Applied Intelligence
https://doi.org/10.1007/s10489-019-01413-8

Level-2 node clustering coefficient-based link prediction

Ajay Kumar[1] · **Shashank Sheshar Singh**[1] · **Kuldeep Singh**[1] · **Bhaskar Biswas**[1]

Abstract

Link prediction finds missing links in static networks or future (or new) links in dynamic networks. Its study is crucial to the analysis of the evolution of networks. In the last decade, lots of works have been presented on link prediction in social networks. Link prediction has been playing a pivotal role in course of analyzing complex networks including social networks, biological networks, etc. In this work, we propose a new approach to link prediction based on level-2 node clustering coefficient. This approach defines the notion of level-2 common node and its corresponding clustering coefficient that extracts clustering information of level-2 common neighbors of the seed node pair and computes the similarity score based on this information. We performed the simulation of the existing methods (i.e. three classical methods viz., common neighbors, resource allocation, preferential attachment, clustering coefficient-based methods (CCLP and NLC), local naive based common neighbor (LNBCN), Cannistrai-Alanis-Ravai (CAR), recent Node2vec method) and the proposed method over 11 real-world network datasets. Accuracy is estimated in terms of four well-known single point summary statistics viz., area under the ROC curve (AUROC), area under the precision-recall curve (AUPR), average precision and recall. The comprehensive experiment on four metric and 11 datasets show the better performance results of the proposed method. The time complexity of the proposed method is also given and is of the order of time required by the existing method CCLP. The statistical test (The Friedman Test) justifies that the proposed method is significantly different from the existing methods in the paper.

Keywords Link prediction · Level-2 node clustering coefficient · Similarity measures · Social network

1 Introduction

A social network is a standard approach to model communication in a group or community of persons. Such networks can be represented as a graphical model in which a node maps to a person or social entity and an edge maps to an association or collaboration between corresponding persons or social entities. The relationships among individuals are continuously changing, so addition and/or deletion of several nodes and edges take place. It results in social networks to be highly dynamic and complex. Lots of issues arise when we study about a social network; some of which are changing association patterns over time, factors that drive those associations, and the effects of those associations to other nodes. Here, we address a specific problem termed as Link Prediction. Formally, the link prediction is stated as [1]: suppose a graph $G_{t_0-t_1}(V, E)$ represents a snapshot of a network during time interval $[t_0, t_1]$ and $E_{t_0-t_1}$, a set of edges present in that snapshot. The task of link prediction is to find set of edges $E_{t'_0-t'_1}$ during the time interval $[t'_0, t'_1]$ where $[t_0, t_1] \leq [t'_0, t'_1]$.

Link prediction idea is useful in several domains of application. Examples include automatic hyperlink creation [2], website hyper-link prediction [3] in the internet and web science domain and friend recommendation on Facebook. Building a recommendation system in e-commerce is an essential task that uses link prediction as a basic building block [4]. In Bioinformatics, protein-protein interactions (PPI) have also been implemented using link prediction [5]. In security concern areas, link prediction is used to find hidden links among terrorists and their organizations.

✉ Ajay Kumar
ajayk.rs.cse16@iitbhu.ac.in

Shashank Sheshar Singh
shashankss.rs.cse16@iitbhu.ac.in

Kuldeep Singh
kuldeep.rs.cse13@iitbhu.ac.in

Bhaskar Biswas
bhaskar.cse@iitbhu.ac.in

[1] Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, 221–005, India

Published online: 04 February 2019

Lots of approaches to link prediction have been proposed in the literature from the last few decades. Similarity-based approaches are the most common ones especially structural similarities which are simple and efficient to compute. For each pair of nodes in the network, an index score is computed using structural information available around the node pair which represents the structural similarity between them. Structural similarity-based methods extract information about the underlying structures range from local to global including quasi-local (more than local and less than global information). Some of which are the neighborhood-based methods (i.e., Common Neighbors [6], Jaccard [7], Adamic/Adar [8], Resource Allocation [9], Preferential attachment [10], etc.) that use local information, path-based methods (i.e., Katz index [11], Inverse path distance [1] Average commute time [12], PageRank [13], Leicht-Holme-Newman Index [14], Random walk with restart [15], etc.) which explore global information of the underlying network.

Difference from existing works Recently some works regarding the effect of clustering coefficient on link prediction task have been introduced. Wu et al. [16] introduces a new measure (CCLP) that consider the clustering ability of common neighbors of a given node pair to assign a similarity score. Liu et al. [17] work show the effect of low/high degree nodes on the clustering ability of a local path and penalizes the longer one to reduce their contribution to the similarity score computation. Further, Wu et al. [18] included link clustering information in addition to node clustering information in similarity calculation, this is called node and link clustering based link prediction (NLC). But, if no common neighbor exists between the seed node and the common node, the contribution of link clustering becomes zero and the NLC method generates to the CCLP. Our work introduces the notion of level-2 common neighbors and level-2 clustering coefficients of a pair of seed nodes. The proposed work explores a large area of the network in the form of clustering ability of level-2 common neighbors of the seed node pair which we called level-2 clustering coefficient. Y. Liu et al. explore networks in the path of length up to three (3) where the proposed work does not have such restriction and it expands the search area with the help of level-1 common neighbors as shown in Fig. 2.

Contribution The major contributions of this work is as follows

- We introduce the notion of level-2 common neighbors and level-2 node clustering coefficients that explore large information of networks compared to the level-1 common neighbors and their corresponding clustering coefficients.
- Based on the above concept, we proposed a novel framework and its corresponding algorithm viz. level-2 node clustering coefficient for link prediction.
- We also contribute its computational procedure and complexity in the paper.
- The experiment results on real network datasets show the superiority of our algorithm against the state-of-the-art-algorithms.

Organization Section 2 presents related work on link prediction. The proposed work with the algorithm has been explained in Section 3. Section 4 discusses experimental study consisting of an evaluation strategy and results of several methods against real network datasets. Finally, Section 5 concludes our work.

2 Literature review

M. E. J. Newman presented a paper on link prediction on collaboration networks in Physics and Biology [6]. In such networks, two authors are considered to be connected if they have at least one paper co-authored by them simultaneously. In the empirical study, the author demonstrated that the likelihood of a pair of researchers teaming up increments with the numbers of different colleagues they have in mutual relation, and the likelihood of a specific researcher acquiring new partners increments with the number of his past teammates. The outcomes give experimental proof in favor of formerly guessed mechanisms for clustering and power-law degree distributions in networks. Later, Liben-Nowell et al. [1] proposed a link prediction models explicitly for a social network. Each node in the network corresponds to a person or an entity, and a link between two nodes shows the interaction between them. The learning paradigm in this environment can be used to extract the similarities between two nodes by several similarity metrics. Ranks are assigned to each pair of nodes based on these similarities, then higher ranked node pairs are designated as predicted links. Further, Hasan et al. [19] expanded this work and demonstrated that there is a significant increase in prediction results when additional topological information about the network is available. They considered different similarity measures as features and performed binary classification task using a supervised learning approach, which is similar to link prediction in their framework. In the relational context [20–22] and in the internet domain [23], refinement graph [24] is constructed from relational database for useful feature generation and binary classification (link prediction) is performed using a regression model. Their framework can acknowledge any relational dataset where there is a relation among objects. In such frameworks, modeling paradigms like probabilistic relational models [25], graphical models

[26], and stochastic relational models [5, 27, 28] have been used. The upsides of these methodologies incorporate the genericity and simplicity where the model can integrate attributes of the entities. On the downside, they are normally intricate and able to contain the excessive number of parameters; a large portion of which may be complex to the user.

The graph embedding is considered as a dimensionality reduction technique in which higher D dimensional nodes in the graphs are mapped to a lower d ($d << D$) dimensional representation space by preserving the node neighborhood structures. Recently, some graph embedding techniques [29–33] have been proposed and applied successfully in link prediction problem. The Laplacian eigenmaps [29] and Logically linear embedding (LLE) [33] are examples based on the simple notion of embedding. such embedding techniques are quite complex in nature and face scalability issues. To tackle the scalability issue, graph embedding techniques have leveraged the sparsity of real-world networks. For example, DeepWalk [32] extracts local information of truncated random walk and embeds the nodes in representation space by considering the walk as a sentence in the language model [34, 35]. It preserves higher order proximity by maximizing the probability of co-occurrence of random walk of length 2k+1 (previous and next k nodes centered at a given node). Node2vec [30] also uses a random walk to preserves higher order proximity but it is biased which is a trade-off between the breadth-first search (BFS) and depth-first search (DFS). The experimental results show that the node2vec performs better than the Deepwalk.

Continuous growing size of social networks such as Myspace, Facebook, Linkedin, Flickr, etc., has shown to be one of the key challenges in link prediction. Prior existing methodologies may not be implemented to such networks because of continuous evolving nature and their huge size, so some other direct methodologies are required to address these issues. As an example, Tylenda et al. [36] show that the timestamps of previous affiliations (that expressly use the genealogy of interactions) can be used to enhance the performance of link prediction. Song et al. [37] considered a social network consisting of around 2 million nodes and 90 million edges and compute similarity measures among these nodes using matrix factorization. Recently, authors Acar et al. [38] implemented tensor as the extension of matrix factorization which is more richer and higher-order models.

This paragraph gives an overview of clustering-based link prediction. Huang [39] presented a paper on graph topology based link prediction where generalized clustering coefficient is used as a predictive parameter. The author introduces a cycle formation model which shows the relationship between link occurrence probability and its ability to form different length cycles. Further, Liu et al. [17] proposed degree related clustering coefficient to quantify the clustering ability of nodes. They applied the same to paths of shorter lengths and introduced a new index Degree related Clustering ability Path (DCP). They performed the degree of robustness (DR) test for their index and showed that missing links have a small effect on the index. Recently, Wu et al. [16] extracted triangle structure information in the form of node clustering coefficient of common neighbors. Their experiments on several real datasets show comparable results to CAR index in [40]. The same concept of clustering coefficient is also introduced in the work presented by Wu et al. [18]. Authors introduce both node and link clustering information in their work [18]. Their experiments on small, middle and large network datasets showed better performance results against existing methods, especially on middle and large network datasets.

