Chapter 6

Link Prediction using Information Diffusion Perspective

In this chapter, an application is designed by leveraging the information diffusion process. The community information of nodes is also utilized to predict future links.

6.1 Introduction

The rapid development of online social networking sites, such as Facebook, Twitter, and Sina Weibo, has attracted significant attention. These networking sites not only focus on individuals making new friends but also help share information or ideas over a network. Therefore, social networks provide a platform for user interaction and information dissemination. In the past several years, link prediction [167–170] has

attracted considerable attention. Link prediction focuses on estimating the likelihood of the existence of a link between two nodes based on the available information. In addition to considering the factors for relationship formation, link prediction is important for predicting the growth of the network. Link prediction can potentially be applied to user recommendation systems, network growth modeling, community detection, interaction mining, and information diffusion.

Nowell and Kleinberg [167] used the network topology for link prediction in a social network. Specifically, the network structure was modeled as a homogeneous graph $G(V,E)$, where V and E represent the set of nodes and edges, respectively. Each node indicates a user in the network, and each edge represents a relationship between individuals. Hasan et al. [168] presented a classification-based approach for link prediction. They used distinct features to estimate the likelihood of a future link and compared the effect of different features. Several standard methods have been used to define indexes for link prediction, such common neighbors (CN) [169], Adamic/Adar [170], and resource allocation (RA) [171]. Furthermore, domain-specific applications of link prediction have been investigated. However, most studies do not consider the utility and effect of information diffusion in link prediction. Information diffusion occurs when a network user starts retransmitting the information that has been received from its neighbors. This, in turn, leads to an information cascading phenomenon. Information dissemination provides an

opportunity to users for propagating and receiving information to and from a region of influence that is beyond the scope of their social circles. Furthermore, this mechanism affects the formation of social links in the network. For example, we consider three users x, y, and z on Twitter. At the start of the diffusion process, x follows y ($x \leftarrow y$), and y follows z ($y \leftarrow z$), but x does not follow z. Let user y repost information posted by user z. Then, there is a possibility that user x starts following user z if x finds the information valuable.

Recently, the problem of predicting network evolution has been considered from the information diffusion perspective [172, 173]. Zhou et al. [172] presented a visibility model by incorporating the diffusion process to address link prediction. Farajtabar et al. [173] directly explored the diffusion process rather than analyzing its influence on link prediction. In general, information diffusion in a network is modeled as a stochastic process. Kempe et al. [11] presented two basic stochastic models: independent cascade (IC) and linear threshold (LT) diffusion. The former is the most suitable model for integrating information diffusion with link prediction. In this model, an edge (x,y) is associated with a probability $p(x,y)$, which represents the possibility of a piece of information received by x from its neighbors at time-stamp t being propagated at time-stamp $t+1$. To incorporate information diffusion into link prediction, diffusion metrics, that is, information received by nodes, should be computed. Chaoji et al. [174] claimed that directly optimizing such a metric is an

NP-hard problem.

Owing to the complex nature of calculating such metrics (NP-hard problem), approximation methods may incur an excessively high computational cost. In addition to the accuracy of link prediction, efficiency should be considered. Therefore, a community-based framework can be useful for reducing the search space of the diffusion process in link prediction. This community-based framework may also improve prediction accuracy as well as efficiency. This improvement is because the information flow paradigm is viewed as a collaborative exercise based on closely linked groups instead of an individual user [175]. The community of individuals plays a pivotal role in opinion formation along with the individual's neighbors. Therefore, community information should be combined with neighbor information to perform link prediction based on information diffusion.

This chapter addresses the link prediction problem from a new perspective, based on the assumption that community structure information combined with information dissemination may improve the predictor. Previous prediction methods ignored the effect of information diffusion on link creation and prediction. Being the first attempt to incorporate community information and information diffusion into link prediction, this study naturally suggests that effectiveness and efficiency should be balanced. Accordingly, we present the community information

based link prediction algorithm CLP-ID using information diffusion. The contributions of this chapter can be summarized as follows.

