Chapter 3

ELP: Link Prediction in Social Networks based on Ego Network Perspective

This thesis presents four quasi-local similarity-based link prediction methods, three of which apply to multiplex networks, and one applies to simple networks. The inspiration for the method used for link prediction on simple networks comes from the fact that local similarity-based methods use very constrained information regarding node neighborhoods and edge relevance. Hence if edge relevance can be estimated using influences from a larger neighborhood of nodes, it can provide a pathway into extending local similarity-based methods in their corresponding quasi-local domains. In this chapter [1], *ELP* (Ego-based Link Prediction) is presented. Of the existing link prediction 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

[1]Published in Physica A: Statistical Mechanics and its Applications, DOI:https://doi.org/10.1016/j.physa.2022.128008

specific link depending on the node proximity. The proposed *ELP* approach utilizes Ego regions to calculate edge relevance and uses this information to extend some traditional local similarity-based approaches.

3.1 Introduction

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 [32]. Some studies utilized interaction dynamics like the amount of time spent in interaction [135] and frequency pattern [136, 137] to predict future links. Lionel et al.[138] 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.[139], authors propose using ego network layers to improve the performance of local similarity-based link prediction algorithms. In Rezaeipanah et al.[140], 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 [141], where authors study the weightage of each ego circle of corresponding nodes. Due to interaction dynamics, ego-centered social networks is utilized 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. A pair ranking approach combined with ego strength is utilized 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, an improved measure of edge relevance in the entire network can be created. This cumulative edge relevance

using ego regions is calculated for existing edges only. To predict unseen links, feature sets like common neighbors are utilized to predict missing links using the edge strengths of surrounding edges. Different feature sets are used 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 [18], three degree-of-influence [142] as well as cumulative influence in triangular clusters [90]. Since the proposed method aims to predict links using cumulative ego edge strengths, it becomes important to determine which of these edges is most relevant for the link prediction problem.

The *ELP* proposal 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[42], NLC[25]), 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.

3.1.1 Measures of Tie Strength in Social Networks

Granovetter et al.[143] 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 [144–146] have been presented in the literature to measure the tie strength. In Gilbert and Karahalios[144], the authors have provided a model to predict relationships in the classification of strong and weak ties. Eric Gilbert[145] has studied contrasting important relationship factors between different social networks, i.e., Facebook and Twitter. In Arnaboldi et al.[146], the authors have proposed a reduced feature set for tie strength prediction and have 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. [147]. Gilbert and Karahalios [144] 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.

3.1.2 Ego Network Model

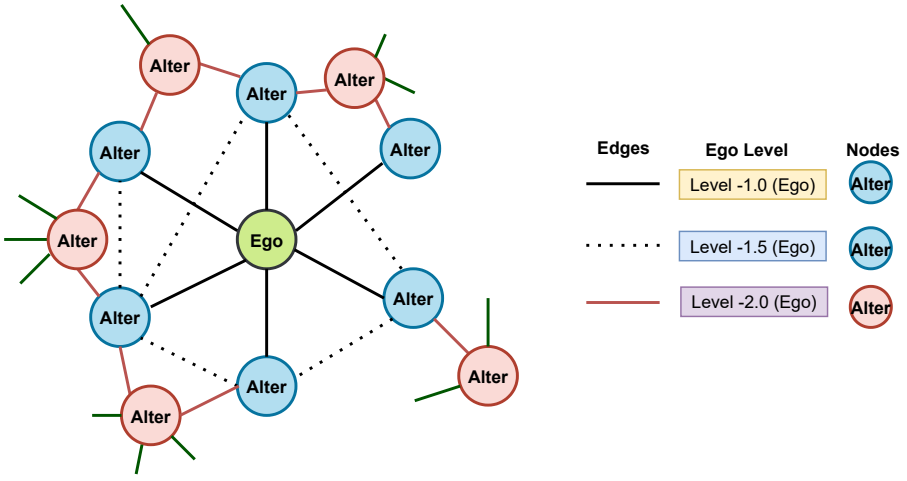


FIGURE 3.1: Ego network structure

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 Figure 3.1. 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. 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 [148], 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 Figure 3.2. 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). Only the region of 3 hops from the node A is considered following the principle of three degrees of influence [142]. Within this region the power law is used 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 $\alpha = 2$ is used. The formal definition of these levels are as follows.

