mathematics

MDPI

Article

A Novel Link Prediction Method for Social Multiplex Networks Based on Deep Learning

Jiaping Cao, Tianyang Lei, Jichao Li * and Jiang Jiang

College of Systems Engineering, National University of Defense Technology, Changsha 410073, China
* Correspondence: lijichao09@nudt.edu.cn

Abstract: Due to the great advances in information technology, an increasing number of social platforms have appeared. Friend recommendation is an important task in social media, but newly built social platforms have insufficient information to predict entity relationships. In this case, platforms with sufficient information can help newly built platforms. To address this challenge, a model of link prediction in social multiplex networks (LPSMN) is proposed in this work. Specifically, we first extract graph structure features, latent features and explicit features and then concatenate these features as link representations. Then, with the assistance of external information from a mature platform, an attention mechanism is employed to construct a multiplex and enhanced forecasting model. Additionally, we consider the problem of link prediction to be a binary classification problem. This method utilises three different kinds of features to improve link prediction performance. Finally, we use five synthetic networks with various degree distributions and two real-world social multiplex networks (Weibo–Douban and Facebook–Twitter) to build an experimental scenario for further assessment. The numerical results indicate that the proposed LPSMN model improves the prediction accuracy compared with several baseline methods. We also find that with the decline in network heterogeneity, the performance of LPSMN increases.

Keywords: social multiplex networks; link prediction; attention mechanism; heterogeneity

MSC: 91D30; 68T07; 05C62

Citation: Cao, J.; Lei, T.; Li, J.; Jiang, J. A Novel Link Prediction Method for Social Multiplex Networks Based on Deep Learning. *Mathematics* **2023**, *11*, 1705. https://doi.org/10.3390/math11071705

Academic Editors: Shuo Yu and Feng Xia

Received: 18 February 2023
Revised: 28 March 2023
Accepted: 29 March 2023
Published: 2 April 2023

1. Introduction

With the continuous improvement in information technology, social networks have become one of the favourite means for people to conduct social relationships and exchange information. People use social networks to share their opinions, as they provide a fast and easy solution for sharing; they can be regarded as complex networks and powerful tools to represent sophisticated relations among entities [1,2]. Network analysis [3] is a new branch of science and is used to analyse both natural and human-made networks. One of the most explored areas in network analysis is link prediction [4], which has wide applications in predicting forthcoming or missing links in a social network.

Complex social networks are typically represented as monolayer networks, which only contain nodes and links of the same type. The entities and connections in many real systems are sophisticated and might, however, develop in multiple layers; hence, they cannot be regarded as homogeneous networks [5]. For a sparse network, such as a newly built social network, it is difficult to make accurate recommendations for users due to the lack of interactive relationships between entities. For instance, Channels is a short video platform that relies on WeChat. Because Channels is a newly built platform, there is a lack of information to make accurate recommendations. Hence, Channels has a disadvantage in market competition; however, WeChat user relationships can be used in Channels to make recommendations, as shown in Figure 1. There is a novel idea that if we can combine information from Channels and WeChat to help make recommendations in Channels, the

user experience in Channels can be improved. This problem can be abstracted as using a mature network's information to support the newly built network to predict links. In this case, it can be regarded as a multilayer network problem. Multilayer networks, often referred to as multiplex networks, heterogeneous networks, or networks of networks, are an advanced approach for modelling such social networks into multiple layers [6]. Two networks are applied to predict links in the newer network, which makes the information more sufficient and improves the performance. If the effect of multiple layers is disregarded and links in only one of the layers are predicted, it might result in severe information loss.

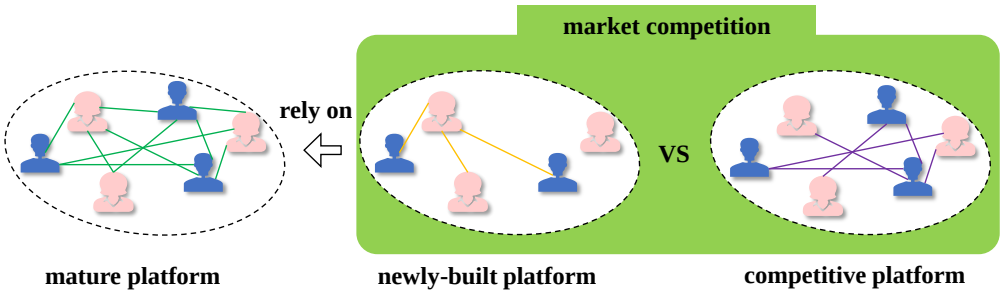


Figure 1. An example of the competition between two platforms.

Link prediction is an interesting research topic, and many scholars have completed many works on this topic. According to the theory within the existing methods, the two major types of link prediction approaches are heuristic and learning-based methods. Heuristic methods, also called similarity-based [7], perform link prediction by extracting the similarities of two nodes based on several similarity metrics [8]. Based on these similarities, ranks are given to each pair of nodes, and higher-ranking node pairs are ultimately classified as projected links [9]. Examples of such indices include common neighbour (CN) [10], Adamic–Adar (AA) [11], and Jaccard indices [12]. The link prediction problem is seen as a binary classification problem by learning-based methods, which first extracts features from a network and then inputs features into a binary classifier [13]. Examples of learning-based methods include DeepWalk [14], LINE [15], and Node2Vec [16]. However, the aforementioned link prediction approaches are all applied on a single layer and cannot fuse information from multiple layers.

Except for only predicting links within a single layer, traditional learning-based methods cannot learn many essential features. For example, a graph neural network (GNN) is not only a typical learning-based approach but also a popular technique for extracting information from graphs [17]. However, a GNN can only consider n-hop node features rather than global structure features; hence, its prediction ability is always limited [18]. Therefore, with the in-depth study of the aforementioned problem, Zhang et al. [7] proposed a novel link prediction framework called subgraph embedding and attributes for link prediction (SEAL), which permits learning from not only structural features but also semantic features. Zhang et al. [19] proposed a deep graph convolutional neural network (DGCNN), which refers to the Weisfeiler–Lehman subtree kernel, to sort subgraph vertices through the SortPooling layer, and this network can obtain global graph topology. Ai et al. [20] proposed a structure-enhanced graph (SEG) neural network that extracts structural features through path labelling from subgraphs. The advantage of using an enclosed subgraph to learn general graph structure features is that it can fully capture the graph's topological features. However, most of the available methods can achieve better results in monolayer networks. In addition, most of the aforementioned methods have poor performance in sparse networks. In reality, an increasing number of social media platforms can be built by relying on existing social media platforms to solve the link prediction problem in a sparse network, which can be regarded as a multiplex network. In this study, we only focus on using current network information to predict future links.

To solve these problems, we propose a novel framework for link prediction in social multiplex networks that can systematically consider both topological and semantic features based on an attention mechanism, namely, **L**ink **P**rediction in **S**ocial **M**ultiplex **N**etworks. Extensive experiments show that LPSMN outperforms baseline link prediction methods. The following is a summary of the contributions in this work:

- We formulate the task of designing a monolayer link prediction framework with the help of multiplex network information. It is beneficial for transferring various types of node information across multiple networks.
- Our method can absorb three types of information, including structural features, semantic features and node attributes, and we use an attention mechanism to fuse information from different layers.
- We construct two kinds of datasets to test our model performance on the task of supporting domain adaptation and conduct experiments on LPSMN and other baselines with these datasets. Moreover, the performance among synthetic different heterogeneity is explored. The result indicates that our model has a leading performance in this task.

The rest of the paper is structured as follows: We introduce the notation definitions in Section 2. In Section 2.1, we demonstrate the principle of our method, including the overall framework and algorithm details. Section 3 discusses the performance and experimental results on seven datasets. Section 4 summarises this paper and suggests directions for future work.

2. Preliminaries

In this section, we introduce the notation definitions and problem description of this paper.

2.1. Definition

Ignoring global structure features or modelling relationships from different platforms into a monolayer network will result in missing information. Several strategies have been given in the literature to handle the above problem [21–25]. However, they have not taken into account global structure features. In this paper, we creatively come up with a framework to methodically consider the global structure features for the multiplex network link prediction task.