- Exploring objective five, a link prediction algorithm CLP-ID is presented by considering community information in addition to node and link information. To improve the effectiveness of the algorithm and provide a better quantization of community information, we incorporate both the effects of positive as well as negative influence.
- A community detection algorithm is proposed to decompose a vast social network into small chunks. The algorithm detect these chunks by considering influence probabilities among the users when assigning community label to users. The independent cascade model is adopted to incorporate information diffusion.
- A probabilistic method is presented to compute the likelihood score of target links based on the assumption that different common neighbors work independently.
- Furthermore, the working of CLP-ID is explained with an example graph. The theoretical complexity of the algorithm is also discussed.
- The experimental result of the proposed algorithm is discussed along with the state-of-the-art algorithms and validate it against different performance metrics. The resulting analysis demonstrates the superiority of the proposed algorithm.

6.2 Proposed Approach

This section presents the CLP-ID algorithm, which adopts a community-based framework from an information diffusion perspective for link prediction. The CLP-ID algorithm can be summarized in three steps as follows. First, a community detection algorithm is used to discriminate between different clusters of the network based on the information diffusion between nodes. Secondly, we incorporate the importance of an individual node's community into the evolving network structure. Finally, we a probabilistic model is used to predict future links among users.

6.2.1 Identification of Community Structure

The main objective of the community detection algorithm (CD) is to divide the network into sub-networks, and then incorporate the community importance to identify future links. The individuals in a particular community have more influence on each other because they have frequent contact. The information diffusion process is viewed as a collaborative process in closely related groups as in a community rather than an individual. Therefore, community information in addition to individual information is also pivotal for predicting future links. The CD algorithm comprises two phases: partition and combination. In the partition phase, the network is divided into communities based on

information diffusion. In the combination phase, unstable communities that are not sufficiently isolated are merged with other suitable communities.

6.2.1.1 Partition phase

To propagate the influence spread, we adopt an IC diffusion model. Initially, every node has a distinct community label. In the CD algorithm, the influenced neighbors of each node are obtained under the IC model. Then, the algorithm iteratively updates the labels of each node based on information diffusion under the IC model. The CD updates the label of a node x based on its influenced neighbors' labels as in [176]. The label of node x can be updated as follows.

Definition 6.2.1. (*Community label [176]*). Given a node $x \in V$, the set of its neighbors is $N(x) = \{y_1, y_2, \ldots, y_n\}$, and the community set of neighbors is $C \leftarrow \{N.C_1, N.C_2, \ldots, N.C_l\}$, where l is the number of communities to which the neighbors belongs. $N.C_j$ denotes the set of neighbors that have the same community label j and are influenced by x. Then, the community label of node x is determined as follows:

$$C_L.x \leftarrow argmax_{1 \leq j \leq l} \left\{ 1 - \prod_{y_i \in N.C_j^{i-1}} (1 - p(x, y_i)) \right\} \tag{6.1}$$

6.2.1.2 Combination phase

In this phase, communities identified in the partition phase are re-examined and merged based on their stability. The algorithm identifies unstable communities according to their detachability index. A community with detachability less than a threshold θ is considered unstable, that is, it is not able to isolate itself from other communities. The detachability index is defined in the context of influence probabilities, as in the case of isolability [177], by considering the connection strength. The detachability index can be calculated as follows.

Definition 6.2.2. *(Detachability).* The detachability of a community defines the quality or degree of being detachable or able to isolate itself from the rest of the network. The detachability of any community C_i is defined in the context of influence probability as follows:

$$D(C_i) \leftarrow \frac{\sum_{u,v \in C_i} p(u,v)}{\sum_{u,v \in C_i} p(u,v) + \sum_{u \in C_i, v \notin C_i} p(u,v)} \tag{6.2}$$

Each unstable community is merged with other suitable and stable communities. **Algorithm 15** presents the pseudocode of the CD algorithm.

