Physica A 605 (2022) 128008

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

ELP: Link prediction in social networks based on ego network perspective

Shivansh Mishra [a,*], Shashank Sheshar Singh [b], Ajay Kumar [c], Bhaskar Biswas [a]

[a] Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, India
[b] Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, India
[c] Department of Computer Science and Engineering, UPES University, Dehradun, India

ARTICLE INFO

Article history:
Received 3 June 2021
Received in revised form 30 April 2022
Available online 4 August 2022

Keywords:
Link prediction
Ego networks
Social influence
Social networks

ABSTRACT

Social network analysis has recently been of much interest to researchers in diverse fields. This increased attention is due to its broad applicability in modeling complex real-world scenarios (problems). Link prediction is a crucial issue in social network analysis, one that finds the likelihood of having a link between two nodes in the network. Of the existing methods, many use topological network properties, while others use algebraic methods, statistical models, node embeddings and, community information. Although some path-based approaches can be said to deal with some nodes' commutative effect at some point, they are not designed to infer the total community effect of all local nodes on a specific link. Hence we present *ELP*, a link prediction method based on the Ego perspective. First, this approach computes each existing edge's ego strength using ego networks, which can be construed as regions of influence of specific nodes. These ego strengths can be abstracted as the total effect of all local nodes on a particular edge. Then we utilize a topological feature set to estimate the prediction scores for target links. This feature set is selected after observing the performance of five different possible topological feature sets. Finally, we perform experiments on real-world networks to validate our algorithm's performance and compare it with state-of-the-art algorithms. The statistical tests justify the significant difference of our proposed method from the state-of-the-art algorithms.

1. Introduction

The rapid development of online social networking (OSN) platforms like Twitter, Facebook, and Sina Weibo has attracted many researchers. Through the analysis of OSNs, many social phenomena are encountered like, link prediction [1–4], information diffusion [5], viral marketing [6–10], community detection [11,12], and anomaly detection [13–16]. The link prediction problem has attracted a lot of attention in the past several years and predicts the target links in an observed static network that may appear in the future. Informally, link prediction can be stated as the investigation of future links among a set of individuals. M. E. J. Newman [1] was the first to present a method to predict target links in collaboration networks in Biology and Physics. Later, the link prediction problem was introduced in social networks by Liben-Nowell et al. [17] as the process of inference of new or still unknown interactions between entities.

* Corresponding author.
E-mail addresses: shivanshmishra.rs.cse18@itbhu.ac.in (S. Mishra), shashankss.rs.cse16@itbhu.ac.in (S.S. Singh), ajayk.rs.cse16@itbhu.ac.in (A. Kumar), bhaskar.cse@itbhu.ac.in (B. Biswas).

https://doi.org/10.1016/j.physa.2022.128008
0378-4371/

Interaction dynamics is a valuable source of information in social network analysis. For example, the type of relationship, such as best friends, family, close friends, extended family members, commercial friendships, etc., can be identified based on interaction frequency and communication distance [18]. Some studies utilized interaction dynamics like the amount of time spent in interaction [19] and frequency pattern [20,21] to predict future links. Lionel et al. [22] further explore the interaction dynamics in a phone call dataset by considering temporal information like timestamp, duration, etc., to predict likely connected pairs. In Toprak et al. [23], authors propose using ego network layers to improve the performance of local similarity-based link prediction algorithms. In Rezaeipanah et al. [24], authors have proposed ego-based features for classification-based link prediction tasks on multiplex networks. Other studies into the behavior of ego networks have also been conducted, such as ones by [25], where authors study the weightage of each ego circle of corresponding nodes. Due to interaction dynamics, we utilize ego-centered social networks to reveal target links in this work. The ego network corresponding to a node comprises a set containing the node itself and its direct and indirect neighbors. We utilize a pair ranking approach combined with ego strength to predict missing links. The highly-ranked nodes (same ego circle nodes) corresponding to a central node are more prone to interact with each other than low-ranked nodes (different ego circle nodes). Each ego network and its levels correspond to how far the influence of a node spreads from its central position. These levels help us quantify the influence of a node over edges in its immediate neighborhood. If node influences of all nodes over all edges are combined, we get an improved measure of edge relevance in the entire network. This cumulative edge relevance using ego regions is calculated for existing edges only. To predict unseen links, we utilize feature sets like common neighbors to predict missing links using the edge strengths of surrounding edges. We use different feature sets to quantify the region of influence between nodes which is most relevant for link prediction. Different paradigms exist for predicting node influence spread away from central nodes such as resource allocation [26], three degree-of-influence [27] as well as cumulative influence in triangular clusters [28]. Since our proposed method aims to predict links using cumulative ego edge strengths, it becomes important to determine which of these edges is most relevant for our link prediction problem.

Contribution - This paper attempts to solve the link prediction problem using a new perspective, believing that the commutative effect of Ego regions of nodes on specific edges can provide a better estimation of the strength of weak edges. Other edge ranking-based approaches ignore the cumulative effect of nodes on ranking edges that are not directly connected to them (clustering-based approaches such as CCLP [29], NLC [30]), hence overlooking the effect some weak edges may have on the overall process of link creation and prediction. This approach can be considered a global similarity-based approach but forgoes the costly matrix operations, which are commonly associated with global similarity and path-based approaches. The major contributions of this work are as follows -

- We introduce the notion of Ego-based edge strengths that simulate all nodes' cumulative effect on all their Ego region edges. This approach is primarily used to understand the actual strength of weak edges, which directly connect two low priority nodes but are an integral part of Ego regions of several nodes.
- We present a new algorithm called ELP based on the above estimation of the strength of existing edges and combine them with different feature sets to predict non-existent links.
- The experimental results show how we choose the best feature sets for link prediction. We also compare the performance of our algorithm with state-of-the-art algorithms to show the benefits of our algorithm. We also perform statistical tests to show that the proposed algorithm is significantly different from other state-of-the-art algorithms.

Organization - The remainder of the paper is organized as follows. Section 3 describes the background information and problem definition. Section 4 presents the framework for ego strength estimation in existing and non existing edges as well as some information on features to be used for link prediction. Section 5 presents the proposed algorithm with example and complexity analysis. Section 7 shows result analysis. Finally, Section 8 is devoted to our conclusion and directions for future work.

2 Related work

Numerous comprehensive studies have been done in order to provide a comprehensive analysis of the link prediction problem and its associated literature [31–37]. These studies have classified link prediction into several classes like similarity-based [26,38], dimensionality reduction-based [39,40], probabilistic and maximum likelihood-based [41,42], learning-based [43], information theory-based [44], etc. The probabilistic link prediction model optimizes an objective function for a particular network in order to construct a model made of several parameters. This model does an excellent job of estimating observed data for the given network. At that point, the probability of a non-existing connection being present is defined as the value of the objective function in the presence of such a link. Techniques for dimensionality reduction fall into two categories: embedding-based and matrix-decomposition-based. This method of link prediction generates feature sets for relevant edges and trains machine learning classifiers on them for classification tasks. While learning-based strategies also make use of machine learning classifiers, the features in this issue category correspond to various attributes of edges on various link prediction indices. Similarity-based link prediction methods are most popular in practice, particularly structural similarities due to their simplicity and efficiency. These methods estimate the similarity index for a pair of individuals based on local, quasi-local, and global topological information. Some local information-based similarity scores (indices) are Common Neighbors [1], Adamic/Adar [38], Resource Allocation [26], Preferential Attachment [1], Jaccard [45], CAR-based Common Neighbor Index (CAR) [46], Local Naive Bayes-based Common Neighbors (LNBCN) [47], Node and Link Clustering coefficient (NLC) [30]. Global similarity indices are computed based on the entire topological structure of the network. Some examples are Katz index [48], Rooted PageRank [49], SimRank [50], etc. Quasi-local similarity indices are the middle ground between local and global indices, which try to incorporate both approaches' important characteristics. Some examples are Local Path [51] and L3 [52].

A different form of solution to the link prediction problem is via graph embedding techniques [39,40,53–55], which can be used in binary classification problems in combination with machine learning algorithms. Logically linear embedding (LLE) [39] and Laplacian eigenmaps [53] are some of the matrix-based graph embedding methods that can be used. However, considering their complicated overall implementation and greater resource intensiveness, they are not scalable. For addressing the scalability issue on large graphs, the sparsity of networks can be used as an improvement factor in targeted algorithms. To deal with the limitations of complete matrix-based embedding methods, DeepWalk, a local embedding-based method that uses local information of random walk, was proposed by Perozzi et al. [40]. DeepWalk preserves higher-order proximity by maximizing the probability of co-occurrence of random walk. The authors of Grover and Leskovec [54] also use a directed random walk model to embed the nodes using a corpus of 2 hop possibilities. For performance improvement, the authors of this Node2Vec algorithm also incorporated the concept of combining both depth-first and breadth-first searches in possible paths. Random walk-based link prediction methods are also considered quasi-local link prediction methods, such as ones by Berahmand et al. [56,57].

The newest subset of link prediction algorithms is community-guided link prediction. A topology-based link prediction algorithm was proposed by Huang Zan [58]. Using a cycle formation model, the generalized clustering coefficient was used as the likelihood score. In 2015, a resolution-based community division-based link prediction approach was presented by Ding et al. [59]. They proposed to use the coarsened resolution to extract community structure in the first step. For computing target link probability, a frequency statistical model is used to distinguish different communities. In 2016, a new similarity feature called community relevance was proposed by Ding et al. [60]. They proposed an amalgamated feature which, in addition to other topological information usually used in other classical link prediction approaches, also uses latent cross-community information. Another recent algorithm was CLP-ID, proposed by Singh et al. [2], that combines a community-based framework and information diffusion principle to calculate the prediction scores for target links. Community detection approaches which use deep learning are also being researched in current times [61–63]. Motif based link prediction has been proposed by Rossi et al. [64]. Multiple similarity based link prediction algorithms were combined with stacked machine learning algorithms to produce improved performance on small datasets by Li et al. [65]. Bastami et al. [66] proposed a gravitation inspired method which combines local, global and community-based features to improve upon the performance of similarity-based methods. In case of signed network link prediction becomes a three class prediction problem but methods which work on unsigned networks have also been successfully used with modification for signed link prediction (Chen et al. [67]). Similarly, Liu et al. [68] used motifs for signed link prediction in complex networks. Due to the advent of cloud computing, link prediction algorithms which are designed specifically for parallel computing platforms are also a growing area of interest. One such proposal was made by Wang et al. [69] which provides community enhanced similarity-based scores to provide an algorithm which can be parallelized across many clusters. Link prediction in other forms of graphs is also a growing topic of research interest, such as in multiplex networks [70].

3 Preliminaries

Social network structures consist of a set of individuals (actors, organizations) and a set of ties (relationships) connecting pairs of individuals. These social structures are represented by graphs $G(V, E)$, where V denotes a set of social actors connected by a set arcs E, which denotes a set of relationships between the actors.

3.1 Measures of tie strength in social networks

Granovetter et al. [71] state that the strength of a tie in social networks is probably a linear combination of four factors emotional intensity, mutual confiding (intimacy), time, and reciprocal services, which characterize the tie. Later, researchers investigated other factors like social distance, emotional support, and structural features. These factors are not equally important, but there is no agreement on their relative importance. Social relationships can be categorized into two classes: weak and strong ties. Strong ties represent more critical relationships, while weak ties are acquaintances. Generally, weak ties are more numerous than strong ties besides their lower strength. Therefore, the cumulative strength of weak ties calculated over the whole network could exceed the directly visible strong ties, and the impact could be substantial.

Several studies [72–74] have been presented in the literature to measure the tie strength. In Gilbert and Karahalios [72], the authors have provided a model to predict relationships in the classification of strong and weak ties. In Gilbert [73], the author has studied contrasting important relationship factors between different social networks, i.e., Facebook and Twitter. In Arnaboldi et al. [74], the authors have proposed a reduced feature set for tie strength prediction and have

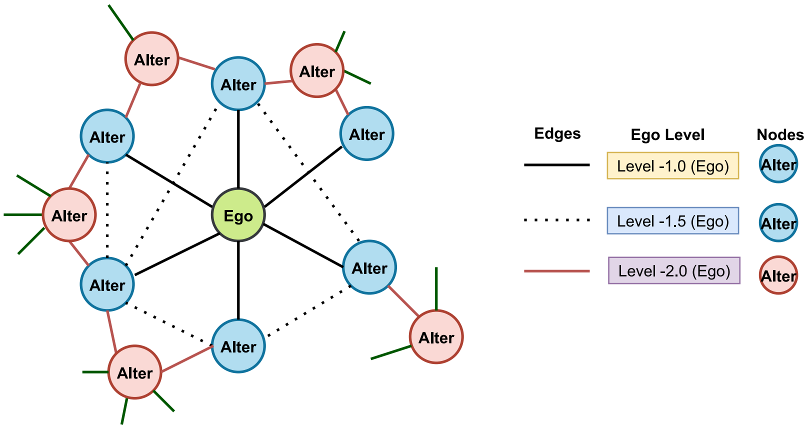


Fig. 1. Ego network structure [18]. Any arbitrary node (Ego) can be envisaged as central node of concentric circles and has relationships with circle nodes (Alters). Each Ego circle has a circle size along with tie strength.

