Improving Link Prediction in Network Representation Learning with Feature Fusion and Local Outlier Factor

Amr Al-Furas [1, 2]*, Mohammed F. Alrahmawy [2], Waleed Mohamed Al-Adrousy [2], Samir Elmougy [2]

[1] Computer Science Department, Ibb University, Ibb, Yemen

[2] Computer Science Department, Faculty of Computers and Information, Mansoura University, Mansoura 35516, Egypt

Emails: amroso783@gmail.com; Mrahmawy@mans.edu.eg; waleed_m_m@mans.edu.eg; mougy@mans.edu.eg.

Abstract

Complex networks are a diverse set of networks found in various fields, such as social, technological, and biological networks. One important task in complex network analysis is link prediction, which involves detecting missing links or predicting future link formation. Many methods based on network structure analysis have been developed for link prediction, including network representation learning (NRL) models that represent nodes in a low-dimensional space. Fusion-based attributed NRL methods are particularly effective, as they capture both content and structure information. However, NRL models for link prediction are binary classification models, which face challenges in identifying negative links and prioritizing predicted links. To address these challenges, we propose a novel approach that treats link prediction as a novelty detection problem. Our approach uses the Local Outlier Factor (LOF) algorithm to quantify the novelty of non-existent links based on the representations of existing links. Our experimental results show that our proposed approach outperforms existing methods, particularly when used with fusion-based attributed NRL models

Keywords: Link Prediction; Network Representation Learning; Complex Network; Feature Fusion; LOF.

1 Introduction

Complex networks such as social, informational, biological and technical networks can be represented as graphs, where the nodes refer to the elements of the network and represent the interactions of the elements and their connections through links. As the relationships between the members of many real-world networks constantly change, the data of many of these networks is incomplete or incorrect, resulting in a limited amount of information that can be gathered from different platforms as new relationships are formed and others disappear[1]. New individuals may appear at any moment within the network, leading to its expansion, while existing individuals may disappear as well. Therefore, link prediction is a central and fundamental issue in the field of complex network analysis [2]. As the name suggests, link prediction is a process that predicts future relationships between network elements that are not connected at present, in other words, it aims to predict who will be connected to whom in the future. Several tasks have been performed using link prediction algorithms, such as the prediction of probable friendship links in social networks [3, 4], the prediction of the possibility of the authors to collaborate in the future [5, 6] as well as the prediction of recommendations [7], where relevant information about the network users and their interactions can be extracted from data and be used for making predictions. According to the structure of the network, different measures of similarity between nodes were proposed in the literature for predicting links between nodes [3, 8, 9]. Depending on the network structure, there are three directions of recent studies proposed to solve the issue of link prediction: similarity based methods [10, 11], probabilistic methods [12, 13], and maximum likelihood methods[14, 15]. In terms of time complexity, each of the previous methods can be classified

Doi: https://doi.org/10.54216/FPA.120210
Received: June 17, 2023 Revised: July 03, 2023 Accepted: July 16, 2023

into local, global, and quasi-local methods[1]. Local methods rely primarily on neighbouring nodes and not the entire network. This allows them to be implemented quickly and in real-time. There are more than one or two nodes in most potential links, which makes them fail, particularly in large-scale networks. Global methods which rely on the network structure are too complex and fail due to this complexity. Furthermore, the link prediction process does not only depend on network structure; sometimes, as in social networks, node characteristics can influence the prediction[16].

Recently, network representation learning (NRL) has been extensively researched. This approach involves representing network nodes in low-dimensional vectors in order to preserve network architecture, node attributes, and other side information[17]. By learning network representation techniques, machine learning technology can be used to analyze these networks in order to take advantage of the high success demonstrated by machine learning techniques in various areas of analysis[18]. A network representation learning can improve analysis tasks such as classifying nodes[19, 20], discovering communities[21, 22], and link prediction[23, 24].

Fusion-based attributed methods combine node attributes or content information with the network structure to learn the node embeddings[25, 26]. One common approach is to concatenate the node attributes with the network structure representation and then apply a neural network to learn the embeddings[27, 28]. Another approach is to use a separate neural network to learn the embeddings for the node attributes and the network structure representation and then concatenate them[29, 30]. These methods have shown improved performance compared to methods that only use the network structure or node attributes alone[31, 32].

In order to obtain link predictions, different NRL methods require different prediction methods. As some NRL methods calculate link probabilities directly[24, 33], while for other methods this has to be learned on top of the node representation. It has been suggested that there are two popular methods. The first reformulate is reshaping the issue as a binary classification task in which the link probability between two node embeddings is analyzed[34, 35]. The second identifying similarities between two node embeddings by their similarity[36, 37]. A method that makes use of different measures of similarity assumes that similar nodes are linked to each other, the higher the similarity, the higher the likelihood of association. However, in heterogeneous networks, this assumption is not correct, and on top of that, nodes may tend to find themselves linked to nodes that are not similar to them. For an algorithm to rely on binary classifiers, it is necessary to create a training set, composed of the existing edges as positive edges. A second set, comprised of non-connected edges as negative edges, is also required for the analysis to be able to rely on binary classifiers. Since nodes that are not currently connected can be connected in the future, determining negative links in the training process is challenging. Furthermore, the binary classification process does not give any degree of correlation probability, making it ineffective in large networks such as social networks where a specific number of correlations is proposed.

In this research, we evaluate the link prediction process from a different perspective, we overcome the shortcomings associated with previous methods, by asserting that link prediction is a novelty detection problem for complex networks. This is by assuming that, any snapshot of the network contains links between nodes, and we want to predict links that do not yet exist based on those links, while also considering other information that influences prediction. So, after representing the network nodes in a low-dimensional space, the links are represented based on those representations, and then a One-class classifier is fed these representations as its training set. Then, we test the non-existent links to determine which links are missing or may be present later. We used Local Outlier Factor (LOF)[38] as an algorithm to gauge novelty. Each data point is given a degree by LOF, and degrees are determined by "How close this link is to the neighborhood around it".