Definition 3.1.1. **(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

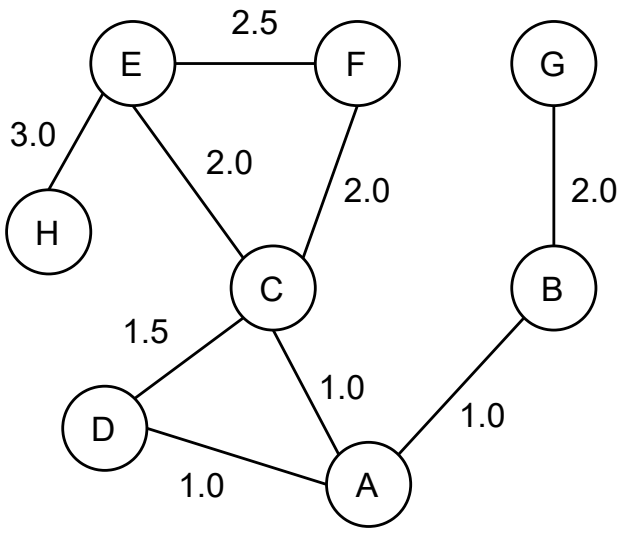


FIGURE 3.2: Ego network structure of node A (Lev.1 is Level-1(A) and so on and so forth) demonstrating the influence node A exerts on edges 3 hop distances away from it.

$y \in V_x$ 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 3.1.2. **(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 3.1.3. **(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 3.1.4. **(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 3.1.5. **(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 $\psi^{2.0}(v)$.

3.2 Proposed work

In this section the proposed method is discussed, 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.

3.2.1 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 [142, 149] suggest that an individual influence is limited to its local region based on small world phenomena. With these studies, the proposed method incorporates the interaction dynamics within the

three-hop area [142] 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.

Definition 3.2.1. **(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.

$$\psi(u,v) = \sum_{i} \sum_{w \in V} \eta_w^i(u,v) \tag{3.1}$$

where,

$$\eta_w^i(u,v) = \begin{cases} 1 & \text{if } (u,v) \in \psi^i(w) \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

The strength of ties in concentric circles of the ego are different, as shown in Fig. 3.1, 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. 3.2 can be used, otherwise use Eq. 3.3. 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. 3.2 is redefined as follows.

$$\eta_w^i(u,v) = \begin{cases} R^i & \textit{if } (u,v) \in \psi^i(w) \\ 0 & \textit{otherwise} \end{cases} \tag{3.3}$$

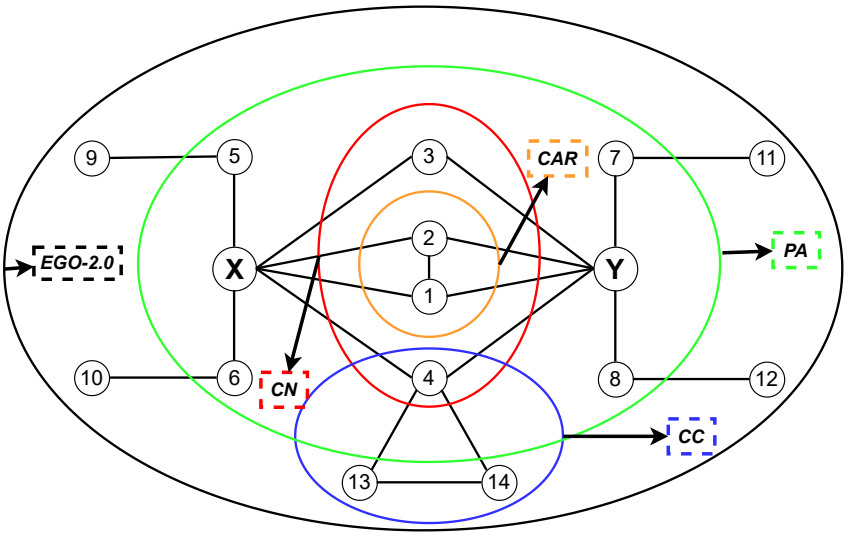


FIGURE 3.3: Example of different regions of feature selection for nodes $X\&Y$ (orange - CAR, red - CN, blue - CC, green - PA, black - Ego-2.0).