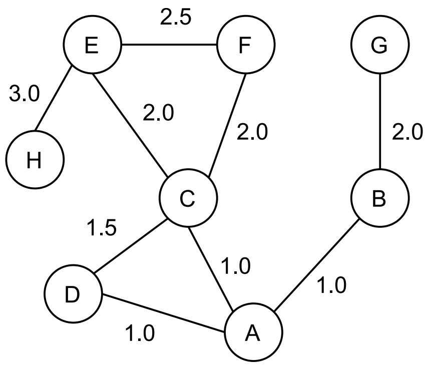


Fig. 2. Ego network structure of node A (Lev.1 is Level-1(A) and so on and so forth) demonstrating the influence node A exerts on edges 3 hop distances away from it. Using edge based ego notation, $Level-1.0(A) = \{A-B, A-C, A-D\}$, $Level-1.5(A) = \{C-D\}$, $Level-2.0(A) = \{C-E, C-F, B-G\}$, $Level-2.5(A) = \{E-F\}$, and $Level-3.0(A) = \{E-H\}$.

shown that the recency of interaction is a much better factor in classifying tie strength than the cumulative closeness of individuals. A study presenting the contrast between time and depth of relationships was given by Marsden et al. [75]. Gilbert and Karahalios [72] work focus on a set of attributes designed by considering all of the seven factors discussed above and presents a study on the Facebook dataset. These studies suggest that some measurable indicators can compute a tie's strength, like frequency of interaction.

3.2 Ego network model

To analyze the micro-level topological features of social networks, some more granular types of subnetworks are considered by researchers. These subnetworks are known as ego networks. The ego networks are formed corresponding to a node (ego) and all the nodes with whom the ego has a connection (alters). The alter nodes are arranged in a series of inclusive groups (circles) in an ego network based on their tie strength. Figuratively, an ego network corresponding to an ego node is depicted in Fig. 1. The initial circle (1) is known as the *support group*, which have alters of strong tie strength with ego node. Informally, the support group nodes are known as best friends and are contacted by the ego in case of financial breakdown, emotional distress, mental stress, etc. The next circle (2) is known as the *sympathy group*, and it contains alters who can be considered close friends. These alters usually contact the ego at least once a month. The last circle (3) is known as the *affinity group*, and contains alters representing casual friends or extended family members.

The literature states that the ego networks are spread out in levels [11], and different levels of ego network of a node A are considered as Level-1.0(A), Level-1.5(A), Level-2.0(A), Level-2.5(A) and Level-3.0(A) as shown in Fig. 2. Level-1.0(A) are the edges connecting A with its direct neighbors (A-B, A-C, A-D) while Level-1.5(A) are edges between these direct neighbors (C-D). Level-2.0(A) are edges connecting direct neighbors with their indirect counterparts (B-G, C-E, C-F) while Level-2.5(A) are those between indirect neighbors (E-F). All the edges at a distance of 3 hops from A which do not belong to Level-2.5(A) are considered as the last circle Level-3.0(A) (E-H). We only consider the region of 3 hops from the node

A following the principle of three degrees of influence [27]. Within this region we use the power law to quantize the influence of each particular level of ego network such that edges belonging to Level-1.0(A), Level-1.5(A), Level-2.0(A), Level-2.5(A) and Level-3.0(A) each have influence equal to $\alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0$ respectively. In this work we use $\alpha = 2$. The formal definition of these levels are as follows.

Definition 1 (*Level-1.0 Ego Network*)**.** The Level-1.0 ego network $\psi^{1.0}(x)$ corresponding to an ego x contains a subnetwork $g(x, V_x, E_x)$ such that if $\exists y \in V$ then $y \in V_x$ iff $\exists (x, y) \in E$, and $\forall (u, v) \in E_x$ satisfies following conditions:

1. $u = x$
2. $v \in V_x$

Definition 2 (*Level-1.5 Ego Network*)**.** The Level-1.5 ego network $\psi^{1.5}(x)$ corresponding to an ego x contains a subnetwork $g(x, V_x, E_x)$ such that if $\exists y \in V$ then $y \in V_x$ iff $\exists (x, y) \in E$, and $\forall (u, v) \in E_x$ satisfies following conditions:

1. $(u = x) \vee (u \in V_x)$
2. $v \in V_x$

Definition 3 (*Level-2.0 Ego Network*)**.** The Level-2.0 ego network $\psi^{2.0}(x)$ corresponding to an ego x contains a subnetwork $g(x, V_x, E_x)$ such that if $\exists y \in V$ then $y \in V_x$ iff $\exists (x, y) \in E$, and satisfies following conditions:

1. $\forall (u, v) \in E_x$ iff $((u = x) \vee (u \in V_x)) \wedge (v \in V_x)$
2. $V_x = \cup_{\forall v \in V_x}(V_x, V_v)$ and $E_x = \cup_{\forall v \in V_x}(E_x, E_v)$, where $g(v, V_v, E_v)$ is $\psi^{1.0}(v)$.

Definition 4 (*Level-2.5 Ego Network*)**.** The Level-2.5 ego network $\psi^{2.5}(x)$ corresponding to an ego x contains a subnetwork $g(x, V_x, E_x)$ such that if $\exists y \in V$ then $y \in V_x$ iff $\exists (x, y) \in E$, and satisfies following conditions:

1. $\forall (u, v) \in E_x$ iff $((u = x) \vee (u \in V_x)) \wedge (v \in V_x)$
2. $V_x = \cup_{\forall v \in V_x}(V_x, V_v)$ and $E_x = \cup_{\forall v \in V_x}(E_x, E_v)$, where $g(v, V_v, E_v)$ is $\psi^{1.5}(v)$.

Definition 5 (*Level-3.0 Ego Network*)**.** The Level-3.0 ego network $\psi^{3.0}(x)$ corresponding to an ego x contains a subnetwork $g(x, V_x, E_x)$ such that if $\exists y \in V$ then $y \in V_x$ iff $\exists (x, y) \in E$, and satisfies following conditions:

1. $\forall (u, v) \in E_x$ iff $((u = x) \vee (u \in V_x)) \wedge (v \in V_x)$
2. $V_x = \cup_{\forall v \in V_x}(V_x, V_v)$ and $E_x = \cup_{\forall v \in V_x}(E_x, E_v)$, where $g(v, V_v, E_v)$ is $\psi^{2.0}(v)$.

3.3 Problem definition

For predicting the most probable links in the network, the link prediction procedure estimates each possible link's likelihood score. The ego network structures are utilized to incorporate an individual's local influence on its neighbors and vice-versa based on small-world phenomena. Therefore, the link prediction problem based on the ego network perspective is defined as follows.

Definition 6 (*Link Prediction*)**.** If $G(V, E)$ is a network graph, where E is current existing edges between V nodes, then the link prediction algorithm investigates the expected future (target, missing) links using likelihood score of non-existing links $\{L_S(x, y); (x, y) \in U \setminus E\}$ based on the ego score of existing links $\{\psi(u, v); (u, v) \in E\}$. Here U is the set of all possible edges.

4 Proposed work

This section discusses the proposed method, which adopts the ego-centric framework by considering interaction dynamics. The ELP algorithm can be divided into three steps. In the first step, the ego strength of each existing link is estimated. Secondly, feature sets are defined for non-existing links based on different topological features. Finally, the algorithm computes the likelihood score of target links.

4.1 Ego strength estimation of existing links

For evaluating the strength of a tie, the pace (interaction frequency) and length of communications (ego distance) are utilized. Some studies [27,76] suggest that an individual influence is limited to its local region based on small world phenomena. With these studies, our proposed method incorporates the interaction dynamics within the three-hop area [27] on ego networks, i.e., length of communications is considered within level 3.0 ego network. Moreover, the pace of interaction is estimated using ego strength $\psi(u, v)$ of an existing link (u, v) and defined as follows.

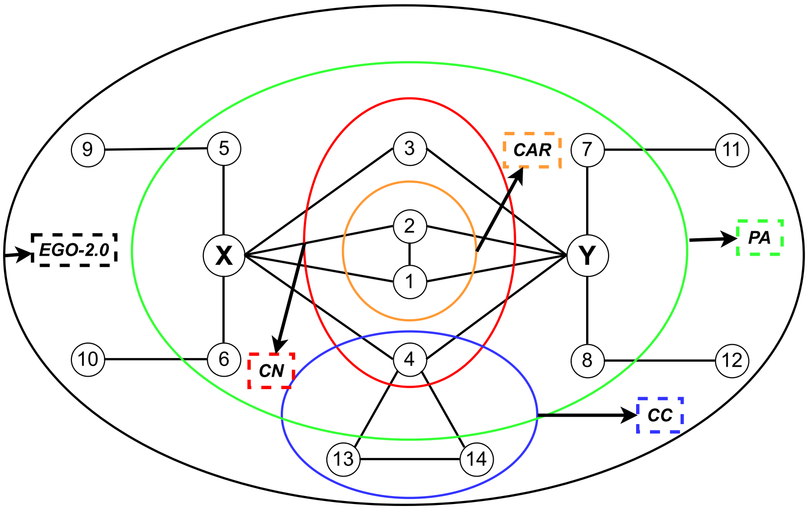


Fig. 3. Example of different regions of feature selection for nodes $X\&Y$ (orange — CAR, red — CN, blue — CC, green — PA, black — Ego-2.0) such that $CN(X,Y) = \{1,2,3,4\}$, $CC(X,Y) = \{4,13,14\}$, $CAR(X,Y) = \{1,2\}$, $PA(X,Y) = \{1,2,3,4,5,6,7,8\}$, and $Ego-2.0(X,Y) = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14\}$.

Definition 7 (*Ego Strength*)**.** If $G(V,E)$ is a network graph, then the ego strength $\psi(u,v)$ of an existing link (u,v) is defined as the average number of existence of an edge (u,v) in local ego networks $i \in \{1.0, 1.5, 2.0, 2.5, 3.0\}$ of an ego w and computed as follows.

$$\psi(u,v) = \sum_{i}\sum_{w\in V} \eta^i_w(u,v) \tag{1}$$

where,

$$\eta^i_w(u,v) = \begin{cases} 1 & \text{if } (u,v) \in \psi^i(w) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The strength of ties in concentric circles of the ego are different, as shown in Fig. 1, due to the frequency of interaction between the ego and its alters. The inner-circle ties corresponding to an ego have more strength than the outer circle ties. To incorporate this behavior, ELP utilizes a ranking strategy that considers higher ranking for inner circles, i.e., $R^i > R^j$ for $i < j$. Both of these strength defining strategies can be used depending on the situation. One possible example is in case of high relevance nodes large amount of information can be shared to larger regions such that Eq. (2) can be used, otherwise use Eq. (3). Ego strength of an edge can also be viewed as the sum of influence of all nodes at a 3-hop distance from the nodes creating the edge. Therefore, Eq. (2) is redefined as follows.

$$\eta^i_w(u,v) = \begin{cases} R^i & \textit{if}(u,v) \in \psi^i(w) \\ 0 & \textit{otherwise} \end{cases} \tag{3}$$

4.2 Feature selection

Now, the proposed algorithm identifies the feature set $\gamma(x,y)$ for each non-existing edge (x,y) based on topological features. These features are explained with example in Fig. 3. Here $CN(X,Y) = \{1,2,3,4\}$, $CC(X,Y) = \{4,13,14\}$, $CAR(X,Y) = \{1,2\}$, $PA(X,Y) = CN(X,Y) \cup \{5,6,7,8\}$, and $Ego-2.0(X,Y) = PA(X,Y) \cup \{9,10,11,12\}$. These are the different topological features which are utilized, and are defined as follows.

1. **Common Neighbors (CN).** In real world, nodes are highly clustered locally with small world phenomena, i.e., nodes with more common neighbors tend to be connected [1]. The common neighbors feature set $\gamma_{CN}(n_1,n_2)$ for a non-existing pair (n_1,n_2) is defined as the set of nodes that are connected to both n_1 and n_2.

$$\gamma_{CN}(n_1,n_2) \leftarrow \{n | n \in \{N(n_1) \cap N(n_2)\}\} \tag{4}$$

where $N(n_1)$ and $N(n_2)$ denotes the neighbors of node n_1 and n_2 respectively.

2. **Preferential Attachment (PA)**. Barabasi et al. [28] considered that nodes with overall more connections, more likely to receive new connections. Therefore, the preferential attachment feature set $\gamma_{PA}(n_1,n_2)$ for a non-existing pair (n_1,n_2) is defined as the set of nodes that is connected to anyone, either n_1 or n_2.

$$\gamma_{PA}(n_1,n_2) \leftarrow \{n | n \in \{N(n_1) \cup N(n_2)\}\} \tag{5}$$

3. **Clustering Coefficient (CC)**. The clustering coefficient is the measure of the degree of those nodes which tend to form the cluster together [29]. Therefore, The clustering coefficient feature set $\gamma_{CC}(n_1, n_2)$ for a non-existing pair (n_1, n_2) is defined as the set of neighbor nodes that tend to form triangles.

$$\gamma_{CC}(n_1, n_2) \leftarrow \{n | n \in \Delta_m\} \tag{6}$$

where Δ_m denotes set of nodes which forms triangles passing through node m, $m \in \{N(n_1) \cap N(n_2)\}$.