6.2.2 Incorporation of Community Importance

The influence of an individual node x in associated community nodes (which belong to the same community) is higher than that of nodes belonging to different communities. Therefore, to incorporate the importance of associated community in link prediction, we use a community index for any node pair (x,y) [178]. The community index $CI(x,y)$ can be computed as follows:

$$CI(x,y) = \begin{cases} +\frac{|C_x|}{|V|}, & \text{if } C_L.x = C_L.y \\ -\frac{|C_x|}{|V|}, & \text{otherwise} \end{cases} \tag{6.3}$$

where, C_x denotes the community to which node x belongs. Incorporating the community information in link prediction ensures a positive influence on the scores if the associated nodes belong to the same community. If the nodes are from different communities, then the influence is negative.

6.2.3 Computation of Likelihood Score

To define the importance of a node x relative to its neighbors, we use a probabilistic model for capturing the influence probabilities. Specifically, each node x independently influences its outgoing neighbors $y \in N^{out}(x)$ with influence probability $p(x,y)$. Similarly, each node y is independently influenced by its incoming neighbors $x \in N^{in}(y)$ with influence probability $p(x,y)$. In an undirected graph, both x and y influence each other, that is,

$N^{out}(x) = N^{in}(x) = N(x)$. Therefore, we adopt the IC diffusion model to compute the similarity index between individuals using these independent features. The similarity index between nodes x and y can be calculated as follows:

$$SI(x,y) = p(x,y) + (1 - \prod_{z \in N^{out}(x) \cap N^{in}(y)} (1 - p(z,y))) \tag{6.4}$$

Therefore, the overall importance $OI(x,y)$ of node x with respect to node y is computed as follows:

$$OI(x,y) = CI(x,y) \times SI(x,y) \tag{6.5}$$

We now compute the likelihood score of non-existing links (x,y). To this end, we adopt a feature set based on common neighbors, that is, we consider the importance of common neighbors to/from both nodes. Let $CN(x,y) = \{z | z \in (N^{out}(x) \cap N^{in}(y))\}$. Then, the likelihood score $LS(x,y)$ of non-existing links (x,y) can be calculated as follows:

$$LS(x,y) = \sum_{z} (OI(x,z) + OI(z,y)) \tag{6.6}$$

Incorporating the concepts mentioned above, we present the CLP-ID algorithm. The corresponding pseudocode is outlined in **Algorithm 16**.

Algorithm 15: CD(G,τ): Community Detection Algorithm

Input: $G(V,E)$: Social graph, τ: Number of iterations
Output: C_S: Community structure of graph

$i \leftarrow 1$ ▷ Partition Phase
$C_S \leftarrow \phi$
for *each* $x \in V$ **do**
  $C_L.x \leftarrow$ a distinct community label
  $z \leftarrow C_L.x$
  $C_z \leftarrow x$
  $C_S \leftarrow C_S \cup C_z$
  for *each neighbor* y_j *of* x , $j \in [1,2,\ldots,|N(x)|]$ **do**
    if *IsInfluence(x,*y_j*)==True* **then**
      $A_x[j] = 1$
    else
      $A_x[j] = 0$
while $i \leq \tau$ **do**
  for *each* $x \in V$ **do**
    $C_L.x \leftarrow argmax_{1\leq j\leq l}\left\{1 - \prod_{y_i \in N.C_j^{i-1}}(1 - p(x,y_i))\right\}$ ▷ Label Propagation
    $z \leftarrow C_L.x$
    $C_z \leftarrow x$
    $C_S \leftarrow C_S \cup C_z$
  $i \leftarrow i+1$ ▷ Combination Phase
for *each* $C_z \in C_S$ **do**
  $D_z \leftarrow$ Estimate detachability of community C_z
  if $D_z < \theta$ **then**
    $N_E \leftarrow$ Find each node $w \in N(C_z) \cap (V \setminus C_z)$
    $T_c \leftarrow$ Find set of number of times each $w \in N_E$ appears in $w \in N(C_z) \cap (V \setminus C_z)$
    $C_{max} \leftarrow argmax_{w\in N_E}(T_c(w))$
    $N_S(C_{max}) \leftarrow$ Set of all nodes those have C_{max}
    $CS \leftarrow$ Set of all communities those have node $w \in N_S(C_{max})$
    $M_{ID} \leftarrow -\infty$
    for *each* $C_i \in CS \setminus C_z$ **do**
      $T_{ID} \leftarrow D(C_z \cup C_i) - D(C_i)$
      if $T_{ID} > M_{ID}$ **then**
        $M_{ID} \leftarrow T_{ID}$
        $t \leftarrow i$
    $C_t \leftarrow (C_t \cup C_z)$
    $C_S \leftarrow (C_S \setminus C_z)$
Return C_S

