

Research Article

An Interlayer Link Prediction Method Based on Edge-Weighted Embedding

Hefei Hu ^{1,2}, Sirui Zhang ¹ and Yanan Wang¹

¹School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

²Beijing Key Laboratory of Work Safety Intelligent Monitoring, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Sirui Zhang; 2847613146@qq.com

Received 25 September 2023; Revised 6 November 2023; Accepted 16 November 2023; Published 8 December 2023

Academic Editor: Siew Ann Cheong

Copyright © 2023 Hefei Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Presently, users usually register accounts on online social networks (OSNs). Identifying the same user in different networks is also known as interlayer link prediction. Most existing interlayer link prediction studies use embedding methods, which represent nodes in a common representation space by learning mapping functions. However, these studies often directly model links within the pre-embedding layer as equal weights, fail to effectively distinguish the strength of edge relationships, and do not fully utilize network topology information. In this paper, we propose an interlayer link prediction model based on weighted embedding of connected edges within the network layer, which models the links within the network layer as weighted graphs to better represent the network and then uses appropriate embedding methods to represent the network in a low-dimensional space. After embedding, vector similarity and distance similarity are used as comprehensive evaluation scores. This paper has conducted a large number of simulation experiments on actual networks. The results show that our proposed model has higher prediction accuracy in all aspects than current advanced models and can achieve the highest accuracy when the training frequency is low, which proves the validity of the proposed model.

1. Introduction

The ubiquity of the Internet has ushered in a new era of online social interaction, with platforms such as Twitter, Facebook, and Instagram becoming integral to our daily lives. In reality, due to the different functions of different OSNs, people tend to join multiple different OSNs and register their own accounts on each network. A user may be active on different OSNs using different accounts. Since different OSNs are usually maintained by different providers, we cannot know whether accounts across different OSNs belong to the same person, which means there are many potential users on different social networks. Matching user accounts across different OSNs is also known as interlayer link prediction[1]. With the enrichment of people's online lives, there are more and more users on different OSNs, which create more opportunities for network mining and learning and also bring more

significance and economic benefits to the research of interlayer link prediction [2]. Interlayer link prediction can link the identity of a single user on multiple OSNs, getting rid of the limitation that analyzing the identity of a single social media user may not be able to fully understand the user's personality and interests. It can fully understand the user's interests, enabling service providers to provide accurate and effective services to users. In addition, interlayer link prediction can also help combat some illegal and criminal activities [3]. For example, some criminals may register a large number of different accounts on OSNs and carry out illegal activities such as spreading viruses or committing fraud on these applications. If there is an effective method to determine the corresponding relationship between accounts of the same user on different OSNs, we can model their illegal network behavior, lock their geographical location, and even determine their true identity, thus effectively cracking down on them.

With the development of network embedding technology [4], researchers are considering using network embedding methods for interlayer link prediction [5–10]. Most of them first unify all nodes into a common latent space and then use prior interlayer links to train the approximate mapping function after obtaining the latent representation. However, such a method is difficult to train a mapping function with perfect performance, because each layer of the latent space is an independent individual, and they are unknown. Especially, when the amount of training is small, the performance of this method will be worse. In addition, existing methods often consider the connections within the embedded multiplex network as equal weights or fail to effectively distinguish the strength of the connection relationship, which ignores the information within the network layer and fails to fully utilize the network structure information. However, node pairs in different layers may have different relationship strengths. The higher the relationship strength between nodes, the closer the embedding vectors in the latent space.

In this paper, the interlayer link prediction model we proposed is divided into two stages: embedding and matching. Unlike other methods that treat connections within the network as equal weights, in the embedding phase, our model can well measure the weight of connecting edges in the network layer by combining the edge betweenness and degree centrality and weight the strength of connecting edges between different node pairs. In this way, the vector of the latent space will make more in-depth uses of the known information of the network, which can improve the prediction accuracy of interlayer links. Tang et al. [11] use the first-order proximity of the line [12] embedding algorithm to embed the network, while our model uses the first-order and second-order proximities of the line [12] embedding algorithm to embed the network. This can obtain a more effective representation of nodes in a low-dimensional space, capture network topology information again, and filter out irrelevant factors. The first-order and second-order proximity degrees trained by the embedding algorithm are connected to obtain the vector representation of each layer of the network node in a low-dimensional space. In the matching phase, since the two networks are embedded in different vector spaces, the approximate mapping function is trained using the prior interlayer links so as to learn a stable cross-network mapping and unify the nodes of the two layers into the common latent space. Finally, by comparing the vector similarity and distance similarity of the embedding vectors of unmatched nodes in the common latent space, we can determine the nodes most likely to correspond to the current network in the other layer network, thus predicting the unobserved interlayer links.

In this study, we developed a model for interlayer link prediction in the multiplex network. The major contributions of this paper can be summarized as follows:

- (i) We will model each layer of the multiplex network as a weighted graph using a combination of edge betweenness and degree centrality to obtain a representation based solely on the network structure. In this way, the vector of the latent space will further

utilize the structural information of the network, which can improve the accuracy of interlayer link prediction.

- (ii) We have innovatively incorporated the second-order proximity of the line [12] embedding algorithm into the representation of embedding vectors, which can obtain more effective node representations in a low-dimensional space, capture network topology information again, and filter out irrelevant factors
- (iii) To reduce time complexity, we have employed matrix multiplication to optimize the computation process of vector consistency and distance consistency for all nonmatching node pairs. This means that by cleverly utilizing matrix operations, we can more efficiently handle the calculation of distance and vector similarities for all nonmatching node pairs, thus improving the algorithm's efficiency.

The rest of this paper is organized as follows. Section 2 summarizes the related works. Section 3 formalizes the problem of interlayer link prediction and elaborates on the details of our proposed algorithm. Section 4 carries out the experimental evaluation, and finally, Section 5 concludes the paper.

2. Related Works

As more and more people study network embedding technology, the technology has become more accurate and efficient. Liu et al. [13] proposed a representation learning model aimed at learning alignment network embedding of multiple networks, explicitly modeling each user's follower and followee as input and output contexts. In this model, both given and potential anchor links can be used as hard and soft constraints in the unified learning framework. They further improved their proposed method by incorporating structural diversity in [14]. Structural diversity mainly focuses on the influence of prior matching nodes from different communities. Man et al. [15] proposed an anchor link prediction model based on embedding and matching. This model adopts a network embedding method to maintain the main structural laws of the network while being aware of supervised anchor chains and then learns stable cross-network mapping for anchor chain prediction. Zhou et al. [16] proposed a semisupervised method based on deep reinforcement learning to study the UIL problem by utilizing the duality of mappings between any two networks, thereby improving prediction accuracy.

There are two methods to solve the problem of interlayer link prediction: (i) based on feature extraction [17–21] and (ii) based on network structure [14, 22–32]. In recent years, research has mainly focused on network structure-based methods. For example, COSNET [22] proposed an energy-based model that links user identities by considering local and global consistency, first extracting distance-based contour features and neighbor-

based network features, and then using aggregation algorithms to obtain local consistency. MAH [23] used hypergraphs to model embedded network features and proposed an embedding method that mapped user identity to a low-dimensional space. Distance-based profile features (such as usernames) were also utilized in the MAH model to achieve better performance. PCT [24] aimed to infer potential corresponding connections that connected multiple shared entities simultaneously across networks by combining contour and network features. The authors in [31] proposed a new link prediction method, namely, weighted common neighbors (WCNs), which predicts the formation of new links in multiplex networks based on common neighbors and various types of centrality measures. Therefore, in this model, each public neighbor has a different impact on the likelihood of node connectivity. Our model inherits the ideas of this paper and makes improvements based on it. Collective random walk (CRW) [25] predicted the formation of social links between users in the target network and the alignment of anchor links between the target network and other external social networks, which included two stages: (1) collective link prediction of anchor points and social links; (2) propagate predicted links through collective random walks in partially aligned probability networks. The authors in [32] proposed an extended version of local random walk based on pure random walking for solving link prediction in the multiplex network, referred to as the multiplex local random walk (MLRW). It explored approaches for leveraging information mined from interlayer and intralayer in a multiplex network to define a biased random walk for finding the probability of the appearance of a new link in one target layer.

