shortcomings. For example, when gauging the connection likelihood of a node pair, the method of Sharma et al. [33] only considers whether there are links between these two nodes in other layers of the network, but ignores the topological properties in target layer. In addition, the setting for tunable parameter in the method proposed by Yao et al. [31] is a quite hard job.

In this paper, we propose a supervised learning-based method to accurately resolve the link prediction problem in multiplex networks. The motivations behind our method are: (1) integrating the structural information from all layers; and (2) avoiding the issue of parameter setting. In the proposed method, the structural information of all layers are taken into consideration by using the feature vectors of node pairs extracted from all layers. Moreover, the problem of parameter determination as in [31,34] is addressed by taking advantage of supervised learning. Besides adopting a group of well-known similarity indexes, such as CN [18], RA [20] and Jaccard [12], as features, we also design two new features, namely *friendship of neighbors* (FoN) and *friendship in auxiliary layers* (FAL), for node pairs. Based on these features, classification models are trained and evaluated on six different multiplex networks. The experimental results manifest that our method outperforms the traditional link prediction methods that are originally designed for single-layer networks and the state-of-the-art methods for multiplex networks.

The main contributions of this paper are summarized as follows:

(1) To address the problem of link prediction in multiplex networks, we propose a new supervised learning-based method, which takes into account the structural information of all layers in terms of a set of features.
(2) Two new features, i.e., FoN and FAL, are designed in this paper, which can capture the additional friendships between target nodes, and thus are conducive to link prediction.
(3) Extensive experiments conducted on six real multiplex networks demonstrate that the proposed method outperforms the compared methods and the new features facilitate link prediction in multiplex networks.

The rest of this paper is structured as follows. Section 2 reviews some related work about link prediction. Section 3 describes the link prediction problem in multiplex networks, introduces three baselines, and defines the evaluation metrics. The proposed method is presented in Section 4. Extensive experimental results are shown and analyzed in Section 5. Finally, Section 6 concludes this paper.

2. Related work

As a research hotspot in complex network analysis, the problem of link prediction has been extensively explored in the last decade. This section briefly reviews the new developments of link prediction in dynamic networks, heterogeneous information networks and multiplex networks. Please refer to some recent surveys [13,35–37] for more work of link prediction.

2.1. Link prediction in dynamic networks

Most traditional link prediction methods aim at uncovering unobserved links in static networks. In truth, many networks in real world are dynamic, which change and evolve over time [38]. One big challenge in link prediction over dynamic networks is to capture the evolving nature. Bu et al. [38] proposed a semi-supervised learning framework base on both survival analysis and game theory. In their framework, the Cox Proportional Hazard Model is used to study the link dynamics rule, and a game theory is proposed to predict future links. Chiu and Zhan [39] presented a deep learning-based approach, in which traditional similarity metrics are selected to represent features of node pairs, and weak estimators are employed to train a deep learning network for link prediction. Goyal et al. [40] proposed the model of dyngraph2vec to predict unseen links. Their model learns the structure of evolution in dynamic graphs by using a deep architecture composed of dense and recurrent layers. Inspired by the success of the generative adversarial network in generating fake images, Chen et al. [41] proposed a generative dynamic link prediction method by modeling the link prediction task as a network generation process. Chi et al. [42] designed a dynamic link prediction approach on the basis of the levels of nodes and the attraction force between nodes. The level of a node is assigned in line with its local influence. To perform dynamic link prediction, De Winter et al. [43] extended the node2vec [44] to dynamic networks by combining temporal aspects of dynamic networks. Motivated by the outstanding performance of LSTM in handling time series, Chen et al. [45] proposed the E-LSTM-D model to forecast dynamic links end to end. The model has the capability to deal with long-term prediction problems, and is suitable for the networks of different scales with fine-tuned structure.

2.2. Link prediction in heterogeneous information networks

Heterogeneous information networks (HINs) [46,47], which consist of multiple types of nodes and relationships, tote more information in comparison with homogeneous networks, and hence more fruitful to link prediction [35]. To tackle link prediction task in HINs, Shakibian and Charkari [48] proposed some new meta-path based statistical similarity measures using the substantial information provided by meta-paths. For link prediction in schema-rich HIN, Cao et al. [49] proposed a link prediction method i.e., LiPaP, in which the Automatic meta-Path Generation algorithm is designed to automatically extract meta-paths from schema-rich HINs. In [50], Fu et al. devised a representation learning model, namely HIN2Vec, for HINs, the core of which is a neural network designed to capture the rich semantics embedded in HINs. To predict possibly existing sentiment links in HINs, Wang et al. [51] proposed an end-to-end Signed Heterogeneous Information Network Embedding (SHINE) framework, which uses multiple deep autoencoders to learn user's latent representations. For the purpose of predicting arbitrary types of academic relations in heterogeneous academic networks, Lu et al. [52] designed a unified link prediction framework, which adopts logistic regression model to learn the features of different meta-paths.

2.3. Link prediction in multiplex networks

Multiplex networks, which model different types of relationship between the same group of entities [29], are a special kind of HINs. In a multiplex network, different layers usually exhibit some extent of correlation [27,28,30]. This suggests that proper use of the information from other layers can facilitate link prediction in target layer. Sharma et al. [33] predicted missing links in target layer by considering whether corresponding links appear in other layers. Pujar et al. [53] proposed a supervised learning based approach to predict co-authorship by leveraging information from different layers in general bibliographical multiplex networks. Hajibagheri et al. [54] presented a comprehensive method to predict missing links in the target layer by taking into account the information of all layers. Yao et al. [31] designed a method to predict missing links by combining both intra-layer and inter-layer information via interlayer relevance. Mandal et al. [32] used several supervised learning methods to solve the link prediction problem in a two-layer network composed of Twitter and Foursquare. Najari et al. [34] put forward a probability-based model, which uses Logistic Regression to obtain the probability of link existence based on intra-layer features. Samei et al. [55] defined a similarity index that combines intra-layer similarity and inter-layer similarity to predict spurious links in a multiplex network. Later, Samei et al. [56] proposed another similarity index, based on hyperbolic geometry, to predict both spurious and missing links in multiplex networks. Chen et al. [57] detected the potential and future relations between individuals in multi-relational social networks by solving the matrix completion problem with a max-norm constrained formulation. Abdolhosseini-Qomi et al. [58] reconstructed the target layer by using the information of other layers that are similar to it.

Research gap: A number of researchers have investigated the link prediction problem in multiplex networks, however, there are some defects in their work. For example, the method of Sharma et al. [33] does not make full use of the information of target layer. The methods proposed by Yao et al. [31] and Najari et al. [34] suffer from the difficulty of parameter setting and cannot benefit from multiple features. Mandal et al. [32] explored multiplex link prediction on only one network.

3. Preliminaries

3.1. Problem description

To better understand the link prediction problem in multiplex networks, this subsection introduces the concept of multiplex networks and formalizes the problem of link prediction in multiplex networks.

A multiplex network with k layers and N vertices can be denoted by $G = (G_1, G_2, \dots, G_k)$, where $G_i = (V_i, E_i)$ represents the network of layer i; V_i and E_i are the node set and edge set in layer i, respectively. Each layer of the network G has the same set of nodes, i.e., $V_1 = V_2 = \cdots = V_k = V$ and $|V| = N$ [29,59]. Fig. 1 shows a simple multiplex network example with two layers. The right part of the figure is the converged network. Let layer α be the target layer, denoted as G_α, and the remaining layers be the auxiliary layers, denoted as $G_{\beta_1}, G_{\beta_2}, \dots, G_{\beta_k - 1}$. The link prediction problem in multiplex networks is to forecast the missing or future links in target layer by using information extracted from the target layer and auxiliary layers.

3.2. Link prediction methods in multiplex networks

3.2.1. Node Similarity Index based on Layer Relevance (NSILR)

NSILR [31] indicates that the similarity scores of node pairs in target layer α are not only contributed by the intra-layer structure information from target layer, but also depend on the inter-layer structure information extracted from other layers. The higher the correlation between auxiliary layer β_i and target layer α, the greater the contribution of layer β_i [31]. For a node pair (x, y) in target layer α, NSILR first calculates its similarity score within each layer based on the traditional methods, such as CN and RA. Then, all scores are aggregated to estimate the similarity of node pair (x, y) in the multiplex network, which is defined as

$$S^{\alpha}(x, y) = (1 - \varphi)\, sim^{\alpha}(x, y) + \varphi \sum_{\beta_i} \mu^{\alpha\beta_i} sim^{\beta_i}(x, y), \tag{1}$$

where $sim^{*}(x, y)$ is the similarity score according to the intra-layer information from layer $*$. $\mu^{\alpha\beta_i}$ denotes the relevance between layers α and β_i, which can be acquired by calculating the Global Overlap Rate (GOR) or Pearson Correlation Coefficient (PCC) between layers. The tunable parameter φ, which lies in the interval [0, 1], is used to adjust the influence of information from intra-layer and inter-layer.

3.2.2. Likelihood assignment algorithm (LAA)

LAA [33] also takes the information from all auxiliary layers of the network into account. The final score is assigned as a weighted combination of scores of different layers. The weights are estimated by checking the link correspondence between two layers using the likelihood that a link presents in the target layer given the corresponding link presenting in the auxiliary layer [33]. The formal definition of LAA is

$$S^{\alpha}(x, y) = \sum_{\beta_i} w_{\beta_i} I(x, y, \beta_i), \tag{2}$$

where w_{β_i} is the weight of layer β_i. If the link (x, y) is found to be present in layer β_i, the value of $I(x, y, \beta_i)$ is 1; otherwise, the value is 0.

3.2.3. Link prediction accounting interlayer similarity (LPIS)

LPIS [34] proposes a probabilistic-based model to calculate the probability of link existence by means of intra-layer features. Then, the probability along with inter-layer similarity are used to calculate the final probability of link existence [34]. The mathematical expression of LPIS is

$$P^{\alpha}(x, y) = (1 - \varphi)\, P^{\alpha}_{intra}(x, y) + \varphi P^{\alpha}_{inter}(x, y), \tag{3}$$

with

$$P^{\alpha}_{inter}(x, y) = \begin{cases} \sum_{\beta_i} P^{\beta_i}_{intra}(x, y) \times R^{\alpha\beta_i}(x, y), & \text{if } I(x, y, \beta_i) = 1 \\ \sum_{\beta_i} \left(1 - P^{\beta_i}_{intra}(x, y)\right) \times \left(1 - R^{\alpha\beta_i}(x, y)\right), & \text{if } I(x, y, \beta_i) = 0 \end{cases} \tag{4}$$

where $P^{*}_{intra}(x, y)$ is the existence probability of link (x, y) in layer $*$. $R^{\alpha\beta_i}$ is the similarity of link (x, y) between layers α and β_i, which can be obtained by calculating interlayer similarity, such as Average Similarity of the Neighbors (ASN) [60] and Asymmetric Average Similarity of the Neighbors (AASN) [34].

3.3. Evaluation metrics

In this paper, link prediction is regarded as a binary classification problem. The label of a node pair is positive if a link connects them, and negative otherwise. Before presenting the evaluation metrics, we first introduces the following common terms.

