

# NeuLP: An End-to-end Deep-learning Model for Link Prediction

Zhiqiang Zhong<sup>1</sup>, Yang Zhang<sup>2</sup>, and Jun Pang<sup>1,3</sup>

<sup>1</sup> Faculty of Science, Technology and Medicine,  
University of Luxembourg, Esch-sur-Alzette, Luxembourg

<sup>2</sup> CISA Helmholtz Center for Information Security,  
Saarland Informatics Campus, Saarbrücken, Germany

<sup>3</sup> Interdisciplinary Centre for Security, Reliability and Trust,  
University of Luxembourg, Esch-sur-Alzette, Luxembourg

**Abstract.** Graph neural networks (GNNs) have been proved useful for link prediction in online social networks. However, existing GNNs can only adopt shallow architectures as too many layers will lead to over-smoothing and vanishing gradient during training. It causes nodes to have indistinguishable embeddings if locally they have similar structural positions, and this further leads to inaccurate link prediction. In this paper, we propose a unified end-to-end deep learning model, namely Neural Link Prediction (NEuLP), which can integrate the linearity and non-linearity user interactions to overcome the limitation of GNNs. The experimental evaluation demonstrates our model’s significant improvement over several baseline models. Moreover, NEuLP achieves a reliable link prediction given two users’ different types of attributes and it can be applied to other pairwise tasks. We further perform in-depth analyses on the relation between prediction performance and users’ geodesic distance and show that NEuLP still can make accurate link prediction while two users are far apart in the networks.

## 1 Introduction

Link prediction is one of the key problems in online social networks (OSNs) mining, which aims to estimate the likelihood of the existence of an edge (relationship) between two nodes (users) [14]. It has been widely applied for friendship recommendation in OSNs, which is important to increase user engagement and leads to more satisfactory user experience. Besides OSNs, link prediction has drawn the attention from many different application domains and researches in various fields, such as in the study of protein-protein interaction networks [10], and identifying hidden or missing criminals in terrorist networks [3].

The existing methods for link prediction in OSNs can be classified into three categories, ranging from the early methods focusing on hand-crafted features (e.g., see methods based on user profiles [2] and graph structure [14, 15]); shallow network embedding-based methods, such as LINE [22] and node2vec [4]; to recently emerged graph neural network (GNN)-based methods [9, 5, 23]. Among

these, GNNs is currently the most popular paradigm, largely owing to their efficiency learning capability. In contrast, hand-crafted features methods are limited in extracting latent information in the OSNs, and shallow network embedding-based methods cannot incorporate user attributes.

GNN-based methods follow a common scheme that explores user information from either social graph or user attributes as node embeddings and then calculates user similarity for making link prediction. However, we argue that it is insufficient since the classic GNNs cannot capture the global information for a given user pair in an OSN. Current GNNs generally follow message-passing mechanism and assume that with the use of enough GNN layers (times of interactive aggregation), the model can capture long-range dependencies. However, recent research shows that GNNs are not robust to multiple layers since they would become over-smoothing and lead to vanishing gradient during training [13, 11]. Therefore, existing GNNs have shallow architecture, 2–4 layers. It causes nodes to have indistinguishable embedding vectors if locally they are at similar positions of an OSN. To fill this gap, You et al. propose a novel GNN model (P-GNNs) which could capture the position of a given node with respect to all other nodes in the graph [24]. However, the introduced anchor-set sampling brings with high computational costs and performs less well on attributed networks.

In this paper, we propose a unified end-to-end deep learning model, namely Neural Link Prediction (NEULP), which could efficiently integrate the linearity and non-linearity user interactions to overcome the above limitations, and NEULP is flexible to handle different types of user attributes for link prediction. We firstly use two shallow GNNs to transform user information (their social relations and attributes) to embeddings which interactively aggregate user features from neighbours. We then add a fusion layer to merge embeddings of two users into one vector, which performs element-wise product on two user embeddings to capture the linearity interactions between two users. However, a simple element-wise product does not account for the non-linearity interactions between users, so we further apply multilayer perceptron (MLP) on the vector to map it to low dimensions. The MLP can explore the non-linearity interactions between two users, where each layer performs a non-linear transformation with an activation function. This architecture is inspired by the recent progress of recommendation systems [6] which is useful for mining online user behaviours. In the end, we utilise a linear transformation layer to summarise the extracted features of a pair of users as the possibility of existing a link between them.