4. **CAR**. Cannistraci et al. [46] stated that nodes which belong to the same local community are more likely to have a connection. Therefore, CAR feature set $\gamma_{CAR}(n_1, n_2)$ for a non-existing pair (n_1, n_2) is defined as follows.

$$\gamma_{CAR}(n_1, n_2) \leftarrow \Big\{ n | n \in \big\{ \forall n' \in \{N(n_1) \cap N(n_2)\}, N(n_1) \cap N(n_2) \cap N(n') \big\} \Big\} \tag{7}$$

5. **Ego** $\psi^{2.0}$. This Ego can be said to encompass a significant area away from the central node such that all 2 hop nodes can be said to be influenced by the central node. This feature can be viewed as a combination of *PA* feature set with all nodes which fall at 2 hops from nodes $X\&Y$. It can also be viewed as a union set of nodes falling in Level-2.0 ego regions of nodes $X\&Y$ $(Level - 2.0(X) \cup Level - 2.0(Y))$. Hence, the Ego $\psi^{2.0}$ feature set $\gamma_{EGO-2.0}(n_1, n_2)$ for a non-existing pair (x, y) is defined as follows.

$$\gamma_{EGO-2.0}(n_1, n_2) \leftarrow \{n | n \in \{N(n_1) \cup N(n_2) \cup N(N(n_1)) \cup N(N(n_2))\}\} \tag{8}$$

4.3 Computation of likelihood score of non-existing links

Finally, ELP computes the likelihood score $S_L(x, y)$ of each non-existing link (x, y) based on selected feature set $\gamma(x, y)$ (feature sets and their formulations can be selected from Section 4.2) and ego strength of existing links. For calculating this, we do a summation over all nodes of $\gamma(x, y)$ set (can be selected from CN, CC, CAR, PA, Ego-2.0) and all these nodes are used to calculate a fraction representing the relevance of existing edges between the intermediate node and nodes between which link likelihood has to be calculated ($x\&y$). The denominator of this fraction is the sum of ego strengths of all edges incident on this node and the numerator represents ego strengths of the incident edges from $x\&y$. The ego strength can be calculated using Eq. (1). The likelihood score $S_L(x, y)$ of a non-existing link (x, y) is computed as follows.

$$S_L(x, y) = \sum_{z \in \gamma(x,y)} \frac{\psi(z, x) + \psi(z, y)}{\sum_{a \in N(z)} \psi(z, a)} \tag{9}$$

Algorithm 1: ELP: Ego-centric Link Prediction Algorithm

```
    Input: Social graph: G(V, E)
    Output: Likelihood score of non-existing links: S_L
1   for each existing link (u, v) ∈ E do
2       ψ(u, v) ← Compute ego strength of edge (u, v) using Eq. (1);
3   for each non-existing link (x, y) ∈ U \ E do
4       γ(x, y) ← Estimate the feature of a pair of nodes (x, y);
5       S_L(x, y) ← Compute likelihood score of (x, y) using Eq. (9);
6   Return S_L;
```

5 Algorithm

Algorithm 1 takes a social network graph as input and estimates the likelihood of target links and returns those computed likelihoods as output. The **for** loop in lines 1–3 computes the likelihood of all existing links using Eq. (1). The **for** loop in lines 4–7 estimates the likelihood value of target links using Eq. (9) based on selected feature set. It has been stated in Stolz and Schlereth [25] that there three types of revealed preferences which can be used as predictors of edge strength — similarity of user attributes, interaction among peers and the overall network structure. Our algorithm *ELP* can also be seen to consists of three such comparable parts — for predicting cumulative strength of existing edges we take the whole network structure into account, each edge has a cumulative effect of multiple interactions of node influences and the final link prediction is the calculation of similarity for non existing edges.

5.1 Applying the algorithm

To explain the working of the proposed algorithm ELP, we use an example graph as shown in Fig. 4. The given example graph has 9 nodes and the lines show the connection between them. The proposed algorithm works in three phases given as follows.

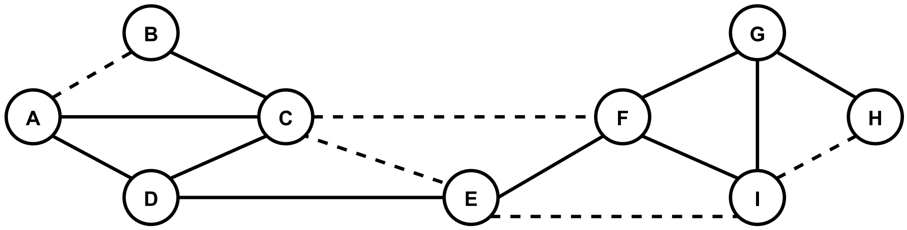


Fig. 4. Example Network for demonstrating the working of *ELP* algorithm for link prediction.

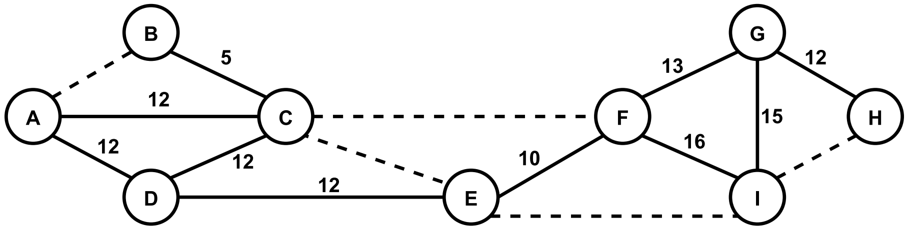


Fig. 5. Ego strength of example network.

- **Ego Strength Estimation:** In this phase, ELP computes the ego strength $\psi(u, v)$ of each existing edge (u, v) using Eq. (1). For example, ego strength of (B, C) can be calculated as $\psi(B, C) = \sum_i \sum_{w \in V} \eta_w^i(B, C)$, where $i \in \{1.0, 1.5, 2.0, 2.5, 3.0\}$. The edge (B, C) exists in ego networks of $\{B, C, A, D, E\}$ under level 3.0 of ego network, so $\psi(B, C) = 5$. Similarly, we can estimate the ego strength of other existing edges as shown in Fig. 5.
- **Feature Selection:** Now, the algorithm computes the feature set $\gamma(x, y)$ for each non-existing pair (x, y) using different topological features as shown in Table 1. For example, the common neighbors (CN) feature set for non-existing edge (A, B) is $\gamma(A, B) = \{C\}$ from Eq. (4) and PA feature set can be computed as $\gamma(A, B) = \{C, D\}$ by Eq. (5). Similarly, we can compute other feature sets CC, CAR, and Ego $\psi^{2.0}$ using Eqs. (6), (7) and (8) respectively.
- **Likelihood Score Computation:** In this phase, ELP estimates likelihood score for each non-existing pair (x, y) based on feature set and ego strength using Eq. (9) as shown in Table 2. For example, the likelihood score of (A, B) can be computed as $S_L(A, B) = \sum_{z \in \gamma(A,B)} \frac{\psi(z,A)+\psi(z,B)}{\sum_{a \in N(z)} \psi(z,a)} = (12 + 5)/(12 + 12 + 5) = 0.5862068$. After that, we need to normalize the likelihood score of each non-existing link by maximum of computed likelihood score. Therefore, the normalize $S_L(A, B) = 0.119847$ and same is shown in Table 2 for all non-existing links based on different feature sets.
- **Predicting Missing Links:** Finally the algorithm ELP predicts missing links based on likelihood score. The standard process for converting this score into predicted label is setting a threshold probability which defines the margin of separation between prediction of edges and non edges. Usually this probability margin is kept at 0.5. The predicted labels are then matched with actual labels of edges to check for accuracy of the proposed approach. Some performance metrics like AUC and AUPR also use the predicted probabilities directly to create a curve with varying thresholds on different axes (Precision/Recall and TPR/FPR) and then take area of this curve as a measure of performance.

5.2 *Complexity analysis*

Based on different topological features, the proposed algorithm presents different variants: EGO-CN, EGO-PA, EGO-CC, EGO-CAR, and EGO-2.0. Assuming the average degree of a node is D_{avg}, the feature creation process takes worst time complexities of $\mathcal{O}(D_{avg}^2)$, $\mathcal{O}(D_{avg}^2)$, $\mathcal{O}(D_{avg}^3)$, $\mathcal{O}(D_{avg}^3)$ and $\mathcal{O}(D_{avg}^4)$ respectively for CN, PA, CC, CAR and $\psi^{2.0}$ feature sets ($|FC|$). The calculation of probability over this node set is represented by FS. In Algorithm 1, the first steps in lines 1–2 is the initial strength calculation part for existing edges. For this, all nodes and their respective Ego regions would have to be taken into account to correctly estimate the cumulative strength of edges. The total complexity of these steps would be $\mathcal{O}(|V| * D_{avg}^3)$ for the whole graph. Here, $|V|$ is the total number of nodes and D_{avg}^3 represents visit into $\psi^{3.0}$ region around the node. The next phase of the algorithm is predicting the likelihood score of non existent edges. For each edges, the complexity can be divided into two major parts, one is generating features (line 4) and second is strength estimation (line 5). So the overall complexity for calculation of each likelihood score would be $\mathcal{O}(|FC| + |FS| * D_{avg})$, where the first term $|FC|$ is for feature set calculation and the second term $|FS| * D_{avg}$ is for calculation of probability over nodes of those feature sets. Essentially, we can see that time taken for calculation of likelihood is directly dependent on number of nodes in feature set. Hence the overall complexity of the algorithm would be $\mathcal{O}((|V| * D_{avg}^3) + |V^2| * (|FC| + |FS| * D_{avg}))$

Table 1
Feature set selection.

Non existing	Feature sets				
Edges	CN	PA	CC	CAR	Ego $\psi^{2.0}$
A-B	'C'	'C', 'D'	'A', 'C', 'D'	'D'	'E', 'C', 'D'
A-E	'D'	'C', 'D', 'F'	'A', 'C', 'D'	'C'	'I', 'G', 'B', 'C', 'D', 'F'
A-F	–	'I', 'G', 'C', 'D', 'E'	–	–	'I', 'G', 'H', 'B', 'C', 'D', 'E'
A-G	–	'I', 'H', 'C', 'D', 'F'	–	–	'I', 'H', 'B', 'C', 'D', 'F', 'E'
A-H	–	'G', 'C', 'D'	–	–	'I', 'G', 'B', 'C', 'D', 'F', 'E'
A-I	–	'G', 'C', 'D', 'F'	–	–	'G', 'H', 'B', 'C', 'D', 'F', 'E'
B-D	'C'	'A', 'E', 'C'	'A', 'C', 'D'	'A'	'A', 'E', 'C', 'F'
B-E	–	'C', 'D', 'F'	–	–	'A', 'I', 'G', 'C', 'D', 'F'
B-F	–	'I', 'E', 'G', 'C'	–	–	'A', 'I', 'G', 'H', 'C', 'D', 'E'
B-G	–	'I', 'C', 'H', 'F'	–	–	'A', 'I', 'H', 'C', 'D', 'F', 'E'
B-H	–	'G', 'C'	–	–	'A', 'I', 'G', 'C', 'D', 'F'
B-I	–	'G', 'C', 'F'	–	–	'A', 'G', 'H', 'C', 'D', 'F', 'E'
C-E	'D'	'A', 'F', 'B', 'D'	'A', 'C', 'D'	'A'	'A', 'I', 'G', 'B', 'D', 'F'
C-F	–	'A', 'I', 'G', 'B', 'D', 'E'	–	–	'A', 'I', 'G', 'H', 'B', 'D', 'E'
C-G	–	'A', 'I', 'H', 'B', 'D', 'F'	–	–	'A', 'I', 'H', 'B', 'D', 'F', 'E'
C-H	–	'A', 'G', 'B', 'D'	–	–	'A', 'I', 'G', 'B', 'D', 'F', 'E'
C-I	–	'A', 'G', 'B', 'D', 'F'	–	–	'A', 'G', 'H', 'B', 'D', 'F', 'E'
D-F	'E'	'A', 'I', 'G', 'C', 'E'	–	–	'A', 'I', 'G', 'H', 'B', 'C', 'E'
D-G	–	'A', 'I', 'H', 'C', 'F', 'E'	–	–	'A', 'I', 'H', 'B', 'C', 'F', 'E'
D-H	–	'A', 'E', 'G', 'C'	–	–	'A', 'I', 'G', 'B', 'C', 'F', 'E'
D-I	–	'A', 'G', 'C', 'F', 'E'	–	–	'A', 'G', 'H', 'B', 'C', 'F', 'E'
E-G	'F'	'I', 'H', 'D', 'F'	'I', 'G', 'F'	'I'	'A', 'I', 'H', 'C', 'D', 'F'
E-H	–	'G', 'D', 'F'	–	–	'A', 'I', 'G', 'C', 'D', 'F'
E-I	'F'	'G', 'D', 'F'	'I', 'G', 'F'	'G'	'A', 'G', 'H', 'C', 'D', 'F'
F-H	'G'	'I', 'G', 'E'	'I', 'G', 'F'	'I'	'I', 'G', 'E', 'D'
H-I	'G'	'G', 'F'	'I', 'G', 'F'	'F'	'G', 'E', 'F'

Table 2
Likelihood score computation.