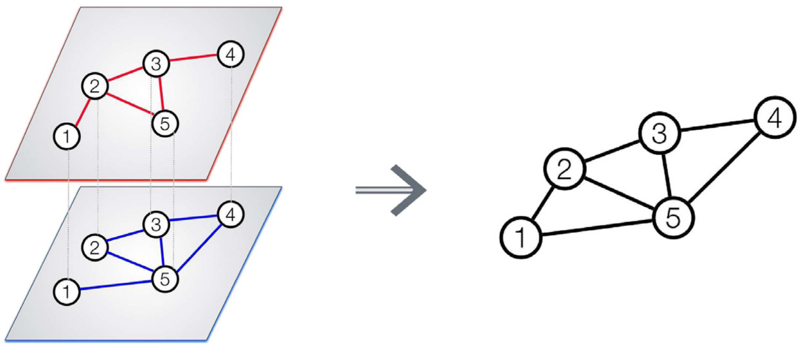


Fig. 1. A simple example of multiplex networks with 2 layers.

(1) True Positive (TP): The number of node pairs that have links are correctly identified as positive.
(2) False Positive (FP): The number of node pairs that do not have links are incorrectly classified as positive.
(3) False Negative (FN): The number of node pairs that have links are incorrectly recognized as negative.
(4) True Negative (TN): The number of node pairs that do not have links are correctly judged as negative.

There are many metrics to evaluate the performance of classification. In what follows, we describe the six metrics used in this paper.

(1) *Precision* indicates the proportion of the actual number of node pairs with links in the node pairs predicted as positive instances. The higher the value of Precision, the better the prediction performance. The calculation formula is

$$Precision = \frac{TP}{TP + FP}. \tag{5}$$

(2) *Recall* is the proportion of correctly predicted node pairs among all the actual linked node pairs. Mathematically,

$$Recall = \frac{TP}{TP + FN}. \tag{6}$$

(3) *F1-score* is the harmonic average of Precision and Recall, which is

$$F1 - score = \frac{2Precision \times Recall}{Precision + Recall}. \tag{7}$$

(4) *Accuracy* is the proportion of all correctly predicted node pairs, which is defined as

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}. \tag{8}$$

(5) *MAE* (Mean Absolute Error) is used to reflect the prediction error rate. The smaller the value, the better the prediction performance. The mathematical expression of MAE reads as

$$MAE = \frac{1}{m}\sum_{i=1}^{m} |y_i - \widehat{y_i}|, \tag{9}$$

where y_i and $\widehat{y_i}$ represent the actual class label and predicted class label of instance i, respectively. m is the number of instances to be predicted.
(6) *RMSE* (Root Mean Square Error) is employed to measure the deviation between the predicted value and the actual value. More explicitly,

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m} (y_i - \widehat{y_i})^2}. \tag{10}$$

4. The proposed method

To date, researchers have proposed a number of link prediction methods for multiplex networks. However, there is still some room for improvement. For example, to compute the connection likelihood of a node pair, LAA only focuses on whether there are links of the node pair in other layers, but neglects the information of target layer [33]. In this work, we propose a new link prediction method based on supervised learning. We regard the link prediction problem as a binary classification problem, in which the class label is determined by the existence of links. When there is a link connecting two nodes, the label is positive (denoted by 1); otherwise, the label is negative (denoted by 0). Owing to the advantage of supervised learning, we do not need to consider the choice of parameters as in [31,34]. In addition, the feature vector of a node pair is composed of the topological features extracted from all layers. Therefore, the method proposed in this paper considers the information provided by each layer more comprehensively.

4.1. Features of node pair

Essentially, link prediction is a binary classification problem for node pairs with their features derived from the structures of networks [61]. Let $G_\alpha = (V_\alpha, E_\alpha)$ represent the target layer to be predicted, and the remaining layers $G_{\beta_1}, G_{\beta_2}, \ldots, G_{\beta_{k-1}}$ be auxiliary layers. In order to predict whether there is a link of a node pair in the target layer, we define a set of topological features for it. This subsection describes all the features used in this work.

4.1.1. Features based on common neighborhood

(1) *Common Neighbors* (CN) [18]. CN measures the possibility of generating links between nodes by calculating the number of common neighbors. The more common neighbors two nodes share, the more likely there is a link between them. Let $\Gamma_i(x)$ denote the neighbor set of node x in layer i, then the value of CN between nodes x and y in layer i is calculated as

$$CN(x, y;\ i) = |\Gamma_i(x) \cap \Gamma_i(y)|. \tag{11}$$

(2) *Resource Allocation* (RA) [20]. RA is based on the idea of resource allocation in networks. It assumes that each node has a certain mass of resource and can transmit its resources to others. Given nodes x and y in layer i, RA defines

the similarity between x and y as the amount of resources received by y through their common neighbors, which is

$$RA(x, y;\ i) = \sum_{z\in\Gamma_i(x)\cap\Gamma_i(y)} \frac{1}{k_z^i}, \tag{12}$$

where k_z^i is degree of node z in layer i.

(3) *Jaccard Coefficient* (JC) [12]. JC computes the similarity of two nodes as the number of common neighbors divided by the total number of their neighbors. The value is defined as

$$JC(x, y;\ i) = \frac{|\Gamma_i(x)\cap\Gamma_i(y)|}{|\Gamma_i(x)\cup\Gamma_i(y)|}. \tag{13}$$

4.1.2. Features based on node degree

(1) *Preferential Attachment* (PA) [23]. PA considers that the probability of an edge connecting to a node is proportional to the degree of the node, which is defined as

$$PA(x, y; i) = k_x^i k_y^i. \tag{14}$$

(2) *Local Assortativity* (LA) [62]. Assortativity is usually adopted to describe the mixed properties of networks. In [62], the concept is extended to the local assortativity of node pairs. The LA score of node pair (x, y) in layer i is calculated as

$$LA(x, y; i) = \frac{4k_x^i k_y^i - k_x^i - k_y^i}{2(k_x^i)^2 + 2(k_y^i)^2 - k_x^i - k_y^i}. \tag{15}$$

4.1.3. Features based on clustering coefficient

(1) *Average Clustering Coefficient* (ACC). ACC is calculated as

$$ACC(x, y; i) = \frac{CC_x^i + CC_y^i}{2}, \tag{16}$$

where CC_x^i is the clustering coefficient of node x in layer i, which is defined as the ratio of the number of connected edges among all neighbors of x to the maximum possible number of connected edges [10]. Formally, CC_x^i is computed as

$$CC_x^i = \frac{2l_x^i}{k_x^i\left(k_x^i - 1\right)}, \tag{17}$$

where l_x^i is the number of connected edges among all neighbors of x in layer i.

(2) *Clustering Coefficient for Link Prediction* (CCLP) [63]. The value of CCLP is the sum of the clustering coefficients of common neighbors between nodes x and y, which is

$$CCLP(x, y; i) = \sum_{z\in\Gamma_i(x)\cap\Gamma_i(y)} CC_z^i. \tag{18}$$

4.1.4. Features based on "friendship"

(1) *Friendship of neighbors* (FoN). This feature takes into account the friendship between neighbors of target nodes. Given target nodes x and y in layer i, FoN counts the number of friendship between x's neighbors and y's neighbors. The formal expression reads as

$$FoN(x, y; i) = \sum_{u\in\Gamma_i(x)} \sum_{v\in\Gamma_i(y)} e(u, v; i), \tag{19}$$

where $e(u, v; i)$ is defined as

$$e(u, v;\ i) = \begin{cases} 1, & \text{if } (u, v) \in E_i \\ 0, & \text{otherwise} \end{cases}$$

(2) *Friendship in auxiliary layers* (FAL). Multiplex networks describe different types of interactions between the same set of nodes. We take the attitude that the "friendship" between two nodes in other layers can affect their relationship in target layer. The feature value of FAL can be computed as

$$FAL(x, y; i) = \begin{cases} 1, & \text{if } (x, y) \in E_i \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

Table 1
Features for node pair.

	Feature	Extracted from target layer	Extracted from auxiliary layers
Features based on common neighborhood	CN	√	√
	RA	√	√
	JC	√	√
Features based on node degree	PA	√	√
	LA	√	√
Features based on clustering coefficient	ACC	√	√
	CCLP	√	√
Features based on "friendship"	FoN	√	√
	FAL	×	√

In real world, the same individuals may have different types of interaction, which can be described by multiplex networks. We hold the view that the connection likelihood of one kind of interaction may be affected by other kinds of interaction. To verify this conjecture, we conduct corresponding experiments in Section 4. Based on the above ideas, when computing the feature vector of a node pair, we take advantage of topological features of the pair not only in target layer but also in auxiliary layers. Therefore, the feature vector of each node pair contains $8 + 9(k - 1)$ features. Since feature *FAL* reflects the friendships of a node pair in auxiliary layers, the node pair has eight features in target layer; in each auxiliary layer, it has nine features. The features used for the proposed method are summarized in Table 1.

4.2. Construction of training set and testing set

In order to use supervised learning algorithms, the training set and testing set should be generated firstly. Here, let U_α denote the set of all possible edges in target layer α, E_α be the set of all existing edges in layer α, and $\bar{E}_\alpha$ represent the set of node pairs that have not yet been linked in layer α. Certainly, $U_\alpha = E_\alpha \cup \bar{E}_\alpha$. We randomly divide E_α into two parts: E_α^T and E_α^P, such that $E_\alpha^T \cup E_\alpha^P = E_\alpha$ and $E_\alpha^T \cap E_\alpha^P = \emptyset$. E_α^T is used for training, and E_α^P is used for testing.

However, E_α^T only contains existing edges. To construct a training set with balanced class distribution, one can randomly extract the same number of node pairs as E_α^T from $\bar{E}_\alpha$. In this paper, we borrow the idea in [39] to construct the training set. We randomly sample the same number of node pairs as E_α^T from $\left(U_\alpha - E_\alpha^P\right)$, which is represented as U_α^T. The training set is the union of E_α^T and U_α^T, i.e., $D_\alpha = E_\alpha^T \cup U_\alpha^T$. As stated in [39], the sparsity of most of naturally-occurring networks results in the low present probability of randomized edges. Therefore, randomly extracting node pairs from U_α is approximately equivalent to yet computationally cheaper than extracting node pairs from $\bar{E}_\alpha$. The literature [26] pointed out that even if a training set with balanced class distribution is used, the imbalance of testing set will still affect the performance measures like Precision. So, when generating the testing set, we also ensure the balance of class distribution. The construction method is as follows. We randomly sample the same number of node pairs as E_α^P from $\left(U_\alpha - (E_\alpha^T \cup U_\alpha^T)\right)$, which is denoted as U_α^P. The testing set is composed of E_α^P and U_α^P, i.e., $P_\alpha = E_\alpha^P \cup U_\alpha^P$. Algorithm 1 shows the construction of training set and testing set.