We conduct extensive experiments on five benchmark datasets and two Instagram datasets in various pairwise prediction tasks to evaluate the performance of NEULP with respect to several state-of-the-art baselines, including (1) link prediction with only network structures and (2) link prediction on attributed social network datasets with one/two type(s) of user attributes. The experimental results demonstrate our model’s significant improvement over the baselines, with up to 5.8% improvement in terms of the AUC score, and its capability in leveraging user attributes and interactions when compared with GNN-based baselines. Moreover, NEULP achieves a strong link prediction when given users’

two different types of attributes. We further perform in-depth analyses on the relation between prediction performance and the graph geodesic of two users in OSNs and this demonstrates that our model successfully used user interactions to improve GNN-based baselines.

## 2 Related Work

We briefly review the state-of-the-art on link prediction in OSNs, including methods based on hand-crafted features, shallow network embedding and GNNs [19]. The hand-crafted features-based methods extract the feature hidden inside the user attributed and edges' structures. A heuristic score is used to measure the similarity and connectivity between two users [14, 15, 2]. These methods are pellucid and efficient, but cannot explore the latent information in OSNs.

Shallow network embedding-based methods learn a link's formation mechanism from the network other than assuming a particular mechanism (e.g., common neighbours), and they can be summarised into two groups, i.e., DeepWalk-based methods and matrix factorisation-based methods. DeepWalk [20] pioneers network embedding methods by considering the node paths traversed by random walks over the network as sentences and leveraging Skip-gram model [18] for learning node representations. On the other hand, some works adopt the idea of matrix factorisation for network embedding [21, 16]. However, these methods cannot deal with user attributes naturally and cannot optimise parameters for a specific task.

In the past few years, many studies have adopted GNNs to aggregate node information in arbitrary graph-structured data for link prediction. Most of the existing GNN models (e.g., see [9, 5, 23]) rely on a series of graph message-passing architectures that perform the convolution in the graph domain by aggregating node feature messages from its neighbours in the graph and stacked multiple GNN layers can capture the long-range node dependencies. However, existing GNNs can only adopt the shallow architecture, with 2–4 layers, which is not enough to get global network structure information. You et al. [24] recently proposed a global position-aware information capture mechanism to overcome this limitation. However, the introduced anchor-set sampling incurs high computational costs and makes the approach perform less well on attributed datasets.

## 3 Problem Definition

A social network is denoted as  $G = (U, E)$ , where  $U = \{u_1, \dots, u_n\}$  represents the set of users and  $E \subseteq U \times U$  is the set of links between users. Besides, we can use the adjacency matrix  $A = (a_{ij}) \in \mathbb{R}^{n \times n}$  to present the social graph, where we have  $a_{ij} = 1$  if  $(u_i, u_j) \in E$ , otherwise  $a_{ij} = 0$ . We formally define the social network with user-shared information (i.e., attributes) as  $G = (U, E, H)$ , where  $H \in \mathbb{R}^{n \times k}$ ,  $k$  is the dimension of user attributes. Each node  $u_i \in U$  is associated with some types of features vectors and a feature type mapping

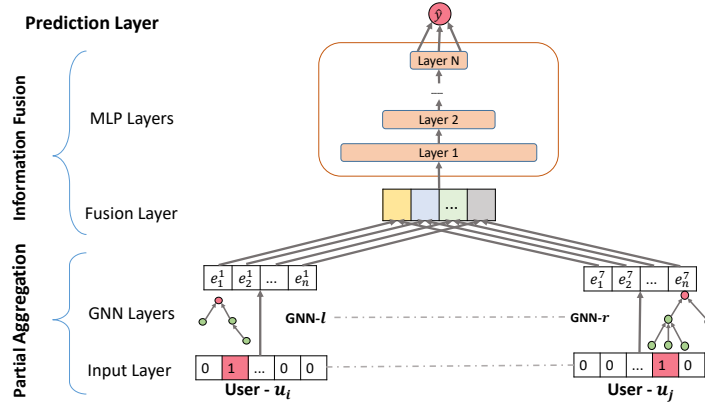


Fig. 1: NEULP’s model architecture. The two input user feature vectors (left:  $u_i$ , right:  $u_j$ ) are firstly transformed with two GNNs and further fused with multiple propagation layers, and the output is the predicted possibility of existing friendship between  $u_i$  and  $u_j$ .