2 The Proposed Model

2.1 Notations and Definitions

- **Network**: A network is the set of nodes and edges represented by the graph $G = (V, E)$, where V represents the nodes and E represents the edges. If $u \; and \; v \in V$, and they have a link, then $e(u, v) \in E$. When nodes have auxiliary information, we refer to this network as the attributed network $G_X = (V, E, X)$, where S stands for network auxiliary information related to each node.

A network can be used as a representation of the interactions and relationships between different types of entities, where entities are referred to as nodes or vertices while their interaction or connection is referred to as edges or links.

Doi: https://doi.org/10.54216/FPA.120210
Received: June 17, 2023 Revised: July 03, 2023 Accepted: July 16, 2023

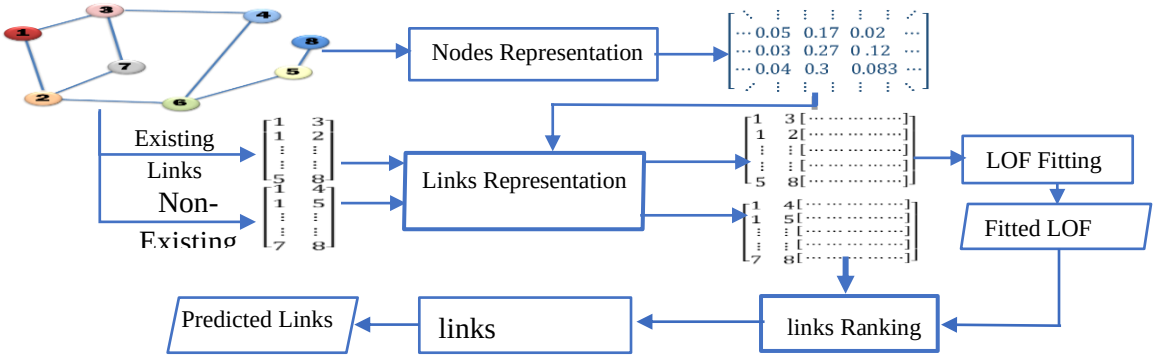

Figure 1: The framework of Link Prediction using Network Representation Learning and Local Outlier Factor

- **Network representation learning**: given a network $G = (V, E)$, is about learning a mapping function $f: V \rightarrow H \in R^{|V|*d}$, which maps each node $v \in V$ to a to a real-valued feature vector $H(v) \in R^d$, where $1 \leq d \ll |V|$is the dimensionality of the node in representation space. As a result of mapping function f, each node in the attributed network G_X is mapped into a low-dimensional representation using the network structure G and the attribute matrix X.
- **Link predication**: link prediction aims to identify the real unobserved links $\acute{E}$ in time snapshot τ that can appear in the network at time $\tau + 1$ given the network $G = (V, E)$ in a time snapshot τ with a set of observed real links E.
- **Novelty detection**: Novelty detection can be defined as the process of determining whether or not the test data are similar or different in some respects from the data that were available during the training process.

2.2 PROPOSED MODEL Stages

The proposed model includes three stages, listed and explained below, as shown in the Figure 1.

- Stage1 (Node Representation): In this stage, all network nodes are represented in a low-dimensional space. Our proposed method is general to all network representation learning models, so we did not specify a model for network nodes' representation.
- Stage2 (Link Representation): Based on the representation of the nodes in the previous stage, the links between the network nodes are also represented, regardless of whether these links exist or not. For each pair of nodes $u, v \in V$,we define a representation function $g: V \times V \rightarrow \tilde{H} \in R^{|V|^{\check{d}}}$ where $\check{d}$ is the vector size representing of the link (u, v) in link representation space. To represent network links, we can use of the equations given by Grover et.al.,[39] which can be listed below:

Average function: $g_1(u, v) = \frac{f(u)+f(v)}{2}$ (1)

Hadamard function: $g_2(u, v) = f(u) * f(v)$ (2)

Weighted-L1 : $g_3(u, v) = |f(u) - f(v)|$ (3)

Weighted-L2 : $g_4(u, v) = |f(u) - f(v)|^2$ (4)

Additionally, we suggest a new operator for representing link (u, v) by merging the pair representations of nodes$\big(f(u), f(v)\big)$:

$g_5(u, v) = f(u) \mid f(v)$ (5)

- Stage3 (Link prediction using LOF): Local Outlier Factor (LOF) is a model for detecting anomalies by considering the density of nodes neighborhood[38]. In this proposed model, LOF measures the novelty of the links of nodes on a numerical scale, which gives an indication of how similar the link under test is to its neighbors (Links that have asymptotic representation). n more detail, the existing links representation is used as training data for fitting a LOF model. In other words, fitting a LOF model means that the model memorizes

Doi: https://doi.org/10.54216/FPA.120210
Received: June 17, 2023 Revised: July 03, 2023 Accepted: July 16, 2023

all the links representations in the training dataset. Next, the novelty measures are calculated by computing the LOF for test datasets (nonexistent links representations). Let U be the set of representations of existing links, and $\tilde{U}$ be the set of representations of nonexistent links, for any element $x \in \tilde{U}$ the LOF score of x can be shown as:

$$LOF(x) = \frac{\sum_{y \in N_k(x)} \frac{lrd_k(x)}{lrd_k(y)}}{k} \quad (6)$$

Where k is the number of neighbors determined by a user-defined parameter. According to k neighbors of an element x, the local reachability density ($lrd_k(x)$) is the inverse of the average reachability distance, $N_k(x)$ consists of set of elements within $k - distance(x)$ of object x.