Algorithm 16: CLP-ID(G): Link Prediction Algorithm

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

1 $C_S \leftarrow CD(G,\tau)$ ▷ See **Algorithm 15**
2 **for** *each link* $(x,y) \in E$ **do**
3 $\quad CI(x,y) \leftarrow$ Compute community index using equation 6.3
4 $\quad SI(x,y) \leftarrow$ Compute similarity index using equation 6.4
5 $\quad OI(x,y) \leftarrow$ Compute overall importance using equation 6.5
6 **for** *each non-existing link* $(x,y) \in U \setminus E$ **do**
7 $\quad z \leftarrow N^{out}(x) \cap N^{in}(y)$
8 $\quad LS(x,y) \leftarrow$ Compute likelihood score using equation 6.6
9 **Return** LS

6.3 Algorithm

Algorithm 15 takes two inputs: a social graph G and the number of iterations τ. CD determines the community structure of the graph based on information diffusion. The algorithm operates in two phases: partition (lines 1–19) and combination (lines 20–35). The partitioning phase divides the network into smaller clusters. Lines 1 and 2 assign the value 1 and the empty set to the integer i and the community structure C_S, respectively. The **for** loop in lines 3–12 assigns a distinct community label and identifies the influential neighbors of each node. The **while** loop in lines 13–19 iteratively updates the community label of each node using label propagation under the IC diffusion model. In the combination phase, merging of unstable communities with most suitable and stable communities is performed so that the detachability of the community structure is improved. The **for** loop in lines 20–35 combines each unstable community with a stable community to improve the isolability of the new

community. Lines 23–26 identify all neighbor nodes of the community with the highest number of appearance in some other communities. Line 27 identifies the set of communities with the highest appearance node. Lines 28–35 find the best suitable and stable community C_t for C_z, and then merge them and remove the unstable community C_z from the list of all communities C_S. Finally, line 36 returns the community structure C_S of G.

The main **Algorithm 16** takes a social graph as input. CLP-ID determines the likelihood score of each target link. Line 1 calls **Algorithm 15** to divide the graph into clusters. The **for** loop in lines 2–5 computes the overall importance of each link in the graph using Equation 6.5. The **for** loop in lines 6–8 determines the likelihood score of each non-existing or target link. Finally, line 9 returns the likelihood score of each target link.

6.3.1 Applying the Algorithm

To explain the proposed algorithm, we take as an example the graph shown in FIGURE 6.1. The given graph G has 16 nodes, and each edge is associated with an influence probability. CLP-ID operates in three steps as follows.

1 *Detection of community structure*. The CD algorithm is used for community detection and operates in two phases: partition and combination. In the partitioning phase, the network is divided into