function.  $H = \{x_i \mid u_i \in U\}$  is the set of node features for all nodes, where  $x_i$  is the associated node feature of the node  $u_i$ .

Given a social network  $G = (U, E)$  or an attributed social network  $G = (U, E, H)$ , the goal of the link prediction problem is to learn a scoring function  $s: U \times U \rightarrow \mathbb{R}$  for predicting a new link between an unlabelled pair of users in a set  $E_p := (U \times U) \setminus E$ .

## 4 Proposed Approach

### 4.1 Model overview

Our proposed model NEULP has three main components as shown in Figure 1: partial aggregation, information fusion and model prediction. In the partial aggregation part, we use the input layer to get the encoded user attributes and feed them into GNN layers. Then, the GNN could iteratively aggregate information from a user’s local neighbourhoods in the social network and update the user presentation (embedding). After obtaining user embeddings through partial aggregation simultaneously, we add a fusion layer to merge embeddings of two users into one vector and further apply MLP on the vector to explore the non-linearity interactions between these two users. At last, we adopt a prediction layer to linearly summarise the extracted feature of two users as the possibility of existing an edge between them.

### 4.2 Partial aggregation

In this subsection, we describe the details of adopting GNNs to iteratively aggregate the partial information from a user’s local network neighbourhood and update its embedding.

**Input layer.** The bottom input layer consists of two feature vectors  $h_i$  and  $h_j$ , where  $h_i, h_j \in H$ , that describe user  $u_i$  and user  $u_j$ , respectively. They can be customised to support a wide range of user attributes, such as location check-ins and posted hashtags.

**GNN encoding.** We adopt the GNN to iteratively aggregate neighbour information for each user in its sub-network and update their embeddings. More specifically, we use Graph Convolutional Networks (GCN) [9] as an example GNN, due to its widespread use and remarkable contributions. GCN operates directly on a graph and induces node feature vectors from the properties of their neighbourhoods. The model can be built by stacking multiple convolutional layers to involve information from farther neighbours.

Formally, given an attributed social network  $G = (U, E, H)$ , and its adjacent matrix  $A$ , a GCN layer is a nonlinear transformation that maps from  $H$  to  $H^{(1)} \in \mathbb{R}^{n \times k'}$ :

$$H^{(1)} = \phi(\hat{A}HW^{(0)} + b^{(0)}) \quad (1)$$

where  $W^{(0)} \in \mathbb{R}^{k \times k'}$ ,  $b^0 \in \mathbb{R}^{k'}$  are model parameters.  $\phi$  is a non-linear activation function.  $k'$  is the output feature dimension and  $\hat{A}$  is the normalised symmetric adjacency matrix which is defined as:

$$\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \quad (2)$$

Here,  $\tilde{A} = A + I_N$  and  $\tilde{D}$  is the degree matrix.  $I_N$  is an identity matrix with dimension of  $n$  and  $\tilde{D}_{ij} = \sum_j \tilde{A}_{ij}$ .

With multiple convolutional layers, the GCN model is then described as:

$$H^{(\ell)} = \phi(\tilde{A}H^{(\ell-1)}W^{(\ell-1)} + b^{\ell-1}) \quad (3)$$

where  $\ell$  denotes the layer number, and  $H^{(\ell)}$  is the node embedding matrix passing by layers. The intuition is that at each layer node aggregates information from their local neighbours. Therefore, we get two embedding matrices  $H_1^{(\ell)}$  and  $H_2^{(\ell)}$  from the left and the right GCNs, respectively.

### 4.3 Information fusion

We continue to describe the information fusion layer and how to explore non-linearity user interactions with the MLP layers.