Clearly, node and link clustering information play an essential role in the evolution of complex networks. The above paragraph shows some research works on link prediction using this property and still more efforts need to be applied. Our work is also an effort in this direction.

3 Proposed work

Evidences [41, 42] suggest that many real networks demonstrate consistent topological features across different domains viz., small-world [43, 44], clustering, and scale-free [45]. Their corresponding basic measures are path length, clustering coefficient, and degree distribution. Most empirically observed networks' behavior resembles small-worlds in which any two nodes can find each other in a few steps even if the network is large enough, i.e., the diameter increases logarithmically with the number of nodes in such networks. Small-world networks are highly clustered and characterized by the clustering coefficient. Our work focuses on clustering coefficient measure which is extended up to next level. This work exploits more local information as level-2 common neighbors and clustering properties of such nodes in the network.

This work relaxes the notion of CAR index where only common neighbors and link information among them (i.e. local communities) [40] are considered and extends the notion of clustering information of the CCLP index. Our work considers level-1, level-2, and level-3 link information (also higher level links in some cases) to extract level-2 triangles (clustering information). It selects level-2 and level-3 link information to a greater extent in the triangle formation as compared to level-1 links. The proposed method explores a large portion (global to some extent) of the underlying network (Figs. 1 and 2).

How it extracts global information (to some extent) While selecting level-2 common neighbors (CN^2), there exist

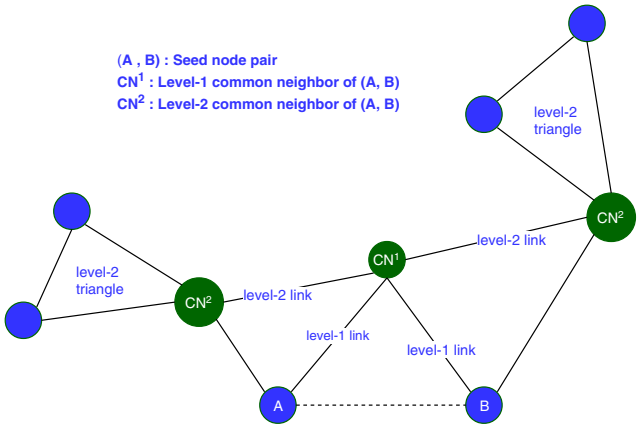


Fig. 1 Notion of level-2 clustering coefficient

some possibilities that level-2 common neighbors are also treated as level-1 common neighbors as shown in Fig. 2. In such cases, level-2 common neighbors are extended to next level in the network and continue until further identification of level-2 common neighbors as level-1 common neighbors.

Definition 1 (**Clustering Coefficient**) It is a measure of the degree to which nodes of a graph tends to be clustered. In graph theory, the clustering coefficient of a node represents its neighbor's tendency to become a clique or complete graph. Mathematically, this measure [42] is expressed as

$$C(i) = \frac{2|\{e_{jk}:v_j,v_k \in N(i), e_{jk} \in E\}|}{k_i(k_i-1)} \tag{1}$$

where k_i is the degree of the node, i and N(i) is immediate neighbors of i. We refer this measure as Level-1 clustering coefficient, based on which Wu et al. [16] presented a paper on link prediction.

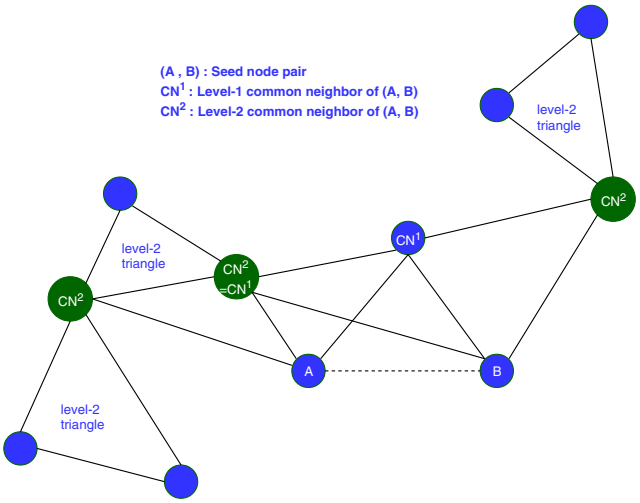


Fig. 2 Expanding local to global architecture

Definition 2 (**Level-2 node clustering coefficient**) We extend the definition of node clustering coefficient [16] to next level named level-2 node clustering coefficient which exploits level-2 common neighbors and their clustering information for every pair of nonexistent nodes in the network.

Figure 3 represents the best explanation of level-2 node clustering coefficient. For the seed node pair (A, B), level-1 common neighbors or simply common neighbors (CNs) are C, D, and E shown in the right upper part of the figure. Further, all those pairs are selected in which the first node is either A or B (one of the seed node pair) and the second node is one of the level-1 common neighbors. For all such pairs, level-2 common neighbors are computed based on common nodes of their respective pair. Based on this definition nodes F and G are level-2 common neighbors shown in the right bottom part of the figure. Now, the total number of triangles passing through each level-2 common neighbor is computed and summed over all such neighbors to find level-2 clustering coefficient of the seed node pair.

Link prediction based on Level-2 node clustering coefficient Our work focuses on level-2 clustering coefficient that explores clustering information of level-2 common neighbors which is more informative than the clustering coefficient used in [16]. We extend the notion of clustering coefficient of a node the next level (level-2) in the network. We compute the level-2 node clustering coefficient according to the (2)

$$\begin{aligned} CCLP^2_{(A,B)} &= \sum_{CN^2_A \in \Gamma(A)\cap\Gamma(CN^1)} CC(CN^2_A) \\ &+ \sum_{CN^2_B \in \Gamma(CN^1)\cap\Gamma(B)} CC(CN^2_B) \\ &= \sum_{CN^2 \in (\Gamma(A)\cap\Gamma(CN^1))||(\Gamma(CN^1)\cap\Gamma(B))} CC(CN^2) \end{aligned} \tag{2}$$

where $CC(CN^2)$ is having the usual definition of node (i.e. CN^2) clustering coefficient value and is computed using the (1). CN^2_A is the level-2 common neighbor corresponding to node A and the common node of the pair (A, B). CN^1 is the level-1 common neighbor defined in the literature and is computed as

$$CN^1 = \Gamma(A) \cap \Gamma(B).$$

The pseudo code of the proposed algorithm is presented in Algorithm 1.

Algorithm description For a given simple undirected graph, the algorithm finds top-L missing links. The main crux of the algorithm is to find level-2 common neighbors from which level-2 clustering coefficient can be calculated.

For each pair of nodes (seed node pair (A, B)) having no edge between them, the algorithm finds all common

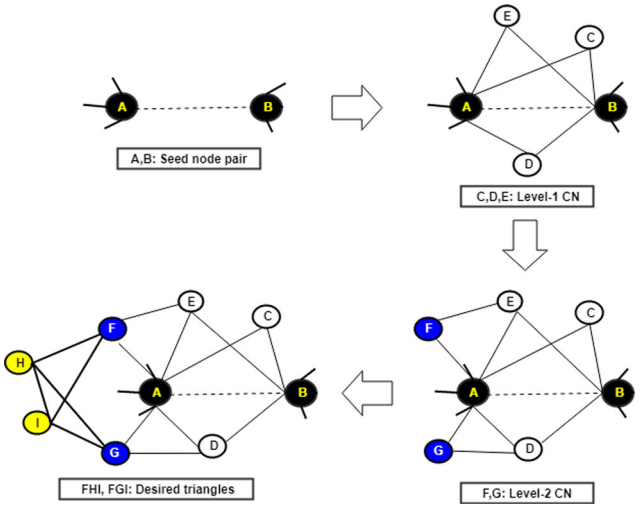


Fig. 3 Computing level-2 node clustering coefficient

neighbors (level-1 CNs) [Line: 1-2]. Level-2 common neighbors are then computed for all those node pairs (p_i, q_i) in which first node p_i belongs to $\{A, B\}$ while second node in level-1 common neighbors of (A, B) [Line: 3-4]. Now for all nodes in level-2 common neighbors, clustering coefficient values are computed and added to get final similarity score for the seed node pair (A, B) [Line: 5-6]. Once scores of all non-existent node pairs have been computed, the next step [Line: 7] arranges them in descending order, and finally, top-L node pairs are returned as predicted links [Line: 8].

4 Experimental study

4.1 Evaluation metrics

The link prediction problem is treated as a binary classification task [19] so most of the evaluation metrics of any binary classification task can be used in link prediction evaluation. The evaluation of a binary classification task having two classes can be represented as a confusion matrix [46].

In the confusion matrix,

- True Positive (TP): positive data item predicted as positive
- True Negative (TN): negative data item predicted as negative
- False Positive (FP): negative data item predicted as positive
- False Negative (FN): positive data item predicted as negative

Based on the confusion matrix, several metrics can be derived as follows [46].

True Positive Rate (TPR)/Recall/Sensitivity

$$TPR = \frac{\#TP}{\#TP + \#FN} \tag{3}$$

False Positive Rate (FPR)

$$FPR = \frac{\#FP}{\#FP + \#TN} \tag{4}$$

True Negative Rate (TNR)/Specificity

$$TNR = \frac{\#TN}{\#TN + \#FP} \tag{5}$$

$$\textbf{Precision} = \frac{\#TP}{\#TP + \#FP} \tag{6}$$

Our approach is evaluated on four metrics viz., Area under the ROC curve (AUROC) [47, 48], Area under the precision-recall curve (AUPR) [49], Average precision [46] and *Recall* [46].

Area under the Receiver Operating Characteristics Curve (AUROC) A roc curve is a plot between the true positive rate (sensitivity) on the y-axis and the false positive rate (1-specificity) on the x-axis. The true positive rate and false positive rate can be evaluated using (3) and (4) respectively. The area under the roc curve [48] is a single point summary statistics between 0 and 1 that can be computed using the trapezoidal rule which sums all the trapezoids under the curve. The value of the auroc of a predictor should be greater than 0.5 which is the value of a random predictor, i.e., higher the value of auroc better the performance of the predictor.