Algorithm 1 Generating training set and testing set.

```
Input: A multiplex network $G$ with $k$ layers; target layer $\alpha$; auxiliary layers $\beta_1, \beta_2, \cdots, \beta_{k-1}$; proportion of existing edges allocated to training set $\omega$.
Output: Training set $D_\alpha$ and testing set $T_\alpha$.
1: $U_\alpha \leftarrow$ all node pairs in layer $\alpha$, $E_\alpha \leftarrow$ all existing edges in layer $\alpha$;
2: $E_\alpha^T \leftarrow$ sample($\omega$ of edges from $E_\alpha$);
3: $E_\alpha^P \leftarrow E_\alpha - E_\alpha^T$;
4: $U_\alpha^T \leftarrow$ sample( $|E_\alpha^T|$ edges from $(U_\alpha - E_\alpha^P)$);
5: $U_\alpha^P \leftarrow$ sample( $|E_\alpha^P|$ edges from $(U_\alpha - (E_\alpha^T \cup U_\alpha^T))$);
6: $D_\alpha \leftarrow \emptyset$;
7: $P_\alpha \leftarrow \emptyset$;
8: for each $(x, y) \in E_\alpha^T \cup U_\alpha^T$ do
9:     Compute features for $(x, y)$ in layer $\alpha$;
10:    Compute features for $(x, y)$ in layers $\beta_1, \beta_2, \ldots, \beta_{k-1}$;
11:    if $(x, y) \in E_\alpha^T$ then
12:        $(x, y)$ is labeled as 1;
13:    else
14:        $(x, y)$ is labeled as 0;
15:    end if
16:    $D_\alpha \leftarrow D_\alpha \cup \{(x, y)\}$;
17: end for
18: for each $(x, y) \in E_\alpha^P \cup U_\alpha^P$ do
19:    Compute features for $(x, y)$ in layer $\alpha$;
20:    Compute features for $(x, y)$ in layers $\beta_1, \beta_2, \ldots, \beta_{k-1}$;
21:    if $(x, y) \in E_\alpha^P$ then
22:        $(x, y)$ is labeled as 1;
23:    else
24:        $(x, y)$ is labeled as 0;
25:    end if
26:    $P_\alpha \leftarrow P_\alpha \cup \{(x, y)\}$;
27: end for
28: return $D_\alpha, P_\alpha$;
```

4.3. Complexity analysis

Actually, the multiplex link prediction method proposed in this paper consists of two parts: generating training set and testing set, and training prediction model. In real application, generating testing set is not included. The computational time of training prediction model depends the supervised learning algorithm. So, we analyze the time complexity of Algorithm 1 in the following.

Let N and $\langle k \rangle$ be the number of nodes in network G and the average degree of all nodes in k layers, respectively. The time complexity for calculating U_α in line 1 is $O(N^2)$, and the time complexities of other operations in line 1 $\sim$ 7 are not larger than $O(N^2)$. Suppose the number of edges in target layer α is M_α, then $|D_\alpha| = |E_\alpha^T \cup U_\alpha^T| \leq 2\omega M_\alpha$ and $|P_\alpha| = |E_\alpha^P \cup U_\alpha^P| \leq 2(1-\omega)M_\alpha$. Given an node pair (x, y), the computational complexity for extracting the feature values is $O(k\langle k \rangle^3)$ (see details in Appendix). Because $k \ll M_\alpha$ and ω is a constant, time complexity for generating training set (line 8 $\sim$ 17) is $O(M_\alpha \langle k \rangle^3)$. In the same way, time complexity for generating testing set (line 18 $\sim$ 27) is $O(M_\alpha \langle k \rangle^3)$. In result, time complexity of Algorithm 1 is $O(N^2 + M_\alpha \langle k \rangle^3)$.

5. Experimental results and analysis

5.1. Experimental setting

As mentioned in Section 4.2, when constructing the training set and testing set, we randomly divide the existing edges in target layer α into two parts: E_α^T and E_α^P. In the following experiments, we randomly select 80% edges from E_α as E_α^T, and the remaining edges are marked as E_α^P. The training set and testing set are constructed according to Algorithm 1. All experimental results are the average of the results of 10 independent divisions.

Table 2
The basic topological characteristics of six multiplex networks. In this table, N is the number of nodes and $|E|$ indicates the number of edges in each layer. r and C are the assortative coefficient and clustering coefficient, respectively. $\langle k \rangle$ is the average degree, H represents the degree heterogeneity ($H = \langle k^2 \rangle / \langle k \rangle^2$), and e denotes the efficiency of network.

Network	layer	N	$\lvert E \rvert$	r	C	$\langle k \rangle$	H	e
Vicker	1	29	240	−0.161	0.754	16.552	1.099	0.796
	2	29	126	−0.152	0.681	8.690	1.275	0.635
	3	29	152	−0.110	0.713	10.483	1.250	0.666
Lazega	1	71	717	0.020	0.522	20.197	1.166	0.633
	2	69	399	0.079	0.498	11.565	1.314	0.528
	3	71	726	−0.079	0.509	20.451	1.167	0.640
CKM	1	215	449	−0.137	0.260	4.177	1.494	0.122
	2	231	498	−0.098	0.260	4.312	1.348	0.115
	3	228	423	0.102	0.211	3.711	1.237	0.100
CElegans	1	253	517	−0.116	0.202	4.087	2.163	0.253
	2	260	888	−0.081	0.186	6.831	1.788	0.331
	3	278	1703	−0.078	0.288	12.252	1.668	0.409
CS	1	60	193	0.005	0.673	6.433	1.213	0.398
	2	32	124	0.003	0.540	7.750	1.227	0.591
	3	25	21	0.017	0.268	1.680	1.389	0.097
	4	47	88	−0.010	0.392	3.745	1.514	0.347
	5	60	194	−0.213	0.640	6.467	1.665	0.475
TF	1	1564	14090	−0.098	0.131	18.018	3.386	0.327
	2	1508	18471	−0.041	0.344	24.497	3.775	0.353

All of the link prediction algorithms in our experiments were coded in Python 3.7 with the graph package of NetworkX[1] and the machine learning package of scikit-learn.[2] Three supervised-learning methods, namely SVM, RandomForest and Adaboost were employed to identify missing links in our algorithm. Most parameters in these algorithms are set to their default values, except for the following ones. The penalty coefficient C and the kernel coefficient γ in SVM are determined by Grid Search. The candidate value of C is in $[1e-3, 1e-2, 1e-1, 1, 10, 100, 1000]$, the candidate value for γ is in $[0.01, 0.001, 0.0001]$. The value of *max_features* in RandomForest is set to 0.8. The learning rate in Adaboost is set to 0.7, and the *max_depth* of base classifiers is set to 2.

5.2. Datasets

To verify the effectiveness of our method, we perform experiments on six multiplex networks. In what follows, brief description for each network is listed.

(1) Vicker [64]: This multiplex network represents a social network between 29 seventh grade students in a school in Victoria, Australia. It contains 3 layers that correspond to contact, best friend and co-working.
(2) Lazega [65,66]: This multiplex social network consists of 3 kinds of relationship, i.e., co-working, friendship and advice, between partners and associates of a corporate law partnership. It has 71 nodes in total.
(3) CKM [67]: It is a multiplex network about the connections between physicians. It contains 3 layers and 246 nodes. Each layer corresponds to the relations of seeking advice, discussing cases and friendship.

1 http://networkx.github.io/.
2 https://scikit-learn.org/stable/.

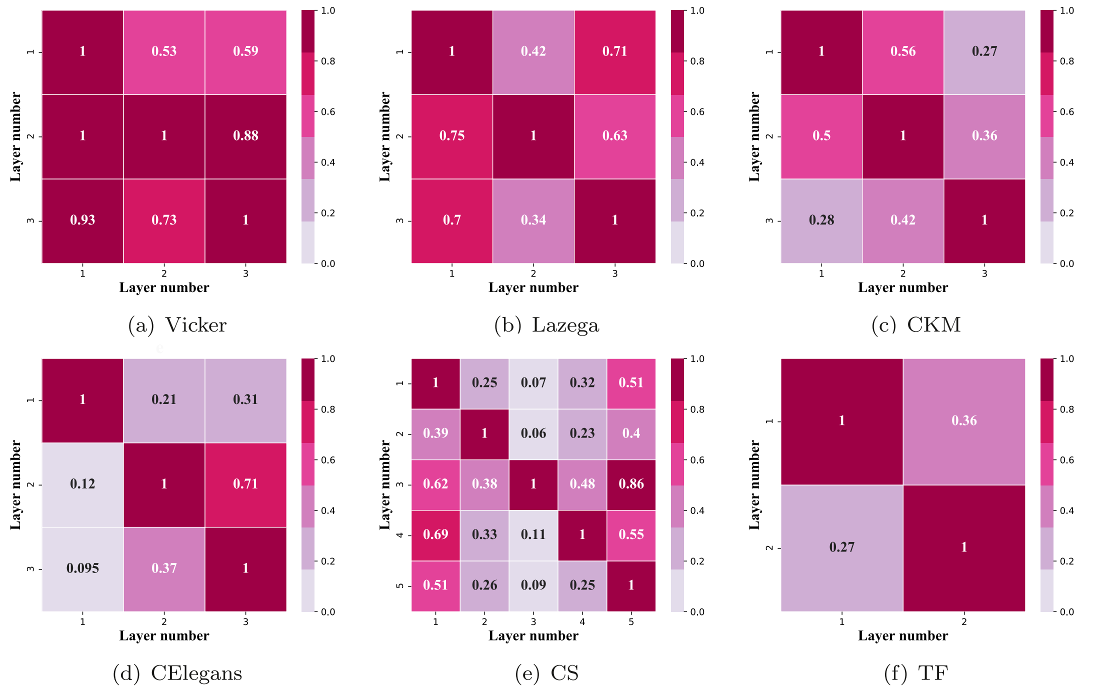


Fig. 2. Overlap rates between layers of six multiplex networks.

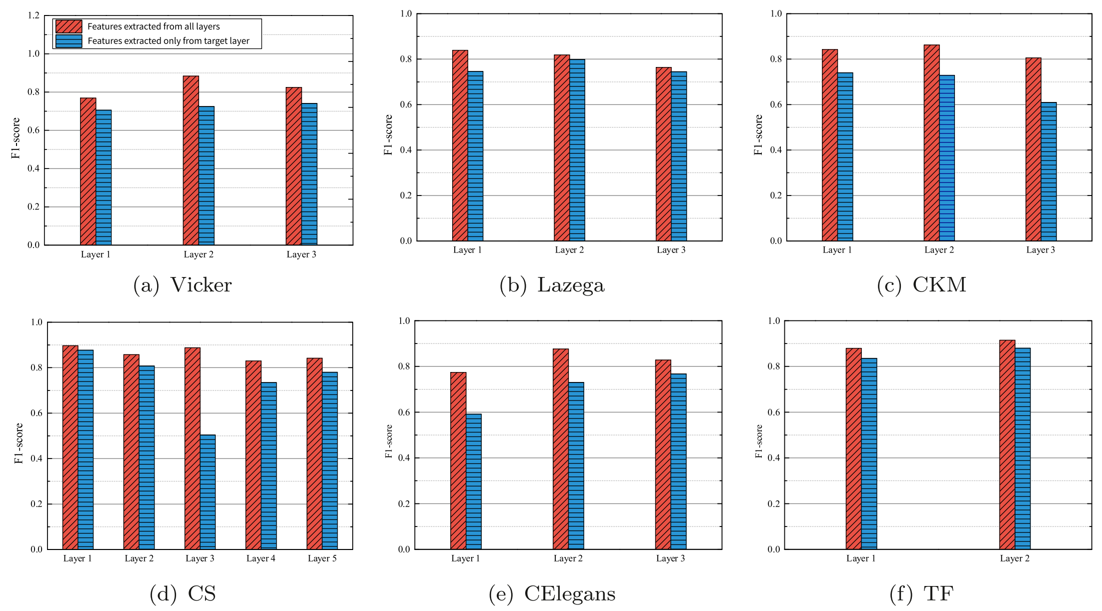


Fig. 3. F1-score results of the proposed method with features only from target layer or from all layers. In this experiment, SVM is used for classification. The results are the average of 10 independent realizations.