**Fusion layer.** In this layer, we merge embeddings of two users ( $h_i^{(\ell)} \in H_1^{(\ell)}$ ,  $h_j^{(\ell)} \in H_2^{(\ell)}$ ) into one vector. In detail, inspired by the Neural Matrix Factorisation model, NeuMF [6], we define the fusion function as:

$$z_0 = \phi(W^T h_i \odot h_j) \quad (4)$$

where  $\odot$ ,  $\phi$  and  $W$  denote the element-wise product of vectors, activation function and edge weights of the output layer, respectively. The benefit of adopting the fusion layer instead of concatenating two embeddings together is that we can

extract the linearity pair-wise interaction between two users. This compensates the insufficiency of GCNs to acquire global user interactions.

**MLP layers.** Since the fusion layer analogues the matrix factorisation model to capture the linear interaction between two users. However, simply capturing linearity user interaction is not enough. To address this issue, we propose to add hidden layers on the fused vector, using a standard MLP to extract the non-linear interactions for pairs of users. In this sense, we can endow the interactions between  $u_i$  and  $u_j$ , rather than the way of fusion layer that uses only a fixed element-wise product on them. More precisely, the MLP layers in our NEULP model is defined as:

$$\begin{aligned} z_1 &= \phi_1(W_1^T z_0 + c_1) \\ z_2 &= \phi_2(W_2^T z_1 + c_2) \\ &\dots \\ z_m &= \phi_m(W_m^T z_{m-1} + c_m) \end{aligned} \tag{5}$$

where  $W_m$ ,  $c_m$ , and  $\phi_m$  denote the weight matrix, bias vector, and activation function for the  $m$ -th layer’s perceptron, respectively.

#### 4.4 Model prediction

After the partial aggregation and information fusion layers, we obtain a representation vector  $r_{ij} = z_m$  for each pair of users ( $u_i$  and  $u_j$ ), which presents the latent features between them. As such, in the last prediction layer, we summarise the latent features as a predicted score which represents the possibility of existing a link between these two users:

$$\hat{y}_{ij} = \sigma(W_f^T \phi_{m+1}(z_m)) \tag{6}$$

where  $\sigma(x) = \frac{1}{1+e^{-x}}$  to scale the output into  $(0, 1)$ . With this, we successfully sum up the comprehensive information between  $u_i$  and  $u_j$  into a similarity score  $\hat{y}_{ij}$  for link prediction.

#### 4.5 Optimisation

To learn model parameters, we optimise the Binary Cross Entropy Loss (BCELoss) which has been intensively used in link prediction models. We adopt the mini-batch Adam [8] to optimise the prediction model and update the model parameters. In particular, for a batch of randomly sampled user tuple  $(u_i, u_j)$ , we compute the possibility  $\hat{y}_{ij}$  of existing a link between them after partial aggregation and information fusion, and then update model parameters by using the gradient of the loss function.

## 5 Experiments

In this section, we conducted intensive experiments in order to answer the following research questions:

Statistics	USAir	Yeast	Cora	Email	Citeseer	Instageam-1	Instagram-2
# of Nodes	332	2,375	2,708	799	3,312	6,815	12,944
# of Edges	2,126	11,693	5,429	10,182	4,660	36,232	61,963
Node features	No	No	Yes	No	Yes	Yes	Yes
Node label	No	No	Yes	Yes	Yes	No	No

Table 1: Statistics summary of the seven datasets.

- RQ1: How effective is NEULP when compared with the state-of-the-art baselines focusing on network structures?
- RQ2: How does NEULP perform when compared with the state-of-the-art baselines for attributed social network datasets with only one type of user attributes?
- RQ3: Can we utilise NEULP for link prediction with two different types of user attributes? How effective will it be?