Area under the precision-recall curve (AUPR) The precision-recall curve is more useful and informative when applied to binary classification on imbalanced datasets [50]. So we have also considered the area under the precision-recall curve (AUPR). This value is computed based on the precision-recall curve which is a plot between the precision values on the y-axis and the recall values on the x-axis. The precision and recall values can be computed using (6) and (3) respectively.

Average precision This metric is also a single point summary value which is computed based on varying threshold[1] values. The average precision value is equal to the precision averaged over all values of recall between 0 and 1 i.e.,

$$AveragePrecision = \int_{r=0}^{1} p(r)dr \tag{7}$$

[1] https://sanchom.wordpress.com/tag/average-precision/

Recall This metric[2] intuitively finds all positive samples (existence of links in this case) in the data and equates to the metric given in (3).

4.2 Datasets

This work used 11 network datasets from various fields to study the performance of our approach. Macaque[3] [51]: is a biological network of cerebral cortex of Rhesus macaque. Football[4] [52]: American football games network played between Division IA colleges during regular season Fall 2000. Celegansneural [42]: A neural network of C. Elegans compiled by D. Watts and S. Strogatz in which each node refers a neuron and, an edge joins two neurons if they are connected by either a synapse or a gap junction. USAir97[5] is an airline network of US where a node represents an airport and an edge shows the connectivity between two airports. Political blogs [53] is a directed network of hyperlinks in political blogs leaning towards the conservatives and the democrats preceding the US election 2004. Yeast[6] [54] is also a biological network of proteins in a cell where a node represents a protein and edge denotes the interaction between two proteins. Amazon web graph [55] is an informational network of web pages of Amazon.com and its sister companies. Power grid [42] is an undirected and unweighted network of power grid located in western states of the United States. ca-GrQc,[7] ca-HepTh, and ca-HepPh are collaboration networks of arXiv General Relativity, High Energy Physics Theory, and High Energy Physics respectively.

Table 1 shows some basic topological properties of the considered networks datasets. N and E are the total numbers of nodes and edges of the networks respectively. D represents node pairs average shortest distance, K, the average degree and C, the average clustering coefficient of the network.

Table 1 Topological informations of real-world network datasets

Datasets	N	E	D	K	C
Macaque	91	1401	1.658	30.791	0.742
Football	115	613	2.486	10.661	0.403
Celegansneural	297	2148	2.447	14.456	0.308
USAir97	332	2126	2.738	12.807	0.749
Political blogs	1490	16718	2.738	22.440	0.361
Yeast	2361	7182	4.376	6.084	0.271
Amazon web graph	2880	3904	3.433	2.711	0.818
Power grid	4941	6594	18.989	2.669	0.107
ca-GrQc	5242	14496	6.049	5.531	0.687
ca-HepTh	8361	15751	7.025	3.768	0.636
ca-HepPh	12008	118521	4.673	19.74	0.699

4.3 Baseline methods

1. *Common Neighbor (CN).* [6]

This is an intuitive notion of formulating similarity between any two nodes in the underlying network where the similarity score is computed based on the number of mutual/common friends.

$$S(a, b) = |\Gamma(a) \cap \Gamma(b)| \tag{8}$$

where, $\Gamma(a)$ and $\Gamma(b)$ are the size of the neighbors of the node a and b respectively and S(a, b) is the similarity score of the node pair (a, b).

2. *Preferential Attachment (PA).* [10]

Preferential attachment is considered as the basis of network growth model [56] in which the number of edges evolving from a node depends on the degree of that node. Newman [6] and Barabasi et al. [10] have extended this basic notion of preferential growth to a pair of nodes and state that the probability of co-authorship between two nodes is related to the product of the degrees of both nodes. i.e.,

$$S(a, b) = k_a * k_b \tag{9}$$

where k_a and k_b are the degrees of nodes a and b respectively.

3. *Resource Allocation (RA).* [9]

Motivated by the resource allocation process [57] in complex networks, Zhou et al. [9] introduced resource allocation index for link prediction. This work suggests that the penalization imposed to larger degree nodes are not sufficient to the existing work of the Adamic/Adar index. The authors imposed a heavy penalty for larger degree nodes to improve the accuracy. The similarity score between two nodes a and b based on this approach is given by

$$S(a, b) = \sum_{c \in \Gamma(a) \cap \Gamma(b)} \frac{1}{k_c} \tag{10}$$

[2] https://ils.unc.edu/courses/2013_spring/inls509_001/lectures/10-EvaluationMetrics.pdf

[3] https://neurodata.io/project/connectomes/

[4] http://www-personal.umich.edu/~mejn/netdata/

[5] http://vlado.fmf.uni-lj.si/pub/networks/data/

[6] https://icon.colorado.edu/#!/networks

[7] https://snap.stanford.edu/data/

4. *Node and Link Clustering Coefficient (NLC).* [18]

It is based on the clustering property of nodes and edges of the network. It considers both node and link clustering coefficients for the score computation. For any non-observed node pair in the graph, the NLC score can be computed as follows

$$S(a,b) = \sum_{c \in \Gamma(a) \cap \Gamma(b)} \frac{CN(a,c)}{k_c - 1} \times C(c) + \frac{CN(b,c)}{k_c - 1} \times C(c) \tag{11}$$

where k_c is the degree of the node c, CN(a,c) is the number of common neighbors of the nodes a and c, C(c) is the clustering coefficient of the node c.

5. *Local Naive Bayes based Common Neighbor (LNBCN).* [58]

This method is based on the Naive Bayes theory and arguments that different common neighbors play different role in the network and hence contributes differently to the score function computed for non-observed node pairs.

$$S(a,b) = \sum_{c \in \Gamma(a) \cap \Gamma(b)} [\log(\frac{C(c)}{1 - C(c)}) + \log(\frac{1-\rho}{\rho})] \tag{12}$$

where ρ is the network density expressed as]

$$\rho = \frac{E}{N(N-1)/2}$$

6. *CAR Index.* [40]

The CAR method [40] is based on the intuition that the two nodes are likely to be connected if their common neighbors are members of a local community (LC). Such common neighbors are weighted more in this method. Cannistraci et al. proposed CAR variants of Common neighbors, Adamic/Adar, Resource allocation etc.

$$\begin{aligned} S(a,b) &= CN(a,b) \times LCL(a,b) \\ &= CN(a,b) \times \sum_{c \in \Gamma(a) \cap \Gamma(b)} \frac{|\gamma(c)|}{2} \end{aligned} \tag{13}$$

where $LCL(a,b)$ refers to local community links defined in [40]. $\gamma(c)$ is the subset of neighbors of node c that are also common neighbors of a and b.

7. *Clustering Coefficient based Link Prediction (CCLP).* [16]

The method selects the common neighbors of the seed node pair and considers the clustering coefficients of these common neighbors to compute the similarity score of the pair. This method shows good performance on the networks with low correlation between the number of common neighbors and the number of links among them. The similarity score between two disconnected seed node pair can be computed as follows

$$S(a,b) = \sum_{c \in \Gamma(a) \cap \Gamma(b)} C(c) \tag{14}$$

$$C(c) = \frac{t(c)}{k_c(k_c - 1)}$$

where k_c is the degree of node c and t(c) is the total triangles passing through the node c.

8. *Node2vec.* [30]

Node2vec is a low dimensional feature representation technique in which nodes are mapped in lower space such that the network neighborhood of the nodes are preserved. It can also be referred to as network embedding technique which tries to preserve the neighborhood structure by mapping similar nodes in the input space to nearby in the representation or embedding space. It is a semi-supervised algorithm that uses the flexible notion of a biased random walk (sampling strategy) to explore the diverse neighborhood of nodes. The sampling strategy accepts 4 inputs viz., number of walks, walk length, return (p) and in-out (q) hyperparameter. The hyperparameter p controls the probability of revisiting the initial node and q explores the undiscovered part of the network. Thus, the algorithm outputs node embedding of length walk length for each node.

The parameter setting for the node2vec algorithm is as follows: the return (p) and in-out (q) hyperparameters are set the default to 1. The window size is 10 and the number of walks per source 10, each of length 80. Finally, the embedding dimension is set to 128.

4.4 Results analysis

This section investigates the effectiveness of our proposed work on different network datasets against the baseline methods. Our method is tested on four well-known accuracy measures of link prediction namely area under the ROC curve (auroc), area under the precision-recall curve (aupr), average precision, and recall as explained in the Section 4.1. Five sets of probe links (i.e., percentage of removed links = 10, 20 ,30 ,40 ,50) (sparsification levels) are used to evaluate each performance metric. Increasing the percentage of removed links beyond 50% may disconnect the graph, so we consider the sparsification level up to 50% only. The fraction of removed links and the individual metric are displayed on x-axis and y-axis respectively. We demonstrate our results (Figs. 4, 5 6, and 7) based on the clustering