3.2.2 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. 3.3. 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 $Ego-2.0(X,Y) = PA(X,Y) \cup \{9,10,11,12\}$. Hence, the large node features are $PA(X,Y) = \{1,2,3,4,5,6,7,8\}$, and $Ego-2.0(X,Y) = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14\}$. There are different topological features which are utilized, 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 [16]. 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 is connected to both from n_1 and to n_2.

$$\gamma_{CN}(n_1,n_2) \leftarrow \{n | n \in \{N(n_1) \cap N(n_2)\}\} \tag{3.4}$$

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. [90] 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, from n_1 or to n_2.

$$\gamma_{PA}(n_1,n_2) \leftarrow \{n|n \in \{N(n_1) \cup N(n_2)\}\} \tag{3.5}$$

3. **Clustering Coefficient (CC)**. The clustering coefficient is the measure of the degree of those nodes which tend to form the cluster together [42]. 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.

$$\gamma_{CC}(n_1,n_2) \leftarrow \{n|n \in \Delta_m\} \tag{3.6}$$

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 [20] 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.

$$\gamma_{CAR}(n_1,n_2) \leftarrow \Big\{n|n \in \big\{\forall n' \in \{N(n_1) \cap N(n_2)\}, N(n_1) \cap N(n_2) \cap N(n')\big\}\Big\} \tag{3.7}$$

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 after 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.

$$\gamma_{EGO-2.0}(n_1,n_2) \leftarrow \{n|n \in \{N(n_1) \cup N(n_2) \cup N(N(n_1)) \cup N(N(n_2))\} \tag{3.8}$$

3.2.3 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 3.2.2) 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. 3.1. The likelihood score $S_L(x,y)$ of a non-existing link (x,y) is computed as follows.

$$S_L(x,y) = \sum_{z \in \gamma(x,y)} \frac{\psi(z,x) + \psi(z,y)}{\sum_{a \in N(z)} \psi(z,a)} \tag{3.9}$$

3.3 *ELP* 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 1-2 computes the likelihood of all existing links using Equation 3.1. The **for** loop in lines 3-5 estimates the likelihood value of target links using Equation 3.9 based on selected feature set. It has been stated in Stolz and Schlereth[141] 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. The algorithm *ELP* can also be seen

to consists of three such comparable parts - for predicting cumulative strength of existing edges the whole network structure is taken 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.

Algorithm 1: ELP: Ego-centric Link Prediction Algorithm

Input: Social graph: $G(V,E)$
Output: Likelihood score of non-existing links: S_L

for *each existing link* $(u,v) \in E$ **do**
  $\psi(u,v) \leftarrow$ Compute ego strength of edge (u,v) using Equation 3.1;
for *each non-existing link* $(x,y) \in U \setminus E$ **do**
  $\gamma(x,y) \leftarrow$ Estimate the feature of a pair of nodes (x,y);
  $S_L(x,y) \leftarrow$ Compute likelihood score of (x,y) using Equation 3.9;
Return S_L;

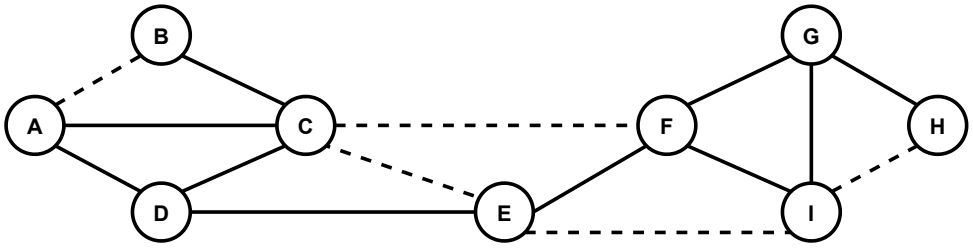


FIGURE 3.4: Example Network for demonstrating the working of *ELP* algorithm for link prediction.

3.3.1 An Illustrative Example

To explain the working of the proposed algorithm ELP, an example graph is used as shown in Figure 3.4. 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.