(4) CElegans [68,69]: It is a multiplex network of Caenorhabditis elegans. There are 279 nodes and 3 layers in total. Three layers represent different synaptic junctions: electric, chemical monadic and polyadic.

(5) CS [70]: This multiplex social network consists of 5 kinds of on-line and off-line relationships, namely Facebook, Leisure, Work, Co-authorship, and Lunch, between the 61 employees of Computer Science department at Aarhus.

(6) TF [34]: This multiplex network has 1565 users and is composed of two layers, which are collected from Twitter and Foursquare, respectively.

In this paper, all networks are treated as undirected and unweighted. The basic topological characteristics of these networks are listed in Table 2. From this table, we can see that the characteristics of these networks are various. For instance, all layers of Vicker have high clustering coefficients, whereas the clustering coefficients of CElegans are low.

In addition, Fig. 2 shows the overlap rates between layers of each network. Given two layers i and j in a multiplex network, if two nodes simultaneously connect in both layers i and j, the edge between them is an overlapping edge. The overlap rate of layer i with respect to layer j is the ratio of the number of overlapping

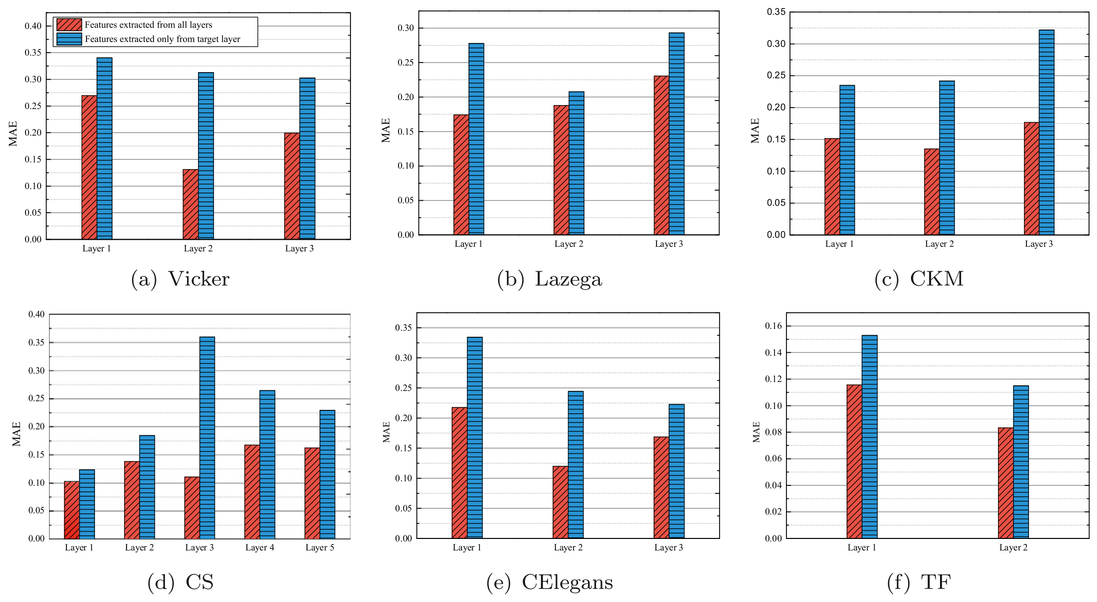


Fig. 4. MAE results of the proposed method with features only from target layer or from all layers. In this experiment, SVM is used for classification. The results are the average of 10 independent realizations.

edges to the number of edges in layer i, which is defined as

$$OR_{ij} = \frac{O_{ij}}{L_i}, \tag{21}$$

where O_{ij} represents the number of overlapping edges observed in both layers i and j, and L_i is the number of edges in layer i. From Eq. (21), OR_{ij} is asymmetric. Fig. 2 manifests that the overlap rates of different networks are also diverse. On the whole, Vicker has the highest overlap rates, indicating high layer relevance between layers in it. As it can be seen from Fig. 2(a), the overlap rate of layer 2 with respect to layer 1 is 1. In other words, all edges in layer 2 are observed in layer 1. Similarly, 93% edges in layer 3 appear in layer 1. On the contrary, the network of CS has the lowest overlap rates for most layer pairs. This result implies that most layers are less correlation in CS.

5.3. Influence of auxiliary layers

Multiplex networks reflect the different kinds of connection between nodes. In our viewpoint, the structural information in auxiliary layers has positive influence on the formation of links in target layer. To verify the assumption, we conduct an experiment, in which the prediction results according to the SVM classifier with or without features extracted from auxiliary layers are compared.

Fig. 3 manifests the results under the metric of F1-score. It is evident from the figure that, for every network, the prediction performance achieved by using the features from all layers is better than that obtained by only using features from the target layer. To further demonstrate the effect of information in auxiliary layers to prediction performance, Fig. 4 depicts the predicted results in terms of MAE. MAE represents the prediction error rate. It can be seen from Fig. 4 that using features derived from all layers can reduce the error rate. Both experimental results bear out our assumption. As a consequence, auxiliary layers' information plays a big role when predicting links for the target layer. Therefore, in our following experiments, features extracted from all layers are adopted.

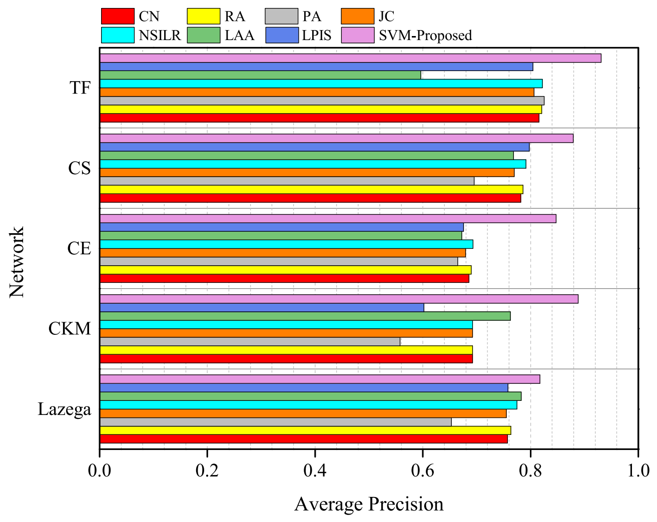


Fig. 5. Precision results of the proposed method and 7 similarity methods on 5 multiplex networks.

5.4. Comparison with similarity-based methods

In this subsection, we implement an experiment for the purpose of evaluating the performance of the proposed method by comparing it with seven existing methods: NSILR [31], LAA [33], LPIS [34], CN [18], RA [20], PA [23] and JC [12]. All these methods can be categorized as similarity-based ones. In [31], the authors did not give the optimal value of the parameter φ, the most suitable method for calculating the relevance between layers, and the best basic similarity index in the NSILR method. In order to determine the configuration that makes the NSILR method work well, we performed an experiment over the six multiplex networks. In the experiment, PCC and GOR were used to calculate layer relevance; the value of φ was switched from $[0.0, 0.1, \ldots, 1.0]$; CN [18], RA [20], PA [23] and AA [22] were selected as the basic similarity indexes. According to the experimental results, we found that when the value of φ is 0.5, PCC is the method

Table 3
Comparison of the proposed method with existing similarity-based methods on Vicker. The results are the average of 10 independent implementations.

Method	Layer	Accuracy	Average accuracy	Precision	Average precision	MAE	Average MAE
	1	0.7305		0.8152		0.2695	
SVM-Proposed	2	0.8689	0.8000	0.8376	0.8162	0.1311	0.2000
	3	0.8006		0.7958		0.1994	
	1	0.7766		0.8002		0.2234	
RF-Proposed	2	*0.8910*	0.8396	*0.8881*	**0.8611**	*0.1090*	0.1604
	3	*0.8511*		*0.8951*		*0.1489*	
	1	*0.8002*		*0.8290*		*0.1998*	
AdaBoost-Proposed	2	0.8775	**0.8400**	0.8666	0.8549	0.1225	**0.1600**
	3	0.8422		0.869		0.1578	
	1	0.7770		0.8167		0.2230	
NSILR	2	0.7831	0.7813	0.7962	0.8043	0.2169	0.2187
	3	0.7838		0.8000		0.2162	
	1	0.7513		0.7958		0.2487	
LAA	2	0.8728	0.8083	0.8808	0.8309	0.1272	0.1917
	3	0.8009		0.8161		0.1991	
	1	0.7134		0.7646		0.2866	
LPIS	2	0.7750	0.7366	0.7885	0.7650	0.2250	0.2634
	3	0.7215		0.7419		0.2785	
	1	0.7086		0.7604		0.2914	
CN	2	0.7417	0.7205	0.7577	0.7501	0.2583	0.2795
	3	0.7113		0.7323		0.2887	
	1	0.7417		0.7875		0.2583	
RA	2	0.7953	0.7574	0.8077	0.7833	0.2047	0.2426
	3	0.7352		0.7548		0.2648	
	1	0.7110		0.7625		0.2890	
PA	2	0.6525	0.6914	0.6731	0.7226	0.3475	0.3086
	3	0.7108		0.7323		0.2892	
	1	0.7113		0.7625		0.2887	
JC	2	0.7329	0.7141	0.75	0.7440	0.2671	0.2859
	3	0.6981		0.7194		0.3019	

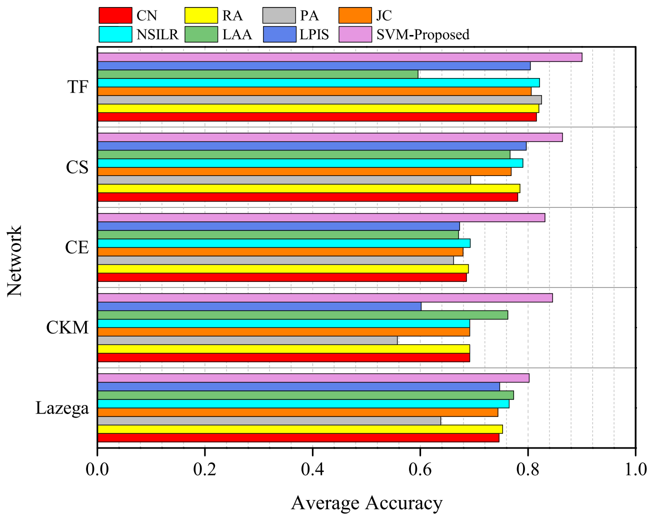


Fig. 6. Accuracy results of the proposed method and 7 similarity methods on 5 multiplex networks.