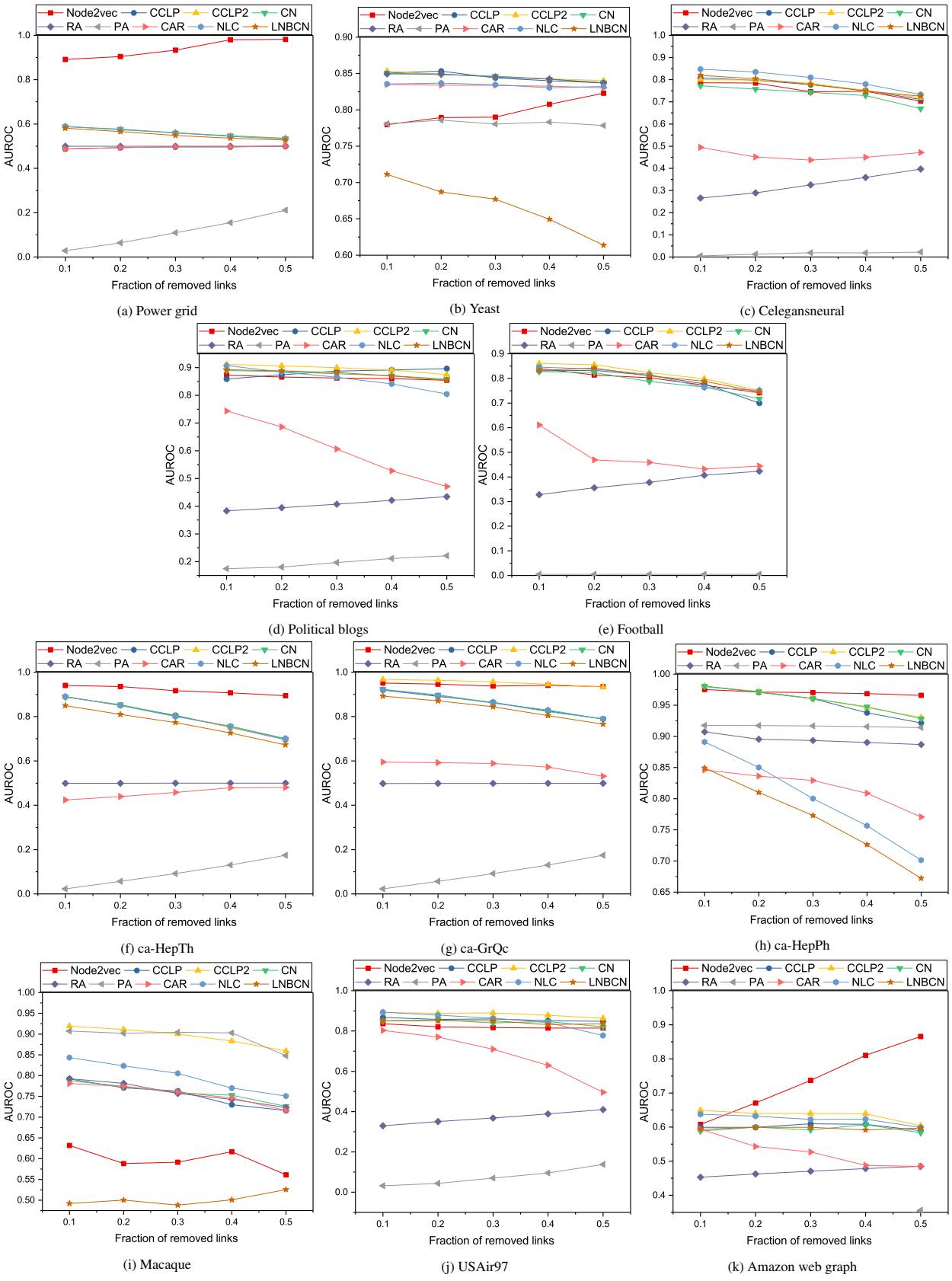


Fig. 4 AUROC results

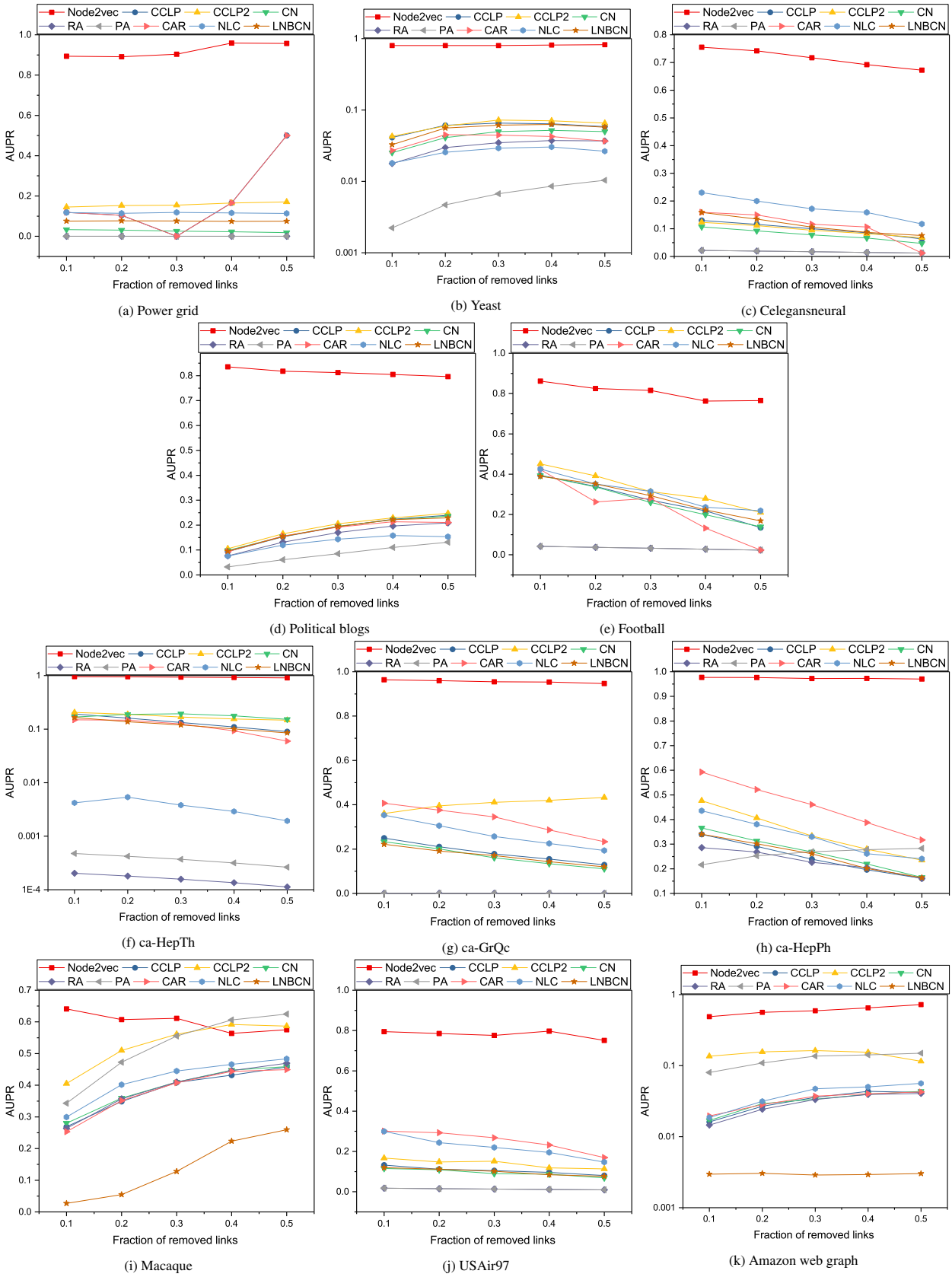


Fig. 5 AUPR results

values of the networks. First, three of each figure (i.e. figure (a), figure (b), and figure (c)) with low clustering, next two figures (figure (d) and figure (e)) having medium clustering values, and last six figures are shown with high clustering values. The proposed method entitled "Level-2 node clustering-coefficient" is abbreviated as "CCLP2" and other baseline methods are also abbreviated in accordance with the abbreviation given in Section 4.3.

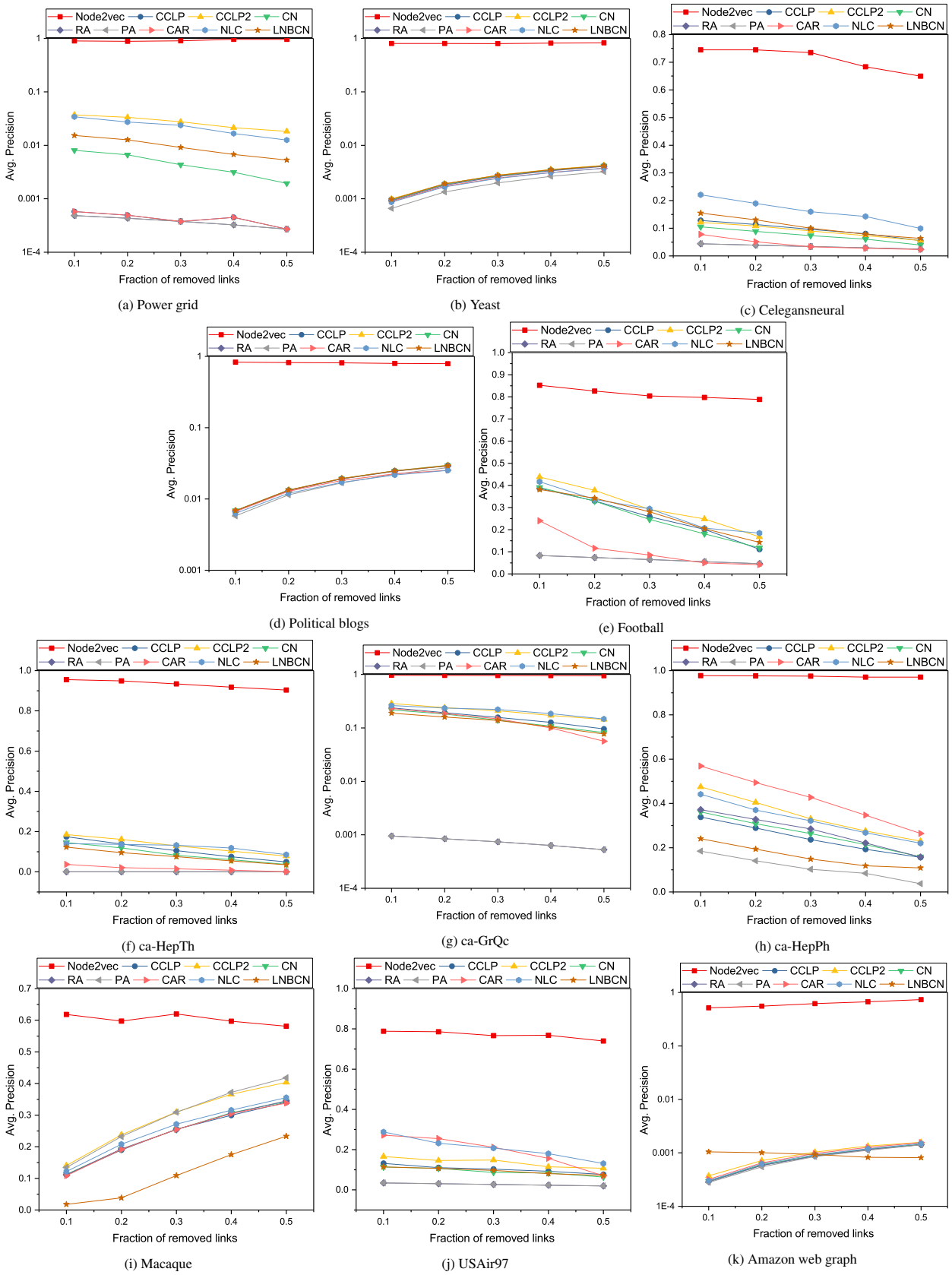
AUROC Figure 4 shows the auroc results of different methods (proposed+baseline) on 11 real-world network datasets. The x-axis represents the five sets of probe links (or fraction of removed links), and auroc is shown on the y-axis. With low clustered ($C \leq 0.3$) networks, the proposed method (CCLP2) shows comparable results with CN, and NLC on power grid (Fig. 4a), CCLP, CN, and RA on yeast (Fig. 4b), and CCLP, and LNBCN on celegansneural data (Fig. 4c). Our method performs better than remaining methods accordingly. The CCLP2 performs overall best on political blogs (Fig. 4d) and football (Fig. 4e) networks (networks with medium clustering values ($0.3 < C \leq 0.5$)). The Node2vec and the NLC perform best on the power grid and celegansneural data respectively. The NLC method also shows good results on these networks but slightly lags CCLP2. For large clustered networks ($C > 0.5$), the CCLP2 performs overall best on collaboration network ca-GrQc (Fig. 4g), macaque (Fig. 4i), and usair97 (Fig. 4j) datasets. Our method outputs comparable results with CCLP and CN on ca-HepTh (Fig. 4f)and ca-HepPh (Fig. 4h). Moreover, it is significantly better than the remaining methods. The Node2vec shows best results on ca-HepTh (Fig. 4f) and amazon web graph (Fig. 4k) where our method is the second-best performer on amazon web graph.