The methods mentioned above mostly focus on optimizing the learning framework, enhancing algorithm efficiency, and using multiple attributes for embedding. However, it is difficult to train mapping functions with high accuracy. In addition, before embedding, the connections within the network are often directly considered equal weight, or the strength of the connection relationship is not effectively distinguished, which ignores the information within the network layer and fails to fully utilize the network structure information. According to [33, 34], in this paper, we propose a model that can measure the weight of the connected edges in the network layer by combining edge betweenness and degree centrality. By weighting the edge strength between the different node pairs, we can get the vector of the latent space which will make more in-depth use of the known information of the network and improve the prediction accuracy of the interlayer link. After that, the first-order proximity and second-order proximity of the line [12] embedding algorithm are used to embed the network, obtain the effective representation of nodes in the low-dimensional space, capture the network topology information again, and filter out the insignificant factors. In conclusion, we propose an interlayer link prediction method based on edge-weighted embedding for interlayer link prediction.

3. Preliminaries and Problem Statement

3.1. Definitions. In general, we use $G(V, E)$ to represent a layer of the network, where V represents the point set and E represents the edge set. E reflects the connections between nodes. A two-layer network can form a multiplex network. In such a scenario, we refer to one network as the source network $G^s(V^s, E^s)$ and the other network as the target network $G^t(V^t, E^t)$. In the model of this section, the paper overlooks the situation where the same user registers multiple accounts within a social network, meaning that individuals within a layer of the network are different. The relevant concepts involved in the study are as follows.

3.1.1. Intralayer Link and Interlayer Link (Anchor Links).

For a multiplex network consisting of two-layer networks $G^s(V^s, E^s)$ and $G^t(V^t, E^t)$, E^s and E^t are the sets of intralayer links of the source network and the target network, respectively. If there is an edge set $E^{st} \subseteq V^s \times V^t$, where one endpoint of an edge in the set belongs to V^s and another endpoint belongs to V^t , then the edge is called an interlayer link. The interlayer links provided in advance are called a priori interlayer link or a priori interlayer node pair, while the other interlayer links are called unobserved interlayer links. Meanwhile, the node pairs in the interlayer link are called matched nodes, and the other nodes in the network are called unmatched nodes.

3.1.2. Network Embedding Model.

For a layer of network in a multiplex network, such as $G^s(V^s, E^s)$, the network embedding model learns low-dimensional network node representations while maintaining the network structure, representing nodes from different layers into latent spaces. After network embedding, a node $v_i^s \in V^s$ is represented as a d -dimensional vector $\mathbf{v}_i^s \in R^d$, where d represents the chosen dimension for the embedding method. The source network G^s is embedded into the vector space Z^s , and the target network G^t is embedded into the vector space Z^t .

3.1.3. Common Matched Neighbor (CMN).

For a prior interlayer node pair (v_i^s, v_j^t) , if there is a node v_a^s connected to a node v_i^s in the source network G^s and a node v_b^t connected to v_j^t in the target network G^t , then the prior interlayer node pair (v_i^s, v_j^t) is called a common matched neighbor (CMN) of the node v_b^t and the node v_j^t .

3.1.4. Mapping Function.

After performing network embedding, the two-layer network is represented as two low-dimensional vector spaces. In order to match later, it is necessary to use prior interlayer links in the training set for obtaining the mapping relationship between the two networks. If we define ϕ as a mapping function, then for the prior interlayer node pairs (v_i^s, v_j^t) , there is $\phi(\mathbf{v}_i^s) = \mathbf{v}_j^t$. After training the mapping function, the two-layer low-dimensional vector spaces Z^s, Z^t can be unified into a common latent space.

The interlayer link prediction of double-layer networks aims to identify whether each node in the source network has a corresponding node in the target network, that is, selecting any unmatched interlayer node pair to determine whether there is an interlayer link given $G^s(V^s, E^s)$, $G^t(V^t, E^t)$, and a known interlayer link $E^{st} = \{(v_i^s, v_j^t) | v_i^s \in V^s, v_j^t \in V^t\}$.

Supervised learning is used in both the embedding and matching stages. During the embedding stage, nodes are represented in a low-dimensional vector space, namely, $G^s \rightarrow \mathbf{Z}^s$ and $G^t \rightarrow \mathbf{Z}^t$. In the matching stage, a prior interlayer link is used to train the mapping function and obtain the mapping relationship between two vector spaces, namely, $\phi(v_i^s) = v_j^t$. To make the final unified common space more realistic in reflecting network information, it is necessary to minimize the loss function in these two stages, which means that the objective function is

$$\min\{L_1(G^s, G^t, \mathbf{Z}^s, \mathbf{Z}^t) + L_2(\phi, \mathbf{Z}^s, \mathbf{Z}^t, E^{st})\}, \quad (1)$$

where $L_1(G^s, G^t, \mathbf{Z}^s, \mathbf{Z}^t)$ is the loss function represented by the source network G^s and the target network G^t in the low-dimensional space \mathbf{Z}^s and \mathbf{Z}^t processes, respectively. $L_2(\phi, \mathbf{Z}^s, \mathbf{Z}^t, E^{st})$ is the loss function in the matching process, reflecting the accuracy of the precise prediction of the prior interlayer links in the training set. However, due to the interdependence of \mathbf{Z}^s , \mathbf{Z}^t , and ϕ , it is difficult to jointly optimize the abovementioned objective functions. Therefore, a two-step optimization method was adopted in the subsequent model. Table 1 lists the main symbols and their respective meanings used in the models proposed in this section. We set it according to common symbol conventions. Among them, bold uppercase letters represent matrices, bold lowercase letters represent vectors, and lowercase letters represent scalars.

3.2. Problem Statement. We propose the similarities of weighted line embedded vector (SWLEV) model for interlayer link prediction based on intralayer edge weighting. The model includes the following parts: intralayer edge weighting, network embedding, vector space matching and similarity calculation, distance similarity calculation, and comprehensive evaluation prediction. Figure 1 shows the schematic diagram of the frame. The individuals connected by vertical brown solid lines between layers belong to the same user in real life, and the corresponding relationship between individuals is known, while the dotted lines between layers indicate that the corresponding relationship between individuals is unknown. The purpose of interlayer link prediction is to determine the corresponding relationship of other individuals. Figure 1(b) shows the corresponding low-dimensional vector space obtained after the two networks are embedded. Before embedding, it is necessary to weigh the links within the network layer. Figure 1(c1) performs neural network training to uniformly represent two low-dimensional vector spaces into a common latent space. Figure 1(c2) performs vector similarity calculation, i.e., calculates the score s_1 mentioned later in this section. Figure 1(d) calculates the distance similarity score,

i.e., the s_2 score, of two low-dimensional vectors. We do not need to carry out the unity of the vector space and only need to calculate the corresponding Euclidean distance relations. Finally, the vector similarity score and the distance similarity score are combined with certain control parameters to represent the prediction score. The details of each part are described as follows.