$$N_k(x) = \{y \in U \backslash x : dest(x,y) < k - distance(x)\} \quad (7)$$

The local reachability density for element x, $lrd_k(x)$, can be calculated as follows:

$$lrd_k(x) = \frac{k}{\sum_{y \in N_k(x)} \text{reachDist}_k(x,y)} \quad (8)$$

The reachability distance of element x with regard to element y, $\text{reachDist}_k(x,y)$, is assigned as:

$$\text{reachDist}_k(x,y) = \max\{k - distance(y), dest(x,y)\} \quad (9)$$

Where $k - distance(x)$ is calculated as Euclidean distance between x and the nearest k-element of the previously memorized elements, Euclidean distance between x and y can be defined as:

$$dest(x,y) = \sqrt{\sum_{i=1}^{\tilde{d}} (x_i - y_i)} \quad (10)$$

Finally, each element is assigned a LOF score based on the average ratio between its local reachability density and its k-nearest neighbors.

LOF(x) is then used to determine whether an element represents a possible link or not, which means that if the LOF(x) value is greater than a user-defined threshold score θ, the element does not represent a potential link, whereas if it is less, then the element might represent a potential link. The new links are predicted by arranging the elements representing the links in terms of the LOF score, so that the elements with a low score represent the links that are predicted to emerge.

3 RESULTS AND DISCUSSION

In this section, we present first a brief description of the real-world datasets used in the experiments, then we introduce the link prediction methods that have been compared with our model to evaluate it. Also, we present the different network representation learning models used within our model in the different evaluation experiments. Finally, a detailed analysis of the experimental results is presented.

Table 1: Statistical presentation of the datasets used in the experimental evaluation.

Dataset	Network size		
	$\|V\|$	$\|E\|$	$\|X\|$
Facebook	22,470	171,002	4714
Wikipedia	11,631	35,324	13183
LastFM	7624	27,806	7842
Twitch	7126	35324	3170

Doi: https://doi.org/10.54216/FPA.120210
Received: June 17, 2023 Revised: July 03, 2023 Accepted: July 16, 2023

3.1 Datasets

To assess the effectiveness and efficiency of the proposed link prediction method, five real-world networks are used; these datasets are listed below and they are available for free on SNAP[40].

- Facebook: The nodes in the network are Facebook pages, and the links between them are reciprocal likes. Features are derived from the page attributives.
- Wikipedia Crocodiles: A node in this network represents a Wikipedia page, whereas edges indicate reciprocal links between Wikipedia pages, and the vertex features describe the presence of nouns within those pages.
- LastFM Asia: In the LastFM network, Asia users represent the nodes, and their reciprocal follower relationships represent the links, while node features are represented by their favorite musicians.
- Twitch England: Nodes in this network are comprised of Twitch users from England and edges in it comprise reciprocal friendship between these users, while node features are concluded from their streaming history.

Specification information of these dataset networks are outlined in Table I.

A. Baseline methods

To evaluate our proposed model, we compare it with some other well-known link prediction methods in the field. These methods are presented next.

3.2 Heuristic -based link prediction baselines

- Common Neighbors (CN))[41]: Similarity can be calculated as the number of common neighbors between each pair of nodes (v, u), The Common Neighbors similarity measure is formulated as:

$$CN_s(v,u) = |\mathcal{N}(v) \cap \mathcal{N}(u)| \tag{11}$$

Where $\mathcal{N}(v)$ shows the set of neighbors of a node v.

- Jaccard Coefficient (JC) [42]: Using the proportion of common neighbours a node pair shares compared to all of their neighbours, this method determines how similar the node pair is to one another. Jaccard Coefficient similarity measure is formulated as:

$$Jc_s(v,u) = \frac{|\mathcal{N}(v) \cap \mathcal{N}(u)|}{|\mathcal{N}(v) \cup \mathcal{N}(u)|} \tag{12}$$

- Resource Allocation (RA) [9]: Common neighbors of pair of nodes (v, u) are regarded as resource transmitters since each one of them provides an entity that enables to convey of resources to the other. RA is defined as follows:

$$RA_s(v,u) = \sum_{w \in \{\mathcal{N}(v) \cap \mathcal{N}(u)\}} \frac{1}{|\mathcal{N}(w)|} \tag{13}$$

- Adamic-Adar index (AA)[8]: Similar to resource allocation, this measure estimates similarity more heavily when common neighbors have lower degrees, both approaches penalize nodes with higher degrees differently. Adamic-Adar similarity measure is formulated as:

$$AA_s(v,u) = \sum_{w \in \{\mathcal{N}(v) \cap \mathcal{N}(u)\}} \frac{1}{\log(|\mathcal{N}(w)|)} \tag{14}$$

- Preferential Attachment (PA) [43]: Speculates that the likelihood of a new connection connecting to v is proportional to $|v|$. As a result, the preferential attachment of v and u is inversely correlated with the number of v and u's neighbours. Preferential Attachment similarity measure is formulated as:

- $$PA_s(v,u) = |\mathcal{N}(v)| \cdot |\mathcal{N}(u)| \tag{15}$$

Doi: https://doi.org/10.54216/FPA.120210
Received: June 17, 2023 Revised: July 03, 2023 Accepted: July 16, 2023

3.3 Network Representation Learning baselines

Network representation learning (NRL) methods can be divided into two main categories, plain representation methods that rely only on the network structure in the network representation, and fusion-based attributed network representation learning methods that rely on combining the network structure features with the node's attributes. Next, we describe a set of network representation methods for each category that are used in the evaluation.