### 5.1 Dataset description

We evaluate our model on five widely used datasets for graph-related machine learning tasks: USAir, Yeast, Cora, Email and Citeseer. They are publicly accessible on the websites, together with their dataset descriptions.<sup>4</sup> We further use an Instagram dataset, which we collected from Instagram relying on its public API.<sup>5</sup> Our data collection follows a similar strategy as the one proposed by Zhang et al [26]. Concretely, we sample users from New York by their geotagged posts. Then, for each user, we collected all her/his posted hashtags, and further performed following preprocessing to filter out the users matching any of the following criteria: (i) users whose number of followers are above the 90th percentile (celebrities) or below the 10th percentile (bots); (ii) users without location check-ins or posted hashtags. Then we generate two datasets Instagram-1 and Instagram-2 according to different filter conditions:

- Instagram-1: users with no less than 100 location check-ins and with no less than 10 friends.
- Instagram-2: users with no less than 100 location check-ins and with no less than 5 friends.

The reason for generating these two Instagram datasets is that we need datasets with different types of user attributes to discuss the possibility of applying NEULP to attributed social networks with different types of user attributes (RQ3). The statistics of the seven datasets are given in Table 1.

For the experiments, we use two sets of 10% existing links and an equal number of nonexistent links as test and validation sets. We use the left 80% existing links and an equal number of nonexistent links as the training sets.

<sup>4</sup> <https://linqs.soe.ucsc.edu/data>, and <http://snap.stanford.edu/data/index.html>

<sup>5</sup> The dataset was collected in 01/2016 when Instagram’s API was publicly available.

	USAir	Yeast	Cora	Instagram-1	Instagram-2
MF	$0.831 \pm 0.010$	$0.903 \pm 0.003$	$0.816 \pm 0.005$	$0.740 \pm 0.013$	$0.714 \pm 0.008$
NeuMF	$0.801 \pm 0.015$	$0.907 \pm 0.005$	$0.686 \pm 0.009$	$0.744 \pm 0.001$	$0.727 \pm 0.001$
node2vec	$0.805 \pm 0.002$	$0.905 \pm 0.008$	$0.770 \pm 0.005$	$0.785 \pm 0.004$	$0.738 \pm 0.002$
GCN	$0.903 \pm 0.006$	$0.938 \pm 0.003$	$0.819 \pm 0.006$	$0.804 \pm 0.004$	$0.773 \pm 0.003$
GraphSAGE	$0.897 \pm 0.007$	$0.933 \pm 0.004$	$0.838 \pm 0.008$	$0.802 \pm 0.005$	$0.770 \pm 0.002$
GAT	$0.902 \pm 0.006$	$0.935 \pm 0.004$	$0.839 \pm 0.008$	$0.796 \pm 0.005$	$0.767 \pm 0.002$
P-GNNs	$0.911 \pm 0.018$	$0.940 \pm 0.006$	$0.852 \pm 0.008$	$0.734 \pm 0.007$	OOM
HGANE	$0.901 \pm 0.004$	$0.932 \pm 0.006$	$0.845 \pm 0.010$	$0.797 \pm 0.007$	$0.771 \pm 0.008$
NEULP	<b><math>0.952 \pm 0.009</math></b>	<b><math>0.965 \pm 0.003</math></b>	<b><math>0.894 \pm 0.006</math></b>	<b><math>0.813 \pm 0.001</math></b>	<b><math>0.790 \pm 0.003</math></b>

Table 2: Link prediction performance comparison with baseline methods on network datasets (AUC). OOM: out of memory.

## 5.2 Experimental settings

**Evaluation metrics.** Like existing link prediction studies [4], we adopt the most frequently-used metrics “Area under the receiver operating characteristic” (AUC) to measure the performance.

**Baselines.** To demonstrate the effectiveness of NEULP, we compare its performance with several baseline methods: MF [17], NeuMF [6], node2vec [4] and several state-of-the-art GNN variants, including GCN [9], GraphSAGE [5], GAT [23], and P-GNNs [24]. Besides, we also adopt HGANE [7], which is a collective network embedding framework with a hierarchical graph attention mechanism.

In order to study the performances of NEULP on the attributed datasets (link prediction using one same type or two different types of user attributes), we additionally compare our model with walk2friend [1] and tag2friend [25].

**Implementation details.** For the node2vec related approaches, i.e., node2vec, walk2friend, tag2friend, we use the default settings as in [4]. All the neural network-related baselines follow their original paper’s code at their GitHub pages if available; otherwise, we implement it by ourself following their original paper, e.g., HGANE. For NEULP, we use a simple GNN model, GCN, for partial aggregation and the layer numbers of GCN and MLP are 2 and 4, respectively.<sup>6</sup> To make a fair comparison, all methods are set to have a similar number of parameters and trained for the same number of epochs.