3.2.1. Intralayer Edge Weighting. Most existing embedded methods ignore the relationship strength of intralayer edges before embedding, treating the intralayer edges of single-layer networks as equal weights, or failing to provide a good representation of the intralayer edge strength. In fact, connections within different layers may have different relationship strengths. The higher the strength of edges between nodes, the higher the similarity of the embedding vectors represented in the latent space after embedding and the more fully utilized the network structure information. Therefore, this paper weights the edges between nodes based on the network structure.

The edge betweenness represents the proportion of the shortest path between nodes connected by this edge to all shortest paths, which can reflect the importance of the edge well. The larger the edge betweenness, the more important and stronger the edge. However, the degree of an edge endpoint also affects the edge betweenness. Considering this factor, we propose a network layer edge-weighted model as follows:

$$w_{ij} = \frac{C_B(e_{ij})}{\sqrt{C_D^i \times C_D^j}} \cdot e_{ij}, \quad (2)$$

where $C_B(e_{ij})$ is the normalized edge betweenness and C_D^i and C_D^j are the degree centrality of the two endpoints v_i and v_j of the edge, respectively, represented as follows:

$$C_B(e) = \sum_{s,t \in V} \frac{\sigma(s,t|e)}{\sigma(s,t)}, \quad (3)$$

$$C_D^i = \frac{k_i}{N-1},$$

where V is the set of nodes, $\sigma(s,t)$ represents the number of shortest paths between nodes s and t , the number of edges e passing through these shortest paths is represented as $\sigma(s,t|e)$, k_i is the degree of the node v_i , and $N = |V|$ is the total number of nodes. After representing the two-layer network with intralayer edge weighting, we embed the network in the form of a weighted graph.

3.2.2. Network Embedding. Network embedding is a commonly used method in network research. Due to the complexity and sparse network in reality, directly processing them can result in high computational complexity. Network embedding learns the low-dimensional representation of network nodes while maintaining the network structure, representing nodes from different layers into different low-dimensional spaces, thereby extracting low-dimensional and compact

TABLE 1: The main symbols and meanings of the model.

Symbols	Meanings
G^s, G^t	Source network, target network
w	Weight of intralayer edge
v	Nodes in the network
\mathbf{v}	The final embedding vector representation of the node v
d	Dimension of embedded vectors
$\mathbf{Z}^s, \mathbf{Z}^t$	Low-dimensional space after embedding source and target networks
ϕ	Mapping function
E^{st}	Prior interlayer links
L_1, L_2	Loss function in embedding and mapping stages
e_{ij}	Intralayer link
$C_B(e_{ij})$	Edge betweenness
C_D	Degree centrality of nodes v_i
\mathbf{u}_i	The vector represented by nodes in second-order proximity
\mathbf{u}_i^t	Context representation in second-order proximity
E^T, E^P	Training set, testing set
Γ	Neighbor set of nodes
s_1, s_2, s	Vector similarity score, distance similarity score, comprehensive evaluation score
HV, HD, S	Vector similarity matrix, distance similarity matrix, comprehensive evaluation matrix
d_{ai}^s	Euclidean distance of embedded vectors
ε	Weight control factor

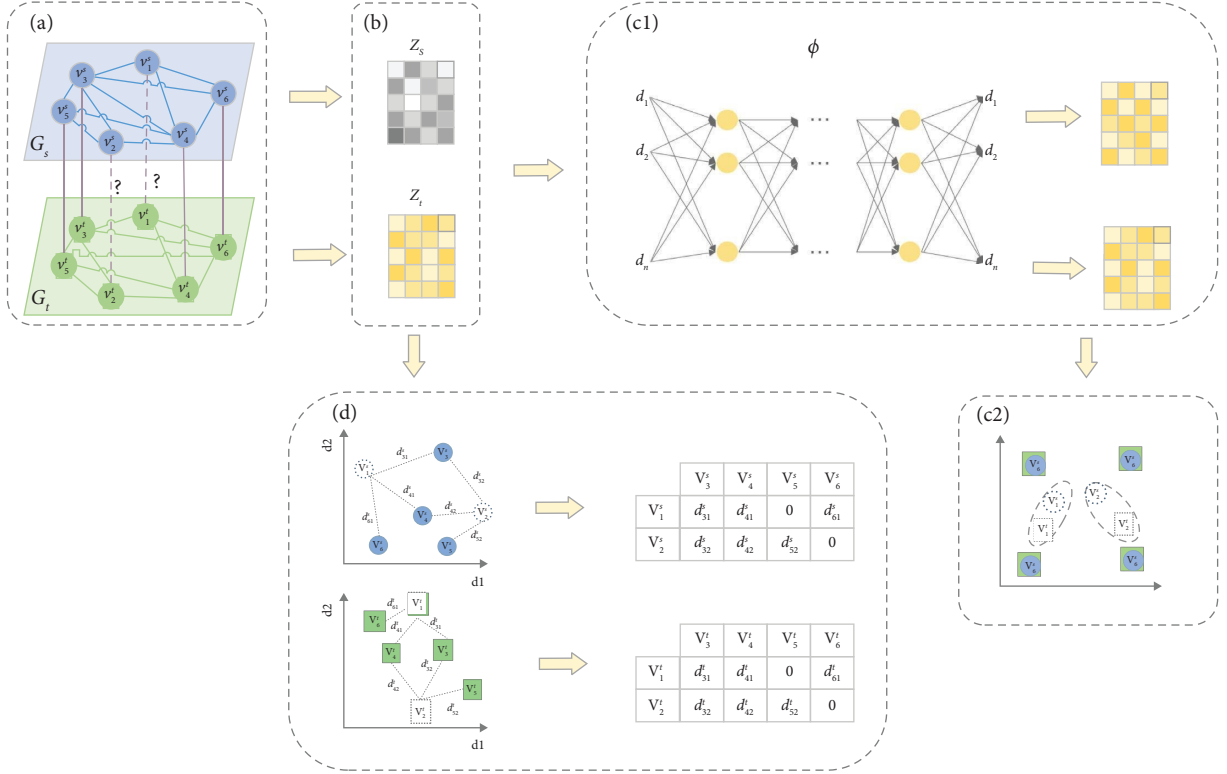


FIGURE 1: Schematic diagram of SWLEV framework. (a) Double-layer network structure diagram. (b) Network embedding. (c1) Neural network training. (c2) Vector similarity calculation. (d) Distance similarity calculation.

features from high-dimensional spaces. This helps us to perform high-level analysis tasks more efficiently in time and space, greatly improving computational efficiency. The idea of embedding is to minimize the distance between vectors corresponding to adjacent nodes, while maximizing the distance between vectors corresponding to nonadjacent nodes.