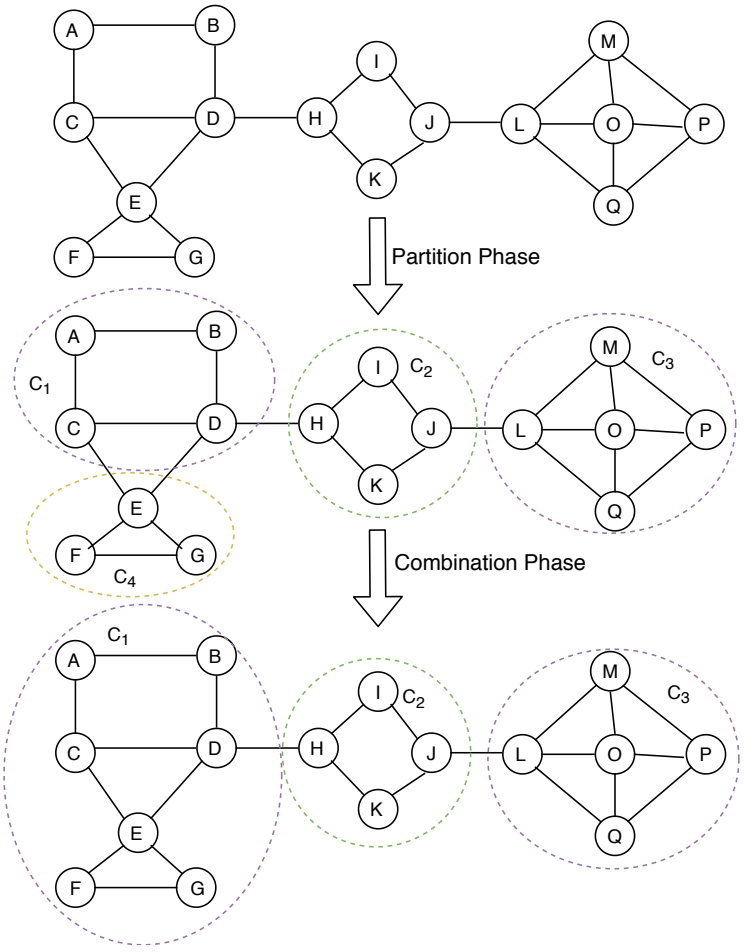


FIGURE 6.1: The Working of CD Algorithm using An Example Graph

communities using label propagation under the IC model. In the combination phase, unstable communities are merged based on the detachability metric. FIGURE 6.1 shows the operation of the CD algorithm: the graph is divided into four clusters (partition phase), and clusters C_1 and C_4 are merged (combination phase). Finally, the example graph consists of three communities C_1, C_2, and C_3.

2 *Overall index computation for existing links.* The CLP-ID algorithm now computes the overall index of each existing link. For example, to compute $OI(A,B)$ of a node pair A and B, we first compute the

community index $CI(A,B)$. Thus, $CI(A,B)$ is calculated as $CI(A,B) \leftarrow +\frac{|C_1|}{|V|} \leftarrow +\frac{7}{16} \leftarrow 0.4375$. The similarity index $SI(A,B)$ can be estimated as $SI(A,B) \leftarrow p(A,B)+(1-\prod_{z\in N^{out}(A)\cap N^{in}(B)}(1-p(z,B))) \leftarrow 0.28252$. We can now normalize the similarity index $SI(A,B)$ as $SI(A,B) \leftarrow \frac{SI(A,B)}{max_{\forall x,y}(SI(x,y))} \leftarrow 0.17974$. Subsequently, we calculate the overall index $OI(A,B)$ as $OI(A,B) = CI(A,B) \times SI(A,B) \leftarrow 0.4375 \times 0.17974 \leftarrow 0.07864$. Similarly, we can compute the overall index for each existing link (x,y), as shown in TABLE 6.1.

3 *Likelihood score computation for target links*. We now compute the likelihood score $LS(x,y)$ of each non-existing link (x,y). For example, to compute $OI(E,B)$ of a node pair E and B, we should first identify the common neighbors $z \leftarrow N(E) \cap N(B) \leftarrow \{D\}$. The likelihood score is $LS(E,B) = \sum_z(OI(E,z) + OI(z,B)) \leftarrow 0.19798 + 0.06302 \leftarrow 0.26100$.