Non existing	Likelihood score				
Edges	CN	PA	CC	CAR	Ego $\psi^{2.0}$
A-B	0.119847	0.172097	0.455022	0.164118	0.175515
A-E	0.114001	0.221181	0.445778	0.172534	0.225574
A-F	0	0.365226	0	0	0.293032
A-G	0	0.435459	0	0	0.344798
A-H	0	0.181597	0	0	0.096068
A-I	0	0.268316	0	0	0.094888
B-D	0.084892	0.281266	0.480667	0.280368	0.286852
B-E	0	0.133975	0	0	0.093347
B-F	0	0.271278	0	0	0.170978
B-G	0	0.341524	0	0	0.338619
B-H	0	0.074904	0	0	0.043289
B-I	0	0.180623	0	0	0.095074
C-E	0.114001	0.786209	0.635289	0.356832	0.801825
C-F	0	0.930254	0	0	0.869283
C-G	0	0.928829	0	0	0.927415
C-H	0	0.668725	0	0	0.672318
C-I	0	0.761686	0	0	0.677504
D-F	0.214226	0.54006	0	0	0.550787
D-G	0	0.694935	0	0	0.62929
D-H	0	0.363161	0	0	0.264686
D-I	0	0.456134	0	0	0.385746
E-G	0.109251	0.474963	0.550356	0.280368	0.484397
E-H	0	0.14993	0	0	0.109619
E-I	0.123501	0.236162	0.517244	0.219589	0.240852
F-H	0.124841	0.373148	0.692044	0.38232	0.38056
H-I	0.099872	0.171123	0.452178	0.223797	0.174522

for calculating link likelihoods over all possible edges. In this formulation if we substitute for the best feature set, i.e., CN, we get $\mathcal{O}((|V| * D_{avg}^3) + (|V^2| * D_{avg}^2))$, which can be further simplified to $\mathcal{O}(|V^2| * D_{avg}^2)$. Here $|FC| \leftarrow D_{avg}^2$ and $|FS| \leftarrow D_{avg}$. Since *CN*, *PA*, *CCLP* (representing clustering coefficient *CC* based method) and *CAR* are also link prediction approaches on their own, we have attempted to provide a quantitative comparison of running time of these link prediction approaches with our proposed *ELP* ($EGO - CN$ variation used for representation in these running time calculation context) algorithm in Table 3. The experiment have been run on five different *Ratio* values between 0.1&0.5 which are ratio of testing to

Table 3
Running time analysis (in seconds) for different *Ratio* values representing testing to total edges percentage in five datasets.

Dataset	Ratio	CN	CAR	PA	CCLP	ELP
Karate	0.1	0.038	0.049	0.026	0.056	0.045
	0.2	0.037	0.045	0.026	0.052	0.043
	0.3	0.036	0.043	0.027	0.049	0.041
	0.4	0.036	0.042	0.026	0.045	0.036
	0.5	0.036	0.041	0.026	0.044	0.035
Jazz	0.1	2.404	4.221	0.598	5.051	6.342
	0.2	2.356	3.972	0.566	4.350	5.211
	0.3	1.919	3.232	0.567	3.700	4.444
	0.4	1.766	2.758	0.582	3.110	3.616
	0.5	1.564	2.373	0.618	2.587	2.709
Celegansneural	0.1	3.361	5.251	1.258	5.457	6.903
	0.2	3.217	4.467	1.257	4.965	5.676
	0.3	2.939	3.995	1.244	4.479	4.643
	0.4	2.778	3.557	1.243	3.839	3.787
	0.5	2.607	3.117	1.241	3.382	3.099
Airlines	0.1	1.821	2.979	0.809	3.464	4.898
	0.2	1.741	2.847	0.777	3.040	4.172
	0.3	1.668	2.415	0.768	2.750	3.371
	0.4	1.565	2.139	0.763	2.302	2.697
	0.5	1.441	2.063	0.792	2.062	2.143
SmaGri	0.1	32.351	40.045	15.497	41.275	56.216
	0.2	30.855	36.967	15.463	38.237	48.018
	0.3	29.411	34.361	15.376	35.323	38.863
	0.4	28.064	31.984	15.331	32.661	32.499
	0.5	26.775	29.818	15.338	30.172	27.745

Table 4
Statistical information of real-world datasets.

Dataset	N	E	D	K	C
Karate	34	78	2.34	4.59	0.57
Jazz	198	2742	2.224	27.697	0.617
Airlines	235	1297	2.31	11.04	0.558
Celegansneural	297	2148	2.45	14.47	0.29
Political blogs	1490	16 718	2.738	22.44	0.361
SmaGri	1059	4917	2.981	9.286	0.349
GrQc	5242	14 496	6.047	5.531	0.53

total edges of dataset. The running times for our algorithm can been observed to be comparable to other standard link prediction algorithms. The description for these small datasets can be found in Section 6.1.

6 Experimental setup

6.1 Datasets

To validate our proposed algorithm ELP, we use six real-world networks to compare its performance with other methods. Karate[1] [77] dataset contains social ties among the members of a university karate club collected by Wayne Zachary in 1977. Jazz[2] [78] is a collaboration network of jazz musicians which shows musicians as nodes and edges represent their respective collaborations with each other. Airlines[3] [79] is a US airline network where nodes and edges represent airports and the connectivity between airports. Celegansneural[4] [80] is a neural network of C. Elegans. Nodes represent neurons, and edges denote connections by either a synapse or a gap junction. Political blogs[5] [38] is a network of hyperlinks between weblogs on US politics, recorded in 2005 by Adamic and Glance. SmaGri[6] [81] is the undirected form of a citation network from Garfield collection which represents the results searches in Web of Science. It was made using HistCite software. GrQc[7] is collaboration network from arXiv for research in General Relativity and Quantum Cosmology

1 http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/ucidata.htm
2 http://networkrepository.com/jazz.php
3 http://vlado.fmf.uni-lj.si/pub/networks/data/
4 https://neurodata.io/project/connectomes/
5 http://www-personal.umich.edu/mejn/netdata/
6 http://vlado.fmf.uni-lj.si/pub/networks/data/cite/default.htm
7 https://snap.stanford.edu/data/ca-GrQc.html

category Table 4 provides the statistical information about datasets used for experimental analysis. N, E, D, K, and C represent the number of nodes, number of edges, average shortest distance between a pair of node, the average degree of node, and average clustering coefficient of the network respectively. All of the experiments performed on a 64-bit Linux Mint 19.3 PC with Intel(R) Core(TM) i7-4770 CPU@ 3.40 GHz processor and 32 GB memory. Python was used as language of programming all the algorithms. The code for link prediction using the proposed ELP algorithm is available on Github at https://github.com/shivansh-mishra/linkpredict-static. Each experiment was executed for each algorithm and testing edges to total edges percent value (*Ratio*) 20 times.

6.2 Evaluation metrics

Hasan et al. [82] structured the link prediction issue as a binary classification problem, allowing for the application of the majority of associated evaluation metrics. A confusion matrix may be used to illustrate the assessment of a binary classification issue with two classes [83]. Three measures are used to assess the proposed method *ELP*: accuracy, area under the precision–recall curve (AUPR), and area under the curve (AUC).

- **Accuracy -** Accuracy score is the simplest kind of measure that may be used to evaluate the effectiveness of a classification system. It is just the sum of all accurate predictions divided by the whole sample size. The link prediction problem's valid predictions may be specified using the link existence probability. If the likelihood is greater than 0.5 and an actual connection exists in the graph, the classification is regarded accurate.
- **AUPR -** For binary classification issues, AUPR is more informative and beneficial [3,83]. As a result, we employed it as a forecasting measure. The AUPR values are determined using the precision–recall curve, in which the x- and y-axes reflect the recall and precision values, respectively.
- **AUC -** The area under the receiver operating characteristics curve (AUROC/AUC) [3,83] plots TPR (y-axis) against FPR (x-axis). The AUC value is a single-point statistical summary with a range of 0–1 and is estimated using the trapezoidal rule.

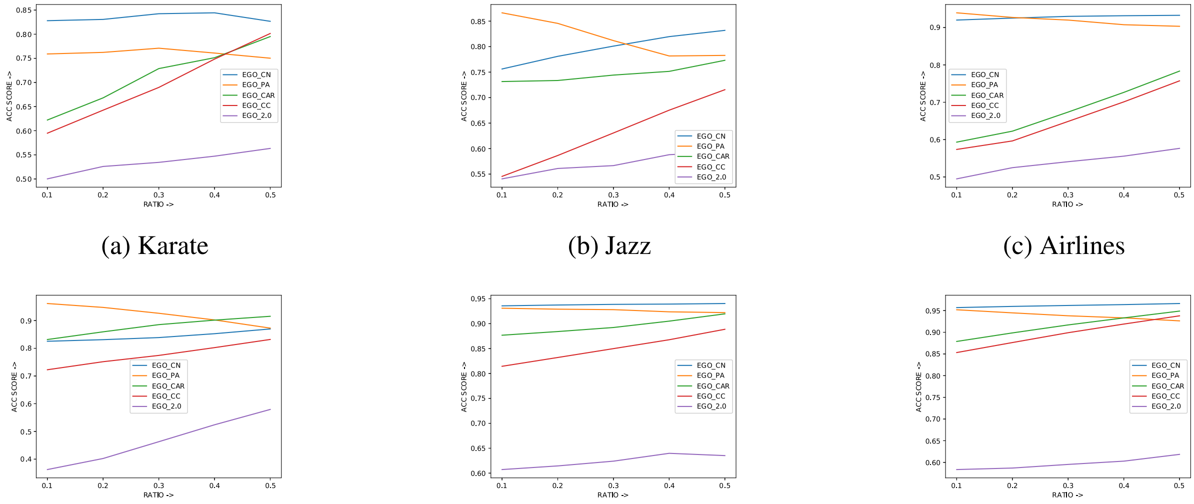
6.3 Performance comparison methods

Newman et al. [1] stated that the similarity score between a pair of nodes is dependent on the number of common neighbors between them and called the method Common Neighbor (CN). Zhou et al. [26] proposed the RA index for link prediction using diffusion model to penalize higher-degree nodes and called it Resource Allocation (RA). Cannistraci et al. [46] suggested a similarity score using the local community paradigm. They proposed CAR variants of CN, JA, AA, and RA of which we use the CN variant. Wu et al. [29] used the clustering coefficient to get a better understanding of the strength of a possible link and proposed Clustering coefficient link prediction ($CCLP$). Jaccard Coefficient (JC) is one of the oldest metric proposed by Jaccard [45] and is a bit similar to Common Neighbor based similarity score. Barabasi et al. [28] considered an opposite approach to the diffusion paradigm specifically for co-authorship networks and called it Preferential attachment (PA). *PageRank* is an adjustment of *Katz* centrality that takes into consideration the fact that the centrality gain using a link from an important node should be decreased depending on how many nodes the central node is connected to (resource allocation). Grover et al. [54] presented a network embedding based method for link prediction. $N2V(Node2vec)$ creates embeddings for nodes in a low dimensional space and then formulates edge embeddings from these node embeddings. These are used as training data for a logistic regression based classification model. Singh et al. [2] proposed an algorithm, $CLP - ID$, which combines a community-based framework and information diffusion to calculate the prediction scores for target links. Finally, ELP is our proposed algorithm. We also try to investigate the relationship between our algorithm's performance based on different feature sets ($EGO-CN$, $EGO-PA$, $EGO-CAR$, $EGO-CC$, $EGO-2.0$). We use three metrics in these experiments: Accuracy Score, AUPR and AUC. We also consider five different ratios (0.1, 0.2, 0.3, 0.4, 0.5) of testing set edges to total edges of graph datasets while the remaining ones are considered training datasets.

7 Empirical analysis

7.1 Accuracy score on features

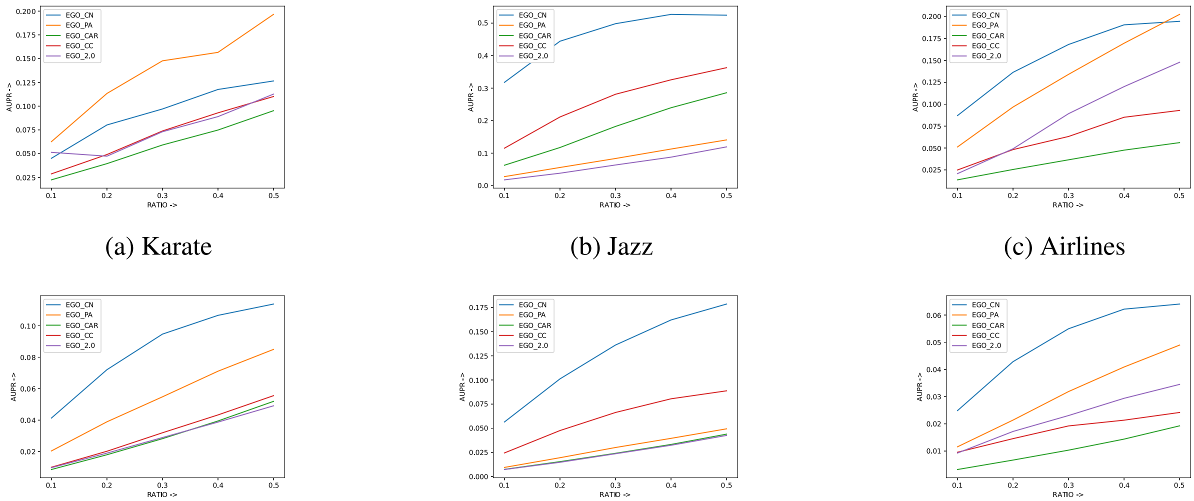
As is evident from Fig. 6 in all datasets and all algorithms, the $EGO - 2.0$ based algorithm performs worst. This shows that the feature set generated using $EGO - 2.0$ is spread too far from the relevant edge influence to provide a reliable estimation of its effect. $EGO - CC$ and $EGO - CAR$ are usually in the middle-of-the-pack based on performance, while $EGO - CN$ is best in most cases and trades places with $EGO - PA$ in few others. They also present the most negligible variation based on different datasets' ratios, making them a reasonably good choice for the Accuracy Score metric.



(a) Karate (b) Jazz (c) Airlines

(d) Celegansneural (e) Political blogs (f) SmaGri

Fig. 6. Accuracy score comparison for different feature sets on six datasets.