After each layer of the network is represented with weighted intralayer edges, we use the line [12] embedding model for node representation. Here, we use the joint first-order and second-order proximity of the line embedding algorithm. The meaning of first-order proximity is that if there are two nodes directly connected to an edge with a higher degree, the distance between the embedded vectors will be relatively close. The idea of the second-order proximity is that vertices that share many neighbors with other nodes are similar to each other. For two-layer networks, we embedded the first-order proximity and the second-order proximity, respectively:

(a) First-order proximity embedding

For the connection of two points within a layer, such as $e_{ij}^s = (v_i^s, v_j^s)$, the joint probability distribution of nodes v_i and v_j is defined as follows:

$$p_1(v_i^s, v_j^s) = \frac{1}{1 + \exp(-(\mathbf{V}_i^s)^T \mathbf{V}_j^s)}, \quad (4)$$

where the vectors $\mathbf{v}_i^s, \mathbf{v}_j^s \in R^d$ in the denominator are the vector representations of nodes v_i^s, v_j^s in a low-dimensional space and $(\cdot)^T$ represents the

transposition of the matrix. The empirical probability distribution is as follows:

$$\hat{p}_1(v_i^s, v_j^s) = \frac{w_{ij}^s}{W}, \quad (5)$$

where w_{ij}^s is the weight of the intralayer edge e_{ij}^s , which can be calculated from intralayer edge weighting. W is the sum of the intralayer edge weights in the network layer, expressed as $W = \sum_{(i,j) \in E^s} w_{ij}^s$. To maintain first-order proximity, the direct method is to minimize the distance between these two distributions, namely,

$$O_1 = d(\hat{p}_1(\cdot, \cdot), p_1(\cdot, \cdot)). \quad (6)$$

KL-divergence is a method of representing the similarity between two distributions. We use KL-divergence instead of the above distance to find the KL-divergence of the joint probability distribution $p_1(v_i^s, v_j^s)$ and empirical probability distribution $\hat{p}_1(v_i^s, v_j^s)$ for all connected edges in a layer of the network, which is

$$O_1 = - \sum_{\forall (v_i^s, v_j^s) \in E^s} \text{KL}(p_1(v_i^s, v_j^s), \hat{p}_1(v_i^s, v_j^s)). \quad (7)$$

After omitting some constants, the objective function in (7) can be expressed as follows:

$$O_1 = - \sum_{\forall (v_i^s, v_j^s) \in E^s} w_{ij}^s \log p_1(v_i^s, v_j^s). \quad (8)$$

The goal of first order-proximity embedding is to find the vector representation $\{\mathbf{v}_i^s\}_{i=1..|V|}$ of each node with the smallest (8) and learn the low-dimensional vectors of each node through the random gradient descent method. The nodes in the target network's G^t layer are also embedded with the first-order proximity according to the same steps.

(b) Second-order proximity embedding

The first-order proximity embedding only utilizes the observed edges in the network for representation, and the larger the weight of the edges between nodes, the tighter the vector representation in the embedding space. However, using only first-order proximity cannot reflect the global network structure because the information is not enough. The second-order proximity embedding considers the shared neighborhood structure between nodes, thus supplementing the network information. The idea is that if two nodes are not directly connected but share a large number of neighboring nodes, then these two nodes have a high second-order proximity and should also be tightly represented in the embedding space. The specific definition of the second-order proximity is as follows.

The second-order proximity of the line embedding algorithm refers to the proximity between two text units that are important for semantic similarity in a text corpus. The basic principle is to express text based on embedded vectors, normalize the root mean square of the text vector, and then calculate the cosine similarity of the two vectors as their similarity index.

Due to the study of undirected networks in this paper, in the processing of the second-order proximity, undirected edges are set as two directed edges with opposite directions and equal weights, and each node is represented as two vectors: its own \mathbf{u}_i and the specific "context" \mathbf{u}'_i of other nodes. If two nodes are distributed similarly in the "context," then they are similar to each other.

In directed edge $e_{ij} = (v_i, v_j)$, the probability distribution of context v_j generated by the node v_i is expressed as follows:

$$p_2(v_j | v_i) = \frac{\exp(-\mathbf{u}'_j \cdot \mathbf{u}_i)}{\sum_{k=1}^{|V|} \exp(-\mathbf{u}'_k \cdot \mathbf{u}_i)}. \quad (9)$$

Due to the assumption of the second-order proximity, the more common neighbors two nodes have, the more similar they are. That is, nodes with similar distributions in the context have tighter vectors in the embedding space, and the empirical distribution is expressed as follows:

$$\hat{p}_2(v_j | v_i) = \frac{w_{ij}}{W_i}, \quad (10)$$

w_{ij} is the weight of the intralayer edge e_{ij} , calculated from the intralayer edge weighting. $W_i = \sum_{k \in \Gamma(v_i)} w_{ik}$. $\Gamma(v_i)$ is the set of all neighbors of the node v_i . The undirected network is represented as two directed edges with opposite directions and equal weights, but when calculating W_i , the weight of the connecting edges between the same node pair is only calculated once. Similarly, we need to minimize the distance between the conditional probability distribution mentioned above and the empirical distribution. Using KL-divergence to represent the approximation of two distributions, the objective function is obtained as follows:

$$O_2 = - \sum_{\forall (v_i, v_j) \in E} w_{ij} \log p_2(v_j | v_i). \quad (11)$$

By learning $\{\mathbf{u}_i\}_{i=1..|V|}$ and $\{\mathbf{u}'_i\}_{i=1..|V|}$ to minimize the objective function O_2 in (11), we can obtain the second-order proximity vector representing \mathbf{u}_i for each vertex.

Afterwards, we concatenate the vectors obtained by performing first-order and second-order adjacency embedding on each node in the two-layer network and reweight the dimensions to balance the two vectors obtained by first-order and second-order embedding. The obtained results are represented as vectors for that node. After this step, the nodes of the two-layer network can be correspondingly embedded into two low-dimensional vector spaces.

3.2.3. Vector Space Matching and Similarity Calculation.

After embedding the two-layer network, the source network and the target network are represented in two different latent spaces. In order to obtain the corresponding relationship between two spaces, the paper constructed a neural network with three hidden layers and trained the mapping function using prior interlayer links $(v_i^s, v_j^t) \in E^{st}$. In order to test the final prediction performance of the algorithm, we divided the observed interlayer links into a training set E^T and a testing set E^P , and the edges in the training set are also known as prior interlayer links.

The constructed neural network layer is a fully connected network, consisting of an input layer, an output layer, and three hidden layers between these two layers. This eliminates the requirement for linear mapping and provides greater flexibility in capturing latent relationships between two vector spaces, making the mapping function more realistic. Among them, the number of neurons in the input and output layers is the dimension of the node in the latent space, and 1200 neurons are placed in each hidden layer. The neural network diagram is shown in Figure 2.

Based on the prior interlayer linkage, this neural network is used to learn the mapping function from the vector space \mathbf{Z}^s of the source network to the vector space \mathbf{Z}^t of the target network. The input of this network is the embedding vector of the prior interlayer link in the training set at the source network node, and the output is the vector representation of the corresponding node of the link in the target network. That is, for each prior interlayer node pair $(v_i^s, v_j^t) \in E^T$, \mathbf{v}_i^s is used as the input and \mathbf{v}_j^t is used as the target output. The

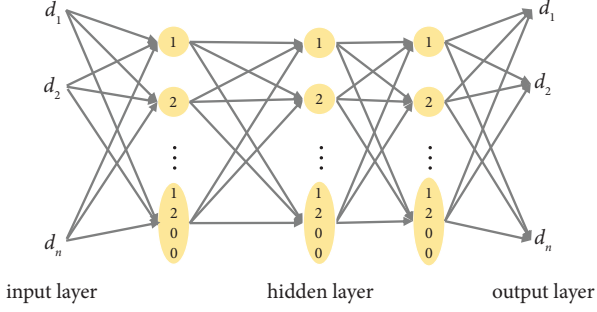


FIGURE 2: Schematic diagram of the neural network.

neural network is trained to obtain the mapping function ϕ , in which the loss function is defined as follows:

$$L_2(\phi, Z^s, Z^t, E^T) = \sum_{\forall (v_i^s, v_j^t) \in E^T} 1 - \cos(\phi(v_i^s), v_j^t). \quad (12)$$

Among them, $\cos(\cdot)$ is cosine similarity, and the larger the value is, the closer the vectors are to each other; $\phi(v_i^s)$ is the representation of nodes in the source network mapped to the target network. By minimizing the loss function, we obtain the function ϕ that maps from the source network vector space to the target network vector space. Afterwards, two low-dimensional vector spaces can be unified into the same space.