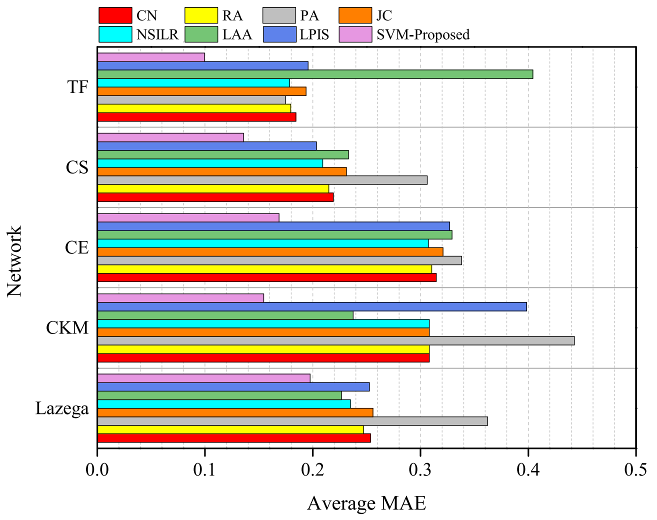


Fig. 7. MAE results of the proposed method and 7 similarity methods on 5 multiplex networks.

for layer relevance, and RA is the basic similarity index, the NSILR method can get the best performance on most networks. As a consequence, we keep the above settings in the following experiments. As per the experimental results in [34], the LPIS method performs best when AASN is employed to measure inter-layer similarity. However, the optimal value for parameter φ is not suggested. In accordance with a simple experiment, we set the value of φ as 0.3, and the basic similarity index to be RA.

In Table 3, we list the prediction results of different methods on the network of Vicker as an example. The experimental results in the table are the average of ten independent realizations. Here, *SVM-Proposed* denotes that in our proposed method, the SVM classification algorithm is used. Likewise, *RF-Proposed* and *AdaBoost-Proposed* indicate that the classification algorithms are RandomForest and AdaBoost, respectively. For each evaluation metric, we highlight the best average performance for all layers in boldface, and the best performance for each layer in italics. It is evident from Table 3 that our methods obtain the best performance not only for the whole network but also for each layer under the metrics of Accuracy, Precision and MAE. In a

Table 4
Comparison of the proposed method with supervised NSILR on Vicker. The results are the average of 10 independent implementations.

Classification algorithm	Method	Layer	Accuracy	Precision	Recall	F1-score	MAE	RMSE
SVM	Proposed	1	0.7305	0.8152	0.7938	0.7689	0.2695	0.5146
		2	0.8689	0.8376	0.9385	0.8836	0.1311	0.3510
		3	0.8006	0.7958	0.8677	0.8245	0.1994	0.4422
		Average	**0.8000**	**0.8162**	0.8667	**0.8257**	**0.2000**	**0.4359**
	Supervised NSILR	1	0.7697	0.7968	0.8729	0.8233	0.2303	0.4768
		2	0.7385	0.7133	0.8615	0.7744	0.2615	0.5083
		3	0.7218	0.6681	0.9613	0.7876	0.2782	0.5264
		Average	0.7433	0.7261	**0.8986**	0.7951	0.2567	0.5038
Random Forest	Proposed	1	0.7766	0.8002	0.8604	0.8285	0.2234	0.4692
		2	0.8910	0.8881	0.9115	0.8978	0.1090	0.3253
		3	0.8511	0.8951	0.8194	0.8544	0.1489	0.3841
		Average	**0.8396**	**0.8611**	**0.8638**	**0.8602**	**0.1604**	**0.3929**
	Supervised NSILR	1	0.7482	0.7903	0.8167	0.8016	0.2518	0.4975
		2	0.7289	0.7454	0.7308	0.7360	0.2711	0.5149
		3	0.7360	0.7410	0.7839	0.7584	0.2640	0.5125
		Average	0.7377	0.7589	0.7771	0.7653	0.2623	0.5083
AdaBoost	Proposed	1	0.8002	0.829	0.8625	0.8439	0.1998	0.4433
		2	0.8775	0.8666	0.9115	0.8858	0.1225	0.3434
		3	0.8422	0.869	0.8355	0.8509	0.1578	0.3932
		Average	**0.8400**	**0.8549**	**0.8698**	**0.8602**	**0.1600**	**0.3933**
	Supervised NSILR	1	0.7664	0.7774	0.8896	0.8275	0.2336	0.4806
		2	0.722	0.7265	0.7769	0.7457	0.278	0.5242
		3	0.7686	0.7522	0.8548	0.7972	0.2314	0.4786
		Average	0.7523	0.7520	0.8404	0.7901	0.2477	0.4945

nutshell, the average Accuracy achieved by AdaBoost-Proposed is up to 0.84 and the average MAE of it is as low as 0.16, and the average Precision got by RF-Proposed is 0.8611. In comparison with baselines, these results are quite decent. It is worth mentioning that the values of each metric on Vicker vary from layer to layer due to various characteristics in different layers (see Table 2) and inequable overlap rates of layers (see Fig. 2(a)). Besides, LAA obtains the best prediction results among baselines on Vicker. From Fig. 2(a), we know that there are many overlapping edges among layers of Vicker. Especially, all edges in the second layer exist in the first layer, and 88% of these edges occur in the third layer. The idea of LAA is to predict the existence of links in target layer by considering whether the corresponding links present in other layers. Therefore, LAA method can achieve good prediction performance over the network with high overlap rates like Vicker.

To further demonstrate the superiority of our method, Figs. 5–7 separately depict the comparison results on the five other networks in terms of Precision, Accuracy and MAE. Take Fig. 5 as an example. The results in the figure are the average values of Precision. Apparently, our method, i.e., SVM-proposed, obtains the highest values over all networks; all its values exceed 80%, and even 95% on TF. Analogous circumstances can be uncovered in Figs. 6 and 7.

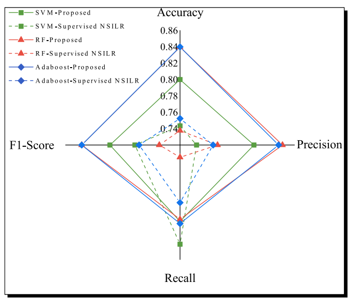
5.5. Comparison with supervised NSILR methods

In this and the following subsections, we perform extensive experiments to comprehensively analyze the effectiveness of our method. In the interest of fairness, we convert the baselines into supervised methods. To do that, the similarity scores of a node pair in different layers of a multiplex network are computed, and then used to construct the feature vector of the node pair. However, not all baselines can be transformed into supervised methods. For example, the similarity scores of a node pair computed by LAA [33] in other layers are related to the class label of the pair in target layer. This can lead to label leaks. As a result, LAA is not suitable to be transformed into supervised method. In addition, the supervised LPIS method designed in [34] uses Logistic Regression to gauge connection likelihood of node pairs, so we do not adopt it in this subsection.

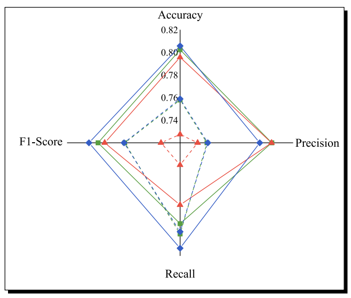
So, we compare our method with the supervised NSILR method in this subsection. Table 4 outlines the results of both methods on the network of Vicker measured by the six evaluation metrics. The better value of the average for each metric is emphasized by boldface. It can be observed from the table that when using the algorithm of SVM, although the performance of the proposed method for the first layer is slightly worse than that of the supervised NSILR, the average performance of the proposed method is more prominent under the metrics of Accuracy, Precision, F1-score, MAE and RMSE. More explicitly, when encapsulating RandomForest and AdaBoost, the proposed method outperforms the corresponding supervised NSILR on not only the whole network but also each layer. Moreover, Fig. 8 graphically presents the comparison results of our proposed method and the supervised NSILR on six networks according to the metrics of Accuracy, Precision, Recall and F1-score. For these four metrics, higher value means better prediction performance. Therefore, the farther from the center a point is, the better the result is in these figures. In general, our method outnumbers the corresponding supervised NSILR base on these four measures. Additionally, Fig. 9 describes the comparison between our method and the supervised NSILR in terms of MAE and RMSE on the five other networks. MAE and RMSE reflect the prediction error rate. So the lower the score, the better the prediction performance. It is apparent from Fig. 9 that our method achieves the lower scores than the corresponding supervised NSILR except on CS. In accordance with the above results, we conclude that the proposed method is superior to the supervised NSILR method.

5.6. Analysis of different features

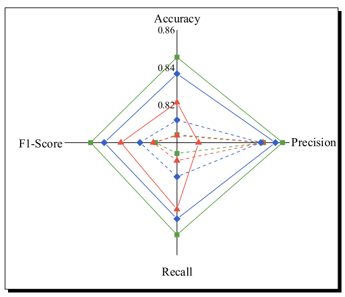
In this subsection, we first evaluate the performance of the proposed method by comparing its results with those got by individual features. In the proposed method, some similarity indexes, such as CN and RA, are used as features of node pairs. In this experiment, we convert four similarity features, i.e., CN, RA,



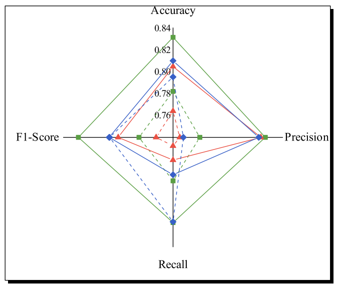
(a) Vicker



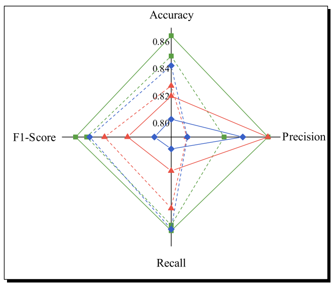
(b) Lazega



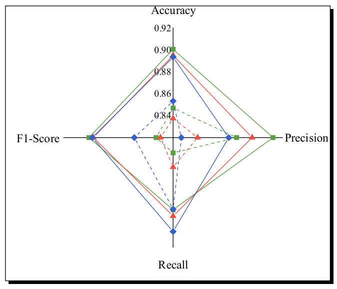
(c) CKM



(d) CElegans

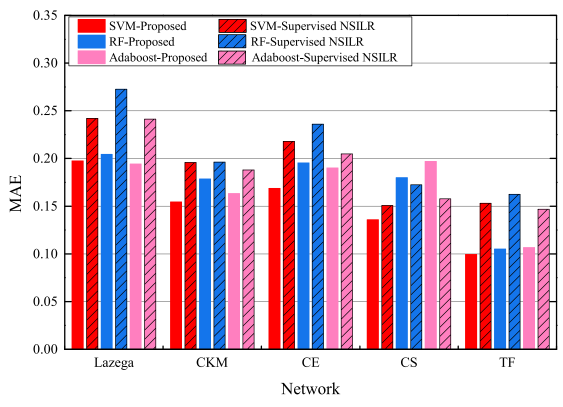


(e) CS

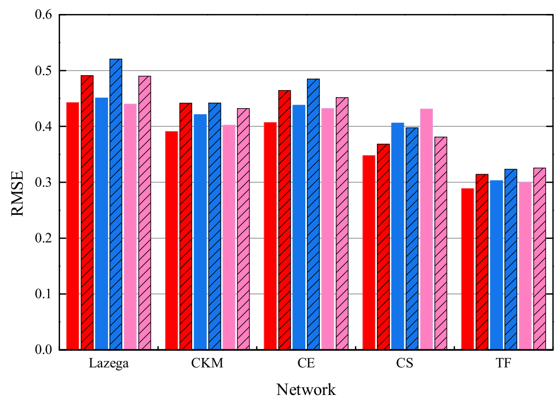


(f) TF

Fig. 8. Comparison of the proposed method with supervised NSILR on six networks in terms of Accuracy, Precision, Recall and F1-score. Each result is the average value of all layers.