AUPR Most of the real-world networks are sparse, i.e., the number of existing links are less compared to the number of non-existing links. In other words, these networks are highly imbalanced, and the literature suggested that the precision-recall curve (aupr) [50] is more informative than the roc curve for the evaluation of such networks. Hence, AUPR is also considered as one of the evaluation approaches of the link prediction.

Figure 5 shows the result of the area under the precision-recall curve (aupr). From the figure, we observe that the Node2vec is the best performing method against all datasets except the macaque where CCLP2 and PA are best when the fraction of removed links are 40% and 50%. After Node2vec, the proposed method (CCLP2) performs best on power grid (at sparsification levels 10%, 20%, 30%) and yeast networks (except at 20%). The CCLP2 also performs best on medium clustered networks (political blogs Fig. 5d and football 5e) and high clustered networks (ca-GrQc Fig. 5g, macaque Fig. 5i, and amazon web graph Fig. 5k). Moreover, it beats all methods except CN on ca-HepTh and CAR on ca-HepPh as depicted in Fig. 5f and h respectively. On celegansneural and usair97 datasets, our method shows average performance compared to others. With the high clustering value of the usair97, our method performs average because of the lower number of common neighbors between the pairs (local airports (LAs), local centers(LCs)), (local airports (LAs), hubs), and between two local airports [58]. The lower performance of common neighborhood-based methods also due to the same reason. Note that we have used sparsification levels and the fraction of removed links interchangeably.

Average precision Figure 6 shows the average precisions results on 11 real-world network datasets. Similar to the aupr result, Node2vec shows outstanding performance on all datasets. The considered methods except for Node2vec show very low average precision results on all datasets. Our methods performs best after Node2vec on power grid (Fig. 6a) and football (Fig. 6e) networks. The CCLP2 and other methods show comparable results on yeast (Fig. 6b), political blogs (Fig. 6d), and amazon web graph (Fig. 6k). On collaboration networks (i.e. ca-HepTh (Fig. 6f) and ca-Grqc (Fig. 6g)), our method show equivalent results as that of the NLC with some fluctuation. The same results are obtained on the macaque network (Fig. 6i), but, the two equivalent methods are CCLP2 and PA. On ca-HepPh, the CCLP2 lags behind the CAR method. Finally, our method shows average performance on celegansneural (Fig. 6c) and usair97 (Fig. 6j) datasets. The average performance of the CCLP2 and other common neighbor based methods are due to the same reason explained for AUPR in the previous paragraph.

Recall Figure 7 shows recall results for all methods (proposed+baseline). With the low clustered networks, the CCLP2 shows its best on yeast (Fig. 7b) network after Node2vec and on celegansneural (Fig. 7c) after CAR method and comparable results with CN and NLC on power grid data (Fig. 7a). It also best performs on political blogs (Fig. 7d) but average performance on football (Fig. 7e). With high clustered networks, our method overall works best on usair97 network (Fig. 7i) and amazon web graph (Fig. 7j), while second-best performing method on macaque and ca-HepPh networks after Node2vec and CN respectively. On amazon web graph and macaque networks, PA shows good results over the CCLP2 when sparsification level is increased to 40% and 50%. On arXiv networks (i.e., ca-HepTh and ca-GrQc), the CCLP2 result is comparable to CN, CCLP, and NLC. Our results in Fig. 7 shows that the Node2vec method is the best performing method on power grid, yeast, ca-HepTh, and macaque networks, and CAR is overall best on the ca-GrQc dataset.



(a) Power grid (b) Yeast (c) Celegansneural

(d) Political blogs (e) Football

(f) ca-HepTh (g) ca-GrQc (h) ca-HepPh

(i) Macaque (j) USAir97 (k) Amazon web graph

Fig. 6 Average precision results

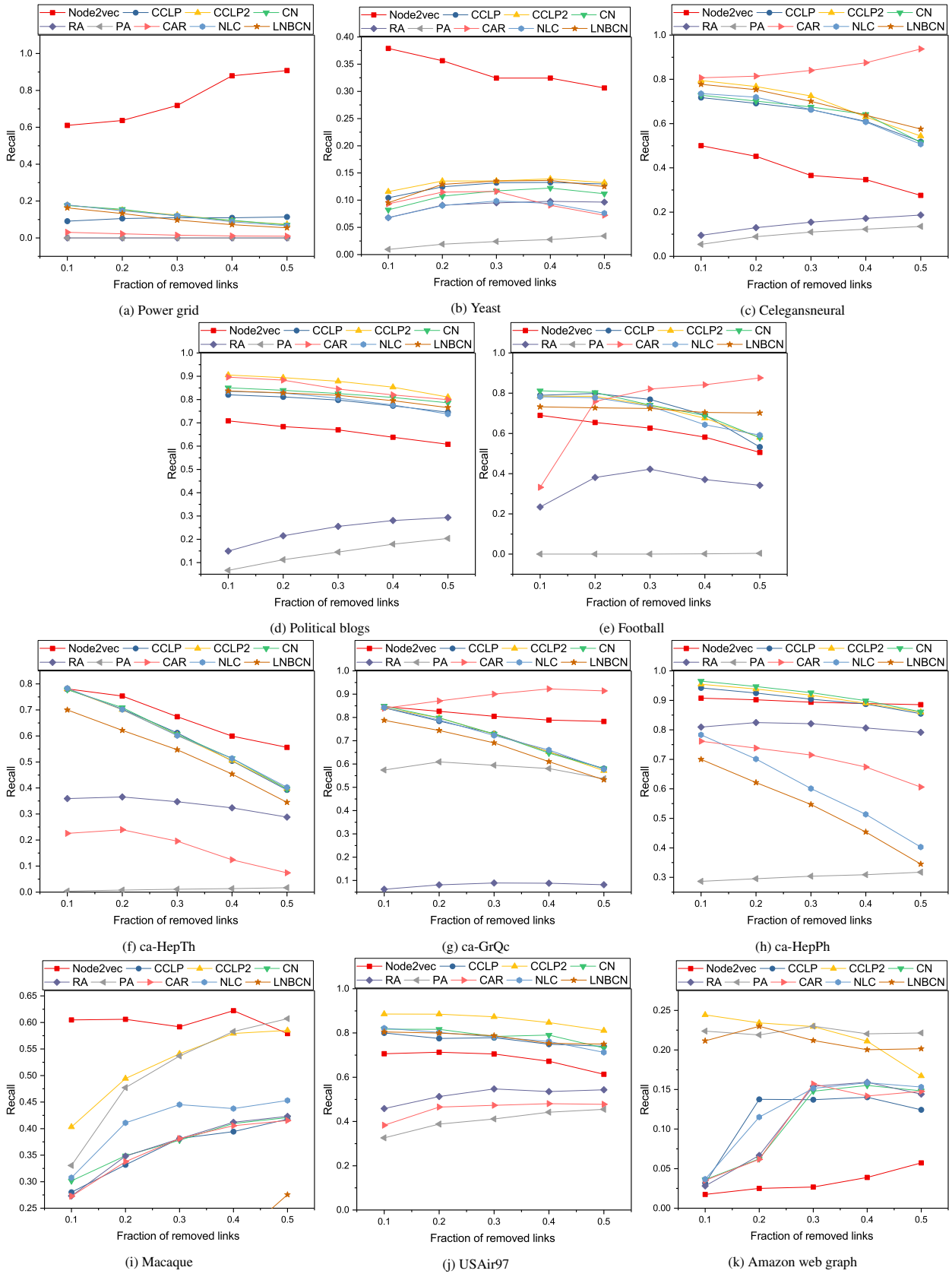


Fig. 7 Recall results

Concluding remarks By analyzing the auroc results, we observe that the proposed method (cclp2) shows comparable results on low clustered networks while best prediction results on medium clustered networks. With high clustered networks, it is best on three datasets (ca-GrQc, macaque, and usair97) and second-best performer on the remaining three datasets (ca-HepTh, ca-HepPh, and amazon web graph). When observing the aupr results, we find that the Node2vec gives best results on all datasets. The proposed method is second best performing one on all networks except the celegansneural, ca-HepPh and usair97. On ca-HepTh, the CN beats our method only on 30% and 40%, lags on 10%, and comparable on 20% and 50% of removed links. Further, Node2vec also performs overall best on average precision metric against 11 network datasets. Our method is the second-best performer on power grid and football, and comparable on high clustered networks except for usair97 where CCLP2 show average performance. The recall result shows that Node2vec is best on 4 datasets (power grid, yeast, ca-HepTh, and macaque). The CCLP2 is overall best on political blogs and usair97 networks while second-best performer on yeast, celegansneural, ca-HepPh, macaque, and amazon web graph. It shows comparable results on power grid, football, ca-HepTh, and ca-GrQc datasets.