Definition 1. *Multiplex network G.We denote the multiplex network as* $G(L_1, L_2, \ldots, L_N)$*, where* $L_i = L(V, E_i)$ *represents one of multiplex network layers, in which V is a set of nodes (the same across the layers),* $E_i(i = 1, 2, \ldots, N)$ *denotes the set of links of the i-th layer and N is the number of layers.*

Definition 2. *Link prediction. The goal is to evaluate the likelihood of pairwise nodes* (v_m, v_n) *to have a link* e_{mn}*. The issue might be expressed as a classification challenge on potential links* E_p *based on observed edges* E_o *and observed node features* X_o*.*

Definition 3. *Node feature matrices* **S**, **L**, *and* **ET**. *We denote* $\mathbf{S} \in \mathbb{Z}^{N \times C}$, $\mathbf{L} \in \mathbb{R}^{N \times C}$, *and* $\mathbf{ET} \in \mathbb{R}^{N \times C}$ *as the graph structure feature matrix, latent feature matrix, and explicit feature matrix, respectively. The graph structure feature reveals topological information beneath nodes, the latent feature is obtained from matrix factorisation methods, and the explicit feature is the node attributes in the original data set. A node pair embedding is a concatenation of the three types of features of two nodes and is denoted as* $\mathbf{X}^{N \times 6 \times C}$*, where C is the dimension of feature matrices, which is set according to author requirements. In this context, we assume three feature matrices have the same dimensions.*

2.2. Problem Statement

In our study, we initially perform the link prediction task in a sparse network with the support of an external layer with sufficient information. We regard this problem as a

binary classification problem and need a validation layer to examine the prediction results. We model two platforms into a 3-layer multiplex network $G(L_1L_2L_3)$, and each layer has the same users, as shown in Figure 2. The bottom layer $L_1 = L(V, E_1)$ is an external layer that has abundant user relationship information. The middle layer $L_2 = L(V, E_2)$ and the upper layer $L_3 = L(V, E_2{}')$ are representations of the same newly built social network. The middle layer is the validation layer to examine whether the external information can promote the link prediction effectiveness of LPSMN. The upper layer is a prediction layer, and we extract newly built network node features from this layer. L_2 represents a newly built network at t_1, and L_3 represents a newly built network at t_2, while $t_1 < t_2$. The only difference between them is that the validation layer has more user relationships than the prediction layer, which is denoted as $|E_2| > |E_2{}'|$, where $|E_2|$ and $|E_2{}'|$ denote the number of edges in the validation layer and the prediction layer, respectively. If two platforms have the same users, we assume that these two platforms should have similar relationships.

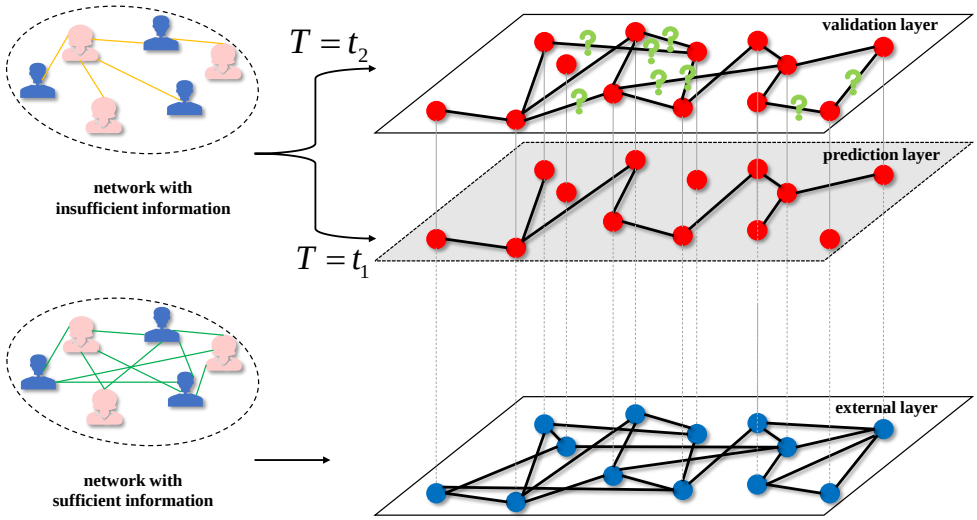


Figure 2. Multiplex network structure of two social platforms. The multiplex network has three layers, including an external layer, prediction layer and validation layer. The external layer is modelled from the platform with sufficient information, and the validation layer is modelled from the platform with insufficient information. The prediction layer is generated by randomly removing several edges from the validation layer. In our model, we use the external layer and the prediction layer to predict links in the validation layer.

In this paper, the problem is combining node features of external layer ${X_1}^{N\times 3\times C}$ and prediction layer ${X_2}^{N\times 3\times C}$ to perform link prediction tasks in the validation layer L_3. For a monolayer link prediction problem, we use prediction layer information X_2 to perform link prediction tasks in the validation layer rather than only using prediction layer information to predict links. We generally consider both structural and semantic features within the external and prediction layers, which can be expressed as:

$$[X_1,\ X_2;\ G(L_1, L_2,\ L_3)] \xrightarrow{\zeta(\bullet)} [E_3] \tag{1}$$

where $[X_1,\ X_2;\ G(L_1, L_2,\ L_3)]$ is the input, $[L_1, \ldots, L_n]$ is the output, $\zeta(\bullet)$ is the model to be learned, and L_n denotes the result of sample n.

3. Methodology

Our link prediction method can be divided into 5 steps: (1) samples are extracted from the validation layer; (2) graph structure features and latent features in both the external and

prediction layers are extracted; (3) three types of node features are concatenated to generate sample embeddings in both the external and prediction layers; (4) sample embeddings from the external and prediction layers based on the attention mechanism are aggregated; and (5) a dense layer is used to predict links. The overall framework of LPSMN is shown in Figure 3.

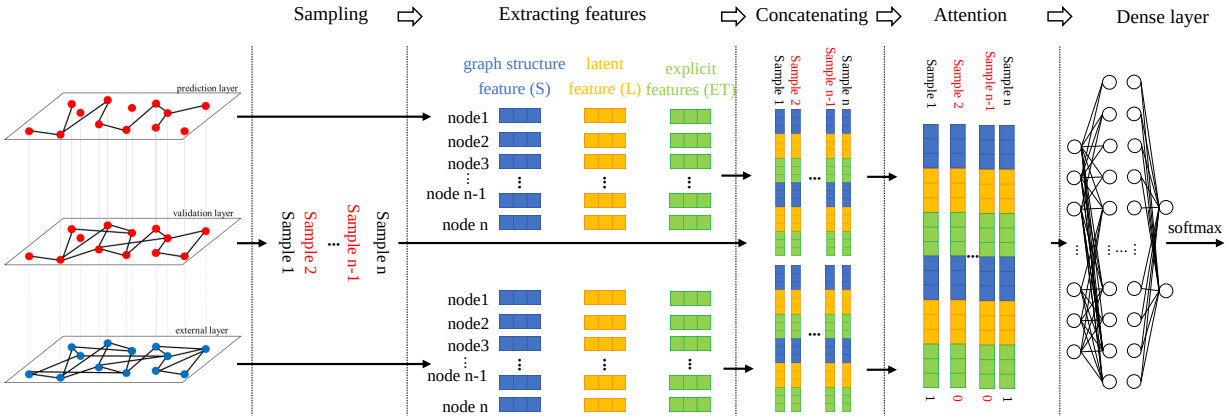


Figure 3. Overall framework of LPSMN. The overall framework includes 5 steps. The first step is sampling on the validation layer, and the next step is extracting three types of features from the external layer and the prediction layer, respectively. Then, sample representations are generated by concatenating the features of each node. The fourth step is using an attention mechanism to fuse the sample representation from the external and prediction layers. Finally, we use a dense layer to predict the label of the sample.