- **Ego Strength Estimation:** In this phase, *ELP* computes the ego strength $\psi(u,v)$ of each existing edge (u,v) using Eq. 3.1. 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) is existed in ego networks of $\{B,C,A,D,E\}$ under level 3.0 of ego

network, so $\psi(B,C) = 5$. Similarly, the ego strength of other existing edges can be estimated as shown in Figure 3.5.

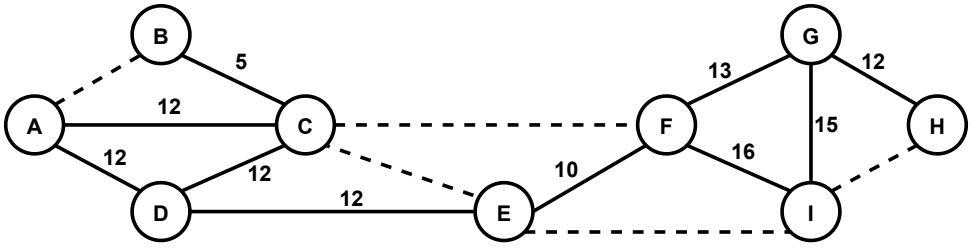


FIGURE 3.5: Ego Strength of Example Network.

- **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 3.1. For example, the common neighbors (CN) feature set for non-existing edge (A,B) is $\gamma(A,B) = \{C\}$ from Eq. 3.4 and PA feature set can be computed as $\gamma(A,B) = \{C,D\}$ by Eq. 3.5. Similarly, other feature sets CC, CAR, and Ego $\psi^{2.0}$ can be computed using Eq. 3.6,3.7 and 3.8 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. 3.9 as shown in Table 3.2. 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, there is 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 3.2 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.

TABLE 3.1: 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 3.2: Likelihood Score Computation

Non Existing Edges	Likelihood Score				
	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

3.3.2 Complexity Analysis

Based on different topological features, the proposed algorithm presents different variants: ELP-CN, ELP-PA, ELP-CC, ELP-CAR, and ELP-$\psi^{2.0}$. Assuming the average degree of a node is D_{avg}, the feature creation process takes worst time complexities of $\mathcal{O}(D^2_{avg})$, $\mathcal{O}(D^2_{avg})$, $\mathcal{O}(D^3_{avg})$, $\mathcal{O}(D^3_{avg})$ and $\mathcal{O}(D^4_{avg})$ respectively for CN, PA, CC, CAR and $\psi^{2.0}$ feature sets ($|FC|$). The calculate 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 $\mathcal{O}(|V| * D^2_{avg})$ for the whole graph. Here, $|V|$ is the total number of nodes and

TABLE 3.3: 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

D^3_{avg} 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 calculation and the second term $|FS| * D_{avg}$ is for calculation strength over those features. Essentially, it is observed 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 $\mathcal{O}((|V| * D^3_{avg}) + |E| * (|FC| + |FS| * D_{avg}))$. In this formulation if we substitute for the best feature set, i.e., CN, the final result is $\mathcal{O}((|V| * D^3_{avg}) + |E| * (D^2_{avg}))$. Here $|FC| \leftarrow D^2_{avg}$ and $|FS| \leftarrow D_{avg}$. Since *CN*,*PA*,*CCLP* (representing clustering coefficient