Finally, we observe that Node2vec is the best performing method on all datasets with some exception. The CCLP2 shows better performance after the Node2vec method on average and high clustered networks except USAir97. Though USAir97 is having high clustering coefficient value, our method performs average (after CAR, and NLC) on AUPR and average precision metrics. The possible reason may be because there is the least number of common neighbors between two local airports (LAs), between local airports (LAs) and local centers (LCs), and between local airports (LAs) and hubs [58]. This results in the least probability of such links to present in the top of the list and hence the precision gets reduced of our method and the common neighborhood-based methods. As a result, there is low performance in AUPR and average precision.

Complexity analysis Here, we estimate the efficiency of the proposed method as well as the baseline predictors (algorithms). Only off-line parts of all algorithms are considered for the time estimation where off-line refers to the similarity matrix computation for all pair of nodes. The time needed to compute level-2 common neighbors is the same as the normal common neighbor of a pair of nodes, i.e., $O(n^2)$ when the data structure is the adjacency matrix and O(n) in case of the adjacency list. The clustering coefficient of a node takes $O(n^3)$ in the worst case and $O(nk^2)$ after applying some optimization, although an approximate algorithm of O(n) time by Schank and Wagner [59] also exists.

The main crux of our algorithm is the computation of the level-2 clustering coefficient in Steps 3-6 where the loop of Step 3 iterates to 2k times. Line 4 costs O(n) to compute CNs for a given pair and O(nk) for computing clustering coefficients, resulting in total $2k \times (O(n) + O(nk))$ time for 4 steps. The outer for loop iterates to $O(n^2)$ in the worst case, so the total time complexity of the proposed algorithm is $O(n^3k^2)$ which is comparable to the $O(n^3k)$ of existing CCLP. The computational complexity of the NLC and the LNBCN are $O(n^4k)$ and $O(n.O(f(z)+nk^3))$ where f(z) is the influence function. The CAR costs $O(nk^4)$ which is more complex as it computes time-consuming local community links (LCL). Other methods like CN, AA, RA estimates $O(nk^3)$ while PA costs $O(nk^2)$ where k is the average degree of the network. The Node2vec [30] method is based on a random walk sampling which is efficient compared to pure BFS/DFS. The effective time complexity of the node2vec is $O(\frac{l}{k'(l-k')})$ per sample where l is walk length and k' is neighborhood size.

Statistical test In this paragraph, we conduct a statistical test [60] to show the significant difference of the proposed method with the baseline methods. We perform the Friedman test [61, 62] to analyze whether there is a significant difference among multiple methods. It is a non-parametric counterpart of the repeated measures ANOVA. If the test result showed a significant difference, we further applied post hoc analysis to check the degree of rejection of each hypothesis. For the post hoc analysis, several methods are available in the literature and we applied post hoc counterpart of the Friedman test known as Posthoc Friedman Conover method. The proposed method CCLP2 is considered as the control algorithm in the posthoc analysis. We select the level of confidence $\alpha_c = 0.05$ and the degree of freedom $D_f = 8$.

The Friedman test result for the metric area under the ROC curve (AUROC) is tabulated in the Table 2. The table shows the computed Friedman test values F_f on different percentage (10, 20, 30, 40, and 50) of removed links (or sparsification levels). The Friedman test rejects the null hypothesis H_0 if the test value F_f is greater than $\chi^2(\alpha_c, D_f)$, i.e. $F_f > 15.51$. We have performed the same test for remaining 3 metrics viz., AUPR, Recall and Average Precision where we found that the null hypothesis is rejected for each of the metrics. We have not shown the remaining here.

Since this test rejects the null hypothesis on each percentage of removed links so we go for the post hoc analysis. The results of the post hoc analysis are shown in Table 3 for all 4 metrics used in this paper. With the confidence level $\alpha_c = 0.05$, we observe that the proposed

Table 2 The Friedman test on Area under the ROC Curve (AUROC)

Removed links (%)	Dataset	IS-value									Test value	State result
		Node2vec	CCLP	CCLP2	CN	RA	PA	CAR	NLC	LNBCN	F_f	Is $F_f > \chi^2$?
10	Macaque	0.63199	0.79209	0.91921	0.7875	0.79259	0.91732	0.78093	0.84349	0.49197	47.466	Null Hypothesis Rejected
	Football	0.84021	0.83049	0.86067	0.8281	0.32826	0.00474	0.61076	0.84617	0.83241		
	Celegansneural	0.78599	0.80817	0.80028	0.77269	0.26587	0.00527	0.49472	0.84701	0.81955		
	USAir97	0.83602	0.86762	0.89145	0.85007	0.32958	0.03225	0.80266	0.89348	0.8508		
	Political blogs	0.87281	0.85881	0.91166	0.88962	0.38327	0.17453	0.7436	0.90752	0.89255		
	Yeast	0.77971	0.85077	0.85354	0.85065	0.84938	0.78088	0.83494	0.83562	0.71119		
	Amazon web graph	0.60775	0.5995	0.64935	0.58821	0.45291	0.08295	0.59378	0.63773	0.59335		
	Power grid	0.89143	0.48755	0.58843	0.5872	0.49946	0.02824	0.48755	0.58948	0.58129		
	ca-GrQc	0.95144	0.91962	0.96748	0.92301	0.49802	0.02237	0.59494	0.92142	0.89244		
	ca-HepTh	0.94038	0.88902	0.88972	0.88813	0.49917	0.02237	0.42399	0.89102	0.84927		
	ca-HepPh	0.97528	0.9801	0.98051	0.98051	0.90737	0.91754	0.84623	0.89102	0.84927		
20	Macaque	0.58817	0.77075	0.91119	0.7742	0.78123	0.91188	0.77379	0.82347	0.50015	41.541	Null Hypothesis Rejected
	Football	0.81396	0.83259	0.85578	0.82336	0.35586	0.0047	0.46912	0.83809	0.84206		
	Celegansneural	0.78504	0.79719	0.79833	0.75735	0.28911	0.01219	0.45128	0.83489	0.80416		
	USAir97	0.82055	0.85718	0.88811	0.85534	0.35042	0.04408	0.77034	0.87852	0.85176		
	Political blogs	0.86617	0.87417	0.90649	0.88584	0.39449	0.18029	0.68585	0.88432	0.88852		
	Yeast	0.78944	0.85339	0.84905	0.84859	0.84942	0.78592	0.83401	0.83678	0.68717		
	Amazon web	0.67059	0.59935	0.63993	0.59964	0.46255	0.16024	0.54331	0.63223	0.59926		
	Power grid	0.9043	0.49376	0.57517	0.57715	0.49957	0.06411	0.49376	0.57362	0.56606		
	ca-GrQc	0.94598	0.89122	0.96368	0.89782	0.49833	0.05681	0.59172	0.8945	0.87084		
	ca-HepTh	0.93529	0.85119	0.85012	0.85396	0.4993	0.05681	0.43903	0.85024	0.81001		
	ca-HepPh	0.97136	0.97089	0.97213	0.9714	0.8955	0.91742	0.83615	0.85024	0.81001		
30	Macaque	0.5915	0.76278	0.90004	0.75815	0.75693	0.90442	0.76048	0.80546	0.4882	44.904	Null Hypothesis Rejected
	Football	0.80292	0.8125	0.82345	0.78719	0.37782	0.00465	0.45868	0.81644	0.81101		
	Celegansneural	0.74513	0.77752	0.78369	0.74319	0.32566	0.019	0.43761	0.80986	0.77897		
	USAir97	0.81677	0.85996	0.88998	0.83981	0.36812	0.07017	0.70941	0.8636	0.85032		
	Political blogs	0.86271	0.88731	0.89981	0.87817	0.4072	0.19667	0.60622	0.86517	0.88266		
	Yeast	0.79004	0.84402	0.84603	0.84651	0.84519	0.78042	0.83386	0.83471	0.67703		
	Amazon web graph	0.73707	0.60955	0.64005	0.59175	0.47066	0.23873	0.5268	0.62252	0.59947		
	Power grid	0.93327	0.49698	0.56191	0.55913	0.49966	0.10937	0.49698	0.56015	0.5482		
	ca-GrQc	0.93761	0.8644	0.95637	0.86322	0.49862	0.09151	0.58845	0.8608	0.84441		
	ca-HepTh	0.91689	0.80546	0.80138	0.80254	0.49943	0.09151	0.45759	0.80022	0.77299		
	ca-HepPh	0.97038	0.96051	0.96084	0.96084	0.89339	0.9169	0.82919	0.80022	0.77299		
40	Macaque	0.61685	0.73003	0.8832	0.75282	0.74258	0.9028	0.74629	0.76991	0.50097	42.62	Null Hypothesis Rejected
	Football	0.77291	0.77777	0.79831	0.76483	0.40681	0.0046	0.43195	0.76349	0.78828		
	Celegansneural	0.74764	0.75169	0.75133	0.72846	0.35868	0.0189	0.44974	0.77997	0.74867		
	USAir97	0.81444	0.85122	0.87774	0.84676	0.38865	0.09531	0.62982	0.84167	0.83187		
	Political blogs	0.86056	0.89142	0.89165	0.87088	0.42104	0.21101	0.5276	0.84148	0.87017		

Table 2 (continued)