The first step in LPSMN is to obtain connected samples and unconnected samples to build the training data based on the validation layer. The approach of producing unconnected samples is to randomly select half of a node's unconnected neighbours and use these node pairs as negative samples. The labels of a connected sample and an unconnected sample are denoted as 1 and 0, respectively. The second step is essential, in which we extract targeted features of external and prediction layers, and we discuss this key step in Sections 3.1 and 3.2. The third and fourth steps are encoding samples. We first concatenate multiple features of each node pair, and then we use an attention mechanism to fuse information from two networks. Finally, embedding links are inserted into the dense layer with softmax to predict the label of the sample. The links in LPSMN are encoded by three components: graph structure features, node embeddings, and node attributes.

3.1. Graph Structure Features

A GNN typically adheres to a message-passing schema [26]; however, only node features are conveyed throughout the message transmission process; node topology is not explicitly considered. In this paper, we extract graph structure features based on the WL algorithm [27]. The WL algorithm has a triplet (V, E, l) to represent a graph, where l is a set of node labels. A node label $l_i(v)$ can be updated in each iteration i. The WL graphs' sequence is as follows

$$\{G_0, \ldots, G_h\} = \{(V, E, l_0), \ldots, (V, E, l_h)\} \tag{2}$$

where $G_0 = G$, $l_0 = l$, and h denote the number of WL iterations. Neither V or E ever change in this sequence. The WL kernel k with h iterations is defined as

$$k_{WL}^{(h)}(G) = \alpha_0 k(G_0) + \alpha_1 k(G_1) + \cdots + \alpha_h k(G_h) \tag{3}$$

where α_i denotes non-negative real weights.

The WL algorithm updates node integers iteratively until a fixed point is achieved. We use a simple example to illustrate this process, as shown in Figure 4. In each iteration, a node aggregates the labels of its neighbours to generate an aggregated label. If the aggregated labels of two nodes are similar, they will have the same label in the next iteration. In this process, integer labels only represent symbols to distinguish nodes' different structural roles, and the absolute value does not have a meaning. We use a node's labels in all iterations as the node graph structure feature. The degree of each node is used as its initial label. In the example, in the WL label sequences of node A and node B, the first WL label sequence match, which means that the neighbourhood of 2 hops around these nodes is isomorphic: in each iteration, 1-hop neighbourhoods are aggregated, so labels at depth n are affected by every node within $2n$ hops. The iteration time in the WL algorithm is set according to the requirement: the more iteration time, the higher the performance of the algorithm. In our experiment, since we set the dimension of the graph structure feature as 64, the WL algorithm needs to iterate 64 times, and we can obtain 64 labels for each node.

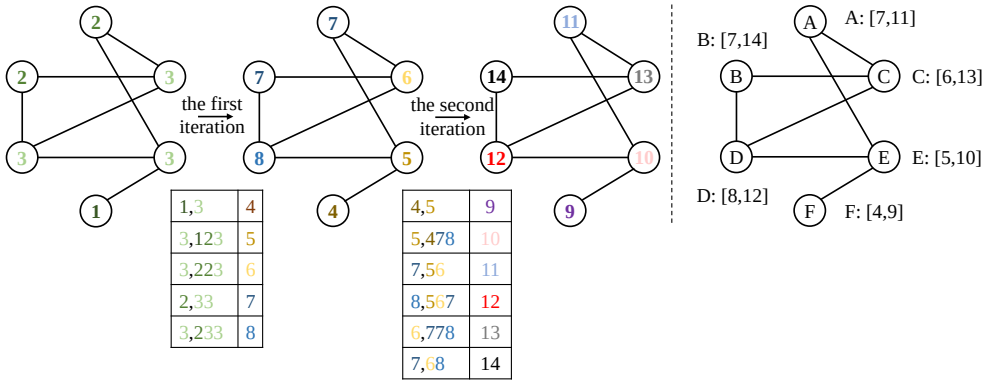


Figure 4. Illustration of generating global structure features based on the WL algorithm. The first iteration is used as an example for illustration. Every node in the first iteration aggregates the label of its neighbour nodes. If two nodes' aggregated labels are the same, in the next iteration, these two nodes will have the same label. The absolute value of the label is just a symbol, and the number has no meaning.

3.2. Latent Features and Explicit Features

LPSMN not only considers graph structure features but also considers latent features and explicit features. Graph embedding uses matrix factorisation to discover the low-dimensional latent representation of network nodes. The low-dimensional latent representation, named the latent features, has wide applications in graph-based tasks [28,29]. Latent features contain the global properties of graphs and represent a graph as a set of vectors $\{u_1, u_2, \ldots, u_n\}$. Each vector $u_i \in \mathbb{R}^d$ is a form of i-th node in the d-dimensional space, as shown in Figure 5.

Nowadays, there are several state-of-the-art embedding approaches. DeepWalk combines random walk and Word2Vec to learn the node representation on large graphs. However, it is unable to learn on weight graphs and focuses mainly on second-order proximity while neglecting first-hop proximity [14]. To address the aforementioned shortcomings, LINE can achieve node embedding on direct and weighted graphs. In addition, it simultaneously considers first-hop closeness and second-hop closeness [15]. Varying from DeepWalk, which is based on the depth-first search (DFS) method, Node2Vec employs a special random walk, which can both learn structural relationships based on a breadth-first search (BFS) and homophily based on DFS [16]. However, it cannot learn enough structural similarity information because of the bias in which random walk has a limited number of steps. Struc2Vec builds a multilayer graph structure to calculate structure-based similarity [30].

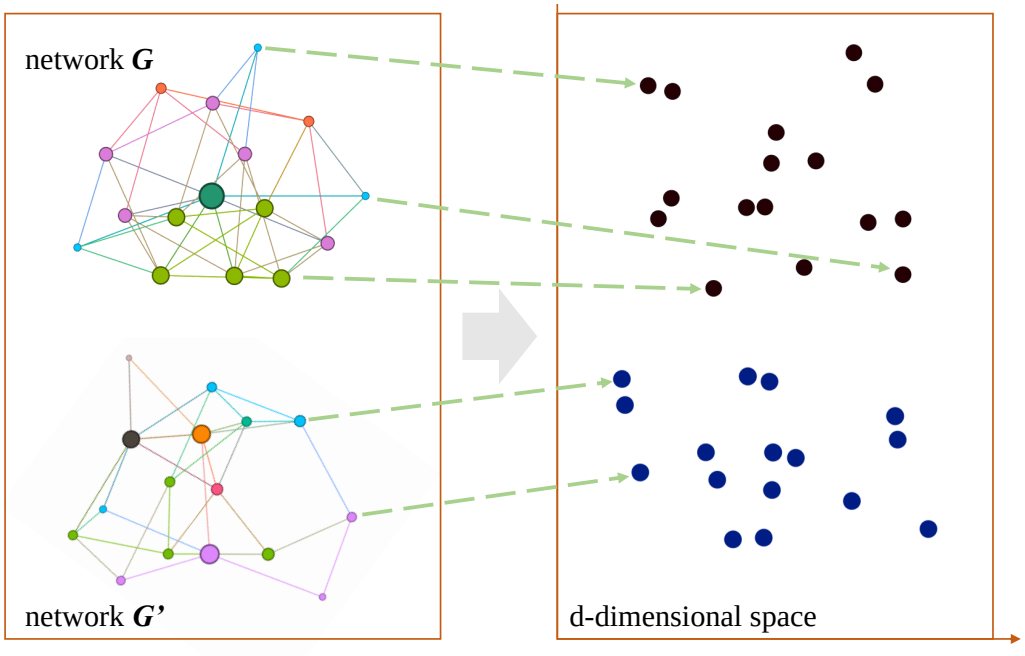


Figure 5. Process of graph embedding. Graph embedding methods can transfer the nodes of two networks to a low-dimensional continuous space while maintaining the essential structure and properties of the networks.

LPSMN is flexible with which node embedding technique is used. In our model, we choose Node2Vec since it considers node structural equivalence at the same time. The purpose of Node2Vec is to extract the feature representation of nodes by training a mapping function f, which maximises a node's log-probability of noticing its network neighbourhood, as in Equation (4):

$$\max_{f} \sum_{u \in V} \log Pr(N_S(u)|f(u)) \tag{4}$$

$$Pr(N_S(u)|f(u)) = \prod_{n_i \in N_S(u)} Pr(n_i|f(u)) \tag{5}$$

$$Pr(n_i|f(u)) = \frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))} \tag{6}$$

where u denotes a node of the network and $N_S(u) \in V$ denotes a neighbourhood set of node u. Equation (5) assumes the independence of observing any neighbourhood node. As shown in Equation (6), there is a symmetric effect on the target node and its neighbours in the feature space. Since networks are non-Euclidean, according to different sampling strategies, we can obtain different neighbour nodes.