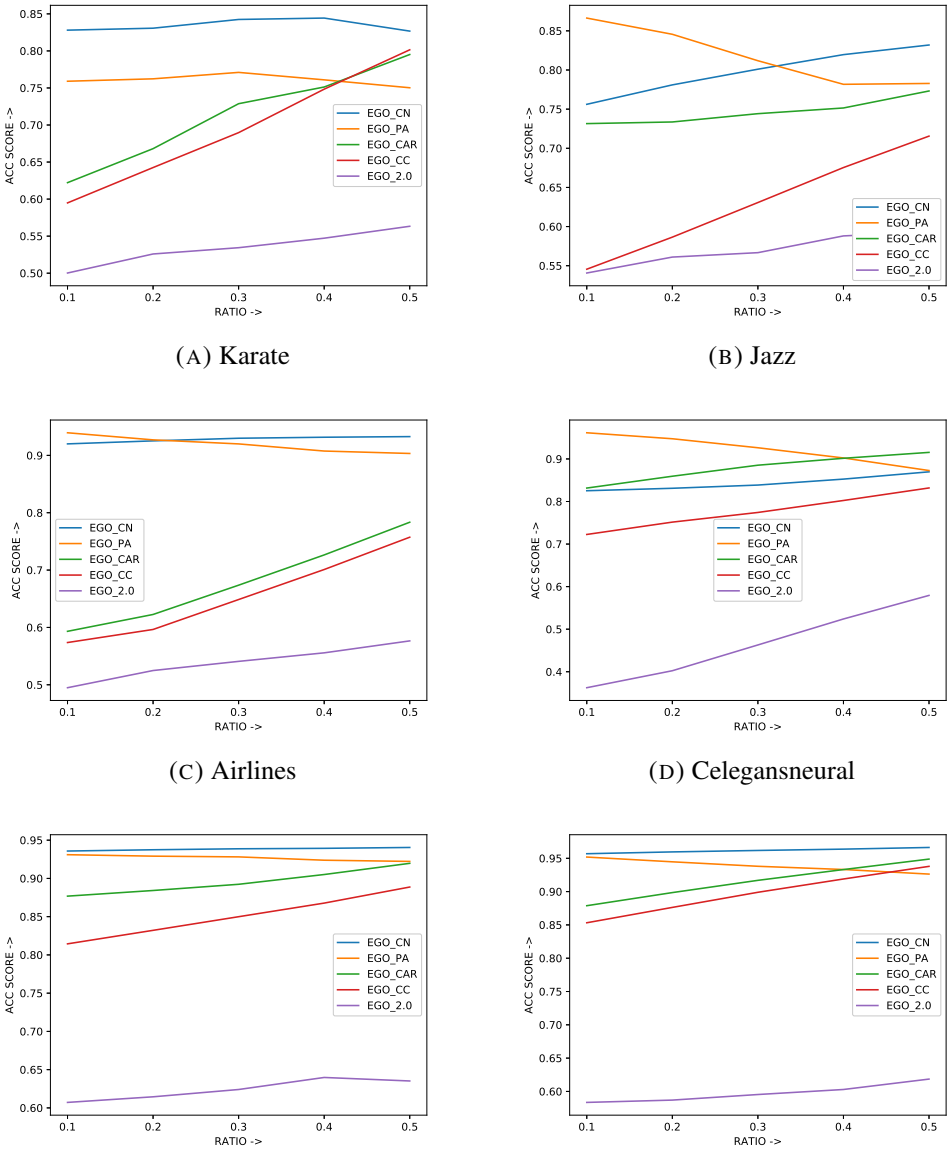
CC based method) and *CAR* are also link prediction approaches on their own, a quantitative comparison of running time of these link prediction approaches with the proposed *ELP* ($EGO-CN$ variation used for representation in these running time calculation context) algorithm is presented in Table 3.3. The experiment have been run on five different *Ratio* values between 0.1&0.5 which are ration of testing to total edges of dataset. The running times for *ELP* algorithm can been observed to be comparable to other standard link prediction algorithms.

3.4 Performance Analysis

In this section *ELP* is the final proposed algorithm. The relationship between the algorithm's performance based on different feature sets ($EGO-CN$, $EGO-PA$, $EGO-CAR$, $EGO-CC$, $EGO-2.0$) is also investigated. Three metrics are used in these experiments: Accuracy Score, AUPR and AUC. Five different ratios (0.1, 0.2, 0.3, 0.4, 0.5) of testing set edges to total edges are considered.

3.4.1 Accuracy Score On Features

As is evident from Fig. 3.6 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 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.