For each unmatched pair of nodes (v_a^s, v_b^t) , their representations in their respective vector spaces are $\mathbf{v}_a^s, \mathbf{v}_b^t$. Based on the mapping function ϕ obtained above, the node v_a^s can be represented in the vector space of the target network. The cosine similarity of vectors $\phi(\mathbf{v}_a^s)$ and \mathbf{v}_b^t is used to represent their similarity, with s_1 being the similarity score of the vector, represented as follows:

$$s_1(v_a^s, v_b^t) = \frac{\phi(\mathbf{v}_a^s)^T \cdot \mathbf{v}_b^t}{\|\phi(\mathbf{v}_a^s)\| \cdot \|\mathbf{v}_b^t\|} = \frac{\sum_{i=1}^d \phi(\mathbf{v}_a^s)_i (\mathbf{v}_b^t)_i}{\sqrt{\sum_{i=1}^d (\phi(\mathbf{v}_a^s)_i)^2} \cdot \sqrt{\sum_{i=1}^d (\mathbf{v}_b^t)_i^2}}, \quad (13)$$

where $\|\cdot\|$ represents the modulus of a vector. For each node in the source network, we can provide a list of nodes in the target network and output the vector similarity matrix $\mathbf{H}\mathbf{V}$ in the form of a matrix. Each row of this matrix represents the vector similarity score between an unmatched node in the source network and all unmatched nodes in the target network.

3.2.4. Distance Similarity Calculation. Since the mapping function is obtained through training, it may not perfectly represent the spatial mapping relationship. In order to avoid the one-sidedness of single vector similarity score evaluation, when two-layer nodes are embedded into the corresponding two vector spaces, we use the positional relationship between unmatched nodes and their neighboring nodes that have already been matched as interlayer

links as a supplement to set the distance similarity score. This can more fully utilize the known network structure information, and the distance similarity score is represented as follows:

$$s_2(v_a^s, v_b^t) = \sum_{\substack{\forall (v_i^s, v_j^t) \in E^T \\ v_i^s \in \Gamma(v_a^s) \\ v_j^t \in \Gamma(v_b^t)}} \exp(-d_{ai}^s \cdot |d_{ai}^s - d_{bj}^t| \cdot d_{bj}^t), \quad (14)$$

where E^T is the interlayer link in the training set, that is, a prior interlayer link, $\Gamma(\cdot)$ represents the neighbor set of nodes, and d_{ai}^s represents the Euclidean distance between the unmatched node v_a^s and the node that has already matched v_i^s in the latent space after embedding. The Euclidean distance between the vectors \mathbf{v}_a^s and \mathbf{v}_i^s in the latent space after embedding is represented as follows:

$$d_{ai}^s = \sqrt{\sum_{k=1}^d ((\mathbf{v}_a^s)_k - (\mathbf{v}_i^s)_k)^2}. \quad (15)$$

For all nodes in the network, we output the distance similarity score matrix $\mathbf{H}\mathbf{D}$, $\mathbf{H}\mathbf{D}_{mn} = s_2(v_m^s, v_n^t)$. Each row of this matrix represents the distance similarity score between an unmatched node in the source network and all unmatched nodes in the target network. In the related concept elaboration above, a prior interlayer node pair (v_i^s, v_j^t) is a common matching neighbor of unmatched nodes v_a^s and v_b^t . The proposal of the distance similarity formula takes into account the Euclidean distances d_{ai}^s and d_{bj}^t between unmatched nodes and their common matching neighbors (CMNs). The smaller the Euclidean distance, the greater the impact of the CMNs on these two unmatched nodes and the more likely the nodes to be similar. $|d_{ai}^s - d_{bj}^t|$ represents the difference in the Euclidean distance between unmatched nodes in a two-layer network and their CMNs. The smaller the difference is, the more similar the two are. In addition, by summing, if there are more CMNs between two unmatched nodes, the greater the distance similarity and the more similar the nodes. The higher the distance similarity score s_2 , the greater the likelihood of connections between nodes.

3.2.5. Comprehensive Evaluation and Prediction. In the first two steps, we can obtain the vector similarity score and the distance similarity score between unmatched node pairs in the two-layer network. We effectively combine these two scores through control factors and propose a comprehensive evaluation score, represented as follows:

$$s(v_a^s, v_b^t) = \varepsilon s_1(v_a^s, v_b^t) + (1 - \varepsilon) s_2(v_a^s, v_b^t), \quad (16)$$

where ε is the factor that controls the weights of the two scores, with $0 < \varepsilon \leq 1$. By controlling the factors, distance similarity and vector similarity are combined as the final comprehensive evaluation score for interlayer link prediction, represented by a matrix as follows:

$$\mathbf{S} = \varepsilon\mathbf{H}\mathbf{V} + (1 - \varepsilon)\mathbf{H}\mathbf{D}. \quad (17)$$

For any unmatched node v_a^s in the source network G^s , the comprehensive evaluation score s is used to calculate the matching degree between all nodes in the target network G^t and that node. Then, a sorted list with scores from high to low can be obtained, and the node with the highest score is finally selected from the list. In this way, for each v_a^s , a node with the highest matching score in the target network will be obtained as a possible interlayer link, which means that the node in the target network corresponding to $\max s$ will be taken as the most likely matching node in the source network, and then, accuracy calculation will be carried out. The SWLEV design flowchart is shown in Figure 3.

Finally, the paper conducts a complexity analysis of the model. When embedding a network, the computational complexity is $O(k_1 d |E|)$, k_1 is the number of iterations, d is the embedding dimension, and $|E|$ is the number of edges in the network. In the stage of executing spatial mapping after embedding, due to the use of neural networks for training, the time complexity is $O(k_2 d |E^T|)$, k_2 is the number of iterations when training the mapping function, and $|E^T|$ is the number of interlayer links in the training set. The time complexity for vector similarity calculation is $O(|V|^2)$, and the time complexity for distance similarity calculation is also $O(|V|^2)$, where $|V|$ represents the total number of nodes in the network.

4. Experiments

4.1. Dataset Introduction. In order to verify the superiority of the proposed model in the paper, a large number of comparative experiments were conducted on real datasets with current advanced prediction models. The dataset used in the experiment is Foursquare-Twitter (FT). It uses two real cross-network datasets collected from users' online social networks Foursquare and Twitter, which were collected by literature [25]. Because some users provide their Twitter accounts in their Foursquare profiles, a real-world representation of the users in the two-layer network is obtained, and some nodes in the two-layer network are aligned. We preprocess the obtained dataset to remove self-loops and multiple edges. Due to the requirement of link prediction connectivity, we extract the most connected subset in each layer of the network as the target dataset for experiments. In addition, tools were used to analyze the characteristic parameters of each dataset, such as network average degree, average distance, clustering coefficient, degree correlation coefficient, heterogeneity, and the number of known interlayer links. The statistical characteristics of the dataset are shown in Table 2.