- HOPE[44]: A plain NRL approach, that instead of employing an adjacency matrix, uses generalized singular value decomposition to try to capture higher-order closeness and asymmetrical transitivity.
- NodeSketch[45]: NodeSketch generates node representation in Hamming space. It is a plain NRL technique preserving high-order node proximity via a recursive scheme based on a data-independent hashing and sketching algorithm.
- DeepWalk[46]: The Skip-Gram model approximated by hierarchical SoftMax is used to represent nodes in low dimension space by random walks with fixed transition probabilities.
- Node2vec[39]: It is an extension of a DeepWalk model, it generates node representation by approximating the Skip-Gram model with negative sampling and short random walks to explore node neighbourhoods. An in-out parameter q and a return parameter p govern the random walk's properties.
- TADW[47]: Text-associated DeepWalk (TADW) uses matrix factorization to learn the representation of vertices based on their text properties.
- SINE[48]: In order to learn node representations, the skip-gram model is applied to the content information as a kind of contextual node. Therefore, SINCE predicts content information from the representation vector of each node.
- DANE-WLA[28]: A model for representing attributed networks, which consists of two stages, in the first stage, the attributes of the nodes and their links are integrated, and in the second, the integrated nodes are represented in a low-dimensional space using a deep autoencoder.

3.4 Experimental Settings

In order to evaluate the proposed method and compare it with existing link prediction methods, some steps must be taken. The first step is to process the original network G and obtain an initial set of links for training and another set for testing. In our experiments, we randomly removed 20% of the existing links, which represent the positive test set E^{+}_{test}, and used the remaining links (positive train set E^{+}_{train}) to build the training network G_{train}. Then a negative test set E^{-}_{test} is randomly selected from the non-existent links in the G, which is equal to the positive test group in the number of items ($|E^{-}_{test}| = |E^{+}_{test}|$). To evaluate the proposed model, we used the generic source code presented by Karate Club[40] for all network representation learning baselines except DANE-WLA that was provided in our previous work[28]. Additionally, the node representation dimensionality is tuned, d = 64, in order to obtain optimal performance. To calculate the LOF value for each link, we set the hyperparameter k to 15.

3.4.1 Experimental Result and Evaluation

This section presents the results of our experimental study and our analysis of them. In Subsection 1 we compare the proposed model with various NRL, based on different link representation operators. We then compare our proposed model with traditional similarity-based link prediction methods in Subsection 2.

3.4.2 Link prediction using LOF with different link representation operators

We examined our proposed LOF-based model with seven NRL models with five links representation operators on five datasets to ascertain the most effective operator for representing links based on node representation. For evaluation, we use two metrics that are widely used to measure link prediction method effectiveness:

Precision: Precision is defined as the percentage of actual links l in all links L that the algorithm predicted, precision is expressed as follows

$$Precision = \frac{l}{L} \tag{16}$$

AUC: AUC can be calculated as described in the context of link prediction. If n_1 is the proportion of times the correctly existing links has a higher score than the nonexistent links, and n_2 is the proportion of times both have

Doi: https://doi.org/10.54216/FPA.120210
Received: June 17, 2023 Revised: July 03, 2023 Accepted: July 16, 2023

the same score, then take into account n random experiments of selecting an existing links and a nonexistent links. AUC score can therefore be calculated as[1]:

$$AUC = \frac{n_1 + 0.5 \times n_2}{n} \tag{17}$$

Based on different link representation approaches, we summarized our results in Table 2, through these results we can make the following observation:

Merge operator achieves better performance when used to represent links with various NRL models on the ACU and Precision metrics, followed by the Hadamard operator. Overall, DANE-WLA and SINE models showed the best results with the Merge operator, while SINE model showed the best results with Hadamard operator. These three models gave the best results in general on all datasets. This superiority of Merge operator results is due to its preservation of the true features of the pair of nodes.

The Fusion-based attributed NRL models provided substantially better link predictions than the plain NRL models. As a result, new links can be proposed or missing links can be predicted using all the data provided by the network, not just the network structure.

For each operator, we count DANE-WLA runtime on four datasets. Figure 2 shows that the fitting time for the Merge operator is longer since the link representation vector is twice as large as the representation vector for the other operators. Furthermore, Avg took a longer time than the other operators because of the shape of the numbers representing the link, as in the other three, the values are nearer zero.

Table 2: Comparison of seven popular NRL methods bootstrapped using seven operators of links representation based on AUC and precision metrics of link prediction for four datasets.