Node2Vec uses a flexible neighbourhood sampling strategy, which finds neighbourhoods with BFS and DFS. The crucial idea of Node2Vec is the biased random walk of fixed length l, as shown in Equation (24):

$$P(c_i = v_j | c_{i-1} = v_i) = \begin{cases} \frac{\pi_{v_i v_j}}{Z} & \text{if} (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where c_i denotes the i-th node, $\pi_{v_i v_j}$ denotes the transition probability of the node pair (v_i, v_j). The constant Z has the function of normalisation. To consider both BFS and DFS, a

biased random walk is defined and is used to direct the walk. $\pi_{v_i v_j} = \alpha_{pq}(t, v_j) \cdot \omega_{v_i v_j}$ is the transition probability, where α is denoted as

$$\alpha_{pq}(t, v_j) = \begin{cases} \frac{1}{p} & , d_{tv_j} = 0 \\ 1 & , d_{tv_j} = 1 \\ \frac{1}{q} & , d_{tv_j} = 2 \end{cases} \tag{8}$$

and d_{tv_j} denotes the shortest path distance between nodes t and v_j.

Node attributes, which describe various types of auxiliary information regarding specific nodes, are frequently available as explicit features. For example, many platforms have personal user information, such as gender, age, and occupation. This additional information can work as node attributes. References [31,32] have shown that fusing graph structure features with latent features and explicit features can make the information more complete, which can improve the performance of LPSMN.

3.3. Link Prediction in Social Multiplex Networks

We use graph structure features, latent features and explicit features to encode sample links (v_i, v_j):

$$x_o = (S_i \oplus L_i \oplus ET_i) \oplus (S_j \oplus L_j \oplus ET_j) \tag{9}$$

where $x_o \in \mathbf{X}_o$ denotes each link embedding, $\mathbf{X}_o{}^{N \times 6 \times C}$ denotes a set of link embeddings, S_i and S_j denote the graph structure feature of $node_i$ and $node_j$, L_i and L_J denote the latent feature of $node_i$ and $node_j$, ET_i and ET_j denote the explicit feature of $node_i$ and $node_j$, $\oplus$ denotes the concatenation of two vectors. We apply an attention mechanism $Attention(\cdot)$ to aggregate two networks' information:

$$\begin{aligned} \mathbf{X}_a &= Attention(\mathbf{X}_o{}^e, \mathbf{X}_o{}^p) \\ &= f(\mathbf{X}_o{}^e) \times \mathbf{X}_o{}^e + f(\mathbf{X}_o{}^p) \times \mathbf{X}_o{}^p \end{aligned} \tag{10}$$

where f is an activation function. In this experiment, we use the "*ReLU* " function, $\mathbf{X}_a \in \mathbb{R}^{N \times c}$ is the output edge embedding matrix, $\mathbf{X}_o{}^e$ denotes link embeddings within the external layer, and $\mathbf{X}_o{}^p$ denotes link embeddings within the prediction layer. After fusing link information from two layers, we use an MLP layer to predict the link label with the softmax function. The LPSMN model defines two hidden layers, and the "*ReLU6*" function is applied as the activation function.

$$x_n = MLP(\omega \times \mathbf{X}_a + b) \tag{11}$$

where ω and b are the parameters to be trained. To prevent overfitting, 20% of the neurons are randomly dropped. X_a is trained using a binary cross-entropy loss function

$$loss = \frac{1}{N} \sum_{n=1}^{N} -y_n \cdot \log x_n - (1 - y_n) \cdot \log(1 - x_n) \tag{12}$$

where x_n refers to the probability score that link n is predicted to be true, Y_n is the label of link n, and N is the number of training edges. Algorithm 1 details how to generate the final prediction results.

Algorithm 1 Link Prediction for Social Multiplex Networks

Require: External layer graph $G1$, Validation layer graph $G2$, Prediction layer graph $G3$, Embedding dimension C
Ensure: Link labels **Label**
/* sampling on $G2$ */ ;
$links \leftarrow Sample(G2)$;
/* extracting graph structure feature */;
$\mathbf{S}_1, \mathbf{S}_2 \leftarrow$ *Weisfeiler-Lehman*$(G1)$, *Weisfeiler-Lehman*$(G3)$;
/* extracting latent feature */;
$\mathbf{L}_1, \mathbf{L}_2 \leftarrow$ *node2vec*$(G1)$, *node2vec*$(G3)$;
prepare explicit feature $\mathbf{ET}_1$, $\mathbf{ET}_2$;
/* encoding links */;
$link_{embedding} \leftarrow$ Equation (10);
/* fusing link embedding from two layers */;
for $link \in links$ **do**
    $link_{training} \leftarrow$ Equation (11);
end for
$label_{pre} \leftarrow MLP\left(link_{training} \mid link \in links\right)$;

4. Experiments

LPSMN can extract not only graph structure features but also latent and explicit node features. Moreover, LPSMN predicts links based on information from multiple layers.

4.1. Datasets

We carry out comparative experiments on seven datasets, 5 of which are synthetic network datasets and 2 of which are real-world network datasets (the Weibo–Douban (WD) social network dataset and Facebook–Twitter (FT) social network dataset) [33]. To analyse the sensitivity of heterogeneity on LPSMN performance, the 5 synthetic multiplex networks are generated by the Price model with various power–law distributions [34]. In our experiments, heterogeneity represents the imbalance of the node degree distribution. Then, we show the processes of the synthetic network generation [35].

4.1.1. Synthetic Networks

The synthetic networks are generated as follows. First, a star graph is generated that has m_0 nodes. Then, there are two ways to add a new node with $m(m \leq m_0)$ edges connected to existing nodes. One involves picking an existing node with probability $1 - p_h$ at random, while the other involves selecting an existing node with probability p_h using the preferred attachment method.

The steps to producing synthetic multiplex networks are shown as follows. First, several external layer edges are randomly removed to produce the validation layer, and then several validation layer edges are randomly removed to generate the prediction layer. Then, some edges are removed randomly, and unconnected node pairs are randomly selected in each layer. The number of selected node pairs is as high as the number of removed edges. The ratio of removed edges P is set to 0.2.

4.1.2. Real-World Networks

The **Weibo–Douban** (WD) dataset is a two-layer network dataset from Weibo and Douban, which are a microblogging service platform and an interest-based social platform, respectively. The **Facebook–Twitter** (FT) dataset is a two-layer network dataset from Facebook and Twitter, both of which are social media platforms. In the WD multiplex network, the Weibo network acts as the external layer, and the Douban network acts as the validation layer. These two networks have the same users. The prediction layer is generated by randomly removing 20 proportions of edges in the validation layer. We carry out the same operation for the FT multiplex network. The details of synthetic and real-world social

multiplex networks are shown in Table 1. Since the datasets lack node attributes, explicit features are not included.

Table 1. The details of the datasets.

Multiplex Network	Layer	#Nodes	#Edges
SF (γ = 2.1)	the first layer	2000	38,080
	the second layer	2000	28,080
	the third layer	2000	23,080
SF (γ = 2.5)	the first layer	2000	38,080
	the second layer	2000	28,080
	the third layer	2000	23,080
SF (γ = 3)	the first layer	2000	38,080
	the second layer	2000	28,080
	the third layer	2000	23,080
SF (γ = 5)	the first layer	2000	38,080
	the second layer	2000	28,080
	the third layer	2000	23,080
SF (γ = 10)	the first layer	2000	38,080
	the second layer	2000	28,080
	the third layer	2000	23,080
WD	the first layer	5000	200,192
	the second layer	5000	17,013
	the third layer	5000	14,613
FT	the first layer	5000	52,139
	the second layer	5000	897
	the third layer	5000	165

4.2. Baseline

We first compare LPSMN with heuristics methods. The baseline includes nine popular heuristics: the CN, Jaccard, hub promoted index (HPI), hub depressed index (HDI), resource allocation (RA), AA, preferential attachment (PA), local path (LP), and Katz heuristics. CN, Jaccard, HPI, HDI, RA, and AA are based on neighbours to calculate the proximity, and PA is based on the preferential attachment similarity; and LP and Katz are based on the path to calculate the similarity. The baselines are only applied on monolayer networks. The heuristics approaches' specifics are as follows [36]:

CN: It describes that if two disconnected nodes have more common neighbours, they are more likely to be linked in the feature. In a network, it has the effect of triadic closure and functions as a typical mechanism in daily life [10], as shown in Equation (13).

$$s_{xy} = |\Gamma(x) \cap \Gamma(y)| \tag{13}$$

Jaccard: It holds the idea that comparing whether two sets are similar is comparing the proportion of the elements they share. Higher sample similarity is indicated by a bigger Jaccard index [12], as shown in Equation (14).