4.2. Comparison Model. We used advanced interlayer link prediction models as a comparison, listed as follows:

- (1) DeepLink [16] proposes a new algorithm based on deep reinforcement learning to study the UIL problem by utilizing the duality of mappings between any two networks. It is an end-to-end network alignment method and semisupervised user identity

link learning algorithm that does not require extensive feature engineering and can easily combine features based on configuration files.

- (2) MulCEV [11] proposes a framework based on the consistency of multiple types of embedded vectors. After the network is represented as a low-dimensional vector space, the distance consistency of node positions is additionally used for prediction supplementation, fully utilizing the effective information of the latent space after embedding.
- (3) IONE [13] proposes a representation learning model to learn alignment network embedding for multiple networks. This model models each user's followers and follower models as input and output contexts, and given and potential anchor links can be used as hard and soft constraints in the unified learning framework in this model, thereby promoting information transmission. In this model, all factors are described by a single objective function, and network embedding and user alignment can be achieved simultaneously when the objective function value is the smallest. In addition, random gradient descent and negative sampling are used to achieve efficient learning of the model. Concerning IONE and its two variant INE [13], INE does not consider the information of the input context of the node.
- (4) IONE-d [14] proposes a representation learning model to learn an aligned network embedding for multiple networks. It explicitly models the followership and followeeship of each user as the input and output context. Both given and potential anchor links can be used in this model as hard and soft constraints in a unified framework for learning.
- (5) CRW [25] proposes a unified link prediction framework to solve the collective link identification problem, which consists of two phases: step (1) collective link prediction of anchor and social links and step (2) propagation of predicted links across the partially aligned "probabilistic networks" with collective random walk.

4.3. Evaluation Indicators. We use indicators to evaluate the accuracy of link prediction algorithms, which are defined as follows:

$$\begin{aligned} \text{Precision@N} &= \frac{|\text{CorrUsers@N}|^{ST}}{|\text{UnAlignedAnchors}|} \\ &= \sum_{i=1}^m 1_i \{\text{success@N}\} / m, \end{aligned} \quad (18)$$

where $|\text{UnAlignedAnchors}|$ is the number of all unmapped users and $|\text{CorrUsers@N}|^{ST}$ is the number of users previously aligned with the source network users found in the top- N list of the target network. m is the number of all unobserved interlayer links, and $1_i \{\text{success@N}\}$ indicates whether the unobserved links in the top- N list are real

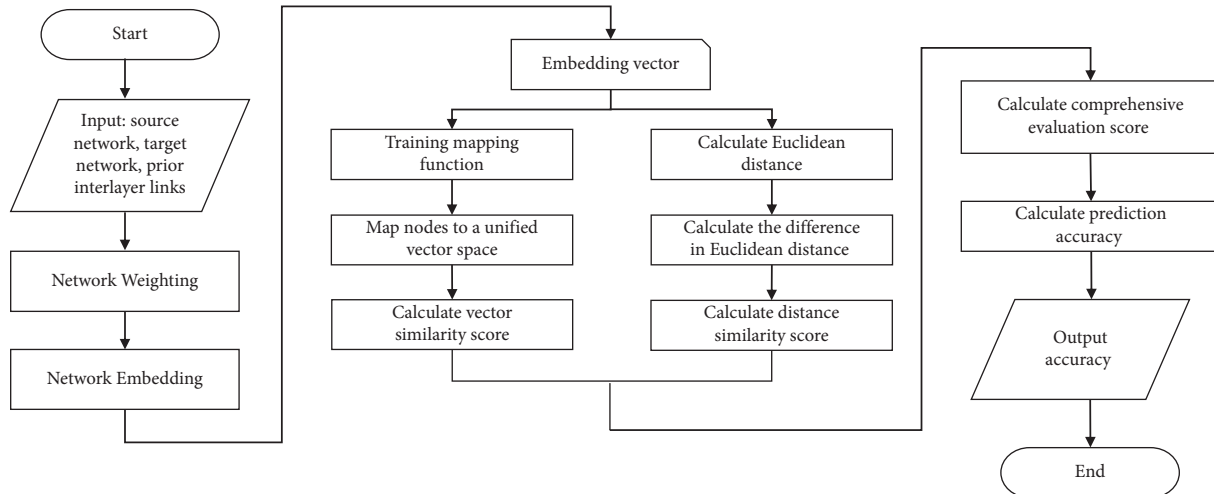


FIGURE 3: The design flowchart of SWLEV.

TABLE 2: Network topological properties of datasets.

Network	$ V $	$ E $	$\langle k \rangle$	$\langle d \rangle$	C	r	H	E^{st}
Foursquare	5311	54232	20.42	3.34	0.23	-0.19	3.45	3148
Twitter	5120	130576	51.01	2.55	0.30	-0.21	4.49	

interlayer links. If so, it is 1; otherwise, it is 0. Obviously, the larger the value of Precision@ N , the better the prediction performance of the algorithm.

4.4. Simulation Results and Performance Analysis. The task of the paper is to discover interlayer links in the test set based on the information in each layer of the training set and the multiplex network. In order to test the accuracy of the model, we randomly divided the known interlayer link sets into a training set E^T and a testing set E^P . The number of edges in the training set to the number of prior interlayer links is called the training ratio. According to [14, 16], we set the training ratio to 0.9. The training set E^T contains 90% of the edges in the set, which are considered as prior interlayer links. The edges in E^T are used to train the model and predict unknown interlayer links in the network. The test set E^P contains the remaining 10% of edges, which are waiting for matching to test the model’s predictive ability. To ensure the universality of the results, we repeated all the experiments 10 times, taking the average accuracy as the experimental result. After preprocessing the dataset, we conduct the experiments in the following aspects and use Precision@ N as an index to measure the predictive ability of the algorithm model.

- (1) Explore the performance of the SWLEV model in the FT dataset under different control factors ε : According to [11], we set the embedding dimension $d = 256$, iteration times = 100,000, and training ratio = 0.9. Since the prediction results are more convincing when N is larger due to that there are less accidental factors, we record the change in prediction accuracy when N is 30 and 100. We record the changes in Precision@30 and Precision@100

when the control factor ε changes from 0.1 to 1.0. The results are shown in Table 3.

From Table 3, it can be seen that the FT network usually achieves the highest accuracy when the control factor ε is 0.9. Therefore, in all subsequent experiments, the control factor ε of the comprehensive evaluation score will be set to 0.9.

- (2) We explore the variation of the prediction accuracy of the proposed model with N and compare it with DeepLink [16], MulCEV [11], IONE [13], INE [13], IONE-d [14], and CRW [25] models. Same as the first part of the experiment, the experimental conditions are set as embedding dimension $d = 256$, with a training ratio of 0.9. According to multiple testing experiments, different models require different iterations to present the optimal results. We set 100,000 iterations for SWLEV, DeepLink, and MulCEV models, 10 million iterations for IONE and INE models, and 100 million iterations for IONE-d models during mapping. The experimental results are shown in Figure 4.

The results show that our proposed model achieves the highest accuracy in all settings. Due to the accidental factors having a small influence when N is large, the prediction results are more convincing when N is large. Therefore, we explain the improved accuracy when N is 30 and 100. When $N = 30$, the accuracy is improved by 41.8% compared to the DeepLink model, 3.5% compared to the MulCEV model, 6.3% compared to the IONE model, 30.0% compared to the INE model, 5.0% compared to the IONE-d model, and 60.5% compared to the CRW model. When $N = 100$, the accuracy is improved by 49.6% compared to the DeepLink model, 3.1% compared to the MulCEV model, 6.5% compared to the IONE model, 25.0% compared to the INE model, 3.7% compared to the IONE-d model, and 20.9% compared to the CRW model. This confirms the

TABLE 3: The variation of prediction accuracy Precision@30 and Precision@100 with control factor ϵ .