Dataset	Model	ACU					Precision				
		Avg	Had	L1	L2	Merge	Avg	Had	L1	L2	Merge
FACEBOOK	NodeSketch	79.37	62.17	64.23	67.03	**83.14**	80.40	66.91	66.39	69.91	**81.24**
	HOPE	70.01	**80.14**	65.43	69.18	78.32	73.86	77.43	68.42	71.67	**72.13**
	DeepWalk	**80.86**	56.27	58.35	56.18	70.12	**81.65**	58.50	61.51	59.94	75.12
	Node2Vec	74.18	**79.68**	73.18	73.98	74.18	80.38	**81.47**	75.13	77.84	74.33
	TADW	81.35	76.27	58.14	56.37	**84.38**	81.65	78.50	65.15	58.94	**83.18**
	SINE	95.14	**95.34**	80.59	77.44	88.35	89.27	90.04	82.37	79.53	**91.54**
	DANE-WLA	77.49	73.13	62.37	58.30	**95.00**	82.38	80.61	73.01	69.21	**92.61**
Wikipedia	NodeSketch	**92.85**	86.17	75.19	80.12	90.05	**86.69**	81.39	70.23	75.60	86.05
	HOPE	84.23	97.14	77.83	74.76	**89.14**	79.21	92.31	73.79	70.56	**80.16**
	DeepWalk	95.18	95.13	95.13	94.12	**96.95**	98.79	88.78	88.91	89.21	**90.62**
	Node2Vec	96.15	95.12	94.69	95.23	**97.01**	90.04	89.22	89.17	89.31	**90.37**
	TADW	97.08	95.28	96.13	93.87	**97.98**	89.63	88.88	89.50	88.26	**90.03**
	SINE	98.17	**99.48**	95.27	94.36	99.21	90.04	**92.32**	88.68	88.31	90.02
	DANE-WLA	97.27	98.06	96.67	95.83	**99.48**	89.80	88.58	90.27	89.74	**92.77**
LastFM	NodeSketch	66.28	66.34	53.73	56.14	**73.19**	**70.52**	58.58	56.46	59.63	55.33
	HOPE	70.14	63.18	61.14	55.89	**76.84**	74.07	75.48	65.50	59.52	**78.84**
	DeepWalk	72.19	73.14	69.12	68.25	**74.38**	74.12	**77.18**	69.15	66.98	75.26
	Node2Vec	67.14	64.29	71.36	55.14	**69.14**	71.15	66.65	61.87	59.81	**76.62**
	TADW	57.14	**79.14**	55.39	52.44	78.42	60.00	**72.23**	57.53	53.03	61.96
	SINE	82.19	**85.37**	85.13	82.14	77.45	74.51	**83.25**	82.47	83.09	81.95
	DANE-WLA	95.18	93.02	88.15	85.78	**98.34**	93.66	93.72	88.15	85.78	**96.68**
TWITCH	NodeSketch	**74.70**	**73.92**	**61.07**	**62.85**	**82.41**	**69.41**	**67.64**	**64.32**	**66.77**	**74.79**

Doi: https://doi.org/10.54216/FPA.120210
Received: June 17, 2023 Revised: July 03, 2023 Accepted: July 16, 2023

	HOPE	76.11	69.15	65.98	60.94	**83.43**	80.42	80.75	70.58	64.89	**83.82**
	DeepWalk	81.57	79.01	72.22	66.38	**92.52**	86.07	89.93	77.01	70.13	**91.23**
	Node2Vec	84.28	84.71	81.96	78.18	**86.83**	86.93	89.18	80.29	77.75	**89.05**
	TADW	78.36	74.40	81.14	64.08	**81.43**	83.00	78.73	72.35	69.33	**89.23**
	SINE	88.17	**96.18**	92.15	91.01	92.38	87.35	**91.13**	89.33	88.79	90.08
	DANE-WLA	90.12	87.60	82.06	80.02	**91.19**	80.24	80.68	80.57	80.02	**82.38**

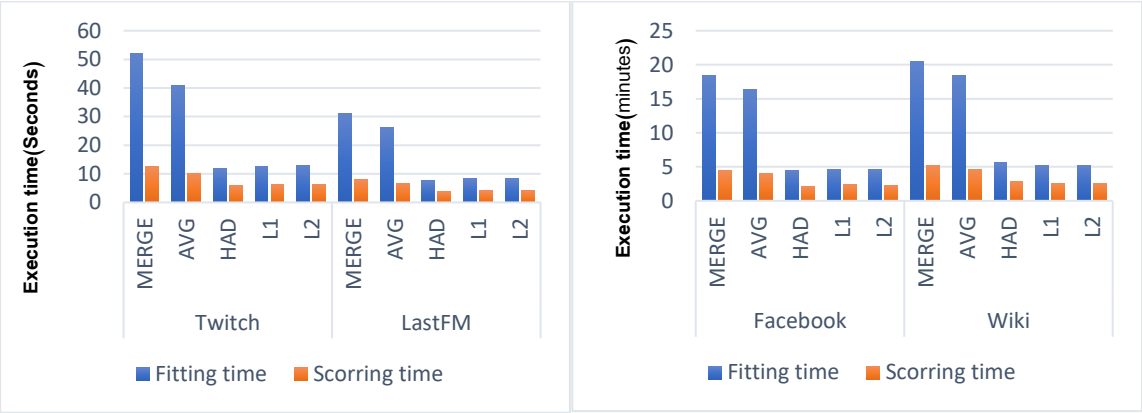

Figure 2: Execution times of DANE-WLA with the different operators on four datasets.

3.4.3 NRL and LOF based Link prediction versus Heuristic -based link prediction.

The results shown in **Error! Reference source not found.**Table 3 compare the effectiveness of the proposed method and the similarity-based method for link prediction. Based on AUC and Precision metrics results, we can clearly see that the proposed method is better than heuristic-based methods. When using DANE-WLA model with the Merge operator, the proposed method outperformed by up to 27.08 in LastFM dataset. Therefore, we conclude that traditional methods based on network structure cannot predict the future links in the network optimally. While LOF and NRL based methods has a higher effectiveness and predicts more reliable links.

Table 3: Comparison with popular heuristic-based baselines based on AUC and precision metrics of link prediction in four datasets.

Model	AUC				Precision			
	FACE	WIKI	LastFM	TWITCH	FACE	WIKI	LastFM	TWITCH
CN	81.13	89.00	69.38	81.18	86.64	85.44	68.37	84.57
JC	88.88	87.65	70.13	85.75	81.62	87.90	66.35	81.07
RA	83.73	90.17	68.19	83.16	80.89	89.61	69.43	79.58
AA	82.98	91.92	71.26	81.63	77.16	89.67	62.66	76.98
PA	79.15	91.98	64.25	78.97	86.64	85.44	68.37	84.57
LOF+SINE	**95.34**	**99.48**	85.37	**96.18**	90.04	92.32	83.25	**91.13**
LOF+ DANE-WLA	95.00	**99.48**	**98.34**	91.19	**92.61**	**92.77**	**96.68**	82.38

In the following experiments, we will conduct multiple experiments to adjust some parameters, so we will suffice to apply them to LastFM and Twitch datasets, since they are smaller and take less time to adjust, as shown in Figure 2. As a result, we believe it provides sufficient indicators for determining the optimal parameters.