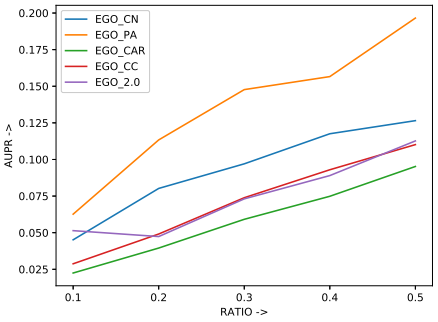


(A) Karate (B) Jazz

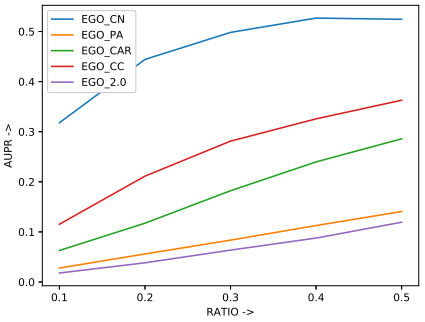
(C) Airlines (D) Celegansneural

(E) Political blogs (F) SmaGri

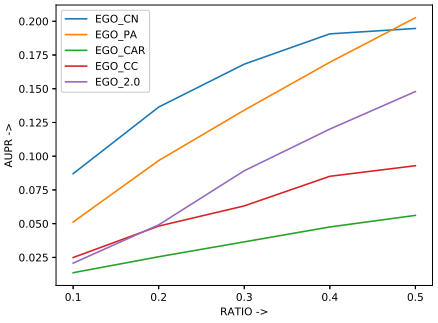
FIGURE 3.6: Accuracy Score comparison of *ELP* variations for different feature sets on six datasets.



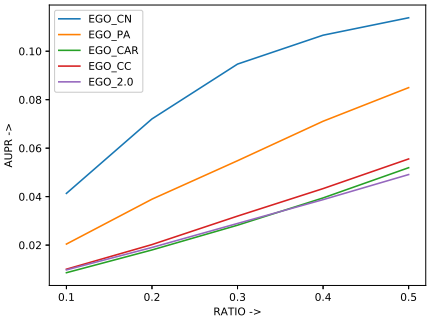
(A) Karate



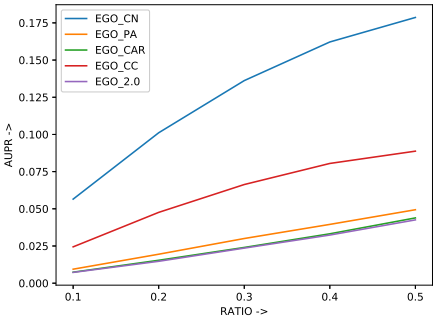
(B) Jazz



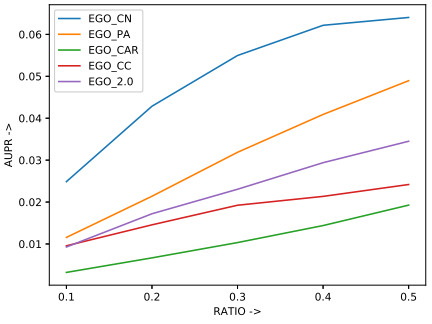
(C) Airlines



(D) Celegansneural



(E) Political blogs



(F) SmaGri

FIGURE 3.7: AUPR comparison of *ELP* variations for different feature sets on six datasets.

3.4.2 AUPR on Features

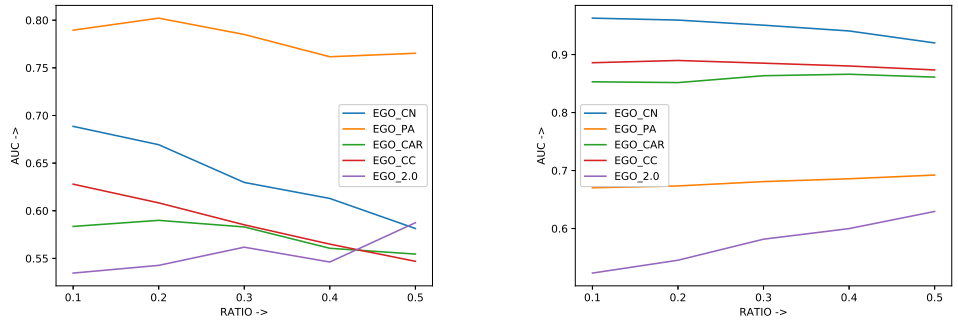
As is evident from Fig. 3.7 in all datasets and all algorithms, $EGO-CN$ performs best in all cases except the Karate dataset, which is comparatively a tiny dataset. For most cases in the $EGO-CAR$ algorithm, the worst performance is seen, while $EGO-PA$, $EGO-CC$, and $EGO-2.0$ can be considered to 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 *ELP* a clear choice for the AUPR metric.