## 5.3 Performance comparison (RQ1)

We first compare the link prediction performance of approaches on the network datasets, which do not include any user-shared information. Here, we use the adjacent matrix  $A$  as user features set. We compare the performance of NEULP with the baselines, except for walk2friend and tag2friend, since the latter two require user-shared information.

<sup>6</sup> Code and datasets are available at <https://github.com/zhiqiangzhongddu/NeuLP>.



Attributes	Methods	Datasets	
		Instagram-1	Instagram-2
Location check-in	walk2friend	$0.831 \pm 0.007$	$0.820 \pm 0.003$
	GCN	$0.808 \pm 0.005$	$0.781 \pm 0.003$
	GraphSAGE	$0.801 \pm 0.006$	$0.775 \pm 0.003$
	GAT	$0.784 \pm 0.006$	$0.758 \pm 0.008$
	P-GNNs	$0.686 \pm 0.012$	OOM
	HGANE	$0.830 \pm 0.007$	$0.799 \pm 0.002$
	NEULP	$0.832 \pm 0.002$	$0.828 \pm 0.001$
	NEULP*	<b><math>0.885 \pm 0.001</math></b>	<b><math>0.880 \pm 0.001</math></b>
Hashtag	tag2friend	$0.724 \pm 0.013$	$0.696 \pm 0.011$
	GCN	$0.812 \pm 0.004$	$0.781 \pm 0.003$
	GraphSAGE	$0.806 \pm 0.004$	$0.775 \pm 0.003$
	GAT	$0.774 \pm 0.012$	$0.763 \pm 0.009$
	P-GNNs	$0.717 \pm 0.010$	OOM
	HGANE	$0.819 \pm 0.005$	$0.781 \pm 0.004$
	NEULP	$0.836 \pm 0.002$	$0.826 \pm 0.002$
	NEULP*	<b><math>0.880 \pm 0.002</math></b>	<b><math>0.875 \pm 0.001</math></b>

Table 3: Link prediction performance comparison between NEULP and baseline methods on attributed social network datasets with one user attribute type (AUC). OOM: out of memory.

From the results shown in Table 2, we first observe that our model consistently outperforms all the baselines with remarkable improvements. It indicates the effectiveness of NEULP for link prediction with only network structures. In particular, there is 4.9% AUC improvement on the Cora dataset. NEULP gets 30.3% AUC improvement over NeuMF (i.e., on the Cora dataset), this indicates the observed network plays an important role in link prediction, as NeuMF does not include the network topology. NEULP also outperforms P-GNNs on the Instagram-1 dataset (with 13.5% AUC improvement), it indicates the advantage of our model on real OSN datasets to capture their global information.

#### 5.4 Attributed OSNs (RQ2 & RQ3)

We continue to discuss the performance of various methods on attributed social networks. More specifically we present the performance of NEULP with one type of attributes (RQ2) and discuss the possibility of applying NEULP on attributed social networks with different types of attributes (RQ3). We use walk2friend & tag2friend as the baselines for OSN datasets with location check-ins and hashtags, respectively, and several variant GNNs as baselines for both datasets.

**Link prediction with one attribute type.** Here we use the encoded user-attributes matrix as user features set (i.e., one-hot encoding), where 1 means the user visited a location (or published a hashtag), and 0 otherwise. NEULP\* means we adopt the embeddings generated by random walk-based methods, e.g., walk2friend, tag2friend, as input user features.

From Table 3, we can see that NEULP outperforms walk2friend, tag2friend and other GNN models with up to 5.8% AUC improvement. Moreover, NEULP\*

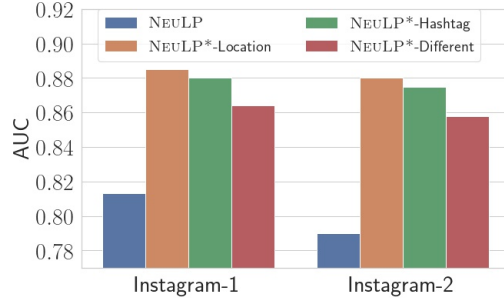


Fig. 2: Link prediction performance comparison of NEULP on the Instagram datasets. NEULP: without user attributes. NEULP\*-feature: adopting random-walk related methods generated embeddings as user features.

gets improvements over NEULP on all three datasets with up to 6.4% AUC improvement, it means the information reflected by location check-ins or hashtag is a good supplement to the social network structure and our model can well capture such information.