Doi: https://doi.org/10.54216/FPA.120210
Received: June 17, 2023 Revised: July 03, 2023 Accepted: July 16, 2023

3.4.4 The k-nearest neighbors Tuning

The local outlier factor method compares a point's density to its neighbor's relative densities. Our goal with this experiment is to determine what is the optimal number of neighboring points in the calculation of the LOF score in order to improve link prediction. We use the DANE-WLA model to represent nodes in a low-dimensional space with various operators to represent links. with k-nearest neighbor points starting at 5 points and increasing by 5 points in each experiment. The k-nearest neighbors points start from 5 points and increase by 5 points in each trial for the fitting of the Local Outlier Factor and the calculation of the AUC. In all the different operators for representing links, as shown in Figure 3, the LOF works best when the k-nearest neighbors are set between 15 and 20 points.

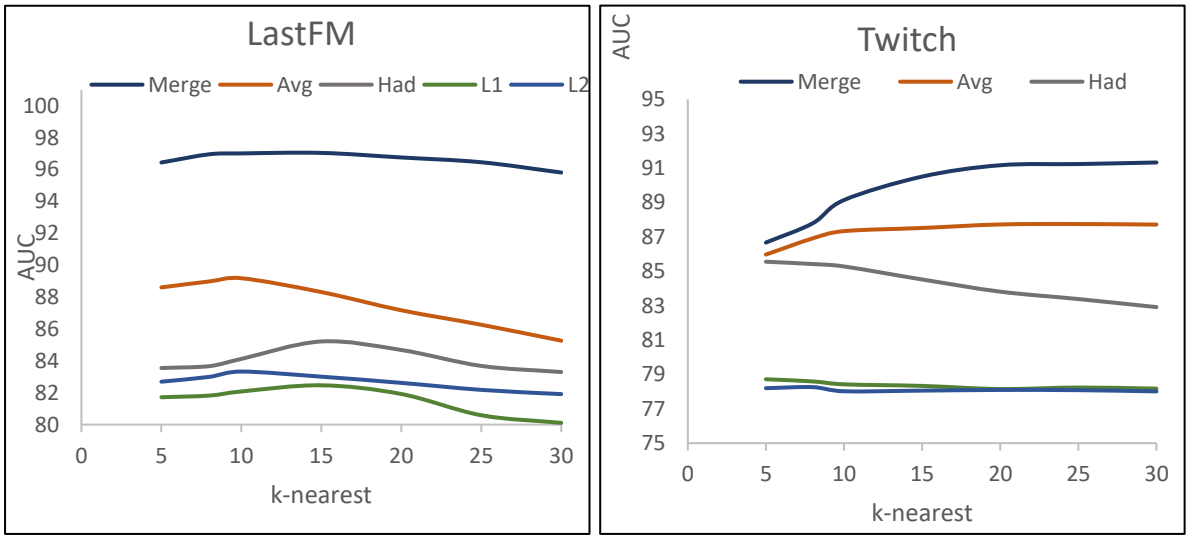

Figure 3: Improvement in AUC of tuning of k-nearest neighbors.

3.4.5 Representation Dimensionality

We evaluate the effect of representation dimensionality on DANE-WLA model performance by modifying representation space and computing the AUC of all operators for $d \in \{32,64,128,256\}$.Results in Figure 4 illustrate an improvement in performance as d increases. We also observe that the Merge operator experiences a low-performance impact when representation space increases, meaning that the Merge operator still produces good results even as the node and therefore link representation dimensions are reduced.

Figure 4: Comparisons of representation dimensions on the LastFM and Twitch datasets with the different operators.

Doi: https://doi.org/10.54216/FPA.120210
Received: June 17, 2023 Revised: July 03, 2023 Accepted: July 16, 2023

3.4.6 Study the proportion of predicted links

This section examines the quality of performance as compared to the predicted percentage of deleted links from the network, in order to estimate the most appropriate percentage. We rank the predicted links according to the calculated LOF value associated with each link.

From Figure 5, we notice that the merge coefficient gives a more stable performance in both datasets. For the LastFM dataset, up to 60%, the percentage of correct links remains higher than 98%, and this is 15% of the links on which LOF was trained. While in the Twitch dataset, the results were lower and the decline began early, but up to 50% of the links, the percentage of correct predicted links was higher than 93%. We observe that it begins less successfully for the operators AVG and Hadamard and then improves after that, whereas for the operators L1 and L2, up to 10% of the predicted links are more stable and then start to fall dramatically after that.