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\Gamma(x) \cup \Gamma(y)} \tag{14}$$

HPI: It describes how much vertex x overlaps with vertex y. Since HPI is only determined by nodes with smaller degrees, it can be seen from this definition that a hub is more likely to be similar to other nodes [37], as in Equation (15).

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\min\{k_x + k_y\}} \tag{15}$$

HDI: Both of the target nodes' degrees and the number of common neighbours determine the similarity score. Since the HDI is only determined by nodes with larger degrees, the hub is penalised [38], as shown in Equation (16).

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\max\{k_x + k_y\}} \tag{16}$$

PA: It first removes one edge from the graph and then adds a new edge that is attached to the nodes already present in the graph in accordance with their degree. A new node is more likely to connect to a node with larger degree [39,40], as shown in Equation (17).

$$s_{xy} = k_x k_y \tag{17}$$

AA: It is used to predict links based on how many common links two nodes have. The smaller the degree of proximity, the more important the common neighbour is [11], as shown in Equation (18).

$$s_{xy} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log k_z} \tag{18}$$

RAI: The largest difference between RAI and AA is the different ways of assigning weights to common neighbour nodes. Instead of using its logarithmic value, RAI directly makes use of the degree magnitude [38,41], as shown in Equation (19).

$$s_{xy} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k_z} \tag{19}$$

LP: To assess if a connection exists, the score is calculated by the number of paths between a certain pair of nodes that are two-hop and three-hop neighbours. Here, $\left(A^2\right)_{xy}$ and $\left(A^3\right)_{xy}$ denote the number of paths of length two and three, respectively [38,42], as shown in Equation (20).

$$LP = \left(A^2\right)_{xy} + \left(A^3\right)_{xy} \tag{20}$$

Katz: It considers all paths in a graph, and the weight of each path is controlled by α. Here, α is less than the inverse of the largest eigenvalue of the adjacency matrix [43], as shown in Equation (21).

$$\mathbf{S} = (\mathbf{I} - \alpha \cdot \mathbf{A})^{-1} - \mathbf{I} \tag{21}$$

Next, we compare LPSMN with four baseline latent feature methods: DeepWalk (DW), LINE, Node2Vec (N2V) and Struc2Vec (S2V). The details of the embedding methods are as follows:

DeepWalk: This approach learns d-dimensional feature representations by simulating uniform random walks. DeepWalk's sampling strategy can be thought of as a special case of Node2Vec with $p = 1$ and $q = 1$ [14].

LINE: This approach learns d-dimensional feature representations by taking into account both 1-hop and 2-hop similarities. By simulating over the close neighbours of nodes, it first learns the 1-hop similarity. The 2-hop similarity is then learned by sampling only nodes that are strictly within a 2-hop distance of the source nodes [15].

Struc2vec: This approach learns d-dimensional feature representations by evaluating the rings' ordered degree sequences at k distances from two nodes. It does not need node location information and label information and only relies on the concept of node degree to construct the multilayer graph [16].

4.3. Metrics

In this subsection, we describe four evaluation metrics used in this research, including the precision, recall, F1 score and area under the curve (AUC) metrics.

Precision: the ability of a classification model to identify only the true positives. *Precision* can be calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

Recall: the capacity of a model to recognise all pertinent samples in a dataset. *Recall* can be calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

F1 score: the $F1$ score is the harmonic mean of the precision and recall values. The absolute values fall between 0 and 1.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{24}$$

where the numbers of true positives, true negatives, false positives, and false negatives are TP, TN, FP, and FN, respectively.

AUC: AUC is prominent in the literature for evaluating link prediction methods [36]. When addressing the imbalance between classes, this metric is extremely helpful.

In this experiment, the precision, recall, F1 score, and AUC measures are used to represent the prediction performance. When analysing the test results, larger values of these four indicators correspond to better performance.

4.4. Experiment Analysis

In this subsection, we assess the LPSMN performance. In our research, we initially compare LPSMN with heuristic methods and latent feature methods. Then, we research the heterogeneity's influence on the performance of our model. Finally, we analyse the sensitivity of the removal ratio and learning rate. In this paper, the node's neighbourhood parameter settings of Node2Vec are as follows: dimension $d = 64$, number of walks $r = 200$, walk length $l = 30$, and neighbourhood size $k = 10$.

4.4.1. Comparison to Heuristic Methods

Table 2 displays the comparison outcomes. We first make a comparison between LPSMN and methods using graph structure features only, including nine popular heuristics: CN, Jaccard, HPI, HDI, PA, AA, RA, LP, and Katz. In LPSMN, the dimensions of the graph structure features, latent features and explicit features are set as 64, so the length of the sample is equal to 256. As the heuristic baselines' properties, we restrict LPSMN$_{str}$ to not include any latent or explicit features.

Table 2. Comparison with heuristic methods (AUC).

Multiplex Network	CN	Jaccard	HPI	HDI	PA	AA	RA	LP	Katz	LPSMN$_{str}$
SF (γ = 2.1)	0.5376	0.5298	0.5328	0.5293	0.6089	0.5370	0.5356	0.6036	0.5859	**0.6203**
SF (γ = 2.5)	0.5241	0.5181	0.5197	0.5191	0.5936	0.5229	0.5228	0.5898	0.5683	**0.6239**
SF (γ = 3)	0.5142	0.5114	0.5126	0.5117	0.5804	0.5144	0.5142	0.5838	0.5569	**0.6327**
SF (γ = 5)	0.5151	0.5126	0.5129	0.5131	0.5699	0.5143	0.5144	0.5684	0.5524	**0.6513**
SF (γ = 10)	0.5111	0.5089	0.5084	0.5093	0.5606	0.5102	0.5095	0.5642	0.5439	**0.5997**
WD	0.7490	0.7291	0.5765	0.7291	**0.8692**	0.7497	0.7500	0.8457	0.8567	0.8232
FT (on validation layer)	0.5513	0.5455	0.4727	0.5448	0.2365	0.5509	0.5508	0.4676	0.4114	**0.7454**
FT (on external layer)	0.6121	0.6059	0.5139	0.6060	0.7385	0.6128	0.6139	0.7427	0.7447	

The bold represents the best AUC score within a network.

As shown in Table 2, we observe that LPSMN generally performs better than the predefined heuristics. Heuristic methods have poor performance on synthetic networks,

most of which obtain an AUC of approximately or slightly higher than 0.5, and LPSMN$_{str}$ has a higher AUC than any other method on those synthetic networks. When comparing these methods on real-world data, the AUC scores of the heuristic methods are also lower than that of LPSMN$_{str}$. Since the experimental results on the validation layer of FT and the AUC of HPI, PA, LP, and Katz are all inadequate, we perform heuristic method experiments on the external layer of FT simultaneously. We find that all AUC scores are better when the model is applied to the external layer, and all AUC scores are higher than those on the validation layer. The quality of the data in the validation layer of FT is poor, that is, the validation layer of FT has 5000 users but only 897 edges, which means that this network is very sparse, and most of the users have no connected neighbours within this network. When facing the cold-start problem, the traditional link prediction methods have very poor performance, but our model has a better performance.