**Link prediction with two different attribute types.** To answer the research question RQ3, we design experiments on attributed social networks with two different types of user attributes, which means performing link prediction given two user’s different types of information, e.g., one user shares location check-ins, and the other user shares hashtags (see NEULP\*-Different in Figure 2). Following the previous settings, we randomly sample two groups of users where the first group users share their location check-ins and the second group sharing hashtags. These two groups account for half of all users. We organise two graphs where users with location check-ins are connected through locations and users with hashtags are connected through hashtags. Then we apply walk2friend and tag2friend to these two graphs to generate user embeddings (model settings are as same as default settings). We use these two generated user embeddings matrix as two user features sets for left and right input, respectively.

From the experimental results are shown in Figure 2, we can see that NEULP on attributed social networks with different types of user attributes gets significantly better performances compared with the results without using user attributes. NEULP\*-Different gets similar prediction performance on attributed social networks with the ones using one user attribute type, i.e., NEULP\*-Location and NEULP\*-Hashtag. It demonstrates the flexibility of our model on making link prediction, even when utilising different types of information that users share in OSNs.

**Performance analyses with geodesic distances.** In order to further investigate the prediction performances of different methods for different users pairs, we perform in-depth analyses on the relation between prediction performance and the graph geodesic of two users in OSNs. The results are drawn in Figure 3, which has two sub-figures to plot the prediction accuracy of the methods (node2vec, GCN, P-GNNs and NEULP) with different user-pair geodesic distances (i.e., the

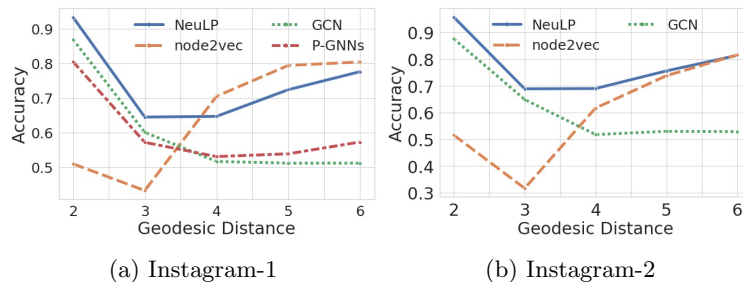


Fig. 3: Prediction accuracy of four link prediction methods on the Instagram datasets with different geodesic distances. P-GNNs leads to OOM on Instagram-2 dataset; thus not showing in figure (b).

number of edges in a shortest path connecting two nodes in the network) for the two Instagram datasets. We see that the accuracy of GCN and P-GNNs significantly decreases when increasing the distance. This means GNN models cannot capture latent information for user pairs of large distances. In contrast, node2vec behaves in an opposite way: it achieves higher accuracy when increasing the distance and it does not perform well when the distance is small. The reason might be that (long) random walks can capture long-distance information. However, NEULP gets the most balanced performances, this demonstrates that it successfully uses both linearity and non-linearity user interactions to improve the performance of GNN-based models for link prediction.

## 6 Conclusion and Future Work

In this paper, we have presented NEULP as an end-to-end deep learning model for link prediction in OSNs. NEULP incorporates linearity and non-linearity user interactions to overcome the limitations of current GNN-based link prediction methods. Compared with the state-of-the-art methods, NEULP can not only achieve significant performance improvements but also have the flexibility of utilising different types of user attributes for link prediction. In future, we plan to extend NEULP for link prediction in temporal (or time-varying) networks [12].

**Acknowledgements.** This work is partially supported by the Luxembourg National Research Fund through grant PRIDE15/10621687/SPsquared.