ϵ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Precision@30	0.634	0.634	0.634	0.641	0.644	0.654	0.651	0.654	0.657	0.246
Precision@100	0.754	0.761	0.754	0.757	0.757	0.761	0.757	0.757	0.764	0.443

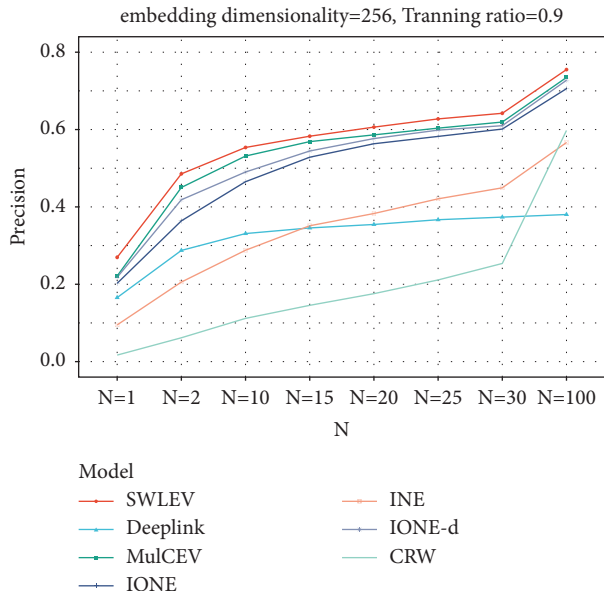


FIGURE 4: Comparison between other models and our proposed model for different @N settings.

effectiveness of the proposed network layer-weighted embedding model, which fully utilizes network topology information and improves the accuracy of interlayer link prediction. From the trend graph of the observation results, it can be seen that as the number increases, the accuracy of the three methods also increases. This is because the larger the value of N , the greater the number of potential matches recommended by different models for each unmatched node, the larger the range of given candidate matches, and the higher the probability of successfully predicting interlayer links.

- (3) We also evaluated the performance of SWLEV and DeepLink, MulCEV, IONE, INE, and IONE-d models under different embedding dimensions. We calculate the prediction accuracy of the six models when the embedding dimensions are 32, 64, 128, and 256, respectively. Same as the first part of the experiment, the experimental conditions are set to $N = 100$, with a training ratio of 0.9. SWLEV, DeepLink, and MulCEV have 100,000 iterations during mapping, 10 million iterations for IONE and INE models, and 100 million iterations for IONE-d models. The experimental results are shown in Figure 5.

The results show that our proposed model has achieved the highest accuracy in almost all dimensions, with an average improvement of 2.4% compared to the better-performing IONE model,

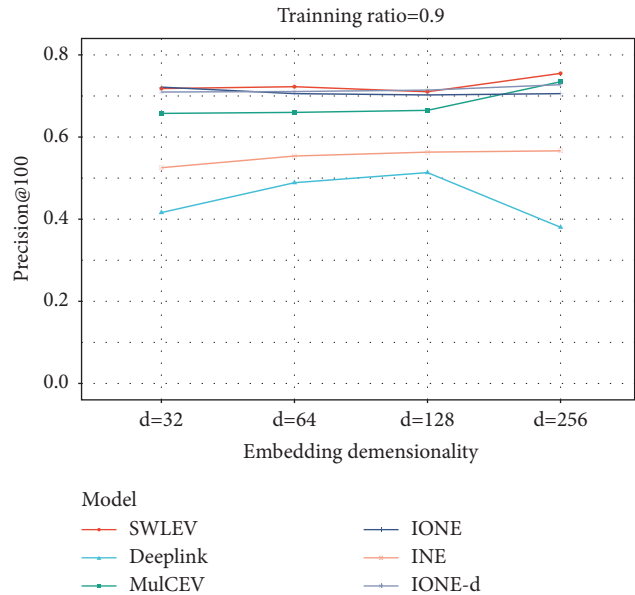


FIGURE 5: Comparison between other models and our proposed model for different dimensionalities.

1.5% compared to the better-performing IONE-d model, and 6.7% compared to the better-performing MulCEV model. Also, it can achieve good predictive performance when the embedding dimension is low. The computational complexity of the algorithm during the embedding process largely depends on the embedding dimension. The smaller the dimension, the lower the complexity, which once again confirms the superiority of the SWLEV model.

- (4) The number of iterations required for training during the mapping process is also an important reference factor, which reflects the time complexity and can verify whether our model can converge. Therefore, the paper explores the performance of model prediction accuracy with the number of iterations. Same as the second part of the experiment, we set $N = 100$, training ratio = 0.9, and embedding dimension = 256 and compare our proposed model with DeepLink, MulCEV, IONE, INE, and IONE-d models. The simulation results are shown in Figure 6.

The results show that our method has significantly better predictive performance than the comparison models, which is consistent with the previous @N experiment. The model proposed in the paper achieved the highest accuracy in all iterations and achieved good prediction performance at low iterations, converging faster than MulCEV and greatly reducing computational complexity. Even when the training

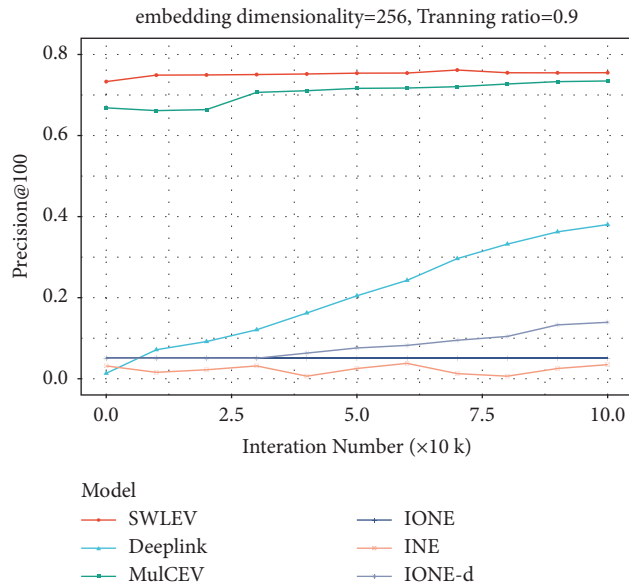


FIGURE 6: Comparison between other models and our proposed model for different interaction numbers.

frequency is 0, there is still a certain degree of predictive ability because a portion of the comprehensive evaluation score is the distance similarity score, which does not require training. After the nodes in the two-layer network are represented in their respective low-dimensional vector spaces, they can be obtained using their position relationships, which add some information to link prediction.

5. Conclusions

In this article, we propose an interlayer link prediction model SWLEV based on the weighted embedding of connected edges within the network layer. This model does not use potentially untrue node attribute information but is entirely based on the network structure for prediction. Before embedding, we model the intralayer links of the single-layer network as a weighted graph. The stronger the node edge relationship, the more similar the vector representation in the low-dimensional space after embedding, thus achieving better network representation. Afterwards, appropriate embedding methods are selected to further capture the structural information of the network. After embedding, vector similarity and distance similarity were used as comprehensive evaluation scores. In addition to introducing the algorithm framework of the paper, this section also discusses simulation design. We have conducted a large number of experiments on real datasets, including the number of iterations, embedding dimensions, and other aspects. The results show that our proposed model has higher prediction accuracy in all aspects than current advanced models and can achieve the highest accuracy when the training frequency is low, reducing computational complexity and verifying the effectiveness and superiority of the model.