4.4.2. Comparison to Latent Feature Methods

Table 3 displays the experimental results. We compare LPSMN with four latent feature methods: DeepWalk, LINE, Node2Vec, and Struc2Vec. They all employ GNNs to learn basic node embeddings. In this experiment, LPSMN includes the 64-dimensional embedding learned from Node2Vec in the node information matrix $\mathbf{X}$. We apply a training and testing set ratio of 60%.

Table 3. Comparison with latent feature methods (AUC).

Multiplex Network	DW	LINE	N2V	S2V	LPSMN
SF (γ = 2.1)	0.5685	0.7732	0.8977	0.7240	**0.9628**
SF (γ = 2.5)	0.5615	0.7673	0.9072	0.7203	**0.971**
SF (γ = 3)	0.5757	0.7630	0.9090	0.7176	**0.971**
SF (γ = 5)	0.5709	0.7665	0.9123	0.7211	**0.9718**
SF (γ = 10)	0.5628	0.7620	0.9076	0.7119	**0.9816**
WD	0.6512	0.8091	0.9765	0.8120	**0.998**
FT	0.5654	0.6617	0.9981	0.6120	**0.9994**

The bold represents the best AUC score within a network.

As shown in Table 3, it is clear that LPSMN performs better than the latent feature approaches. One reason is that by simultaneously learning from two types of features (graph structure feature and latent feature), LPSMN improves latent feature methods. In addition, latent feature methods only use the information in the prediction later; however, LPSMN not only considers the information in the prediction layer but also considers the information from the external layer. Since in a multiplex network each layer has the same nodes, there are a lot of isolated nodes in the prediction layer; however, there are no isolated nodes in the external layer. Another point is that LPSMN significantly outperforms Node2Vec. This indicates that monolayer network embeddings perform worse than using supported information from an external network. Moreover, the most valuable link prediction information hidden in the network might not be fully captured by network embeddings alone. It is also interesting that compared to LPSMN$_{str}$, which uses only structure features (Table 2), performance is improved via joint learning. In order to explore what each part of LPSMN does, we perform an ablation study.

4.4.3. Ablation Study

Table 4 shows the ablation study results of LPSMN. LPSMN$_{L2}$ only considers structure features and latent features on a prediction layer whose performance is poorer than LPSMN, which represents the importance of information from the external layer. LPSMN$_{L2lat}$ and LPSMN$_{L2str}$ only consider the latent feature on the prediction layer and structure feature on the prediction layer, respectively. Both of these two frameworks have poorer performance than LPSMN$_{L2}$, which shows that the combination of structure feature and latent feature can improve the performance of link prediction methods even in a single sparse network.

In Table 2, we have mentioned LPSMN$_{str}$, which only considers the structure feature in the prediction layer and external layer. Comparing it with LPSMN$_{lat}$, which only considers the latent feature in the prediction layer and external layer, we can find that ignoring each of these features can decline the performance of LPSMN. After the ablation study, it is apparent to demonstrate that each part of LPSMN plays an important role in link prediction.

Table 4. Ablation study of LPSMN with the combination of different features and layers.

Multiplex Network	LPSMN$_{L2}$	LPSMN$_{L2lat}$	LPSMN$_{L2str}$	LPSMN$_{lat}$	LPSMN
SF (γ = 2.1)	0.7282	0.5309	0.5109	0.9153	**0.9628**
SF (γ = 2.5)	0.7807	0.5218	0.502	0.915	**0.971**
SF (γ = 3)	0.7352	0.5353	0.509	0.9172	**0.971**
SF (γ = 5)	0.6921	0.5111	0.5038	0.9213	**0.9718**
SF (γ = 10)	0.7584	0.5056	0.5031	0.9343	**0.9816**
WD	0.9757	0.9189	0.7214	0.9966	**0.998**
FT	0.9849	0.9487	0.623	0.9974	**0.9994**

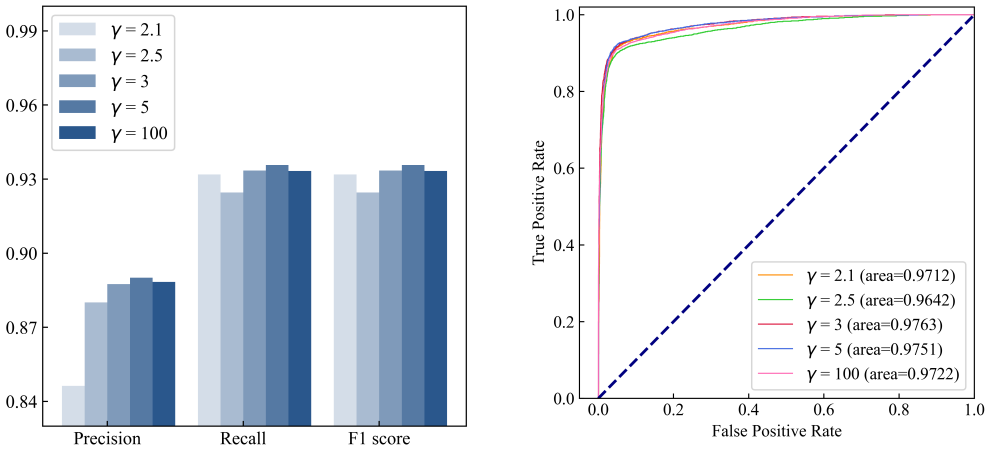
The bold represents the best AUC score within a network.

4.4.4. Complexity Analysis

Heuristic methods only consider local topological structure and are difficult to apply on large-scale graph, which will result in very high complexity. Latent methods adopt different sampling strategies, which effectively reduces both time and space complexity. S2V's complexity is exponential to the number of nodes in the network, which is $\Theta(|V|\log|V|)$. LINE's complexity is only related to the number of edges in the network, which is $O(|E|)$. DW and N2V have the same complexity, which is dependent on the number of nodes and the number of edges of the network, with $O(|E| + |V| + |V|\log|V|)$. The complexity of LPSMN is only related to the number of edges in the network, which is $O(|E|)$. LPSMN has similar complexity to LINE and lower complexity than S2V, DW and N2V.

4.4.5. Impact of Network Heterogeneity

The purpose of this subsection is to study how LPSMN is influenced by network heterogeneity. In Figure 6a, the recall, precision and F1 score metrics are denoted by blue bars, and as the heterogeneity decreases, the blue colour changes from light to dark. By adjusting p from 1 to 0, the power exponent γ can grow from 2 to ∞, where the recall metric shows an upwards trend, and the precision and F1 score metrics display a slight fluctuation. When $\gamma = 2.5$, precision and F1 score metrics are at their lowest points, that is, 0.9246, respectively. After that, a slight increase can be seen when γ changes from 2.5 to 5. When $\gamma = 5$, all three metrics are at their highest points, with values of 0.8901, 0.9357 and 0.9357, respectively. The AUC score is denoted by five different colours with varying power exponents γ. The AUC value fluctuates when adjusting γ from 2.1 to 100, and the AUC data are presented in Figure 6b. The results indicate that heterogeneity, as a universal topological feature, plays a crucial role in network behaviours. As the network heterogeneity decreases, the performance of LPSMN fluctuates.



(**a**) (**b**)

Figure 6. LPSMN performance among different degree distribution networks. (**a**) Recall, precision and F1 score among different degree distribution networks; (**b**) AUC among different degree distribution networks.

4.4.6. Sensitivity Analysis

Since the prediction layer is generated by randomly removing the edges of the validation layer, the removal ratio is an important parameter, and we analyse their sensitivity. Accordingly, we generate the prediction layer by setting the removal ratio to 10, 20, 30, 40, 50, and 60%. The experimental results of WD are shown in Figure 7a,b.

It is concluded that the 10% removal ratio generally outperforms the other removal ratios. Although the 60% removal ratio clearly has the worst performance, LPSMN, with a 60% removal ratio, has more accurate results than the baselines. The prediction performance of LPSMN decreases when the removal ratio changes from 10% to 60%. The LPSMN performance in terms of the AUC score displays a different trend. When the removal ratio is set to 10%, the AUC score of LPSMN is the highest at 0.9991. Then, it continues to decrease to 0.9985 when the removal ratio is 20%. After that, there is an increase to 0.9995.