(a) Karate (b) Jazz (c) Airlines

(d) Celegansneural (e) Political blogs (f) SmaGri

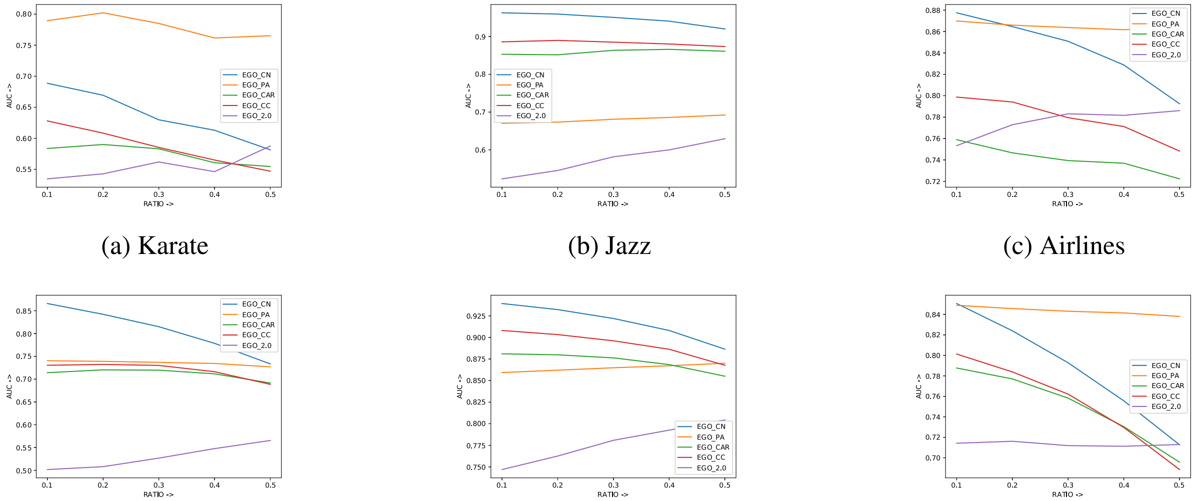
Fig. 7. AUPR Comparison for different feature sets on six datasets.

7.2 AUPR on features

As is evident from Fig. 7 in all datasets and all algorithms, $EGO-CN$ performs best in all cases except the Karate dataset, which is comparatively a tiny dataset. For most cases in the $EGO-CAR$ algorithm, the worst performance is seen, while $EGO-PA$, $EGO-CC$, and $EGO-2.0$ can be considered to be middle-of-the-pack algorithms. A point to be noted here is that in 5 out of 6 datasets, $EGO-CN$'s performance is far above other algorithms for all ratios, making this a clear choice for the AUPR metric.

7.3 AUC on features

As is evident from Fig. 8 in all datasets and all algorithms, $EGO-CN$ performs best for high-density datasets while $EGO-PA$ performs better in mid-density ones. The worst performance is observed in $EGO-2.0$, while $EGO-CC$ and $EGO-CAR$ can be considered the middle-of-the-pack algorithms. It is noted here that $EGO-PA$ shows much more stable performance degradation with an increasing ratio of testing set edges. The other algorithms can be seen to have a sharp



(a) Karate (b) Jazz (c) Airlines

(d) Celegansneural (e) Political blogs (f) SmaGri

Fig. 8. AUC Comparison for different feature sets on six datasets.

dip as the increase of ratio. Henceforth, $EGO - CN$ is the algorithm which is compared with state-of-art algorithms and will be referred as *ELP*.

7.4 Measuring accuracy score

The nature of social networks considered in these experiments is inherently sparse. This is because compared to the total number of possible links for a sizable sized graph, $n * n$ for a set of n nodes, the actual number of interactions would be quite low. Accuracy score is fundamentally a measure of how exactly the set of predicted and actual labels match. In Table 5, we can observe that our proposed algorithm *ELP* performs the best on all datasets except for the Jazz dataset. Even in Jazz, *ELP* is the second-best performing algorithm, with *CAR* being the best here. This can be attributed to the distribution of a higher number of local communities in the Jazz dataset on the occurrence of which *CAR* is based. This is very impressive, especially considering how strict the process of calculation of the Accuracy score is. We can conclude that our algorithm works well with sparse graphs in terms of accuracy at least when the ratio of edges to nodes is less than equal to 10 approximately.

7.5 Measuring AUPR

In Table 6, we can see that overall our proposed *ELP* algorithm can be considered the third-best in performance out of all the algorithms considered. For the smallest Karate dataset, our algorithm's metrics lack behind *CAR*, *PA*, and $Node2V$. It can be observed that $Node2v$ and *PA* show a gradual increase in performance while *CAR*'s metrics can be considered to be analogous. In the Jazz dataset, our algorithm is the third-best algorithm, just behind *RA* and *CCLP*. However, even among these, the numbers are very close and comparable. Our algorithm is the fourth-best performing algorithm in the Airlines dataset, behind *RA*, *CCLP*, and *PA*. In Celegansneural dataset, our algorithm is the best performing out of all the state-of-art algorithms considered for comparison. In the Political Blogs dataset, we can observe our algorithm's worst ranking behind *CN*, *RA*, *CCLP*, and $CLP - ID$. Our algorithm performs better in the SmaGri dataset than most just behind *CAR* and is comprehensively outperformed by *CCLP*. Our algorithm does not perform well in the GrQc dataset and is sixth best algorithm out of all state-of-art algorithms. When considering the overall pattern, it can be seen that for the AUPR metric, our algorithm can be worse than only RA and CCLP in most cases. We can conclude that in terms of AUPR, the overall relative performance of our algorithm decreases as the ratio of edges to nodes increases in a dataset, Karate dataset being an exception to this pattern.

7.6 Measuring AUC

In Table 7, we can see that overall our proposed *ELP* algorithm can be considered the second-best performing algorithm out of all the algorithms considered. Consistently it is only outperformed by PA in some cases. For the Karate dataset, which is the smallest of all datasets, our algorithm is worse than *RA*, *PA*, *PageRank*, and $Node2v$. For the Jazz dataset, our algorithm is ranked behind only *RA*. For the Airlines dataset, our algorithm gives the second-best numbers only behind

Table 5
Comparison of the proposed algorithm ELP with the state-of-the-art algorithms in terms of Accuracy Score (2 best values from each *Rafto* row have been highlighted).

Dataset	Ratio	Algorithm									
		CN	RA	CAR	CCLP	JC	PA	PAGERANK	NODE2V	CLP-ID	ELP
Karate	0.1	0.51329	**0.77008**	0.09862	0.7626	0.63071	0.73248	0.75187	0.692	0.61112	**0.85305**
	0.2	0.58702	**0.76599**	0.18721	0.73014	0.64719	0.73081	0.74884	0.69174	0.63527	**0.84797**
	0.3	0.64685	0.72786	0.25115	0.72863	0.69342	0.70076	**0.75353**	0.68909	0.68187	**0.84246**
	0.4	0.71842	0.725	0.07406	0.75226	0.72585	0.69117	**0.75573**	0.6776	0.72312	**0.84605**
	0.5	0.7628	0.76336	0.0808	**0.80371**	0.75445	0.73878	0.76883	0.68903	0.76855	**0.82644**
Jazz	0.1	0.6864	0.71759	**0.86118**	0.72671	0.70072	0.6187	0.74297	0.75718	0.69823	**0.76226**
	0.2	0.73769	0.74253	**0.89441**	0.74534	0.71859	0.62515	0.75477	0.76592	0.73504	**0.7844**
	0.3	0.69867	0.76514	**0.90701**	0.76446	0.73382	0.63736	0.76466	0.7744	0.73773	**0.80243**
	0.4	0.75868	0.78085	**0.93506**	0.77951	0.74345	0.64285	0.77398	0.78593	0.76158	**0.8204**
	0.5	0.8172	0.79688	**0.93096**	0.79814	0.75439	0.65259	0.7815	0.79869	0.78516	**0.83402**
Celegansneural	0.1	0.5316	0.75317	**0.98236**	0.73179	0.65228	0.68218	0.75481	0.76788	0.65089	**0.82819**
	0.2	0.59847	0.76439	0.71568	0.74387	0.65643	0.68399	0.75923	**0.77721**	0.65211	**0.83321**
	0.3	0.66171	**0.79515**	0.14553	0.75869	0.68701	0.68033	0.76444	0.79068	0.68671	**0.84128**
	0.4	0.72561	**0.80639**	0.04019	0.79099	0.74095	0.69091	0.77322	0.79875	0.73699	**0.85463**
	0.5	0.78532	**0.83175**	0.03458	0.82862	0.78775	0.70028	0.78076	0.80445	0.79165	**0.86894**
Airlines	0.1	0.7331	0.79097	**0.91797**	0.75324	0.62322	0.74891	0.76111	0.74467	0.73224	**0.92641**
	0.2	0.78152	0.81413	**0.95318**	0.76828	0.63724	0.75955	0.77599	0.75852	0.72307	**0.92972**
	0.3	0.57431	**0.81628**	0.35535	0.7876	0.65389	0.77262	0.78294	0.77271	0.68587	**0.9347**
	0.4	0.64761	**0.82686**	0.07379	0.78021	0.68307	0.77939	0.7925	0.77067	0.68802	**0.93354**
	0.5	0.72791	**0.83959**	0.05316	0.76384	0.7465	0.78271	0.80917	0.79239	0.74128	**0.93742**
Polblogs	0.1	0.76697	0.85337	**0.96425**	0.84653	0.79989	0.81685	0.81835	0.82852	0.809	**0.93542**
	0.2	0.79023	0.86031	**0.97544**	0.85229	0.81168	0.8192	0.82258	0.832	0.81078	**0.93676**
	0.3	0.81683	0.8678	**0.98396**	0.85956	0.82847	0.82077	0.82774	0.83542	0.82551	**0.93839**
	0.4	0.84374	0.87924	**0.98928**	0.86565	0.84902	0.82242	0.83407	0.83964	0.84876	**0.93844**
	0.5	0.87347	**0.88647**	0.79735	0.88447	0.8752	0.82462	0.84171	0.83851	0.87548	**0.94014**
SmaGri	0.1	0.82497	0.87312	**0.99701**	0.83958	0.83574	0.78293	0.80849	0.82812	0.82618	**0.95748**
	0.2	0.85348	**0.88238**	0.80072	0.85723	0.85964	0.78286	0.81615	0.83515	0.85432	**0.96024**
	0.3	0.88103	0.88203	0.7012	0.88108	**0.88502**	0.7798	0.82607	0.84293	0.88285	**0.96302**
	0.4	0.90841	0.90761	0.40313	**0.90965**	0.9094	0.78259	0.83636	0.85255	0.90842	**0.96479**
	0.5	0.93079	0.93088	0.2042	**0.93444**	0.93175	0.78842	0.85039	0.86369	0.9315	**0.96679**
GrQc	0.1	0.9959	0.99589	**0.99982**	0.99595	0.9959	0.79237	0.94655	0.94968	0.99597	**0.99598**
	0.2	0.99658	0.99659	**0.99979**	0.99666	0.99659	0.78475	0.95233	0.95148	0.99664	**0.99659**
	0.3	0.99722	0.99721	**0.99973**	0.99731	0.9972	0.78014	0.95858	0.95435	0.99726	**0.99728**
	0.4	0.99777	0.99777	**0.99965**	0.9979	0.99776	0.78273	0.96563	0.95766	**0.99781**	0.99777
	0.5	0.99825	0.99824	**0.99954**	**0.99841**	0.99824	0.77726	0.97275	0.96106	0.99828	0.99824

PA. In Celeganseural dataset, it performs on par with *RA* and is only behind PageRank for some train-to-test ratios. In the Political Blogs dataset, our *ELP* algorithm again performs on par with RA and is only outperformed by *PA*. In the SmaGri dataset, our algorithm is ranked third only behind *PA* and *PageRank*. In the GrQc dataset, our algorithm is ranked third only behind *Node*2v and *PageRank*. We can conclude that the overall relative performance of our algorithm remains mostly consistent for all datasets. The best performing algorithm may change but *ELP* consistently can be considered to be the second best performing algorithm, Karate dataset being the only exception to this pattern.

7.7 Statistical tests

This section compares the different state-of-the-art algorithms with *ELP* and analyzes the significant differences between them. This comparison is made for the metrics of Accuracy, AUPR and AUC. Friedman's test [84] was applied to highlight significant differences between other algorithms compared with *ELP*. The result was hypothesis rejection in all cases. Friedman Siegel's Test [85] was applied as a post hoc procedure to estimate each hypothesis's degree of rejection. *ELP* algorithm was considered as control algorithm, and degree of freedom and confidence level was considered to be 9 and 0.05, respectively. This was done to gain a better measure of the significant difference between our proposed *ELP* algorithm and other algorithms. The statistical tests on different accuracy metrics (Accuracy, AUPR and AUC) demonstrate that the proposed algorithm is significantly different (≤ 0.05) from the state-of-the-art algorithms. From Table 8, we can observe the level of significant differences between our proposed algorithm and other standard algorithms. In Table 8. the combined ratio indicates that the statistical test is performed simultaneously for different sets of observed links.

8 Concluding remarks and future works

This paper presents an Ego-based link prediction algorithm (*ELP*), which uses an ego-based link strength estimation perspective to predict target links. Classical algorithms do not consider the cumulative effect of node-based influence

Table 6
Comparison of the proposed algorithm ELP with the state-of-the-art algorithms in terms of AUPR (2 best values from each *Rafto* row have been highlighted).