3.4.3 Comparing *ELP* performance variation for AUC on Features

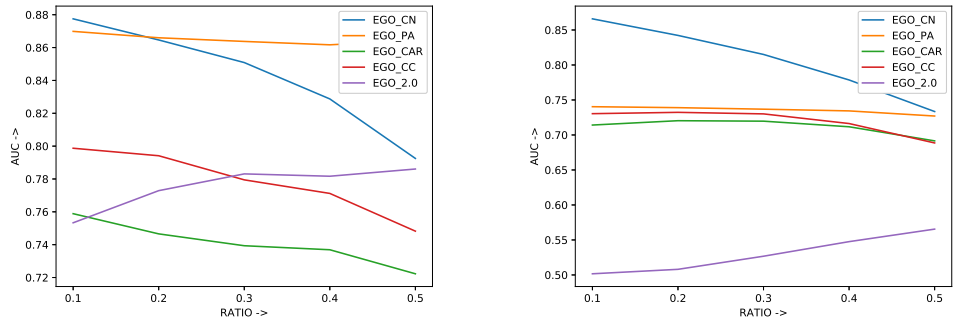
As is evident from Fig. 3.8 in all datasets and all algorithms, $EGO-CN$ performs best for high-density datasets while $EGO-PA$ performs better for mid-density ones. The worst performance is observed in $EGO-2.0$, while $EGO-CC$ and $EGO-CAR$ can be considered the middle-of-the-pack algorithms. It is 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 dip as the increase of ratio. Henceforth, $EGO-CN$ is the algorithm compared with state-of-art algorithms and will be referred to as *ELP*.

3.4.4 Comparing *ELP* performance with baseline algorithms for Accuracy Score

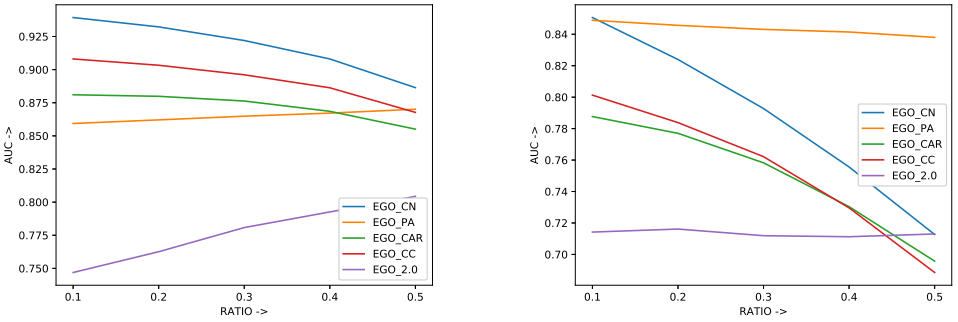
The nature of social networks considered in these experiments is inherently sparse. This is because compared to the total number of possible links for a sizable-sized graph, $n*n$ for a set of n nodes, the actual number of interactions would be quite a few. The accuracy score is fundamentally a measure of how exactly the set of predicted and actual labels match. In Table 3.4, two best values from each *Ratio* row have been highlighted. In this table,



(A) Karate (B) Jazz



(C) Airlines (D) Celegansneural



(E) Political blogs (F) SmaGri

FIGURE 3.8: AUC comparison of *ELP* variations for different feature sets on six datasets.

it is observed that the 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. This can be attributed to the distribution of a higher number of local communities in the Jazz dataset on the occurrence on which *CAR* is based. This is very impressive, especially considering how strict the process of calculation of the Accuracy score is. It can be concluded that the algorithm works well with sparse graphs in terms of accuracy, at least when the ratio of edges to nodes is less than equal to approximately 10.