6.3.2 Complexity Analysis

In this section, we analyze the time complexity of the proposed algorithm. First, we analyze the time complexity of **Algorithm 15**. Lines 1 and 2 perform initialization in $O(1)$ time. Lines 3–12 determine the influenced neighbors of each node in $O(|V|.D_{avg})$ time, where D_{avg} denotes the average degree of a node in the graph. Lines 13–19 update the community

TABLE 6.1: The Computation of Overall Index $OI(x,y)$ of Each Existing Links (x,y) based on CLP-ID Algorithm

Node Pair	Influence Probability	Community Index		Similarity Index		Overall Index	
$x-y$	$p(x,y)$	$CI(x,y)$	$CI(y,x)$	$SI(x,y)$	$SI(y,x)$	$OI(x,y)$	$OI(y,x)$
$A-C$	0.75269	0.43750	0.43750	0.47887	0.47887	0.20951	0.20951
$B-A$	0.28252	0.43750	0.43750	0.17974	0.17974	0.07864	0.07864
$B-D$	0.22642	0.43750	0.43750	0.14405	0.14405	0.06302	0.06302
$C-D$	0.42489	0.43750	0.43750	0.45252	0.84926	0.19798	0.37155
$C-E$	0.90998	0.43750	0.43750	0.76115	0.84926	0.33300	0.37155
$D-E$	0.28639	0.43750	0.43750	0.76115	0.45252	0.33300	0.19798
$F-E$	0.02332	0.43750	0.43750	0.06819	0.48210	0.02983	0.21092
$G-E$	0.08387	0.43750	0.43750	0.06819	0.52062	0.02983	0.22777
$G-F$	0.73445	0.43750	0.43750	0.48210	0.52062	0.21092	0.22777
$H-D$	0.89002	-0.25000	-0.43750	0.56624	0.56624	-0.14156	-0.24773
$H-K$	0.20926	0.25000	0.25000	0.13313	0.13313	0.03328	0.03328
$I-H$	0.03660	0.25000	0.25000	0.02328	0.02328	0.00582	0.00582
$I-J$	0.07083	0.25000	0.25000	0.04506	0.04506	0.01127	0.01127
$J-K$	0.60187	0.25000	0.25000	0.38292	0.38292	0.09573	0.09573
$L-J$	0.22314	-0.31250	-0.25000	0.14196	0.14196	-0.04436	-0.03549
$L-M$	0.54488	0.31250	0.31250	0.50749	0.38112	0.15859	0.11910
$N-L$	0.80161	0.31250	0.31250	0.54445	0.89831	0.17014	0.28072
$N-O$	0.61035	0.31250	0.31250	0.55027	1.00000	0.17196	0.31250
$N-P$	0.80568	0.31250	0.31250	0.64738	0.90090	0.20231	0.28153
$O-L$	0.05417	0.31250	0.31250	0.61323	0.48544	0.19163	0.15170
$O-M$	0.25280	0.31250	0.31250	0.76011	0.32279	0.23753	0.10087
$O-P$	0.21187	0.31250	0.31250	0.75523	0.58577	0.23601	0.18305
$P-M$	0.87243	0.31250	0.31250	0.71588	0.68984	0.22371	0.21558

label in $O(\tau.|V|.D_{avg})$ time, where τ represents the number of iterations in the partition phase. Lines 20–35 merge unstable communities with some other stable communities in $O(l^2.C_{avg}.D_{avg})$ time, where l represents the number of communities after the partition phase. Therefore, CD divides the graph into communities in $O(D_{avg}(\tau.|V|+l^2.C_{avg})$ time.

We now analyze the overall time complexity of the main algorithm CLP-ID (**Algorithm 16**). Line 1 performs community detection using the

CD algorithm in $O(D_{avg}(\tau.|V|+l^2.C_{avg})$ time. The overall importance of each existing link is computed in $O(|E|.D_{avg})$ time (lines 2–5). Lines 6–8 compute the likelihood score of each target link in $O(|E|.D_{avg})$ time. Thus, the overall time complexity of CLP-ID is $O(D_{avg}(|E|+\tau.|V|+l^2.C_{avg}))$. TABLE 6.2 compares the complexity of the proposed algorithm with that of state-of-the-art algorithms.