Dataset	Ratio	Algorithm									
		CN	RA	CAR	CCLP	JC	PA	PAGERANK	NODE2V	CLP-ID	ELP
Karate	0.1	0.02576	0.0515	**0.07502**	0.03449	0.01862	0.04751	0.03847	**0.07433**	0.04461	0.0401
	0.2	0.05695	0.08042	0.08128	0.07755	0.03748	0.08678	0.0637	**0.10402**	0.05773	**0.08847**
	0.3	0.08015	0.10149	0.08579	0.08721	0.0555	**0.12118**	0.0907	**0.11649**	0.09055	0.09949
	0.4	0.08758	0.11348	**0.37281**	0.09712	0.07958	**0.142**	0.10255	0.12704	0.09641	0.10096
	0.5	0.09859	0.12247	**0.42255**	0.11197	0.0861	**0.16942**	0.11326	0.12848	0.115	0.12463
Jazz	0.1	0.33076	**0.33473**	0.32855	**0.33092**	0.27146	0.08406	0.10002	0.12925	0.31694	0.2902
	0.2	0.43159	**0.46065**	0.41725	**0.45577**	0.37989	0.13642	0.181	0.21575	0.42905	0.42394
	0.3	0.48758	**0.51989**	0.43009	**0.5142**	0.4363	0.17861	0.24408	0.26028	0.47659	0.486
	0.4	0.51123	**0.53607**	0.39263	**0.53138**	0.45724	0.21254	0.29509	0.29409	0.49734	0.51486
	0.5	0.50856	**0.54145**	0.32334	**0.53278**	0.45094	0.24085	0.332	0.31156	0.50356	0.5221
Celegansneural	0.1	0.0365	0.0395	0.02884	**0.0419**	0.01593	0.02422	0.02592	0.01933	0.03281	**0.04207**
	0.2	0.0629	**0.0719**	0.05165	**0.07281**	0.02915	0.04538	0.04801	0.03614	0.05965	0.0707
	0.3	0.08032	0.09095	0.07506	**0.09741**	0.03996	0.06321	0.06881	0.04744	0.07836	**0.09375**
	0.4	0.09589	0.10522	0.07608	**0.10762**	0.04868	0.07885	0.08506	0.05612	0.08827	**0.10532**
	0.5	0.10177	**0.10949**	0.08366	0.10964	0.05602	0.09529	0.09809	0.06098	0.09561	**0.1118**
Airlines	0.1	0.09233	0.09983	0.06645	**0.10954**	0.0094	**0.2113**	0.02814	0.01681	0.09087	0.09081
	0.2	0.15564	0.15624	0.08411	**0.16736**	0.01611	**0.28483**	0.05408	0.03035	0.13933	0.14499
	0.3	0.16747	0.18645	0.09411	**0.1943**	0.02348	**0.33529**	0.07924	0.04603	0.16139	0.17225
	0.4	0.17919	0.2005	0.11728	**0.21344**	0.03194	**0.36527**	0.10255	0.04503	0.17521	0.18735
	0.5	0.17755	0.20292	0.12345	**0.21015**	0.04215	**0.4054**	0.11942	0.06162	0.1637	0.19767
Polblogs	0.1	**0.07604**	0.06348	0.06652	0.07439	0.01666	0.0321	0.01982	0.01442	**0.07577**	0.05487
	0.2	0.12866	0.11184	0.10598	**0.13129**	0.03019	0.06069	0.03861	0.02595	**0.13179**	0.09761
	0.3	0.16641	0.14945	0.12936	**0.17203**	0.03997	0.08595	0.05647	0.03519	**0.16987**	0.13195
	0.4	0.19595	0.17468	0.14118	**0.19967**	0.04803	0.11059	0.07352	0.04068	**0.19743**	0.15919
	0.5	0.21155	0.18832	0.15333	**0.21791**	0.05494	0.13111	0.08935	0.04494	**0.21385**	0.1756
SmaGri	0.1	0.02759	0.02749	**0.02827**	**0.03256**	0.00344	0.01399	0.01222	0.00447	0.02548	0.02484
	0.2	0.045	0.0472	**0.05313**	**0.0524**	0.00676	0.02884	0.02248	0.00835	0.04196	0.04215
	0.3	0.05597	0.0563	**0.07643**	**0.06441**	0.01003	0.03978	0.03156	0.01164	0.05344	0.05648
	0.4	0.06185	0.06161	**0.08762**	**0.07142**	0.01401	0.04888	0.03904	0.01398	0.05732	0.06168
	0.5	0.06034	0.06258	**0.0817**	**0.06798**	0.01862	0.05871	0.04393	0.01503	0.05611	0.06393
GrQc	0.1	**0.24113**	0.14225	**0.26749**	0.24026	0.09092	0.01511	0.04081	0.03713	0.23008	0.12941
	0.2	**0.30737**	0.20958	**0.36901**	0.29863	0.12588	0.02595	0.07191	0.06652	0.28935	0.19491
	0.3	**0.34211**	0.2442	**0.42756**	0.33092	0.14979	0.03188	0.09507	0.08854	0.31203	0.22624
	0.4	**0.3527**	0.25575	**0.45468**	0.33903	0.16287	0.03537	0.10865	0.10707	0.33028	0.2417
	0.5	**0.35369**	0.25412	**0.44581**	0.34655	0.1724	0.03749	0.11551	0.11559	0.33249	0.24353

propagation on edges to predict target links, but that is our algorithm's specialty. The closest comparison to this approach can be found in path counting algorithms dependent on adjacency matrix-based operations, which are computationally very expensive. *ELP* performs exceptionally well in the Accuracy metric, a combined representation of the prediction performance of both existent and non-existent edges. For other metrics, i.e., AUPR and AUC, it can be observed that *ELP*'s performance is better on datasets with an average degree greater than 10. This makes our algorithm more suitable for link prediction of networks with the magnitude of edges much larger than nodes. These include social networking site-based datasets like Facebook and Twitter.

Some future possibilities exist to extend this work. Our *ELP* algorithm, at its core, considers an ego to be the region of influence of a node. This influence is combined for all nodes on a single edge to estimate the edge's total significance. Other formulation models of the region of influence can be used, for example, an information diffusion-based perspective. Also, variable-sized regions of influence based on node properties can be explored. Also, since the summation of node influences is our algorithm's basic proposition, dynamic networks might be a good fit for this algorithm.

CRediT authorship contribution statement

Shivansh Mishra: Conceptualization, Methodology, Data Curation, Software, Writing. **Shashank Sheshar Singh:** Data curation, Writing – original draft. **Ajay Kumar:** Data curation, Writing – original draft. **Bhaskar Biswas:** Proof reading, Principal Supervisor.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 7
Comparison of the proposed algorithm ELP with the state-of-the-art algorithms in terms of AUC (2 best values from each *Rafto* row have been highlighted).

Dataset	Ratio	Algorithm									
		CN	RA	CAR	CCLP	JC	PA	PAGERANK	NODE2V	CLP-ID	ELP
Karate	0.1	0.59741	**0.77007**	0.50585	0.66894	0.59037	0.73215	0.6967	**0.83975**	0.64446	0.68136
	0.2	0.64067	0.68187	0.4982	0.68429	0.58995	0.69963	0.66762	**0.77521**	0.63362	**0.71088**
	0.3	0.61682	0.63064	0.48634	0.61975	0.57043	**0.68063**	0.66556	**0.71387**	0.6337	0.64974
	0.4	0.58239	0.61778	0.48828	0.57986	0.56613	**0.68437**	0.62615	**0.66414**	0.60214	0.59264
	0.5	0.5627	0.58794	0.48903	0.561	0.5317	**0.66777**	0.60845	**0.61241**	0.58274	0.57401
Jazz	0.1	0.9535	**0.96444**	0.92283	0.95298	0.9569	0.76487	0.88705	0.90866	0.94891	**0.96021**
	0.2	0.9479	**0.95955**	0.88594	0.95129	0.9491	0.76853	0.88607	0.90155	0.94383	**0.95933**
	0.3	0.9397	**0.95504**	0.807	0.9436	0.93912	0.76836	0.88549	0.89056	0.93651	**0.94967**
	0.4	0.92975	**0.94132**	0.6596	0.9334	0.92767	0.76374	0.88187	0.88188	0.92651	**0.93852**
	0.5	0.91371	**0.92496**	0.4913	0.91686	0.90731	0.76223	0.87625	0.86719	0.90874	**0.92366**
Celegansneural	0.1	0.84661	0.86288	0.47126	**0.86387**	0.78924	0.75489	0.82643	0.80675	0.83403	**0.87036**
	0.2	0.82333	0.84415	0.43939	**0.84218**	0.76924	0.75024	0.81993	0.80109	0.82153	**0.8423**
	0.3	0.7899	**0.81466**	0.43869	0.80947	0.75433	0.75119	0.8142	0.77768	0.79052	**0.81613**
	0.4	0.76471	**0.78059**	0.44734	0.77393	0.72648	0.74354	**0.80571**	0.75605	0.75827	0.77807
	0.5	0.72345	0.73458	0.47016	0.72008	0.69651	**0.74176**	**0.79183**	0.73275	0.7213	0.73185
Airlines	0.1	0.86615	0.87758	0.67254	**0.87801**	0.68851	0.87734	0.78104	0.74528	0.86589	**0.88537**
	0.2	0.85658	0.85923	0.57665	0.86158	0.68137	**0.87638**	0.78146	0.7377	0.8518	**0.86899**
	0.3	0.83433	0.85054	0.4829	0.838	0.68469	**0.87349**	0.78546	0.74043	0.83752	**0.85049**
	0.4	0.81323	0.82567	0.4464	0.82135	0.68366	**0.86938**	0.78792	0.70984	0.81991	**0.82984**
	0.5	0.77537	0.78815	0.43236	0.78791	0.69073	**0.85826**	0.78495	0.71633	0.77888	**0.79546**
Polblogs	0.1	0.93535	**0.93833**	0.74303	0.93676	0.90478	0.9316	0.91358	0.88915	0.93591	**0.93794**
	0.2	0.92992	**0.93264**	0.68279	0.93083	0.8999	0.93075	0.91386	0.88086	0.93125	**0.93189**
	0.3	0.919	**0.92225**	0.608	0.92166	0.89079	**0.93015**	0.91338	0.87576	0.92036	0.92161
	0.4	0.90526	0.90721	0.52933	0.90807	0.87922	**0.92872**	**0.9126**	0.86806	0.90661	0.9071
	0.5	0.88459	0.88643	0.46304	0.88599	0.86089	**0.92644**	**0.91251**	0.86253	0.88479	0.88476
SmaGri	0.1	0.83682	0.84599	0.47605	**0.8506**	0.79177	0.84521	0.84923	0.78622	0.83416	**0.85061**
	0.2	0.81453	0.82075	0.4528	0.81879	0.77838	**0.84489**	**0.84334**	0.77231	0.8145	0.82025
	0.3	0.7876	0.79452	0.45016	0.78731	0.7576	**0.83875**	**0.83503**	0.76674	0.78651	0.7995
	0.4	0.75188	0.75557	0.45846	0.74767	0.73201	**0.82776**	**0.82697**	0.75488	0.75035	0.7573
	0.5	0.70371	0.7087	0.47438	0.69913	0.69674	**0.81938**	**0.81703**	0.7327	0.70994	0.71215
GrQc	0.1	**0.92317**	0.92216	0.59801	0.89274	0.92192	0.74126	0.91282	0.91459	0.92217	**0.92226**
	0.2	0.89417	0.89432	0.59551	0.86405	0.89438	0.73934	0.89482	**0.89622**	0.89528	**0.89612**
	0.3	0.86304	0.86297	0.58704	0.83081	0.86333	0.73797	**0.87636**	**0.87231**	0.86312	0.86316
	0.4	0.82587	0.82714	0.57187	0.79042	0.82579	0.73424	**0.85365**	**0.84837**	0.8287	0.82753
	0.5	0.78622	0.7871	0.53615	0.74926	0.78557	0.73249	**0.82968**	**0.81971**	0.78613	0.78589

Table 8
The Posthoc Friedman Siegel Test (Control method = ELP) corresponding different metrics.

Metric	Ratio	p-value								
		CN	RA	CAR	CCLP	JC	PA	PAGERANK	NODE2V	CLP-ID
ACCURACY	0.1	3.70E−05	0.073366	0.508148	0.149804	0.002725	0.000295	0.009108	0.026506	0.001846
	0.2	0.00016	0.128996	0.067329	0.128996	0.002396	0.000295	0.014196	0.014196	0.001616
	0.3	0.000116	0.17308	0.009108	0.149804	0.019517	0.00016	0.014196	0.021643	0.014196
	0.4	0.001846	0.185686	0.010198	0.436275	0.015807	5.20E−05	0.003971	0.005069	0.023968
	0.5	0.015807	0.212912	0.000938	0.559305	0.015807	5.00E−06	0.001616	0.001616	0.079839
AUPR	0.1	0.697092	0.350201	0.533417	0.185686	0.010198	0.275758	0.061707	0.023968	0.815335
	0.2	1	0.533417	0.87627	0.212912	0.001846	0.161125	0.029273	0.010198	0.533417
	0.3	0.533417	0.697092	0.350201	0.391805	0.000708	0.119471	0.019517	0.003093	0.459559
	0.4	0.96895	0.483522	0.845687	0.227558	0.002725	0.227558	0.056479	0.00449	0.61284
	0.5	0.533417	0.815335	0.436275	0.755497	0.001414	0.139101	0.010198	0.001414	0.436275
AUC	0.1	0.051625	0.755497	1.00E−06	0.119471	0.001077	0.000815	0.04296	0.029273	0.073366
	0.2	0.051625	0.697092	1.00E−06	0.139101	0.002396	0.015807	0.161125	0.161125	0.161125
	0.3	0.119471	0.87627	9.00E−06	0.212912	0.008123	0.119471	0.755497	0.311515	0.139101
	0.4	0.086768	1	6.00E−06	0.161125	0.003093	0.242908	0.697092	0.391805	0.161125
	0.5	0.242908	0.61284	7.20E−05	0.119471	0.008123	0.815335	0.212912	1	0.413686

Compliance with ethical standards

The authors declare no conflicts of interest. The article does not contain any studies with human or animal subjects. This article presents a novel link prediction algorithm using ego network perspective.