Removed links (%)	Dataset	IS-value									Test value	State result
		Node2vec	CCLP	CCLP2	CN	RA	PA	CAR	NLC	LNBCN	F_f	Is $F_f > \chi^2$?
	Yeast	0.80769	0.84003	0.84286	0.84217	0.84236	0.78319	0.83267	0.83036	0.64939		
	Amazon web graph	0.81067	0.60829	0.63898	0.60711	0.47841	0.30388	0.48793	0.62335	0.59182		
	Power grid	0.97994	0.49685	0.54597	0.54742	0.49974	0.15502	0.49685	0.54368	0.53582		
	ca-GrQc	0.94134	0.82526	0.94552	0.8224	0.4989	0.13022	0.57193	0.82966	0.80411		
	ca-HepTh	0.90694	0.75157	0.75261	0.75634	0.49957	0.13022	0.47881	0.75659	0.72636		
	ca-HepPh	0.96875	0.93794	0.94639	0.94741	0.89036	0.91561	0.80858	0.75659	0.72636		
50	Macaque	0.56126	0.71594	0.85942	0.72602	0.72339	0.88796	0.7159	0.75059	0.52559	44.876	Null Hypothesis Rejected
	Football	0.74172	0.69971	0.75182	0.71695	0.42341	0.00456	0.44407	0.75167	0.74535		
	Celegansneural	0.7037	0.7108	0.71897	0.66977	0.39684	0.02219	0.47157	0.73282	0.7257		
	USAir97	0.81418	0.84813	0.86302	0.82049	0.41003	0.13808	0.49573	0.77795	0.8365		
	Political blogs	0.85452	0.8967	0.87475	0.8587	0.43442	0.22092	0.47083	0.8047	0.85456		
	Yeast	0.82299	0.83725	0.84017	0.83749	0.83697	0.77849	0.83032	0.83232	0.61378		
	Amazon web graph	0.86576	0.58945	0.60367	0.58361	0.4849	0.35637	0.48466	0.59915	0.59714		
	Power grid	0.98211	0.5	0.53698	0.53492	0.49982	0.21136	0.5	0.53272	0.52725		
	ca-GrQc	0.93511	0.7897	0.93304	0.78912	0.49914	0.17413	0.53079	0.78822	0.76549		
	ca-HepTh	0.894	0.69597	0.6987	0.69777	0.49968	0.17413	0.4804	0.7012	0.67221		
	ca-HepPh	0.96589	0.92157	0.93042	0.92844	0.88683	0.9142	0.77064	0.7012	0.67221		

method (CCLP2) is significantly different from all the baseline methods except NLC method on AUROC and Average precision. Our method is insignificant on AUROC against 10% of removed links and insignificant on average precision against 30%, 40%, and 50% of removed links. Moreover, the CCLP2 shows its significance on recall against CCLP (except 50%), RA, PA, CAR, NLC (except 10% and 40%), and LNBCN methods. It is insignificant from the CN and the Node2vec (except 10% and 20%). With AUPR, our method is significantly different from the

Table 3 The Posthoc Friedman Conover Test (Control method = CCLP2)

Metric	Removed links (%)	p-value							
		Node2vec	CCLP	CN	RA	PA	CAR	NLC	LNBCN
AUROC	10	0.00219	0.00126	0.00039	3.60E-09	1.00E-11	9.70E-10	0.17335	2.40E-06
	20	0.01193	0.00551	0.06079	3.00E-07	5.70E-10	3.60E-08	0.06079	5.70E-05
	30	0.01079	0.02287	0.00242	5.00E-09	4.90E-11	1.10E-08	0.01468	4.40E-06
	40	0.0346	0.01283	0.0346	5.50E-08	2.40E-10	7.00E-08	0.00811	5.60E-06
	50	0.04009	0.00571	0.00927	6.60E-09	6.50E-11	5.10E-09	0.00673	2.10E-05
Recall	10	0.04495	0.02078	1.37E-01	4.30E-08	1.70E-08	7.40E-05	0.05717	8.93E-03
	20	0.0344	0.03045	0.18656	7.20E-07	5.60E-08	1.07E-02	0.00526	1.54E-03
	30	0.05862	0.02915	0.07288	6.60E-07	1.90E-07	2.58E-02	0.00284	3.30E-03
	40	0.25412	0.04778	0.44606	1.10E-04	9.20E-06	3.06E-02	0.05914	3.84E-02
	50	0.1836	0.0724	0.1292	1.90E-05	9.00E-06	1.89E-02	0.0304	3.82E-02

Table 3 (continued)

Metric	Removed links (%)	p-value							
		Node2vec	CCLP	CN	RA	PA	CAR	NLC	LNBCN
AUPR	10	0.00923	0.00092	2.30E-05	1.60E-12	7.20E-11	3.25E-02	0.00276	1.50E-06
	20	0.01215	0.00024	7.20E-05	2.30E-11	2.60E-10	2.05E-02	0.00578	9.60E-06
	30	0.01221	0.00019	0.00024	2.40E-10	3.70E-08	7.23E-03	0.02358	9.70E-05
	40	0.0245	0.00812	0.00034	4.60E-09	7.50E-07	2.84E-02	0.01323	9.80E-05
	50	0.01669	0.01435	0.00332	3.40E-08	1.00E-05	1.05E-02	0.05125	7.80E-04
Avg. Precision	10	0.01492	0.00088	1.40E-04	6.60E-11	1.60E-12	8.20E-06	0.01244	8.30E-05
	20	0.01115	0.00131	6.90E-05	1.70E-09	1.50E-11	8.80E-05	0.02223	1.31E-03
	30	0.01046	0.00126	2.70E-05	6.70E-09	2.10E-10	5.50E-05	0.11963	6.70E-04
	40	0.00881	0.00515	0.00089	4.00E-08	2.50E-09	2.00E-05	0.16345	1.00E-04
	50	0.01434	0.00051	0.00041	1.00E-08	1.80E-09	1.30E-07	0.13366	4.00E-05

baseline method except for NLC, where it is insignificant only for 50% of the removed links.

5 Conclusion and future works

Motivated by the intuition that more local information of topology of a network may improve the accuracy of link prediction, we extracted common neighbors and clustering information up to next level. The proposed method computes clustering coefficients of level-2 common neighbors of the seed node pair. The similarity score sums over all such common neighbors for the seed node pair. The experiments have been conducted on 11 real-world networks and results are organized as low, medium, and high clustered networks. The comprehensive results show that the proposed method performs better than the baseline methods except for the Node2vec with medium and large average clustering coefficients. Recently, some sophisticated methods like Node2vec [30] and SPM [63] have been proposed which show outstanding performance. Although the prediction performance of these methods are significantly better, however, in the case of large networks the proposed method (CCLP2) should be considered with these methods at least.

In this work, we have considered simple undirected and unweighted networks (datasets) i.e., only one type of relationship between two nodes have been selected. If we consider multiple relationships into account, the prediction performance can be enhanced [64]. In the future, we will try to explore such an idea in a supervised setting. Moreover, we will also validate our method with the networks having negative links (Signed network) like Epinions and Slashdot networks.

References

1. Liben-Nowell D, Kleinberg J The link-prediction problem for social networks. J Am Soc Inf Sci Technol
2. Adafre SF, de Rijke M Discovering missing links in wikipedia. In: Proceedings of the 3rd international workshop on link discovery, LinkKDD '05, pp 90–97
3. Zhu J, Hong J, Hughes JG Using Markov models for web site link prediction. In: Proceedings of the thirteenth ACM conference on hypertext and hypermedia, HYPERTEXT '02, pp 169–170
4. Huang Z, Li X, Chen H Link prediction approach to collaborative filtering. In: Proceedings of the 5th ACM/IEEE-CS joint conference on digital libraries, JCDL '05, pp 141–142
5. Airodi E, Blei D, Xing E, Fienberg S Mixed membership stochastic block models for relational data, with applications to protein-protein interactions. In: Proceedings of international biometric society-ENAR annual meetings
6. Newman MEJ (2001) Clustering and preferential attachment in growing networks. Phys Rev E 64:025102. https://doi.org/10.1103/PhysRevE.64.025102
7. Jaccard P (1901) Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. Bull Soc Vaudoise Sci Nat 37:241–272
8. Lada A, Adar E (2003) Friends and neighbors on the web. Soc Netw 25:211–230. https://doi.org/10.1016/S0378-8733(03)00009-1
9. Zhou T, Lu L, Zhang Y-C (2009) Predicting missing links via local information. Europ Phys J B 71:623–630. https://doi.org/10.1140/epjb/e2009-00335-8
10. Barabasi A, Jeong H, Neda Z, Ravasz E, Schubert A, Vicsek T (2002) Evolution of the social network of scientific collaborations. Physica A Stat Mech Appl 311:590–614. https://doi.org/10.1016/S0378-4371(02)00736-7
11. Katz L (1953) A new status index derived from sociometric analysis. Psychometrika 18(1):39–43
12. Liu W, Lü L (2010) Link prediction based on local random walk. EPL (Europhys Lett) 89(5):58007. http://stacks.iop.org/0295-5075/89/i=5/a=58007

13. Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. In: Proceedings of the seventh international conference on World Wide Web 7, WWW7. Elsevier Science Publishers B. V., Amsterdam, pp 107–117. http://dl.acm.org/citation.cfm?id=297805.297827
14. Leicht EA, Holme P, Newman MEJ (2006) Vertex similarity in networks. Phys Rev E 73:026120. https://doi.org/10.1103/PhysRevE.73.026120
15. Tong H, Faloutsos C, Pan J-Y (2006) Fast random walk with restart and its applications. In: Proceedings of the sixth international conference on data mining, ICDM '06. IEEE Computer Society, Washington, pp 613–622, https://doi.org/10.1109/ICDM.2006.70
16. Wu Z, Lin Y, Wang J, Gregory S (2016) Link prediction with node clustering coefficient. Physica A: Stat Mech Appl 452:1–8. https://doi.org/10.1016/j.physa.2016.01.038
17. Liu Y, Zhao C, Wang X, Huang Q, Zhang X, Yi D (2016) The degree-related clustering coefficient and its application to link prediction. Physica A: Stat Mech Appl 454:24–33. https://doi.org/10.1016/j.physa.2016.02.014
18. Wu Z, Lin Y, Wan H, Jamil W (2016) Predicting top-L missing links with node and link clustering information in large-scale networks. J Stat Mech Theory Exper 8:083202. https://doi.org/10.1088/1742-5468/2016/08/083202
19. Hasan MA, Chaoji V, Salem S, Zaki M (2006) Link prediction using supervised learning. In: Proc. of SDM 06 workshop on link analysis, counterterrorism and security
20. Popescul A, Popescul R, Ungar LH (2003) Statistical relational learning for link prediction
21. Popescul A, Popescul R, Ungar LH (2003) Structural logistic regression for link analysis
22. Taskar B, Wong M-F, Abbeel P, Koller D (2003) Link prediction in relational data. In: Proceedings of the 16th international conference on neural information processing systems, NIPS'03. MIT Press, Cambridge, pp 659–666. http://dl.acm.org/citation.cfm?id=2981345.2981428
23. Sarukkai RR (2000) Link prediction and path analysis using Markov chains1. Comput Netw 33(1-6):377–386
24. Shapiro EY (1983) Algorithmic program debugging. MIT Press, Cambridge
25. Getoor L, Friedman N, Koller D, Taskar B (2002) Learning probabilistic models of link structure. J Mach Learn Res 3:679–707
26. Nallapati RM, Ahmed A, Xing EP, Cohen WW Joint latent topic models for text and citations. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '08, pp 542–550
27. Fu W, Song L, Xing EP Dynamic mixed membership blockmodel for evolving networks. In: Proceedings of the 26th annual international conference on machine learning, ICML '09, pp 329–336
28. Xu Z, Tresp V, Yu S, Yu K (2008) Nonparametric relational learning for social network analysis. In: KDD'2008 Workshop on social network mining and analysis
29. Belkin M, Niyogi P (2001) Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Proceedings of the 14th international conference on neural information processing systems: natural and synthetic, NIPS'01. MIT Press, Cambridge, pp 585–591. http://dl.acm.org/citation.cfm?id=2980539.2980616
30. Grover A, Leskovec J (2016) Node2vec: scalable feature learning for networks. In: Proceedings of the 22Nd ACM SIGKDD international conference on knowledge discovery and data mining, KDD '16. ACM, New York, pp 855-864, https://doi.org/10.1145/2939672.2939754
31. Mehran Kazemi S, Poole D SimplE embedding for link prediction in knowledge graphs. arXiv:1802.04868
32. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '14. ACM, New York, pp 701–710, https://doi.org/10.1145/2623330.2623732
33. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326. https://doi.org/10.1126/science.290.5500.2323. http://science.sciencemag.org/content/290/5500/2323
34. Mikolov T, Chen K, Corrado G, Dean J Efficient estimation of word representations in vector space. arXiv:1301.3781
35. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J Distributed representations of words and phrases and their compositionality. arXiv:1310.4546
36. Tylenda T, Angelova R, Bedathur S Towards time-aware link prediction in evolving social networks. In: Proceedings of the 3rd workshop on social network mining and analysis, SNA-KDD '09, pp 9:1–9:10
37. Song HH, Cho TW, Dave V, Zhang Y, Qiu L Scalable proximity estimation and link prediction in online social networks. In: Proceedings of the 9th ACM SIGCOMM conference on internet measurement, IMC '09, pp 322–335
38. Acar E, Dunlavy DM, Kolda TG (2009) Link prediction on evolving data using matrix and tensor factorizations. In: 2009 IEEE International conference on data mining workshops, pp 262–269. https://doi.org/10.1109/ICDMW.2009.54
39. Zan H (2006) Link prediction based on graph topology: the predictive value of generalized clustering coefficient
40. Cannistraci CV, Alanis-Lobato G, Ravasi T (2013) From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. Sci Rep 3:1613. https://doi.org/10.1038/srep01613
41. Albert R, Barabási A-L (2002) Statistical mechanics of complex networks. Rev Mod Phys 74:47–97. https://doi.org/10.1103/RevModPhys.74.47
42. Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. Nature 393(6684):440–442. https://doi.org/10.1038/30918
43. Kleinberg JM (2000) Navigation in a small world. Nature 406(6798):845
44. Milgram S (1967) The small world problem. Psychol Today 2:60–67
45. Barabasi A-L, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512. https://doi.org/10.1126/science.286.5439.509. http://science.sciencemag.org/content/286/5439/509
46. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, New York
47. Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology 143(1):29–36. https://doi.org/10.1148/radiology.143.1.7063747
48. Fawcett T (2006) An introduction to roc analysis. Pattern Recogn Lett 27(8):861–874. https://doi.org/10.1016/j.patrec.2005.10.010
49. Davis J, Goadrich M (2006) The relationship between precision-recall and roc curves. In: Proceedings of the 23rd international conference on machine learning, ICML '06. ACM, New York, pp 233-240, https://doi.org/10.1145/1143844.1143874
50. Saito T, Rehmsmeier M (2015) The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE 10:1–21. https://doi.org/10.1371/journal.pone.0118432
51. Markov NT, Ercsey-Ravasz MM, Ribeiro Gomes AR, Lamy C, Magrou L, Vezoli J, Misery P, Falchier A, Quilodran R, Gariel MA, Sallet J, Gamanut R, Huissoud C, Clavagnier S, Giroud P, Sappey-Marinier D, Barone P, Dehay C, Toroczkai Z, Knoblauch

K, Van Essen DC, Kennedy H (2014) A weighted and directed interareal connectivity matrix for macaque cerebral cortex. Cereb Cortex 24(1):17–36. https://doi.org/10.1093/cercor/bhs270
52. Girvan MM, Newman EJ (2002) Community structure in social and biological networks. Proc Nat Acad Sci USA 99(12):7821–7826. https://doi.org/10.1073/pnas.122653799
53. Adamic LA, Glance N (2005) The political blogosphere and the 2004 U.S. election: Divided they blog. In: Proceedings of the 3rd international workshop on link discovery, LinkKDD '05. ACM, New York, pp 36–43, https://doi.org/10.1145/1134271.1134277
54. Bu D, Zhao Y, Cai L, Xue H, Zhu X, Lu H, Zhang J, Sun S, Ling L, Zhang N, Li G, Chen R (2003) Topological structure analysis of the protein–protein interaction network in budding yeast. Nucleic Acids Res 31(9):2443–2450. https://doi.org/10.1093/nar/gkg340
55. Šubelj L, Bajec M (2012) Ubiquitousness of link-density and link-pattern communities in real-world networks. Europ Phys J B 85(1):32. https://doi.org/10.1140/epjb/e2011-20448-7
56. Mitzenmacher M (2004) A brief history of generative models for power law and lognormal distributions. Int Math 1:226–251
57. Ou Q, Jin Y-D, Zhou T, Wang B-H, Yin B-Q (2007) Power-law strength-degree correlation from resource-allocation dynamics on weighted networks. Phys Rev E 75:021102. https://doi.org/10.1103/PhysRevE.75.021102
58. Liu Z, Zhang Q-M, Lü L, Zhou T (2011) Link prediction in complex networks: a local naïve bayes model. EPL (Europhys Lett) 96(4):48007. http://stacks.iop.org/0295-5075/96/i=4/a=48007
59. Schank T, Wagner D (2005) Approximating clustering coefficient and transitivity. J Graph Algorithms Appl 9:265–275
60. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30. http://dl.acm.org/citation.cfm?id=1248547.1248548
61. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701. https://doi.org/10.1080/01621459.1937.10503522
62. Friedman M (1940) A comparison of alternative tests of significance for the problem of *m* rankings. Ann Math Statist 11(1):86–92. https://doi.org/10.1214/aoms/1177731944
63. Lü L, Pan L, Zhou T, Zhang Y-C, Stanley HE (2015) Toward link predictability of complex networks. Proc Natl Acad Sci 112(8):2325–2330. https://doi.org/10.1073/pnas.1424644112
64. Wang X, Sukthankar G (2013) Link prediction in multi-relational collaboration networks. In: Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining, ASONAM '13. ACM, New YorkA, pp 1445–1447, https://doi.org/10.1145/2492517.2492584

Ajay Kumar completed his master of technology in Computer Science & Engineering from Samrat Ashok Technological Institute, Vidisha (M.P.) and Bachelor of Technology in Computer Science & Engineering from R.K.D.F Institute of Science and Technology Bhopal (M.P.). He is pursuing Ph.D. in Computer Science and Engineering from Indian Institute of Technology (BHU), Varanasi. His research interests include Link Prediction and Influence Maximization in social/complex networks.

Shashank Sheshar Singh received M.Tech. degree in Computer Science and Engineering from Indian Institute of Technology, Roorkee (IITR). He received B.Tech. degree in Computer Science and Engineering from Kali Charan Nigam Institute of Technology (KCNIT), Banda affiliated to GBTU University, Lucknow. He is working toward the Ph.D. in Computer Science and Engineering from Indian Institute of Technology (BHU), Varanasi. His research interests include Data Mining, Influence Maximization, Link Prediction, and Social Network Analysis.

Kuldeep Singh pursued his Ph.D. in Computer science & Engineering from Indian Institute of Technology (BHU) Varanasi. His research interest includes High utility itemsets mining, social network analysis, and data mining. He received his M.Tech degree in Computer science and Engineering from Guru Jambheshwar University of Science and Technology, Hisar (Haryana). He has 8 years of teaching and research experience. His research interests include Data Mining and Social Network Analysis.

Bhaskar Biswas received Ph.D. in Computer Science & Engineering from Indian Institute of Technology (BHU), Varanasi. He received the B.Tech. degree in Computer Science and Engineering from Birla Institute of Technology, Mesra. He is working as an Associate Professor at Indian Institute of Technology (BHU), Varanasi in the Computer Science & Engineering department. His research interests include Data Mining, Text Analysis, Machine Learning, Influence Maximization, Link Prediction, Social Network Analysis.