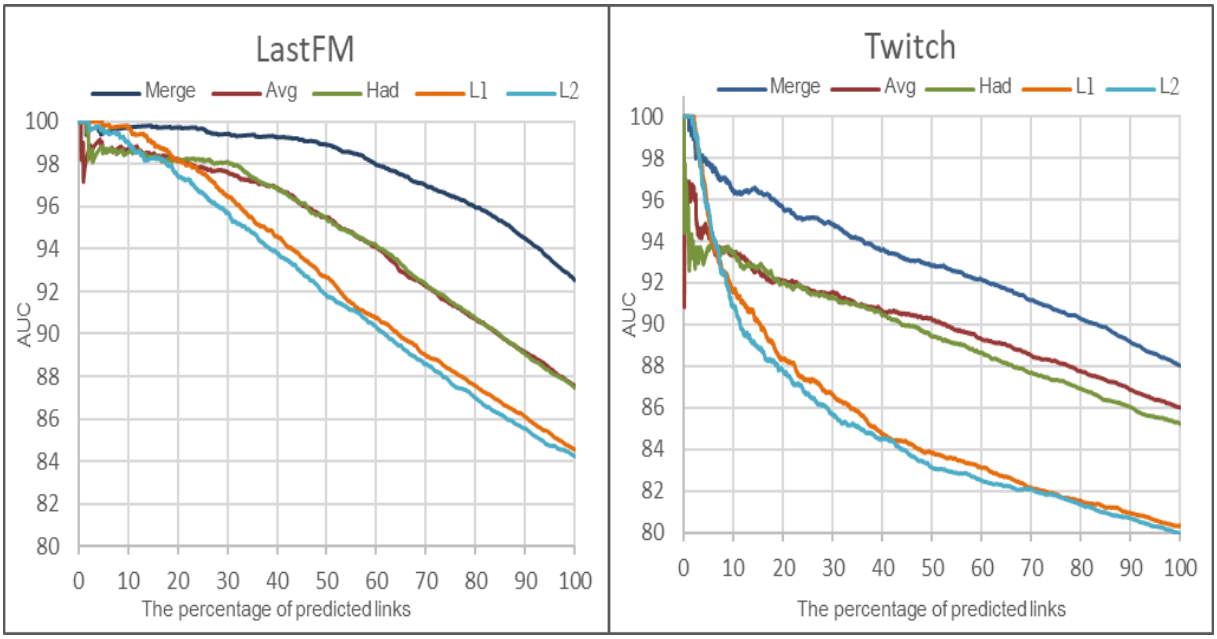

Figure 5: Illustration of the quality of performance in response to the predicted links on LastFM and Twitch datasets.

4. Conclusion

In this paper, we proposed a simple yet effective approach to improve link prediction in complex networks. The problem of link prediction is framed as a problem of novelty detection in this approach. In order to prove this view's validity, we proposed that links can be predicted along three stages: represent network nodes using NRL model, representing the links in low-dimensional spaces, and propose new links based on the LOF model that fitted by the representations of existing links. Based on our results, this approach outperforms baselines for heuristic-based link prediction when applied to different network representation learning models. Also, we noticed that this approach provides better results with the Fusion-based attributed NRL models as compared with the plain NRL models. In light of this, it is evident that the network structure is not enough to identify missing links or suggest new links. Lastly, we argue that the proposed method provides new opportunities for studying link prediction in real-world networks. This argument is driven by both the widespread use of network representation learning to analyse real-world networks, as well as the diversity of novelty detection methods in big data.

Doi: https://doi.org/10.54216/FPA.120210
Received: June 17, 2023 Revised: July 03, 2023 Accepted: July 16, 2023

References

[1] Kumar, A., Singh, S.S., Singh, K., Biswas, B.J.P.A.S.M., Applications, i. Link prediction techniques, applications, and performance: A survey. 2020, 553, 124289.

[2] Daud, N.N., Ab Hamid, S.H., Saadoon, M., Sahran, F., Anuar, N.B.J.J.o.N., Applications, C. Applications of link prediction in social networks: A review. 2020, 166, 102716.

[3] Liben-Nowell, D., Kleinberg, J.J.J.o.t.A.s.f.i.s., technology. The link-prediction problem for social networks. 2007, 58, 1019-31.

[4] Aiello, L.M., Barrat, A., Schifanella, R., Cattuto, C., Markines, B., Menczer, F.J.A.T.o.t.W. Friendship prediction and homophily in social media. 2012, 6, 1-33.

[5] Wohlfarth, T., Ichise, R. Semantic and event-based approach for link prediction. In: International Conference on Practical Aspects of Knowledge Management, Springer, 2008, pp. 50-61.

[6] Chuan, P.M., Son, L.H., Ali, M., Khang, T.D., Huong, L.T., Dey, N.J.A.I. Link prediction in co-authorship networks based on hybrid content similarity metric. 2018, 48, 2470-86.

[7] Eirinaki, M., Gao, J., Varlamis, I., Tserpes, K.J.F.G.C.S. Recommender systems for large-scale social networks: A review of challenges and solutions. Elsevier, 2018, Vol. 78, pp. 413-8.

[8] Adamic, L.A., Adar, E.J.S.n. Friends and neighbors on the web. 2003, 25, 211-30.

[9] Zhou, T., Lü, L., Zhang, Y.-C.J.T.E.P.J.B. Predicting missing links via local information. 2009, 71, 623-30.

[10] Zeng, S.J.P.A.S.M., Applications, i. Link prediction based on local information considering preferential attachment. 2016, 443, 537-42.

[11] Muniz, C.P., Goldschmidt, R., Choren, R.J.K.-B.S. Combining contextual, temporal and topological information for unsupervised link prediction in social networks. 2018, 156, 129-37.

[12] Javari, A., Qiu, H., Barzegaran, E., Jalili, M., Chang, K.C.-C. Statistical link label modeling for sign prediction: Smoothing sparsity by joining local and global information. In: 2017 IEEE International Conference on Data Mining (ICDM), IEEE, 2017, pp. 1039-44.

[13] Das, S., Das, S.K. A probabilistic link prediction model in time-varying social networks. In: 2017 IEEE International Conference on Communications (ICC), IEEE, 2017, pp. 1-6.

[14] Bastami, E., Mahabadi, A., Taghizadeh, E.J.S., computation, e. A gravitation-based link prediction approach in social networks. 2019, 44, 176-86.

[15] Benchettara, N., Kanawati, R., Rouveirol, C. Supervised machine learning applied to link prediction in bipartite social networks. In: 2010 international conference on advances in social networks analysis and mining, IEEE, 2010, pp. 326-30.

[16] Wang, P., Xu, B., Wu, Y., Zhou, X.J.S.C.I.S. Link prediction in social networks: the state-of-the-art. 2015, 58, 1-38.