References

[1] M.E.J. Newman, Clustering and preferential attachment in growing networks, Phys. Rev. E 64 (2001) 025102, http://dx.doi.org/10.1103/PhysRevE.64.025102, URL: https://journals.aps.org/pre/abstract/10.1103/PhysRevE.64.025102.

[2] S.S. Singh, S. Mishra, A. Kumar, B. Biswas, CLP-ID: Community-based link prediction using information diffusion, Inform. Sci. 514 (2020) 402–433, http://dx.doi.org/10.1016/j.ins.2019.11.026, URL: https://www.sciencedirect.com/science/article/abs/pii/S0020025519310734.

[3] A. Kumar, S.S. Singh, K. Singh, B. Biswas, Level-2 node clustering coefficient-based link prediction, Appl. Intell. (2019) http://dx.doi.org/10.1007/s10489-019-01413-8, URL: https://link.springer.com/article/10.1007/s10489-019-01413-8.

[4] A. Kumar, S. Mishra, S.S. Singh, K. Singh, B. Biswas, Link prediction in complex networks based on significance of higher-order path index (SHOPI), Physica A (2019) 123790, http://dx.doi.org/10.1016/j.physa.2019.123790, URL: https://www.sciencedirect.com/science/article/abs/pii/S0378437119321107.

[5] S.S. Singh, K. Singh, A. Kumar, H.K. Shakya, B. Biswas, A survey on information diffusion models in social networks, in: A.K. Luhach, D. Singh, P.-A. Hsiung, K.B.G. Hawari, P. Lingras, P.K. Singh (Eds.), Advanced Informatics for Computing Research, Springer Singapore, Singapore, 2019, pp. 426–439, http://dx.doi.org/10.1007/978-981-13-3143-5_35, URL: https://link.springer.com/chapter/10.1007/978-981-13-3143-5_35.

[6] S.S. Singh, A. Kumar, K. Singh, B. Biswas, C2IM: Community based context-aware influence maximization in social networks, Physica A 514 (2019) 796–818, http://dx.doi.org/10.1016/j.physa.2018.09.142, URL: https://www.sciencedirect.com/science/article/abs/pii/S0378437118312822.

[7] S.S. Singh, K. Singh, A. Kumar, B. Biswas, MIM2: Multiple influence maximization across multiple social networks, Physica A 526 (2019) 120902, http://dx.doi.org/10.1016/j.physa.2019.04.138, URL: https://www.sciencedirect.com/science/article/abs/pii/S037843711930500X.

[8] S.S. Singh, A. Kumar, K. Singh, B. Biswas, LAPSO-IM: A learning-based influence maximization approach for social networks, Appl. Soft Comput. 82 (2019) 105554, http://dx.doi.org/10.1016/j.asoc.2019.105554, URL: https://www.sciencedirect.com/science/article/abs/pii/S1568494619303345.

[9] S.S. Singh, A. Kumar, K. Singh, B. Biswas, IM-SSO: Maximizing influence in social networks using social spider optimization, Concurr. Comput.: Pract. Exper. 32 (2) (2020) e5421, http://dx.doi.org/10.1002/cpe.5421, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5421.

[10] S.S. Singh, K. Singh, A. Kumar, B. Biswas, ACO-IM: maximizing influence in social networks using ant colony optimization, Soft Comput. (2019) 1–23, http://dx.doi.org/10.1007/s00500-019-04533-y, URL: https://link.springer.com/article/10.1007/s00500-019-04533-y.

[11] A. Biswas, B. Biswas, Investigating community structure in perspective of ego network, Expert Syst. Appl. 42 (20) (2015) 6913–6934, http://dx.doi.org/10.1016/j.eswa.2015.05.009, URL: https://www.sciencedirect.com/science/article/abs/pii/S0957417415003292.

[12] A. Biswas, B. Biswas, FuzAg: Fuzzy agglomerative community detection by exploring the notion of self-membership, IEEE Trans. Fuzzy Syst. 26 (5) (2018) 2568–2577, http://dx.doi.org/10.1109/TFUZZ.2018.2795569, URL: https://ieeexplore.ieee.org/document/8263601.

[13] G. Zhang, J. Wu, J. Yang, A. Beheshti, S. Xue, C. Zhou, Q.Z. Sheng, FRAUDRE: Fraud detection dual-resistant to graph inconsistency and imbalance, in: 2021 IEEE International Conference on Data Mining (ICDM), 2021, pp. 867–876, http://dx.doi.org/10.1109/ICDM51629.2021.00098, URL: https://ieeexplore.ieee.org/document/9679178.

[14] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q.Z. Sheng, H. Xiong, L. Akoglu, A comprehensive survey on graph anomaly detection with deep learning, IEEE Trans. Knowl. Data Eng. (2021) 1, http://dx.doi.org/10.1109/TKDE.2021.3118815, URL: https://ieeexplore.ieee.org/document/9565320/.

[15] G. Pang, C. Shen, L. Cao, A.V.D. Hengel, Deep learning for anomaly detection: A review, ACM Comput. Surv. 54 (2) (2021) http://dx.doi.org/10.1145/3439950, URL: https://dl.acm.org/doi/10.1145/3439950.

[16] N. Moustafa, J. Hu, J. Slay, A holistic review of network anomaly detection systems: A comprehensive survey, J. Netw. Comput. Appl. 128 (2019) 33–55, http://dx.doi.org/10.1016/j.jnca.2018.12.006, URL: https://www.sciencedirect.com/science/article/pii/S1084804518303886.

[17] D. Liben-Nowell, J. Kleinberg, The link prediction problem for social networks, in: Proceedings of the Twelfth International Conference on Information and Knowledge Management, in: CIKM 2003, ACM, New York, NY, USA, 2003, pp. 556–559, http://dx.doi.org/10.1145/956863.956972, URL: https://dl.acm.org/doi/10.1145/956863.956972.

[18] V. Arnaboldi, M. Conti, M.L. Gala, A. Passarella, F. Pezzoni, Ego network structure in online social networks and its impact on information diffusion, Comput. Commun. 76 (2016) 26–41, http://dx.doi.org/10.1016/j.comcom.2015.09.028, URL: https://www.sciencedirect.com/science/article/abs/pii/S014036641500465X.

[19] T. Raeder, O. Lizardo, D. Hachen, N.V. Chawla, Predictors of short-term decay of cell phone contacts in a large scale communication network, Social Networks 33 (4) (2011) 245–257, http://dx.doi.org/10.1016/j.socnet.2011.07.002, URL: https://www.sciencedirect.com/science/article/abs/pii/S0378873311000463.

[20] T. Tylenda, R. Angelova, S. Bedathur, Towards time-aware link prediction in evolving social networks, in: Proceedings of the 3rd Workshop on Social Network Mining and Analysis, in: SNA-KDD 2009, Association for Computing Machinery, New York, NY, USA, 2009, pp. 1–10, http://dx.doi.org/10.1145/1731011.1731020, URL: https://dl.acm.org/doi/10.1145/1731011.1731020.

[21] M. Berlingerio, A. Gionis, B. Bringmann, F. Bonchi, Learning and predicting the evolution of social networks, IEEE Intell. Syst. 25 (04) (2010) 26–35, http://dx.doi.org/10.1109/MIS.2010.91, URL: https://ieeexplore.ieee.org/document/5552587.

[22] L. Tabourier, A. Libert, R. Lambiotte, Predicting links in ego-networks using temporal information, EPJ Data Sci. 5 (2016) http://dx.doi.org/10.1140/epjds/s13688-015-0062-0, URL: https://epjdatascience.springeropen.com/articles/10.1140/epjds/s13688-015-0062-0.

[23] M. Toprak, C.L. Boldrini, A. Passarella, M. Conti, Harnessing the power of ego network layers for link prediction in online social networks, IEEE Trans. Comput. Soc. Syst. (2022) 1–13, http://dx.doi.org/10.1109/TCSS.2022.3155946, URL: https://ieeexplore.ieee.org/document/9733385.

[24] A. Rezaeipanah, G. Ahmadi, S. Sechin Matoori, A classification approach to link prediction in multiplex online ego-social networks, Soc. Netw. Anal. Min. 10 (2020) http://dx.doi.org/10.1007/s13278-020-00639-6, URL: https://link.springer.com/article/10.1007/s13278-020-00639-6.

[25] S. Stolz, C. Schlereth, Predicting tie strength with ego network structures, J. Interact. Mark. 54 (2021) 40–52, http://dx.doi.org/10.1016/j.intmar.2020.10.001, URL: https://www.sciencedirect.com/science/article/pii/S1094996820301390.

[26] T. Zhou, L. Lü, Y.-C. Zhang, Predicting missing links via local information, Eur. Phys. J. B 71 (4) (2009) 623–630, http://dx.doi.org/10.1140/epjb/e2009-00335-8, URL: https://link.springer.com/article/10.1140/epjb/e2009-00335-8.

[27] N.A. Christakis, J.H. Fowler, S.K. Walker, Connected: The surprising power of our social networks and how they shape our lives, J. Family Theory Rev. 3 (3) (2011) 220–224, http://dx.doi.org/10.1111/j.1756-2589.2011.00097.x, URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1756-2589.2011.00097.x.

[28] A. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, T. Vicsek, Evolution of the social network of scientific collaborations, Physica A 311 (3) (2002) 590–614, http://dx.doi.org/10.1016/S0378-4371(02)00736-7, URL: https://www.sciencedirect.com/science/article/abs/pii/S0378437102007367.

[29] Z. Wu, Y. Lin, J. Wang, S. Gregory, Link prediction with node clustering coefficient, Physica A 452 (2016) 1–8, http://dx.doi.org/10.1016/j.physa.2016.01.038, URL: https://www.sciencedirect.com/science/article/abs/pii/S0378437116000777.

[30] Z. Wu, Y. Lin, H. Wan, W. Jamil, Predicting top-l missing links with node and link clustering information in large-scale networks, J. Stat. Mech. Theory Exp. 8 (2016) 083202, http://dx.doi.org/10.1088/1742-5468/2016/08/083202, URL: https://iopscience.iop.org/article/10.1088/1742-5468/2016/08/083202.

[31] L. Lü, T. Zhou, Link prediction in complex networks: A survey, Physica A 390 (6) (2011) 1150–1170, http://dx.doi.org/10.1016/j.physa.2010.11.027, URL: https://www.sciencedirect.com/science/article/pii/S037843711000991X.