## References

1. Backes, M., Humbert, M., Pang, J., Zhang, Y.: walk2friends: Inferring Social Links from Mobility Profiles. In: CCS. pp. 1943–1957. ACM (2017)
2. Bhattacharyya, P., Garg, A., Wu, S.F.: Analysis of User Keyword Similarity in Online Social Networks. *Social Network Analysis Mining* **1**, 143–158 (2011)
3. Chen, A.C., Gao, S., Karampelas, P., Alhajj, R., Rokne, J.G.: Finding Hidden Links in Terrorist Networks by Checking Indirect Links of Different Sub-networks. In: Counterterrorism and Open Source Intelligence. pp. 143–158. Springer (2011)

4. Grover, A., Leskovec, J.: node2vec: Scalable Feature Learning for Networks. In: KDD. pp. 855–864. ACM (2016)
5. Hamilton, W.L., Ying, Z., Leskovec, J.: Inductive Representation Learning on Large Graphs. In: NIPS. pp. 1025–1035. NIPS (2017)
6. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.: Neural Collaborative Filtering. In: WWW. pp. 173–182. ACM (2017)
7. Jiao, Y., Xiong, Y., Zhang, J., Zhu, Y.: Collective Link Prediction Oriented Network Embedding with Hierarchical Graph Attention. In: CIKM. pp. 419–428. ACM (2019)
8. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: ICLR (2015)
9. Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. In: ICLR (2017)
10. Lei, C., Ruan, J.: A Novel Link Prediction Algorithm for Reconstructing Protein–protein Interaction Networks by Topological Similarity. *Bioinformatics* **29**(3), 355–364 (2012)
11. Li, G., Müller, M., Qian, G., Delgadillo, I.C., Abualshour, A., Thabet, A.K., Ghanem, B.: DeepGCNs: Making GCNs Go as Deep as CNNs. *CoRR* (2019), abs/1910.06849
12. Li, J., Cheng, K., Wu, L., Liu, H.: Streaming Link Prediction on Dynamic Attributed Networks. In: WSDM. pp. 369–377. ACM (2018)
13. Li, Q., Han, Z., Wu, X.: Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In: AAAI. pp. 3538–3545. AAAI (2018)
14. Liben-Nowell, D., Kleinberg, J.: The Link-prediction Problem for Social Networks. *Journal of the American Society for Information Science and Technology* **58.0**(7), 1019–1031 (2007)
15. Lichtenwalter, R.N., Lussier, J.T., Chawla, N.V.: New Perspectives and Methods in Link Prediction. In: SIGKDD. pp. 243–252. ACM (2010)
16. Liu, X., Murata, T., Kim, K., Kotarasu, C., Zhuang, C.: A General View for Network Embedding as Matrix Factorization. In: WSDM. pp. 375–383. ACM (2019)
17. Menon, A.K., Elkan, C.: Link Prediction via Matrix Factorization. In: ECMLPKDD. pp. 437–452. Springer (2011)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In: ICLR (2013)
19. Mutlu, E.C., Oghaz, T.A.: Review on Graph Feature Learning and Feature Extraction Techniques for Link Prediction. *CoRR* (2019), abs/1901.03425
20. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: Online Learning of Social Representations. In: KDD. pp. 701–710. ACM (2014)
21. Qiu, J., Dong, Y., Ma, H., Li, J., Wang, C., Wang, K., Tang, J.: NetSMF: Large-Scale Network Embedding as Sparse Matrix Factorization. In: WWW. pp. 1509–1520. ACM (2019)
22. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: Large-scale Information Network Embedding. In: WWW. pp. 1067–1077. ACM (2015)
23. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph Attention Networks. In: ICLR (2018)
24. You, J., Ying, R., Leskovec, J.: Position-aware Graph Neural Networks. In: ICML. *JMLR* (2019)
25. Zhang, Y.: Language in Our Time: An Empirical Analysis of Hashtags. In: WWW. pp. 2378–2389. ACM (2019)
26. Zhang, Y., Humbert, M., Rahman, T., Li, C.T., Pang, J., Backes, M.: Tagvisor: A Privacy Advisor for Sharing Hashtags. In: WWW. pp. 287–296. ACM (2018)