3.4.5 Comparing *ELP* performance with baseline algorithms for AUPR

In Table 3.5, two best values from each *Ratio* row have been highlighted. In this table, it is observed that overall the proposed *ELP* algorithm can be considered the third-best in performance out of all the algorithms considered. For the smallest Karate dataset, the 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 analogous. In the Jazz dataset, the algorithm is the third-best algorithm, just behind *RA* and *CCLP*. However, even among these, the numbers are very close and comparable. *ELP* algorithm is the fourth-best performing algorithm in the Airlines dataset, behind *RA*, *CCLP*, and *PA*. In Celegansneural, the *ELP* algorithm is the best performing out of all the state-of-art algorithms considered for comparison. In the Political Blogs dataset, it is observed that the algorithm's worst ranking is behind *CN*, *RA*, *CCLP*, and $CLP-ID$. The algorithm performs better in the SmaGri dataset than most, just behind *CAR*, and is comprehensively outperformed by *CCLP*. The algorithm does not perform well in the GrQc dataset and is the sixth-best algorithm out of all state-of-art algorithms. When considering the overall pattern, it can be seen that for the AUPR metric, the algorithm can be worse than only RA and CCLP in most cases. Regarding AUPR, the overall relative performance of the *ELP* algorithm decreases as the ratio of edges to nodes increases in a dataset, the Karate dataset being an exception to this pattern.

TABLE 3.4: Comparison of the proposed algorithm ELP with the state-of-the-art algorithms in terms of Accuracy Score

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

TABLE 3.5: Comparison of the proposed algorithm *ELP* with the state-of-the-art algorithms in terms of AUPR

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

TABLE 3.6: Comparison of the proposed algorithm *ELP* with the state-of-the-art algorithms in terms of AUC

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 3.7: 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

3.4.6 Comparing *ELP* performance with baseline algorithms for AUC

In Table 3.6, two best values from each *Ratio* row have been highlighted. In this table, it is observed that overall the 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, the smallest of all datasets, the *ELP* algorithm is worse than *RA*, *PA*, *PageRank*, and *Node*2*v*. The *ELP* algorithm is ranked behind only *RA* for the Jazz dataset. For the Airlines dataset, the *ELP* algorithm gives the second-best numbers only behind *PA*. In Celeganseural, it performs on par with *RA* and is only behind PageRank for some train-to-test ratios. In the Political Blogs dataset, the *ELP* algorithm again performs on par with RA and is only outperformed by *PA*. In the SmaGri dataset, the algorithm is ranked third only behind *PA* and *PageRank*. In the GrQc dataset, the *ELP* algorithm is ranked third only behind *Node*2*v* and *PageRank*. It can be concluded that the overall relative performance of the *ELP* algorithm remains mostly consistent for all datasets. The best-performing algorithm may change, but *ELP* consistently can be considered the second best-performing algorithm, the Karate dataset being the only exception to this pattern.

3.4.7 Statistical Tests

This section compares the different state-of-the-art algorithms with *ELP* and analyzes their significant differences. This comparison is made for Accuracy, AUPR, and AUC metrics. Friedman's test[150] was applied to highlight significant differences between other algorithms compared with *ELP*. The result was hypothesis rejection in all cases. Friedman Siegel's Test[151] was applied as a post hoc procedure to estimate each hypothesis's degree of rejection. *ELP* algorithm was considered the control algorithm, and the degree of freedom and confidence level were 9 and 0.05, respectively. This was done to get a better measure of the significant difference between the proposed *ELP* algorithm and other algorithms. The statistical tests on 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 3.7, it is observed that the level of significant differences between the proposed algorithm and other standard algorithms. In Table 3.7. The combined ratio indicates that the statistical test is performed simultaneously for different sets of observed links.

3.5 Concluding Remarks

This chapter 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 strength propagation on edges to predict target links, but that is the algorithm's specialty. The notion of Ego-based edge strengths is introduced that simulates all nodes' cumulative effect on all their Ego region edges. The *ELP* approach is primarily used to understand the strength of weak edges, which directly connect two low-priority nodes but are an integral part of the Ego regions of several nodes. *ELP* is based on estimating the strength of existing edges and combines them with different feature sets to predict non-existent links. The closest comparison to the *ELP* 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 the algorithm more suitable for link prediction networks with the magnitude of edges much larger than nodes. These include social networking site-based datasets like Facebook and Twitter.