Although the performance of the SWLEV model is already excellent, there are still some areas for improvement. First, it only utilizes network structure information for embedding, without fully considering the diverse attributes of

nodes or links. Second, the networks studied by SWLEV are fixed and unchanging, while in reality, social media networks often change constantly. Therefore, our model still lacks the ability to quickly predict unseen nodes or brand new subnets. In the future, we plan to conduct research on more comprehensive embedding methods to simultaneously capture network structures and diverse attribute information. At the same time, we will strive to make efficient predictions while the number of nodes continues to dynamically increase to meet the constantly evolving application needs.

Data Availability

The data that support the findings of this paper are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (Grant no. 61821001).

References

- [1] K. Shu, S. Wang, J. Tang, R. Zafarani, and H. Liu, "User identity linkage across online social networks: a review," *Acm Sigkdd Explorations Newsletter*, vol. 18, no. 2, pp. 5–17, 2017.
- [2] S. Bartunov, A. Korshunov, S. T. Park, W. Ryu, and H. Lee, "Joint link-attribute user identity resolution in online social networks," in *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, Workshop on Social Network Mining and Analysis*. ACM, pp. 12–16, Washington, DC, USA, August 2012.
- [3] A. Narayanan and V. Shmatikov, "Myths and fallacies of 'personally identifiable information,'" *Communications of the ACM*, vol. 53, no. 6, pp. 24–26, 2010.
- [4] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 833–852, 2019.
- [5] J. Zhou and J. Fan, "Translink: user identity linkage across heterogeneous social networks via translating embeddings," in *Proceedings of the 2019-IEEE Conference on Computer Communications*, pp. 2116–2124, Cluj-Napoca, Romania, September 2019.
- [6] X. Zhou, X. Liang, J. Zhao, A. Zhiyuli, and H. Zhang, "An unsupervised user identification algorithm using network embedding and scalable nearest neighbour," *Cluster Computing*, vol. 22, no. 4, pp. 8677–8687, 2019.
- [7] K. Mallick, S. Bandyopadhyay, S. Chakraborty, R. Choudhuri, and S. Bose, "Topo2vec: a novel node embedding generation based on network topology for link prediction," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1306–1317, 2019.
- [8] S. Fu, G. Wang, S. Xia, and L. Liu, "Deep multi-granularity graph embedding for user identity linkage across social networks," *Knowledge-Based Systems*, vol. 193, Article ID 105301, 2020.

- [9] Z. Sun, W. Hu, Q. H. Zhang, and Y. Qu, "Bootstrapping entity alignment with knowledge graph embedding," *IJCAI*, vol. 18, 2018.
- [10] S. Wang, X. Li, Y. Ye et al., "Anchor link prediction across attributed networks via network embedding," *Entropy*, vol. 21, no. 3, p. 254, 2019.
- [11] R. Tang, Z. Miao, S. Jiang, and X. Chen, "Interlayer link prediction in multiplex social networks based on multiple types of consistency between embedding vectors," *IEEE Transactions on Cybernetics*, vol. 25, 2021.
- [12] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 1067–1077, Florence, Italy, May 2015.
- [13] L. Liu, W. K. Cheung, X. Li, and L. Liao, "Aligning users across social networks using network embedding," *Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 16, pp. 1774–1780, 2016.
- [14] L. Liu, X. Li, W. K. Cheung, and L. Liao, "Structural representation learning for user alignment across social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 9, pp. 1–1837, 2019.
- [15] T. Man, H. Shen, S. Liu, X. Jin, and X. Cheng, "Predict anchor links across social networks via an embedding approach," *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, vol. 16, pp. 1823–1829, 2016.
- [16] F. Zhou, L. Liu, K. Zhang, and G. Trajcevski, "Deeplink: a deep learning approach for user identity linkage," in *Proceedings of the 37th IEEE Conference on Computer Communications*, pp. 1313–1321, Burlington, VT, USA, April 2018.
- [17] Y. Sha, Q. Liang, and K. Zheng, "Matching user accounts across social networks based on users message," *Procedia Computer Science*, vol. 80, pp. 2423–2427, 2016.
- [18] R. Zafarani, L. Tang, and H. Liu, "User identification across social media," *ACM Transactions on Knowledge Discovery from Data*, vol. 10, no. 2, pp. 1–30, 2015.
- [19] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff, "Identifying users across social tagging systems," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 5, no. 1, pp. 522–525, 2021.
- [20] D. Zhao, N. Zheng, M. Xu, X. Yang, and J. Xu, "An improved user identification method across social networks via tagging behaviors," in *Proceedings of the 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 616–622, Volos, Greece, June 2018.
- [21] R. Zafarani and H. Liu, "Connecting users across social media sites: a behavioral-modeling approach," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 41–49, Chicago, IL, USA, August 2013.
- [22] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "Cosnet: connecting heterogeneous social networks with local and global consistency," in *Proceedings of the Twenty-First ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD)*, pp. 1485–1494, Long Beach, CA, USA, August 2015.
- [23] S. Tan, Z. Guan, D. Cai, X. Qin, J. Bu, and C. Chen, "Mapping users across networks by manifold alignment on hypergraph," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, pp. 159–165, 2014.
- [24] J. Zhang and P. S. Yu, "Pct: partial co-alignment of social networks," in *Proceedings of the 25th International Conference on World Wide Web*, pp. 749–759, Geneva, Switzerland, April 2016.
- [25] J. Zhang and S. Y. Philip, "Integrated anchor and social link predictions across social networks," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence(IJCAI)*, Palo Alto, CA, USA, July 2015.
- [26] Y. Li, Z. Su, J. Yang, and C. Gao, "Exploiting similarities of user friendship networks across social networks for user identification," *Information Sciences*, vol. 506, pp. 78–98, 2020.
- [27] X. Zhou, X. Liang, X. Du, and J. Zhao, "Structure based user identification across social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1178–1191, 2018.
- [28] X. Zhou, X. Liang, H. Zhang, and Y. Ma, "Cross-platform identification of anonymous identical users in multiple social media networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 411–424, 2016.
- [29] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering," in *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 141–142, Denver, CO, USA, June 2005.
- [30] X. Ding, C. Ma, X. Zhang, H. S. Chen, and H. F. Zhang, "SOIDP: predicting interlayer links in multiplex networks," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 4, pp. 997–1007, 2022.
- [31] E. Nasiri, K. Berahmand, Z. Samei, and Y. Li, "Impact of centrality measures on the common neighbors in link prediction for multiplex networks," *Big Data*, vol. 2, no. 10, 2022.
- [32] E. Nasiri, K. Berahmand, and Y. Li, "A new link prediction in multiplex networks using topologically biased random walks," *Chaos, Solitons & Fractals*, vol. 151, 2021.
- [33] R. Tang, S. Jiang, X. Chen, H. Wang, and W. Wang, "Interlayer link prediction in multiplex social networks: an iterative degree penalty algorithm," *Knowledge-Based Systems*, vol. 194, Article ID 105598, 2020.
- [34] A. Narayanan, E. Shi, and B. I. Rubinstein, "Link prediction by de-anonymization: how we won the kaggle social network challenge," in *Proceedings of the 2011 International Joint Conference on Neural Networks*, pp. 1825–1834, San Jose, CA, USA, June 2011.