Figure 7c,d shows the experimental results on the FT dataset. The precision, recall and F1 score metrics show a downward trend as the percentage of the removed edges increases. It is apparent that even though the information from the external layer does not change, as the number of edges in the prediction layer decreases, the performance of LPSMN decreases. The precision, recall and F1 score on the FT dataset show a similar trend to those on the WD dataset. The AUC scores have the same trends on the two datasets. When the proportion of removed edges rises from 0.1 to 0.6, the AUC score first increases and then decreases. When the removal ratio is 30%, the performance is the best.

Training data volume is an essential parameter, and we choose 60%, 70% and 80% as the division proportions of the dataset to analyse parameter sensitivity. In addition, we set three learning rates—0.001, 0.005 and 0.008.

As shown in Figure 8, we study the heat maps of the three evaluation metrics (precision, recall and F1 score) of our model. The model produces better prediction results when the three measures are larger. The better a value is, the darker the corresponding pickle; hence, the model with an 80% removal ratio and 0.08 learning rate is the optimal choice. The proposed model's success is due to the traditional machine learning methods, according to the analysis of the experimental outcomes.

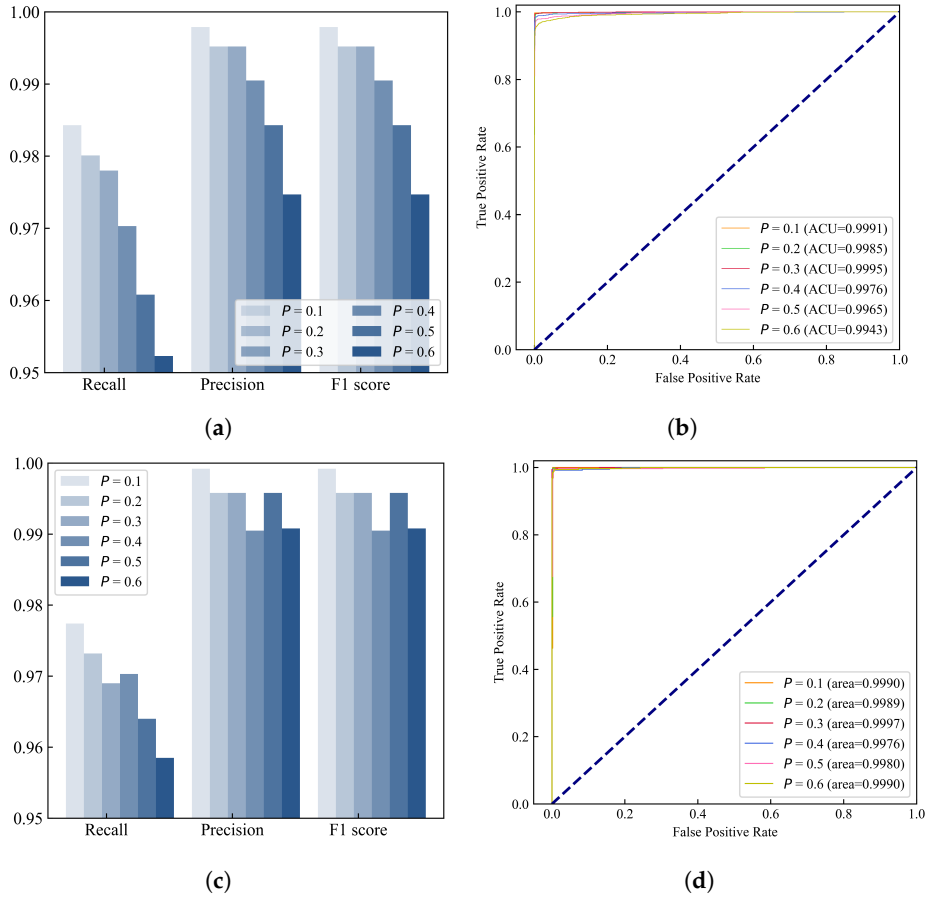


Figure 7. LPSMN performance among different removing-ratio networks on a real-world network. (**a**) Recall, precision and F1 score among different removing-ratio networks on WD; (**b**) AUC among different removing-ratio networks on WD; (**c**) Recall, precision and F1 score among different removing-ratio networks on FT; (**d**) AUC among different removing-ratio networks on FT.

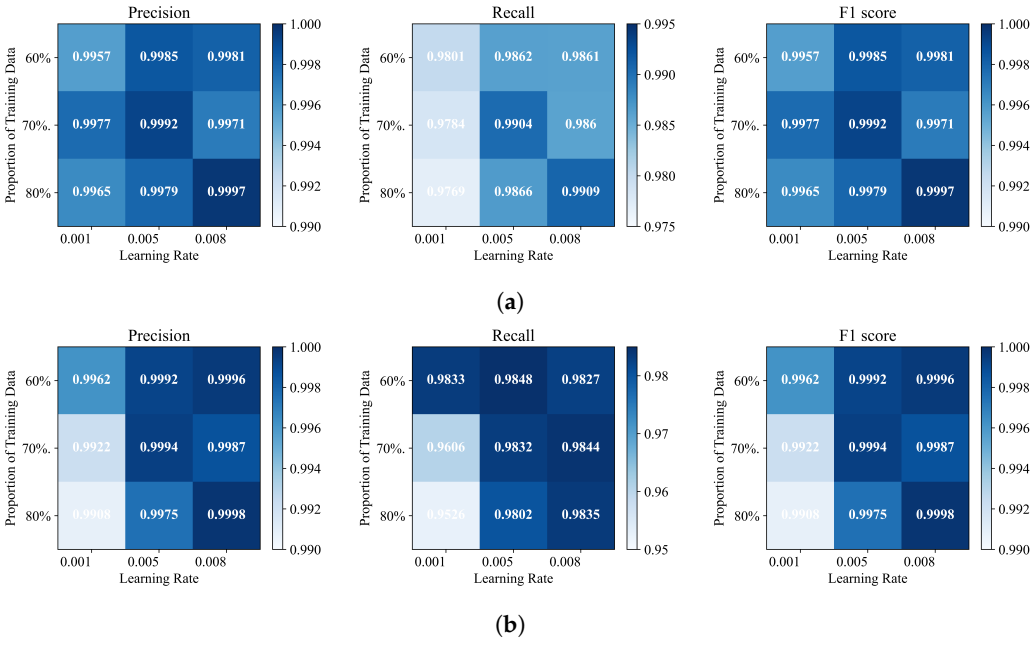


Figure 8. Parameter sensitivity results of the LPSMN model with respect to three metrics. (**a**) Results on WD; (**b**) Results on FT.

5. Conclusions and Future Work

In this paper, based on deep learning, we propose a novel multiplex social network link prediction framework, namely LPSMN, which can predict links in a monolayer network with the help of external layers. Different social platforms should be modelled into different layers rather than monolayers since they have various types of connections for the same user. The experiments focus on feature extraction and how to fuse information from different layers. Traditional monolayer model prediction has limitations when facing a network with insufficient information; hence, the LPSMN model combines global structure features, latent features and explicit features to generate training features and model different social platforms into multiplex networks. In addition, the LPSMN model can effectively synthesise information from different layers, which can promote the prediction results of the model. We simulate five synthetic networks with various degree distributions and two real social networks, and the experiments show that LPSMN has better performance on these networks than the common baseline models.

With the booming of social media, a growing amount of newly built social platforms do not have sufficient information to predict missing or forthcoming links. It is anticipated that our method will have great potential in the field of recommendation for social networks with limited information. However, our model only considers a static network and ignores temporal information. In addition, we only simply concatenate different features and ignore the semantic information hidden in them. In the future, we plan to combine sequential characteristics and other fusing methods to strengthen the prediction performance of the model. In addition, feature-extracting methods play important roles in the overall framework, but our framework lacks an adequate analysis of selecting feature-extracting methods. In the future, we should consider more creativity in feature-extracting methods.

Author Contributions: Conceptualisation, J.C.; methodology, J.C.; software, J.C.; validation, J.C.; formal analysis, J.C.; investigation, J.C.; resources, J.C.; data curation, J.C.; writing—original draft preparation, J.C.; writing—review and editing, T.L.; visualisation, J.C.; supervision, J.L.; project administration, J.J.; funding acquisition, J.L. and J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (NNSFC) under Grant 72001209, 72231011, and 72071206; the Science and Technology Innovative Research Team in Higher Educational Institutions of Hunan Province under Grant 2020RC4046; the Science Foundation for Outstanding Youth Scholars of Hunan Province under Grant 2022JJ20047.