TABLE 6.2: The Comparison of the Complexity of CLP-ID with the State-of-the-art Algorithms

Algorithm	Complexity	Remarks
CN [169]	$O(N.D^3_{Avg})$	Local similarity index
PA [179]	$O(N.D^2_{Avg})$	Local Similarity Index
RA [171]	$O(N.D^3_{Avg})$	Local Similarity Index
LNBCN [180]	$O(N(f(z)+N.D^3_{Avg}))$	Naive Bayes theory
CAR [181]	$O(N.D^4_{Avg})$	Community-based Similarity Index
NLC [182]	$O(N.D^3_{Avg})$	Clustering Coefficient
N2V [183]	$O(\frac{l}{k'(l-k')})$[1]	Network Embedding
CCLP [184]	$O(N^2.D^2_{Avg})$	Clustering Coefficient
CCLP2 [185]	$O(N^3.D^2_{Avg})$	Level-2 Clustering Coefficient
CLP-ID	$O(D_{avg}(M+\tau.N+l^2.C_{avg}))$	Information dissemination

6.4 Empirical Analysis

All the experiments performed on eight real-world network datasets: Football [146], Celegansneural [147], USAir97 [148], Political blogs [149], Amazon web graph [150], NetScience [186], Power [147], and GrQc [141]. The performance of proposed algorithm is tested against nine methods regarding four performance metrics.

[1]The given time complexity of the N2V is defined for per sample where l is walk length

6.4.1 Performance Metrics

Hasan et al. [168] treated the link prediction problem as a binary classification problem, thus employing most of the related evaluation metrics. The evaluation of a binary classification problem with two classes can be represented as a confusion matrix [187].

		Prediction outcome		
		p	**n**	**Total**
Actual value	**p′**	True positive	False negative	P′
	n′	False positive	True negative	N′
	Total	P	N	

1 Recall or True positive rate (TPR) or Sensitivity or Hit rate

$$Recall = \frac{p'p}{P'} \tag{6.7}$$

2 False positive rate (FPR) or Fall-out

$$FPR = \frac{n'p}{N'} \tag{6.8}$$

3 Specificity or True negative rate (TNR) or Selectivity

$$Specificity = \frac{n'n}{N'} \tag{6.9}$$

4 Precision or Positive predictive value

$$Precision = \frac{p'p}{P} \tag{6.10}$$

5 Average Precision

$$AveragePrecision = \int_{r=0}^{1} p(r)dr \tag{6.11}$$

Manning et al. [187] defines the following metrics based on the above confusion matrix. We evaluate the proposed algorithm CLP-ID in terms of four accuracy metrics viz., Area under the precision-recall curve (AUPR) [188], Recall [187], Area under curve (AUC) [189], and Average Precision [187].

- **Area under the precision-recall curve (AUPR).** In binary classification problems, AUPR [185, 187] is more informative and useful. Thus, we used it as an accuracy measure. AUPR values are computed based on the precision–recall curve, where the x- and y-axes represent the recall and precision values, respectively. Precision measures the deviation from true values and its scatter, and is given in Equation 6.10. Similarly, recall measures the deviation from true values and its total relevant values, and is given in Equation 6.7.

- **Recall.** In link prediction, recall [185, 187] is the fraction of target links that are successfully predicted, and is given in Equation 6.7.

- **Area under the Receiver Operating Characteristics Curve (AUROC/AUC).** AUROC/AUC [185, 187] plots TPR (y-axis) against FPR (x-axis). The TPR and FPR values can be computed using Equations 6.7 and 6.8, respectively. The AUC value is a single-point statistical summary with range 0–1 and is estimated using the trapezoidal rule.