[32] P. Wang, B. Xu, Y. Wu, X. Zhou, Link prediction in social networks: the state-of-the-art, Sci. China Inf. Sci. 58 (1) (2015) 1–38, http://dx.doi.org/10.1007/s11432-014-5237-y, URL: https://link.springer.com/article/10.1007/s11432-014-5237-y.
[33] V. Martínez, F. Berzal, J.-C. Cubero, A survey of link prediction in complex networks, ACM Comput. Surv. 49 (4) (2016) http://dx.doi.org/10.1145/3012704, URL: https://dl.acm.org/doi/10.1145/3012704.
[34] S. Haghani, M.R. Keyvanpour, A systemic analysis of link prediction in social network, Artif. Intell. Rev. 52 (3) (2019) 1961–1995, http://dx.doi.org/10.1007/s10462-017-9590-2, URL: https://link.springer.com/article/10.1007/s10462-017-9590-2.
[35] A. Kumar, S.S. Singh, K. Singh, B. Biswas, Link prediction techniques, applications, and performance: A survey, Physica A 553 (2020) 124289, http://dx.doi.org/10.1016/j.physa.2020.124289, URL: https://www.sciencedirect.com/science/article/pii/S0378437120300856.
[36] N.N. Daud, S.H. Ab Hamid, M. Saadoon, F. Sahran, N.B. Anuar, Applications of link prediction in social networks: A review, J. Netw. Comput. Appl. 166 (2020) 102716, http://dx.doi.org/10.1016/j.jnca.2020.102716, URL: https://www.sciencedirect.com/science/article/pii/S1084804520301909.
[37] T. Zhou, Progresses and challenges in link prediction, iScience 24 (11) (2021) 103217, http://dx.doi.org/10.1016/j.isci.2021.103217, URL: https://www.sciencedirect.com/science/article/pii/S2589004221011858.
[38] L.A. Adamic, E. Adar, Friends and neighbors on the web, Social Networks 25 (3) (2003) 211–230, http://dx.doi.org/10.1016/S0378-8733(03)00009-1, URL: https://www.sciencedirect.com/science/article/abs/pii/S0378873303000091.
[39] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326, http://dx.doi.org/10.1126/science.290.5500.2323, URL: https://science.sciencemag.org/content/290/5500/2323.
[40] B. Perozzi, R. Al-Rfou, S. Skiena, DeepWalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD 2014, Association for Computing Machinery, New York, NY, USA, 2014, pp. 701–710, http://dx.doi.org/10.1145/2623330.2623732, URL: https://dl.acm.org/doi/10.1145/2623330.2623732.
[41] C. Wang, V. Satuluri, S. Parthasarathy, Local probabilistic models for link prediction, in: Seventh IEEE International Conference on Data Mining (ICDM 2007), 2007, pp. 322–331, http://dx.doi.org/10.1109/ICDM.2007.108, URL: https://ieeexplore.ieee.org/document/4470256.
[42] K. Yu, W. Chu, S. Yu, V. Tresp, Z. Xu, Stochastic relational models for discriminative link prediction, in: B. Schölkopf, J.C. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems 19, MIT Press, 2007, pp. 1553–1560, URL: http://papers.nips.cc/paper/2998-stochastic-relational-models-for-discriminative-link-prediction.pdf.
[43] J.R. Doppa, J. Yu, P. Tadepalli, L. Getoor, Learning algorithms for link prediction based on chance constraints, in: J.L. Balcázar, F. Bonchi, A. Gionis, M. Sebag (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 344–360, http://dx.doi.org/10.1007/978-3-642-15880-3_28, URL: https://link.springer.com/chapter/10.1007/978-3-642-15880-3_28.
[44] K. Anand, G. Bianconi, Entropy measures for networks: toward an information theory of complex topologies, Phys. Rev. E 80 (4 Pt 2) (2009) 045102, http://dx.doi.org/10.1103/physreve.80.045102, URL: https://journals.aps.org/pre/abstract/10.1103/PhysRevE.80.045102.
[45] P. Jaccard, Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines, Bull. Soc. Vaudoise Sci. Nat. 37 (1901) 241–272.
[46] C.V. Cannistraci, G. Alanis-Lobato, T. Ravasi, From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks, Sci. Rep. 3 (1) (2013) http://dx.doi.org/10.1038/srep01613, URL: https://www.nature.com/articles/srep01613.
[47] Z. Liu, Q.-M. Zhang, L. Lü, T. Zhou, Link prediction in complex networks: A local naïve Bayes model, Europhys. Lett. 96 (4) (2011) 48007, http://dx.doi.org/10.1209/0295-5075/96/48007, URL: https://iopscience.iop.org/article/10.1209/0295-5075/96/48007.
[48] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1) (1953) 39–43, http://dx.doi.org/10.1007/BF02289026, URL: https://link.springer.com/article/10.1007/BF02289026.
[49] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, in: Proceedings of the Seventh International World Wide Web Conference, Vol. 30, Computer Networks and ISDN Systems, 1998, pp. 107–117, http://dx.doi.org/10.1016/S0169-7552(98)00110-X, URL: https://www.sciencedirect.com/science/article/abs/pii/S016975529800110X.
[50] G. Jeh, J. Widom, SimRank: A measure of structural-context similarity, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD 2002, ACM, New York, NY, USA, 2002, pp. 538–543, http://dx.doi.org/10.1145/775047.775126, URL: https://dl.acm.org/doi/10.1145/775047.775126.
[51] L. Lü, C.-H. Jin, T. Zhou, Similarity index based on local paths for link prediction of complex networks, Phys. Rev. E 80 (2009) 046122, http://dx.doi.org/10.1103/PhysRevE.80.046122, URL: https://journals.aps.org/pre/abstract/10.1103/PhysRevE.80.046122.
[52] I.A. Kovács, K. Luck, K. Spirohn, Y. Wang, C. Pollis, S. Schlabach, W. Bian, D.-K. Kim, N. Kishore, T. Hao, M.A. Calderwood, M. Vidal, A.-L. Barabási, Network-based prediction of protein interactions, BioRxiv (2018) http://dx.doi.org/10.1038/s41467-019-09177-y, URL: https://www.nature.com/articles/s41467-019-09177-y.
[53] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, in: NIPS 2001, MIT Press, Cambridge, MA, USA, 2001, pp. 585–591, http://dx.doi.org/10.5555/2980539.2980616, URL: https://dl.acm.org/doi/10.5555/2980539.2980616.
[54] A. Grover, J. Leskovec, Node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD 2016, Association for Computing Machinery, New York, NY, USA, 2016, pp. 855–864, http://dx.doi.org/10.1145/2939672.2939754, URL: https://dl.acm.org/doi/10.1145/2939672.2939754.
[55] S.M. Kazemi, D. Poole, Simple embedding for link prediction in knowledge graphs, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, in: NIPS 2018, Curran Associates Inc., Red Hook, NY, USA, 2018, pp. 4289–4300, URL: https://dl.acm.org/doi/10.5555/3327144.3327341.
[56] K. Berahmand, E. Nasiri, S. Forouzandeh, Y. Li, A preference random walk algorithm for link prediction through mutual influence nodes in complex networks, J. King Saud Univ. - Comput. Inf. Sci. (2021) http://dx.doi.org/10.1016/j.jksuci.2021.05.006, URL: https://www.sciencedirect.com/science/article/pii/S1319157821001099.
[57] K. Berahmand, E. Nasiri, M. Rostami, S. Forouzandeh, A modified DeepWalk method for link prediction in attributed social network, Computing (2021) http://dx.doi.org/10.1007/s00607-021-00982-2, Springer.
[58] Z. Huang, Link prediction based on graph topology: The predictive value of generalized clustering coefficient, 2010, http://dx.doi.org/10.2139/ssrn.1634014,
[59] J. Ding, L. Jiao, J. Wu, Y. Hou, Y. Qi, Prediction of missing links based on multi-resolution community division, Physica A 417 (2015) 76–85, http://dx.doi.org/10.1016/j.physa.2014.09.005, URL: https://www.sciencedirect.com/science/article/abs/pii/S0378437114007638.
[60] J. Ding, L. Jiao, J. Wu, F. Liu, Prediction of missing links based on community relevance and ruler inference, Knowl.-Based Syst. 98 (2016) 200–215, http://dx.doi.org/10.1016/j.knosys.2016.01.034, URL: https://www.sciencedirect.com/science/article/abs/pii/S095070511600054X.
[61] F. Liu, S. Xue, J. Wu, C. Zhou, W. Hu, C. Paris, S. Nepal, J. Yang, P.S. Yu, Deep learning for community detection: Progress, challenges and opportunities, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, in: IJCAI 2020, 2021, http://dx.doi.org/10.5555/3491440.3492133, URL: https://dl.acm.org/doi/abs/10.5555/3491440.3492133.
[62] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin, Q.Z. Sheng, P.S. Yu, A comprehensive survey on community detection with deep learning, IEEE Trans. Neural Netw. Learn. Syst. (2022) 1–21, http://dx.doi.org/10.1109/TNNLS.2021.3137396, URL: https://ieeexplore.ieee.org/document/9732192.

[63] F. Liu, Z. Li, B. Wang, J. Wu, J. Yang, J. Huang, Y. Zhang, W. Wang, S. Xue, S. Nepal, Q.Z. Sheng, ERiskCom: an e-commerce risky community detection platform, VLDB J. (2022) http://dx.doi.org/10.1007/s00778-021-00723-z, URL: https://link.springer.com/article/10.1007/s00778-021-00723-z.
[64] R.A. Rossi, A. Rao, S. Kim, E. Koh, N.K. Ahmed, G. Wu, From closing triangles to higher-order motif closures for better unsupervised online link prediction, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Association for Computing Machinery, New York, NY, USA, 2021, pp. 4085–4093, http://dx.doi.org/10.1145/3459637.3481920, URL: https://dl.acm.org/doi/abs/10.1145/3459637.3481920.
[65] K. Li, L. Tu, L. Chai, Ensemble-model-based link prediction of complex networks, Comput. Netw. 166 (2020) 106978, http://dx.doi.org/10.1016/j.comnet.2019.106978, URL: https://www.sciencedirect.com/science/article/pii/S1389128619308710.
[66] E. Bastami, A. Mahabadi, E. Taghizadeh, A gravitation-based link prediction approach in social networks, Swarm Evol. Comput. 44 (2019) 176–186, http://dx.doi.org/10.1016/j.swevo.2018.03.001, URL: https://www.sciencedirect.com/science/article/pii/S2210650217304704.
[67] G. Chen, C. Xu, J. Wang, J. Feng, J. Feng, Nonnegative matrix factorization for link prediction in directed complex networks using PageRank and asymmetric link clustering information, Expert Syst. Appl. 148 (2020) 113290, http://dx.doi.org/10.1016/j.eswa.2020.113290, URL: https://www.sciencedirect.com/science/article/pii/S0957417420301159.
[68] S.-Y. Liu, J. Xiao, X.-K. Xu, Link prediction in signed social networks: From status theory to motif families, IEEE Trans. Netw. Sci. Eng. 7 (3) (2020) 1724–1735, http://dx.doi.org/10.1109/TNSE.2019.2951806, URL: https://ieeexplore.ieee.org/document/8892638.
[69] J. Wang, Y. Ma, M. Liu, W. Shen, Link prediction based on community information and its parallelization, IEEE Access 7 (2019) 62633–62645, http://dx.doi.org/10.1109/ACCESS.2019.2907202, URL: https://ieeexplore.ieee.org/document/8673750.
[70] S. Mishra, S.S. Singh, A. Kumar, B. Biswas, MNERLP-MUL: Merged node and edge relevance based link prediction in multiplex networks, J. Comput. Sci. 60 (2022) 101606, http://dx.doi.org/10.1016/j.jocs.2022.101606, URL: https://www.sciencedirect.com/science/article/pii/S1877750322000369.
[71] M.S. Granovetter, The strength of weak ties, Am. J. Sociol. 78 (6) (1973) 1360–1380, http://dx.doi.org/10.1086/225469, URL: https://www.journals.uchicago.edu/doi/10.1086/225469.
[72] E. Gilbert, K. Karahalios, Predicting tie strength with social media, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, in: CHI 2009, Association for Computing Machinery, New York, NY, USA, 2009, pp. 211–220, http://dx.doi.org/10.1145/1518701.1518736, URL: https://dl.acm.org/doi/10.1145/1518701.1518736.
[73] E. Gilbert, Predicting tie strength in a new medium, in: Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, in: CSCW 2012, Association for Computing Machinery, New York, NY, USA, 2012, pp. 1047–1056, http://dx.doi.org/10.1145/2145204.2145360, URL: https://dl.acm.org/doi/10.1145/2145204.2145360.
[74] V. Arnaboldi, A. Guazzini, A. Passarella, Egocentric online social networks: Analysis of key features and prediction of tie strength in Facebook, Comput. Commun. 36 (10) (2013) 1130–1144, http://dx.doi.org/10.1016/j.comcom.2013.03.003, URL: https://www.sciencedirect.com/science/article/abs/pii/S0140366413000856.
[75] P.V. Marsden, K.E. Campbell, Measuring tie strength, Social Forces 63 (2) (1984) 482–501, http://dx.doi.org/10.1093/sf/63.2.482, URL: https://www.jstor.org/stable/2579058.
[76] S. Pei, L. Muchnik, J.S. Andrade Jr., Z. Zheng, H.A. Makse, Searching for superspreaders of information in real-world social media, Sci. Rep. 4 (2014) 5547, http://dx.doi.org/10.1038/srep05547, URL: https://www.nature.com/articles/srep05547.
[77] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. (1977) 452–473.
[78] P.M. Gleiser, L. Danon, Community structure in Jazz, Adv. Complex Syst. 06 (04) (2003) 565–573, http://dx.doi.org/10.1142/S0219525903001067.
[79] V. Batagelj, A. Mrvar, Pajek program for analysis and visualization of large networks reference manual list of commands with short explanation version BE, 1999.
[80] D. Watts, S. H. Strogatz, Collective dynamics of small world networks, Nature 393 (1998) 440–442, http://dx.doi.org/10.1038/30918, URL: https://www.nature.com/articles/30918.
[81] N.P. Hummon, P. Dereian, Connectivity in a citation network: The development of DNA theory, Social Networks 11 (1) (1989) 39–63, http://dx.doi.org/10.1016/0378-8733(89)90017-8, URL: https://www.sciencedirect.com/science/article/abs/pii/0378873389900178.
[82] M.A. Hasan, V. Chaoji, S. Salem, M. Zaki, Link prediction using supervised learning, in: Proc. of SDM 06 Workshop on Link Analysis, Counterterrorism and Security, Vol. 30, 2006, pp. 798–805.
[83] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, New York, NY, USA, 2008.
[84] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Amer. Statist. Assoc. 32 (200) (1937) 675–701.
[85] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18, http://dx.doi.org/10.1016/j.swevo.2011.02.002, URL: https://www.sciencedirect.com/science/article/abs/pii/S2210650211000034.

Shivansh Mishra is currently pursuing his Ph.D. in Computer science and Engineering from Indian Institute of Technology (BHU) Varanasi. His research interest includes social network analysis. He received his M.Tech degree in Computer Science and Engineering from National Institute of Technology, Kurukshetra. He received M.Sc.(TECH) degree (B.E. equiv.) in Information Systems from Birla Institute of Technology and Science, Pilani.

Shashank Sheshar Singh received Ph.D. degree in Computer Science and Engineering from Indian Institute of Technology (BHU), Varanasi and currently working at Thapar Institute of Engineering and Technology, Patiala as assistant professor. Prior to that, he 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. His research interests include Data Mining, Influence Maximization, Link Prediction and Social Network Analysis.

Ajay Kumar received Ph.D. degree in Computer Science and Engineering from Indian Institute of Technology (BHU), Varanasi and is currently working at UPES University, Dehradun as assistant professor. He completed his master of technology in Computer Science and Engineering from Samrat Ashok Technological Institute Vidisha (M.P.) and Bachelor of Technology in Computer Science and Engineering from R.K.D.F Institute of Science and Technology Bhopal (M.P.). His research interests include Link Prediction and Influence Maximization in social/complex networks.

Bhaskar Biswas received Ph.D. in Computer Science and 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 Associate Professor at Indian Institute of Technology (BHU), Varanasi in the Computer Science and Engineering department. His research interests include Data Mining, Text Analysis, Machine Learning, Influence Maximization, Link Prediction, and Social Network Analysis.