Institutional Review Board Statement: This article does not involve ethical research and does not require ethical approval.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The common dataset used in this study can be obtained from https://apex.sjtu.edu.cn/datasets/8 (30 March 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Daud, N.N.; Ab Hamid, S.H.; Saadoon, M.; Sahran, F.; Anuar, N.B. Applications of link prediction in social networks: A review. *J. Netw. Comput. Appl.* **2020**, *166*, 102716. [CrossRef]
2. Tang, C.F.; Rao, Y.; Yu, H.L.; Sun, L.; Cheng, J.M.; Wang, Y.T. Improving Knowledge Graph Completion Using Soft Rules and Adversarial Learning. *Chin. J. Electron.* **2021**, *30*, 623–633.
3. Barabási, A.L. Network science. *Philos. Trans. R. Soc. A* **2013**, *371*, 20120375. [CrossRef] [PubMed]
4. Hasan, M.; Zaki, M. A survey of link prediction in social networks. In *Social Network Data Analytics*, 1st ed.; Aggarwal, C.C., Ed.; Springer US: Boston, MA, USA, 2011; pp. 243–275.
5. Davis, D.; Lichtenwalter, R.; Chawla, N.V. Multi-relational link prediction in heterogeneous information networks. In Proceedings of the International Conference on Advances in Social Networks Analysis and Mining, Kaohsiung, Taiwan, 25 July 2011; pp. 281–288.
6. Ma, J.L.; Sun, Z.C.; Zhang, Y.Q. Enhancing traffic capacity of multilayer networks with two logical layers by link deletion. *IET Control Theory Appl.* **2022**, *16*, 1–6. [CrossRef]

7. Zhang, M.; Chen, Y. Link prediction based on graph neural networks. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montreal, Canada, 3 December 2018; pp. 5171–5181.
8. Yu, C.; Zhao, X.; An, L.; Lin, X. Similarity-based link prediction in social networks: A path and node combined approach. *J. Inf. Sci.* **2017**, *43*, 683–695. [CrossRef]
9. Kumar, A.; Singh, S.S.; Singh, K.; Biswas, B. Link prediction techniques, applications, and performance: A survey. *Physica A* **2020**, *553*, 124289. [CrossRef]
10. Roweis, S.T.; Saul, L.K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **2000**, *290*, 2323–2326. [CrossRef]
11. Adamic, L.A.; Adar, E. Friends and neighbors on the web. *Soc. Netw.* **2003**, *25*, 211–230. [CrossRef]
12. Jaccard, P. Etude de la distribution florale dans une portion des alpes et du jura. *Bull. Soc. Vaudoise Sci. Nat.* **1901**, *37*, 547–579.
13. Jalili, M.; Orouskhani, Y.; Asgari, M.; Alipourfard, N.; Perc, M. Link prediction in multiplex online social networks. *R. Soc. Open Sci.* **2017**, *4*, 1–11. [CrossRef]
14. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24 August 2014; pp. 701–710.
15. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q.Z. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18 May 2015; pp. 1067–1077.
16. Grover, A.; Leskovec, J. Node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 6 August 2016; pp. 855–864.
17. Fu, S.C.; Liu, W.F.; Li, S.Y. Two-order graph convolutional networks for semi-supervised classification. *IET Image Process.* **2019**, *13*, 2763–2771.
18. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6 August 2017; pp. 1263–1272.
19. Zhang, M.; Cui, Z.; Neumann, M.; Chen, Y.X. An end-to-end deep learning architecture for graph classification. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2 February 2018; pp. 383–391.
20. Ai, B.; Qin, Z.; Shen, W.; Li, Y. Structure enhanced graph neural networks for link prediction. *arXiv* **2022**, arXiv:2201.05293.
21. Shan, N.; Li, L.; Zhang, Y.; Bai, S.S.; Chen, X.Y. Supervised link prediction in multiplex networks. *Knowl.-Based Syst.* **2020**, *203*, 106168. [CrossRef]
22. Chen, L.; Qiao, S.J.; Han, N.; Yuan, C.A.; Song, X.; Huang, P.; Xiao, Y. Friendship prediction model based on factor graphs integrating geographical location. *CAAI Trans. Intell. Technol.* **2020**, *5*, 193–199. [CrossRef]
23. Tang, R.; Jiang, S.; Chen, X.; Wang, H.Z.; Wang, W.X.; Wang, W. Interlayer link prediction in multiplex social networks: An iterative degree penalty algorithm. *Knowl.-Based Syst.* **2020**, *194*, 105598. [CrossRef]
24. Nasiri, E.; Berahm, K.; Li, Y. A new link prediction in multiplex networks using topologically biased random walks. *Chaos Soliton. Fract.* **2021**, *151*, 111230. [CrossRef]
25. Malhotra, D.; Goyal, R. Supervised-learning link prediction in single layer and multiplex networks. *Mach. Learn. Appl.* **2021**, *6*, 100086. [CrossRef]
26. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks? *arXiv* **2018**, arXiv:1810.00826.
27. Shervashidze, N.; Schweitzer, P.; van Leeuwen, E.J.; Mehlhorn, K.; Borgwardt, K.M. Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.* **2011**, *12*, 2539–2561.
28. Cai, H.; Zheng, V.W.; Chang, K.C.C. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1616–1637. [CrossRef]
29. Goyal, P.; Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowl.-Based Syst.* **2018**, *151*, 78–94. [CrossRef]
30. Figueiredo, D.R.; Ribeiro, L.F.R.; Saverese, P.H.P. struc2vec: Learning node representations from structural identity. *arXiv* **2017**, arXiv:1704.03165.
31. Nickel, M.; Jiang, X.; Tresp, V. Reducing the rank in relational factorization models by including observable patterns. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Cambridge, MA, USA, 8 December 2014; pp. 1179–1187.
32. Zhao, H.; Du, L.; Buntine, W. Leveraging node attributes for incomplete relational data. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6 August 2017; pp. 4072–4081.
33. Cao, X.Z.; Yu, Y. BASS: A Bootstrapping Approach for Aligning Heterogenous Social Networks. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Riva del Garda, Italy, 19 September 2016; pp. 459–475.
34. Price, D.D.S. A general theory of bibliometric and other cumulative advantage processes. *J. Am. Soc. Inf. Sci.* **1976**, *27*, 510–515. [CrossRef]
35. Liang, B.; Wang, X.; Wang, L. Impact of heterogeneity on network embedding. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1296–1307. [CrossRef]
36. Lü, L.; Zhou, T. Link prediction in complex networks: A survey. *Physica A* **2011**, *390*, 1150–1170. [CrossRef]
37. Ravasz, E.; Somera, A.L.; Mongru, D.A.; Oltvai, Z.N.; Barabási, A.L. Hierarchical organization of modularity in metabolic networks. *Science* **2002**, *297*, 1551–1555. [CrossRef]
38. Zhou, T.; Lü, L.; Zhang, Y.C. Predicting missing links via local information. *Eur. Phys. J. B* **2009**, *71*, 623–630. [CrossRef]

39. Xie, Y.B.; Zhou, T.; Wang, B.H. Scale-free networks without growth. *Physica A* **2008**, *387*, 1683–1688. [CrossRef]
40. Barabási, A.L.; Albert, R. Emergence of scaling in random networks. *Science* **1999**, *286*, 509–512. [CrossRef]
41. Ou, Q.; Jin, Y.D.; Zhou, T.; Wang, B.H.; Yin, B.Q. Power-law strength degree correlation from resource-allocation dynamics on weighted networks. *Phys. Rev. E* **2007**, *75*, 021102. [CrossRef]
42. Lü, L.; Jin, C.H.; Zhou, T. Similarity index based on local paths for link prediction of complex networks. *Phys. Rev. E* **2009**, *80*, 046122. [CrossRef] [PubMed]
43. Katz, L. A new status index derived from sociometric analysis. *Psychometrika* **1953**, *18*, 39–43. [CrossRef]