[17] Zhang, D., Yin, J., Zhu, X., Zhang, C. Network representation learning: A survey. IEEE transactions on Big Data. 2018, 6, 3-28.

[18] Makarov, I., Kiselev, D., Nikitinsky, N., Subelj, L., Elzeki, O.M., Shams, M., et al. Survey on graph embeddings and their applications to machine learning problems on graphs. PeerJ Computer Science. 2021.

[19] Donnat, C., Zitnik, M., Hallac, D., Leskovec, J. Learning structural node embeddings via diffusion wavelets. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1320-9.

[20] Ribeiro, L.F., Saverese, P.H., Figueiredo, D.R. struc2vec: Learning node representations from structural identity. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 385-94.

[21] Hu, B., Wang, H., Yu, X., Yuan, W., He, T. Sparse network embedding for community detection and sign prediction in signed social networks. Journal of Ambient Intelligence and Humanized Computing. 2019, 10, 175-86.

[22] Sun, H., He, F., Huang, J., Sun, Y., Li, Y., Wang, C., et al. Network embedding for community detection in attributed networks. ACM Transactions on Knowledge Discovery from Data (TKDD). 2020, 14, 1-25.

[23] Wang, Z., Chen, C., Li, W. Predictive network representation learning for link prediction. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, 2017, pp. 969-72.

[24] Cai, L., Li, J., Wang, J., Ji, S.J.I.T.o.P.A., Intelligence, M. Line graph neural networks for link prediction. 2021.

[25] Li, G., Li, Q., Liu, J., Zhu, Y., Zhong, M. FANE: A Fusion-Based Attributed Network Embedding Framework. In: Web and Big Data: 5th International Joint Conference, APWeb-WAIM 2021, Guangzhou, China, August 23–25, 2021, Proceedings, Part I 5, Springer, 2021, pp. 53-60.

[26] Liu, T., Yin, J., Qin, Q.J.A.S. MFHE: Multi-View Fusion-Based Heterogeneous Information Network Embedding. 2022, 12, 8218.

Doi: https://doi.org/10.54216/FPA.120210
Received: June 17, 2023 Revised: July 03, 2023 Accepted: July 16, 2023

[27] Yang, H., Chen, L., Pan, S., Wang, H., Zhang, P. Discrete embedding for attributed graphs. Pattern Recognition. 2022, 123, 108368.
[28] Al-Furas, A.T., Alrahmawy, M.F., Al-Adrousy, W.M., Elmougy, S.J.I.A. Deep Attributed Network Embedding via Weisfeiler-Lehman and Autoencoder. 2022, 10, 61342-53.
[29] Pan, Y., Zou, J., Qiu, J., Wang, S., Hu, G., Pan, Z. Joint network embedding of network structure and node attributes via deep autoencoder. Neurocomputing. 2022, 468, 198-210.
[30] Hong, R., He, Y., Wu, L., Ge, Y., Wu, X. Deep attributed network embedding by preserving structure and attribute information. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2019.
[31] Xia, F., Sun, K., Yu, S., Aziz, A., Wan, L., Pan, S., et al. Graph learning: A survey. IEEE Transactions on Artificial Intelligence. 2021, 2, 109-27.
[32] Zhou, J., Liu, L., Wei, W., Fan, J.J.A.C.S. Network representation learning: from preprocessing, feature extraction to node embedding. 2022, 55, 1-35.
[33] Zhang, M., Chen, Y.J.A.i.n.i.p.s. Link prediction based on graph neural networks. 2018, 31.
[34] Saxena, A., Fletcher, G., Pechenizkiy, M.J.E.D.S. NodeSim: node similarity based network embedding for diverse link prediction. 2022, 11, 24.
[35] Pio-Lopez, L., Valdeolivas, A., Tichit, L., Remy, É., Baudot, A.J.S.R. MultiVERSE: a multiplex and multiplex-heterogeneous network embedding approach. 2021, 11, 1-20.
[36] Zhang, H., Qiu, L., Yi, L., Song, Y. Scalable multiplex network embedding. In: IJCAI, 2018, Vol. 18, pp. 3082-8.
[37] Zhang, C., Shang, K.-K., Qiao, J.J.C. Adaptive similarity function with structural features of network embedding for missing link prediction. 2021, 2021.
[38] Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J. LOF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 93-104.
[39] 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, 2016, pp. 855-64.
[40] Rozemberczki, B., Kiss, O., Sarkar, R. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 3125-32.
[41] Chen, J., Geyer, W., Dugan, C., Muller, M., Guy, I. Make new friends, but keep the old: recommending people on social networking sites. In: Proceedings of the SIGCHI conference on human factors in computing systems, 2009, pp. 201-10.
[42] Jaccard, P.J.B.S.V.S.N. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. 1901, 37, 547-79.
[43] Barabâsi, A.-L., Jeong, H., Néda, Z., Ravasz, E., Schubert, A., Vicsek, T.J.P.A.S.m., et al. Evolution of the social network of scientific collaborations. 2002, 311, 590-614.
[44] Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W. Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, 2016, pp. 1105-14.
[45] Yang, D., Rosso, P., Li, B., Cudre-Mauroux, P. Nodesketch: Highly-efficient graph embeddings via recursive sketching. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1162-72.
[46] 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, 2014, pp. 701-10.
[47] Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E.Y. Network representation learning with rich text information. In: IJCAI, 2015, Vol. 2015, pp. 2111-7.
[48] Zhang, D., Yin, J., Zhu, X., Zhang, C. SINE: scalable incomplete network embedding. In: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 737-46.

Doi: https://doi.org/10.54216/FPA.120210
Received: June 17, 2023 Revised: July 03, 2023 Accepted: July 16, 2023