(a)



(b)

Fig. 9. Comparison of the proposed method with the supervised NSILR based on MAE and RMSE. Each result is the average value of all layers.

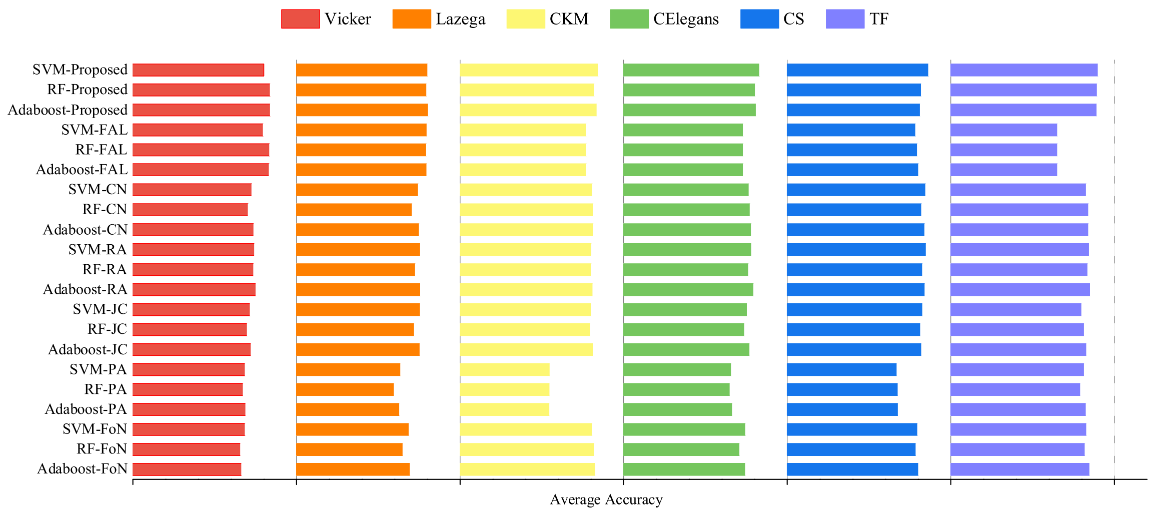


Fig. 10. Accuracy of the proposed method with supervised individual features.

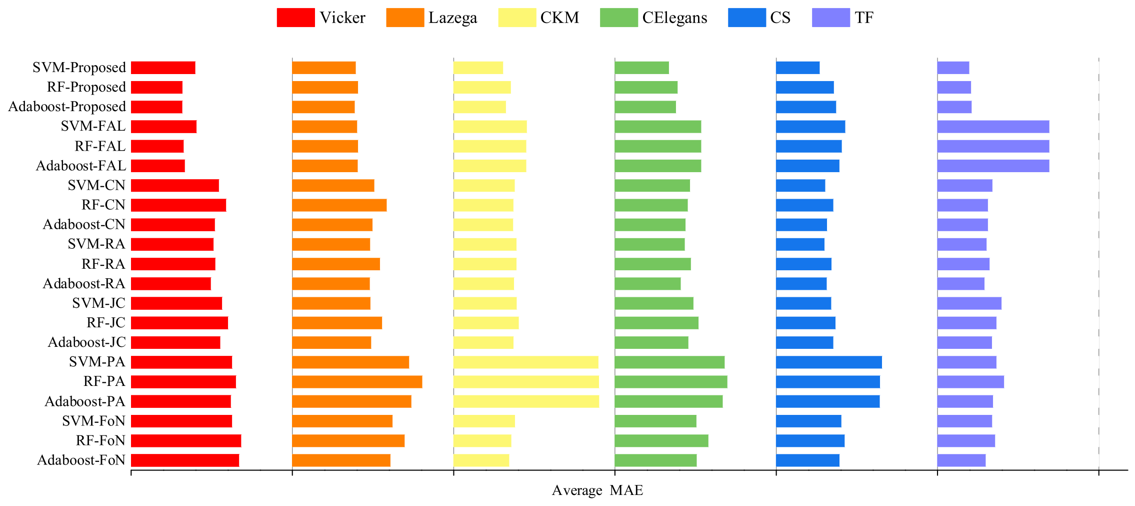


Fig. 11. MAE of the proposed method with supervised individual features.

Table 5
Comparison of the proposed method with supervised individual features under the metric of F1-score. In this experiment, SVM is the supervised-learning algorithm.

Network	Layer	Proposed	CN	RA	JC	PA	FoN	FAL
Vicker	1	0.7689	0.8132	*0.8233*	0.8174	0.7735	0.7781	0.7648
	2	0.8836	0.7484	0.7744	0.7496	0.6961	0.6891	*0.9020*
	3	*0.8245*	0.7832	0.7876	0.7601	0.7564	0.7683	0.8064
	Average	**0.8257**	0.7816	0.7951	0.7757	0.7420	0.7452	0.8244
Lazega	1	0.8389	0.7606	0.7696	0.7829	0.6544	0.6887	*0.8407*
	2	*0.8189*	0.7928	0.8064	0.7966	0.6514	0.7100	0.7655
	3	0.7639	0.7549	0.7606	0.7582	0.6571	0.7197	*0.7823*
	Average	**0.8072**	0.7694	0.7789	0.7792	0.6543	0.7061	0.7962
CKM	1	*0.8424*	0.7809	0.7672	0.7738	0.5310	0.8181	0.7317
	2	*0.8623*	0.8120	0.7957	0.8008	0.4740	0.8126	0.7714
	3	*0.8059*	0.7725	0.7747	0.7684	0.5154	0.7601	0.6239
	Average	**0.8369**	0.7885	0.7792	0.7810	0.5068	0.7969	0.7090
CElegans	1	*0.7740*	0.7375	0.7358	0.7433	0.5527	0.6941	0.4967
	2	*0.8768*	0.7996	0.8099	0.7585	0.6708	0.7527	0.8371
	3	*0.8282*	0.7652	0.7685	0.7572	0.6317	0.7263	0.5675
	Average	**0.8263**	0.7674	0.7714	0.7530	0.6184	0.7244	0.6338
CS	1	*0.8969*	0.8831	0.8926	0.8720	0.5504	0.8257	0.7563
	2	0.8572	*0.8672*	0.8505	0.8597	0.7966	0.7833	0.6552
	3	*0.8878*	0.8572	0.8305	0.8667	0.5061	0.8188	0.8524
	4	*0.8300*	0.8141	0.8205	0.7537	0.6641	0.7568	0.8106
	5	0.8418	0.8339	*0.8488*	0.7742	0.6977	0.7794	0.6970
	Average	**0.8627**	0.8511	0.8486	0.8253	0.6430	0.7928	0.7543
TF	1	*0.8792*	0.7596	0.7888	0.7196	0.7843	0.8054	0.5223
	2	*0.9145*	0.8663	0.8822	0.8498	0.8088	0.8164	0.4269
	Average	**0.8969**	0.8130	0.8355	0.7847	0.7966	0.8109	0.4746

JC, PA, and two new features, i.e., FoN and FAL, into supervised methods. Table 5 lists the results measured by F1-score on the six networks. These results are obtained by encapsulating the SVM algorithm. In Table 5, the best average F1-score for each network is labeled in boldface and the best score for each layer of a network is indicated by italics. From the table, we observe that the proposed method gets the best average F1-scores on all networks, and best F1-scores on most of layers.

Intuitively, for a multiplex network with high overlap rates, if there are other types of interaction between two nodes, they have a big change to form an edge in the target layer. The feature of FAL is designed based on this idea. Fig. 2 tells us that the networks of Vicker and Lazega have higher overall overlap rates than other four networks. Therefore, we believe that FAL will be more prominent on these two networks than other features. The results in Table 5 support this conjecture. In addition, from Fig. 2(d) we find that the average overlap rate of layer 2 is remarkably higher than those of other two layers in the network of CElegans. Correspondingly, the F1-score of layer 2 listed in Table 5 is better compared with those of other two layers. Similar phenomena can also be observed on other networks

Besides, Figs. 10 and 11 present the Accuracy and MAE scores of the proposed method with supervised individual features on six networks. In this experiment, all the three classification algorithms are adopted. The results in this two figures again prove the effectiveness of our method. However, the results of individual features are less than satisfactory.

One of our major contributions is that we designed two new features, i.e., FoN and FAL. As illustrated above, both features outperforms other individual features on some networks. Here, we implement another experiment to further analyze the effectiveness of these two features. To this end, we compare the

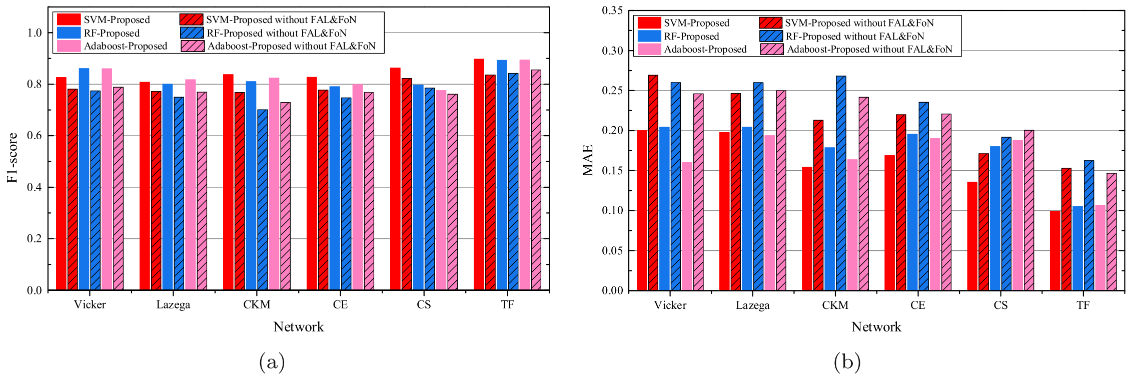


Fig. 12. Comparison of the proposed method with and without FoN and FAL. Each result is the average value of all layers.

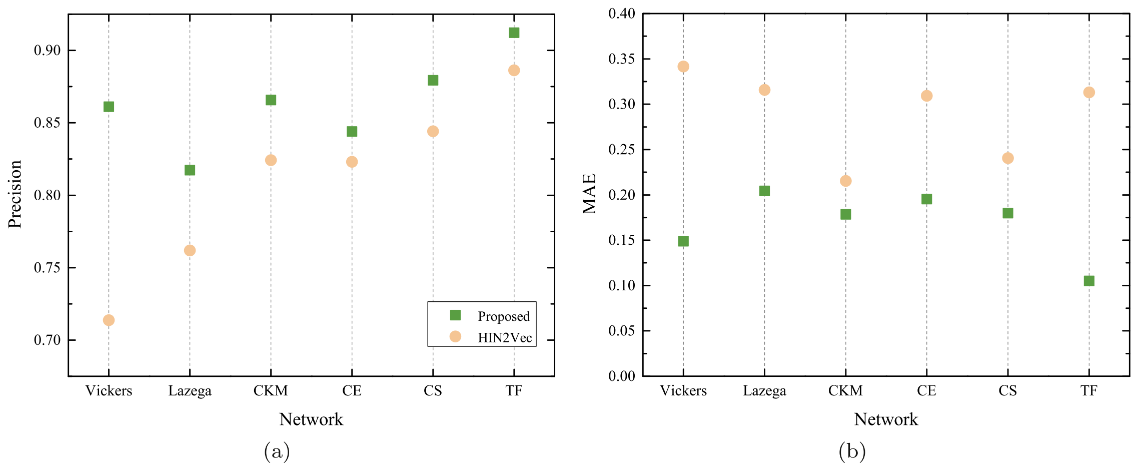


Fig. 13. Comparison of the proposed method with HIN2Vec. Each result is the average value of all layers.

corresponding results of the proposed method with and without these two features. The results in terms of F1-score and MAE are presented in Fig. 12. It can be observed from the figure that these two features can apparently enhance the prediction accuracy and effectively reduce the error rate. The reason is that both features capture the additional friendships for target nodes that other features do not take into account. Particularly, the feature of FAL indicates the friendships between target nodes in auxiliary layers. For a network with high overlap rates, it is indeed conducive to finding missing links in target layer.

5.7. Comparison with HIN2Vec

HIN2Vec [50] is a representation learning model for HINs. Given a group of relationships specified in forms of meta-paths in an HIN, HIN2Vec learns the latent vectors of nodes and edges by maximizing the likelihood of predicting relationships among nodes jointly. Since multiplex networks are a special case of HINs, an experiment is elaborated in this subsection to compare the prediction performance of the proposed method and the method based on HIN2Vec.[3] Node vectors used in the former are composed of the features introduced in this paper, while those in the latter are learned by HIN2Vec. In this experiment, the classification algorithm of RandomForest is adopted.

Fig. 13 shows the prediction results of these two methods under the metrics of Precision and MAE. It can be seen from the figure that the performance of the proposed method is superior to that of the method based on HIN2Vec. Therefore, the features extracted for the proposed method are favorable. In our point of view, the results in Fig. 13 may be caused by the following reasons:

(1) HIN2Vec is essentially designed for HINs. Although multiplex networks are some kind of HINs, they have very few meta-paths, which are the base of HIN2Vec.
(2) Different parameter settings in HIN2Vec can produce different results.
(3) The features in the proposed method are specified as per the purpose of link prediction, and thus conduce to positive prediction.

Furthermore, the main difference between the proposed method and HIN2Vec is: HIN2Vec learns the features of nodes and relationships automatically with the help of meta-paths; while the proposed method manually specifies the features of node pairs. However, in the proposed method, the features can be adjusted as needed.

6. Conclusion

In this paper, we investigated the link prediction problem in multiplex networks, which has not been fully explored so far. We proposed to solve the problem by considering it as a binary classification problem. To this end, we elaborated a set of features for node pairs and employed three supervised learning algorithms. In our point of view, auxiliary layers' information has positive influence on the formation of links in target layer.

[3] In the experiment, parameters of HIN2Vec are set as: dimensionality $d = 10$, negative sampling rate $n = 5$, meta-paths length $w = 2$, and length of random walks $l = 1000$. And vector function of node pairs is Hadamard.

Algorithm 2 Compute features

```
Input: The ith layer of multiplex network $G$; node pair $(x, y)$.
Output: Feature values of $(x, y)$.
1: $S_{CN} \leftarrow 0, S_{RA} \leftarrow 0, S_{CCLP} \leftarrow 0, S_{FoN} \leftarrow 0$;
2: Compute $S_{PA}$, $S_{LA}$ and $S_{ACC}$ according to Eqs. (14), (15) and (16), respectively;
3: for $z \in \Gamma_i(x)$ do
4:     if $z \in \Gamma_i(y)$ then
5:         $S_{CN} \leftarrow S_{CN} + 1$;
6:         $S_{RA} \leftarrow S_{RA} + 1/k_z^i$;
7:         $S_{CCLP} \leftarrow S_{CCLP} + CC_z^i$;
8:     end if
9: end for
10: $S_{JC} \leftarrow S_{CN}/(k_x^i + k_y^i - S_{CN})$;
11: if $x \in \Gamma_i(y)$ then
12:     $S_{FAL} \leftarrow 1$;
13: else
14:     $S_{FAL} \leftarrow 0$;
15: end if
16: for $u \in \Gamma_i(x)$ do
17:     for $v \in \Gamma_i(y)$ do
18:         if $u \in \Gamma_i(v)$ and $u \neq y$ then
19:             $S_{FoN} \leftarrow S_{FoN} + 1$
20:         end if
21:     end for
22: end for
23: return All feature values;
```

Therefore, the proposed method extracts the features of node pairs from all layers of the network.

To verify the superiority of the proposed method, abundant experiments were conducted on six multiplex networks under a series of evaluation metrics. Experimental results indicate that (1) the prediction performance can be improved by using structural information in auxiliary layers; (2) the proposed method outperforms baselines in most of cases; and (3) the feature of FAL performs good on networks with high overlap rates.

CRediT authorship contribution statement

Na Shan: Methodology, Software, Writing - original draft. **Longjie Li:** Conceptualization, Formal analysis, Writing - original draft, Writing - review & editing, Funding acquisition. **Yakun Zhang:** Software, Resources. **Shenshen Bai:** Resources, Formal analysis. **Xiaoyun Chen:** Investigation, Supervision.

Declaration of competing 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.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61602225) and the Fundamental Research Funds for the Central Universities, China (no. lzujbky-2019-90).

Appendix

Here, we discuss the time complexity for extracting feature values for node pair (x, y). Algorithm 2 lists the computation of feature values from the ith layer of multiplex network G. Since a number of features will use the degrees and clustering coefficients of nodes, we assume that both values of nodes are calculated in advance.

Line 2 in Algorithm 2 directly computes the values of PA, LA and ACC using Eqs. (14), (15) and (16), respectively. So, the time complexity is $O(1)$. In line 3 $\sim$ 9, features of CN, RA and CCLP are computed. In the worst case, the complexity is $O(\langle k\rangle^2)$, where $\langle k\rangle$ is the average degree of nodes. The complexities for calculating JC and FAL are $O(1)$ (line 10) and $O(\langle k\rangle)$ (line 11 $\sim$ 15), respectively. Finally, computational complexity for FoN is $O(\langle k\rangle^3)$ (line 16 $\sim$ 22) in the worst case. In result, the time complexity of Algorithm 2 is $O(1) + O(\langle k\rangle^2) + O(\langle k\rangle) + O(\langle k\rangle^3) = O(\langle k\rangle^3)$. For a multiplex network with k layers, the time complexity is $O(k\,\langle k\rangle^3)$.

References

[1] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks, Rev. Modern Phys. 74 (1) (2002) 47–97, http://dx.doi.org/10.1103/RevModPhys.74.47.

[2] H.-W. Shen, A.-L. Barabási, Collective credit allocation in science, Proc. Natl. Acad. Sci. 111 (34) (2014) 12325–12330, http://dx.doi.org/10.1073/pnas.1401992111.

[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.-U. Hwang, Complex networks: Structure and dynamics, Phys. Rep. 424 (4) (2006) 175–308, http://dx.doi.org/10.1016/j.physrep.2005.10.009.

[4] L. Lü, T. Zhou, Link prediction in complex networks: A survey, Physica A 390 (6) (2011) 1150–1170, http://dx.doi.org/10.1016/j.physa.2010.11.027.

[5] V. Martínez, F. Berzal, J.-c. Cubero, A survey of link prediction in complex networks, ACM Comput. Surv. 49 (4) (2017) 1–33, http://dx.doi.org/10.1145/3012704.

[6] L. Li, S. Fang, S. Bai, S. Xu, J. Cheng, X. Chen, Effective link prediction based on community relationship strength, IEEE Access 7 (2019) 43233–43248, http://dx.doi.org/10.1109/ACCESS.2019.2908208.

[7] C.V. Cannistraci, G. Alanis-Lobato, T. Ravasi, From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks, Sci. Rep. 3 (1) (2013) 1613, http://dx.doi.org/10.1038/srep01613.

[8] R. Guimerà, M. Sales-Pardo, Missing and spurious interactions and the reconstruction of complex networks, Proc. Natl. Acad. Sci. 106 (52) (2009) 22073–22078, http://dx.doi.org/10.1073/pnas.0908366106.

[9] R. Albert, I. Albert, G.L. Nakarado, Structural vulnerability of the North American power grid, Phys. Rev. E 69 (2) (2004) 025103, http://dx.doi.org/10.1103/PhysRevE.69.025103.

[10] D.J. Watts, S.H. Strogatz, Collective dynamics of small-world networks, Nature 393 (6684) (1998) 440–442, http://dx.doi.org/10.1038/30918.

[11] A. Cardillo, J. Gomez-Gardeñes, M. Zanin, M. Romance, D. Papo, F. Del Pozo, S. Boccaletti, Emergence of network features from multiplexity, Sci. Rep. 3 (1) (2013) 1344, http://dx.doi.org/10.1038/srep01344.

[12] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, J. Am. Soc. Inf. Sci. Technol. 58 (7) (2007) 1019–1031, http://dx.doi.org/10.1002/asi.20591.

[13] B. Pandey, P.K. Bhanodia, A. Khamparia, D.K. Pandey, A comprehensive survey of edge prediction in social networks: techniques, parameters and challenges, Expert Syst. Appl. 124 (2019) 164–181, http://dx.doi.org/10.1016/j.eswa.2019.01.040.

[14] P. Wang, B. Xu, Y. Wu, X. Zhou, Link prediction in social networks: the state-of-the-art, Sci. China Inf. Sci. 58 (1) (2015) 1–38, http://dx.doi.org/10.1007/s11432-014-5237-y.

[15] Z. Li, X. Fang, O.R.L. Sheng, A survey of link recommendation for social networks, ACM Trans. Manage. Inf. Syst. 9 (1) (2017) 1–26, http://dx.doi.org/10.1145/3131782.

[16] L. Li, S. Bai, M. Leng, L. Wang, X. Chen, Finding missing links in complex networks: A multiple-attribute decision-making method, Complexity 2018 (2018) 1–16, http://dx.doi.org/10.1155/2018/3579758.

[17] S. Bai, L. Li, J. Cheng, S. Xu, X. Chen, Predicting missing links based on a new triangle structure, Complexity 2018 (2018) 1–11, http://dx.doi.org/10.1155/2018/7312603.

[18] M.E. Newman, Clustering and preferential attachment in growing networks, Phys. Rev. E 64 (2) (2001) 4, http://dx.doi.org/10.1103/PhysRevE.64.025102.

[19] E. Ravasz, A.L. Somera, D.A. Mongru, Z.N. Oltvai, A.L. Barabási, Hierarchical organization of modularity in metabolic networks, Science 297 (5586) (2002) 1551–1555, http://dx.doi.org/10.1126/science.1073374.

[20] T. Zhou, L. Lü, Y.-C. Zhang, Predicting missing links via local information, Eur. Phys. J. B 71 (4) (2009) 623–630, http://dx.doi.org/10.1140/epjb/e2009-00335-8.

[21] E.A. Leicht, P. Holme, M.E. Newman, Vertex similarity in networks, Phys. Rev. E 73 (2) (2006) 26120, http://dx.doi.org/10.1103/PhysRevE.73.026120.

[22] L.A. Adamic, E. Adar, Friends and neighbors on the web, Social Networks 25 (3) (2003) 211–230, http://dx.doi.org/10.1016/S0378-8733(03)00009-1.

[23] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512, http://dx.doi.org/10.1126/science.286.5439.509.

[24] A. Pecli, M.C. Cavalcanti, R. Goldschmidt, Automatic feature selection for supervised learning in link prediction applications: a comparative study, Knowl. Inf. Syst. 56 (1) (2018) 85–121, http://dx.doi.org/10.1007/s10115-017-1121-6.

[25] C. Ahmed, A. Elkorany, R. Bahgat, A supervised learning approach to link prediction in Twitter, Soc. Netw. Anal. Min. 6 (1) (2016) 1–11, http://dx.doi.org/10.1007/s13278-016-0333-1.

[26] M. Fire, L. Tenenboim-Chekina, R. Puzis, O. Lesser, L. Rokach, Y. Elovici, Computationally efficient link prediction in a variety of social networks, ACM Trans. Intell. Syst. Technol. (TIST) 5 (1) (2013) 1–25, http://dx.doi.org/10.1145/2542182.2542192.

[27] V. Nicosia, G. Bianconi, V. Latora, M. Barthelemy, Growing multiplex networks, Phys. Rev. Lett. 111 (5) (2013) 058701, http://dx.doi.org/10.1103/PhysRevLett.111.058701.

[28] M. Szell, R. Lambiotte, S. Thurner, Multirelational organization of large-scale social networks in an online world, Proc. Natl. Acad. Sci. 107 (31) (2010) 13636–13641, http://dx.doi.org/10.1073/pnas.1004008107.

[29] S. Boccaletti, G. Bianconi, R. Criado, C.I. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, M. Zanin, The structure and dynamics of multilayer networks, Phys. Rep. 544 (1) (2014) 1–122, http://dx.doi.org/10.1016/j.physrep.2014.07.001.

[30] K.-M. Lee, B. Min, K.-I. Goh, Towards real-world complexity: an introduction to multiplex networks, Eur. Phys. J. B 88 (2) (2015) 48, http://dx.doi.org/10.1140/epjb/e2015-50742-1.

[31] Y. Yao, R. Zhang, F. Yang, Y. Yuan, Q. Sun, Y. Qiu, R. Hu, Link prediction via layer relevance of multiplex networks, Internat. J. Modern Phys. C 28 (08) (2017) 1750101, http://dx.doi.org/10.1142/S0129183117501017.

[32] H. Mandal, M. Mirchev, S. Gramatikov, I. Mishkovski, Multilayer link prediction in online social networks, in: 2018 26th Telecommunications Forum (TELFOR), IEEE, 2018, pp. 1–4, http://dx.doi.org/10.1109/TELFOR.2018.8612122.

[33] S. Sharma, A. Singh, An efficient method for link prediction in complex multiplex networks, in: 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), IEEE, 2015, pp. 453–459, http://dx.doi.org/10.1109/SITIS.2015.93.

[34] S. Najari, M. Salehi, V. Ranjbar, M. Jalili, Link prediction in multiplex networks based on interlayer similarity, Physica A 536 (2019) 120978, http://dx.doi.org/10.1016/j.physa.2019.04.214.

[35] A. Kumar, S.S. Singh, K. Singh, B. Biswas, Link prediction techniques, applications, and performance: A survey, Physica A (2020) 124289, http://dx.doi.org/10.1016/j.physa.2020.124289.

[36] S. Haghani, M.R. Keyvanpour, A systemic analysis of link prediction in social network, Artif. Intell. Rev. 52 (3) (2019) 1961–1995, http://dx.doi.org/10.1007/s10462-017-9590-2.

[37] A. Divakaran, A. Mohan, Temporal link prediction: A survey, New Gener. Comput. 38 (2019) 213–258, http://dx.doi.org/10.1007/s00354-019-00065-z.

[38] Z. Bu, Y. Wang, H.-J. Li, J. Jiang, Z. Wu, J. Cao, Link prediction in temporal networks: Integrating survival analysis and game theory, Inform. Sci. 498 (2019) 41–61, http://dx.doi.org/10.1016/j.ins.2019.05.050.

[39] C. Chiu, J. Zhan, Deep learning for link prediction in dynamic networks using weak estimators, IEEE Access 6 (2018) 35937–35945, http://dx.doi.org/10.1109/ACCESS.2018.2845876.

[40] P. Goyal, S.R. Chhetri, A. Canedo, Dyngraph2vec: Capturing network dynamics using dynamic graph representation learning, Knowl.-Based Syst. 187 (2020) 104816, http://dx.doi.org/10.1016/j.knosys.2019.06.024.

[41] J. Chen, X. Lin, C. Jia, Y. Li, Y. Wu, H. Zheng, Y. Liu, Generative dynamic link prediction, Chaos 29 (12) (2019) 123111, http://dx.doi.org/10.1063/1.5120722.

[42] K. Chi, G. Yin, Y. Dong, H. Dong, Link prediction in dynamic networks based on the attraction force between nodes, Knowl.-Based Syst. 181 (2019) 104792, http://dx.doi.org/10.1016/j.knosys.2019.05.035.

[43] S. De Winter, T. Decuypere, S. Mitrovic, B. Baesens, J. De Weerdt, Combining temporal aspects of dynamic networks with node2vec for a more efficient dynamic link prediction, in: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2018, pp. 1234–1241, http://dx.doi.org/10.1109/ASONAM.2018.8508272.

[44] A. Grover, J. Leskovec, Node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 2016, pp. 855–864, http://dx.doi.org/10.1145/2939672.2939754.

[45] J. Chen, J. Zhang, X. Xu, C. Fu, D. Zhang, Q. Zhang, Q. Xuan, E-LSTM-D: A deep learning framework for dynamic network link prediction, IEEE Trans. Syst. Man Cybern.: Syst. (2019) 1–14, http://dx.doi.org/10.1109/TSMC.2019.2932913.

[46] Y. Sun, Y. Yu, J. Han, Ranking-based clustering of heterogeneous information networks with star network schema, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 797–806, http://dx.doi.org/10.1145/1557019.1557107.

[47] Y. Sun, J. Han, Mining heterogeneous information networks: A structural analysis approach, SIGKDD Explor. Newslett. 14 (2) (2013) 20–28, http://dx.doi.org/10.1145/2481244.2481248.

[48] H. Shakibian, N.M. Charkari, Statistical similarity measures for link prediction in heterogeneous complex networks, Physica A 501 (2018) 248–263, http://dx.doi.org/10.1016/j.physa.2018.02.189.

[49] X. Cao, Y. Zheng, C. Shi, J. Li, B. Wu, Link prediction in schema-rich heterogeneous information network, in: Advances in Knowledge Discovery and Data Mining, in: PAKDD 2016, Springer International Publishing, Cham, 2016, pp. 449–460, http://dx.doi.org/10.1007/978-3-319-31753-3_36.

[50] T.-y. Fu, W.-C. Lee, Z. Lei, Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, in: CIKM '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 1797–1806, http://dx.doi.org/10.1145/3132847.3132953.

[51] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, Q. Liu, SHINE: Signed heterogeneous information network embedding for sentiment link prediction, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, in: WSDM '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 592–600, http://dx.doi.org/10.1145/3159652.3159666.

[52] M. Lu, X. Wei, D. Ye, Y. Dai, A unified link prediction framework for predicting arbitrary relations in heterogeneous academic networks, IEEE Access 7 (2019) 124967–124987, http://dx.doi.org/10.1109/access.2019.2939172.

[53] M. Pujari, R. Kanawati, Link prediction in multiplex networks, Netw. Heterog. Media 10 (1) (2015) 17–35, http://dx.doi.org/10.3934/nhm.2015.10.17.

[54] A. Hajibagheri, G. Sukthankar, K. Lakkaraju, A holistic approach for link prediction in multiplex networks, in: Social Informatics, Springer International Publishing, 2016, pp. 55–70, http://dx.doi.org/10.1007/978-3-319-47874-6_5.

[55] Z. Samei, M. Jalili, Discovering spurious links in multiplex networks based on interlayer relevance, J. Complex Netw. 7 (5) (2019) 641–658, http://dx.doi.org/10.1093/comnet/cnz007.

[56] Z. Samei, M. Jalili, Application of hyperbolic geometry in link prediction of multiplex networks, Sci. Rep. 9 (1) (2019) 12604, http://dx.doi.org/10.1038/s41598-019-49001-7.

[57] L. Chen, M. Gao, B. Li, W. Liu, B. Chen, Detect potential relations by link prediction in multi-relational social networks, Decis. Support Syst. 115 (September) (2018) 78–91, http://dx.doi.org/10.1016/j.dss.2018.09.006.

[58] A.M. Abdolhosseini-Qomi, S.H. Jafari, A. Taghizadeh, N. Yazdani, M. Asadpour, M. Rahgozar, Link prediction in real-world multiplex networks via layer reconstruction method, 2019, arXiv:1906.09422.

[59] M. Kivela, A. Arenas, M. Barthelemy, J.P. Gleeson, Y. Moreno, M.A. Porter, Multilayer networks, J. Complex Netw. 2 (3) (2014) 203–271, http://dx.doi.org/10.1093/comnet/cnu016.

[60] D. Zhao, L. Li, H. Peng, Q. Luo, Y. Yang, Multiple routes transmitted epidemics on multiplex networks, Phys. Lett. A 378 (10) (2014) 770–776, http://dx.doi.org/10.1016/j.physleta.2014.01.014.

[61] B. Chen, Y. Hua, Y. Yuan, Y. Jin, Link prediction on directed networks based on AUC optimization, IEEE Access 6 (2018) 28122–28136, http://dx.doi.org/10.1109/ACCESS.2018.2838259.

[62] T. Feyessa, M. Bikdash, G. Lebby, Node-pair feature extraction for link prediction, in: 2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing, IEEE, 2011, pp. 1421–1424, http://dx.doi.org/10.1109/PASSAT/SocialCom.2011.244.

[63] Z. Wu, Y. Lin, J. Wang, S. Gregory, Link prediction with node clustering coefficient, Physica A 452 (2016) 1–8, http://dx.doi.org/10.1016/j.physa.2016.01.038.

[64] M. Vickers, S. Chan, Representing Classroom Social Structure, Victoria Institute of Secondary Education, Melbourne, 1981.

[65] L. Emmanuel, The Collegial Phenomenon: The Social Mechanisms of Cooperation among Peers in a Corporate Law Partnership, Oxford University Press, 2001.

[66] T.A.B. Snijders, P.E. Pattison, G.L. Robins, M.S. Handcock, New specifications for exponential random graph models, Sociol. Methodol. 36 (1) (2006) 99–153, http://dx.doi.org/10.1111/j.1467-9531.2006.00176.x.

[67] J. Coleman, E. Katz, H. Menzel, The diffusion of an innovation among physicians, Sociometry 20 (4) (1957) 253, http://dx.doi.org/10.2307/2785979.

[68] B.L. Chen, D.H. Hall, D.B. Chklovskii, Wiring optimization can relate neuronal structure and function, Proc. Natl. Acad. Sci. 103 (12) (2006) 4723–4728, http://dx.doi.org/10.1073/pnas.0506806103.

[69] M. De Domenico, M.A. Porter, A. Arenas, Muxviz: a tool for multilayer analysis and visualization of networks, J. Complex Netw. 3 (2) (2015) 159–176, http://dx.doi.org/10.1093/comnet/cnu038.

[70] M. Magnani, B. Micenkova, L. Rossi, Combinatorial analysis of multiple networks, 2013, arXiv:1303.4986.