- **Precision.** In link prediction, precision [185, 187] measures the deviation from true values and its scatter, and can be computed by Equation 6.10. To validate the predicted probability values against the training data, a threshold is required that can act as a boundary between true and false predictions. However, this threshold may be different for different runs of the algorithm. To provide a more standard metric, we used average precision[2]. The average precision metric is a single-point summary value that is computed based on varying threshold values. The average precision value is equal to the precision averaged over all values of recall between 0 and 1 (Equation 6.11).

6.4.2 Methods to Compare

1 **Common Neighbor (CN).** In 2001, Newman et al. [169] stated that the similarity score $S(x,y)$ between a a pair of nodes x and y is

[2] Henceforth, we will use the terms average precision and precision interchangeably, but the formula used for the calculation of the metric is that of average precision

dependent on the number of mutual friends, given as follows.

$$S(x,y) = |N(x) \cap N(y)| \tag{6.12}$$

where $N(x)$ and $N(y)$ denotes the set of neighbor nodes of x and y respectively.

2 **Preferential Attachment (PA).** In 2002, Barabasi et al. [179] considered preferential growth to a pair of nodes for link prediction and presented the probability of co-authorship $S(x,y)$ between a pair of nodes x and y, given as follows.

$$S(x,y) = D_x \times D_y \tag{6.13}$$

where D_x and D_y denotes the degree of x and y respectively.

3 **Resource Allocation (RA).** In 2009, Zhou et al. [171] presented resource allocation index for link prediction by imposing penalty to higher degree nodes. The score $S(x,y)$ between a pair of nodes x and y based on this method given as follows.

$$S(x,y) = \sum_{z \in N(x) \cap N(y)} \frac{1}{D_z} \tag{6.14}$$

4 **Local Naive Bayes based Common Neighbor (LNBCN).** In 2011, Liu et al. [180] suggested that that different common neighbors play different role in the network. With their different contribution in score computation, similarity score $S(x,y)$ between a pair of nodes x and y

is defined as follows.

$$S(x,y) = \sum_{z \in N(x) \cap N(y)} \{ \log(\frac{C(z)}{1-C(z)}) + \log(\frac{\rho}{1-\rho}) \} \tag{6.15}$$

where $C(z)$ is clustering coefficient of node z and ρ is computed as follows.

$$\rho = \frac{|E|}{|V| \times (|V|-1)/2} \tag{6.16}$$

5 **CAR Index.** In 2013, Cannistraci et al. [181] suggested that a pair of nodes possibly have a connection if their common neighbors are members of a local community. They proposed CAR variants of CN, JA, AA, and RA. The common neighbor variant CAR_CN computes similarity score $S(x,y)$ between a pair of nodes x and y as follows.

$$S(x,y) = CN(x,y) \times \sum_{z \in N(x) \cap N(y)} \frac{\gamma(z)}{2} \tag{6.17}$$

where $\gamma(z)$ represents a subset of neighbors of node z such that $\gamma(z) \subseteq N(z) \cap N(x) \cap N(y)$. $CN(x,y)$ is the set common neighbors of a pair of individuals x and y.

6 **Node and Link Clustering Coefficient (NLC).** In 2016, Wu et al. [182] adopted both node and link clustering coefficients. The similarity score of a target link can be computed as follows.

$$S(x,y) = \sum_{z \in N(x) \cap N(y)} \{ \frac{CN(x,z)}{D_z - 1} + \frac{CN(y,z)}{D_z - 1} \} \times C(z) \tag{6.18}$$

7 **Node2vec (N2V).** In 2016, Grover et al. [183] presented a network embedding method to predict future links. N2V mapped nodes to a lower dimensional space.

8 **Clustering Coefficient based Link Prediction (CCLP).** In 2016, Wu et al. [184] utilized clustering coefficient to compute similarity score of a pair of individuals based on common neighbors. The similarity index between a target link (x,y) can be computed as follows.

$$S(x,y) = \sum_{z \in N(x) \cap N(y)} C(z) \tag{6.19}$$

The clustering coefficient $C(z)$ of node z can be computed as follows.

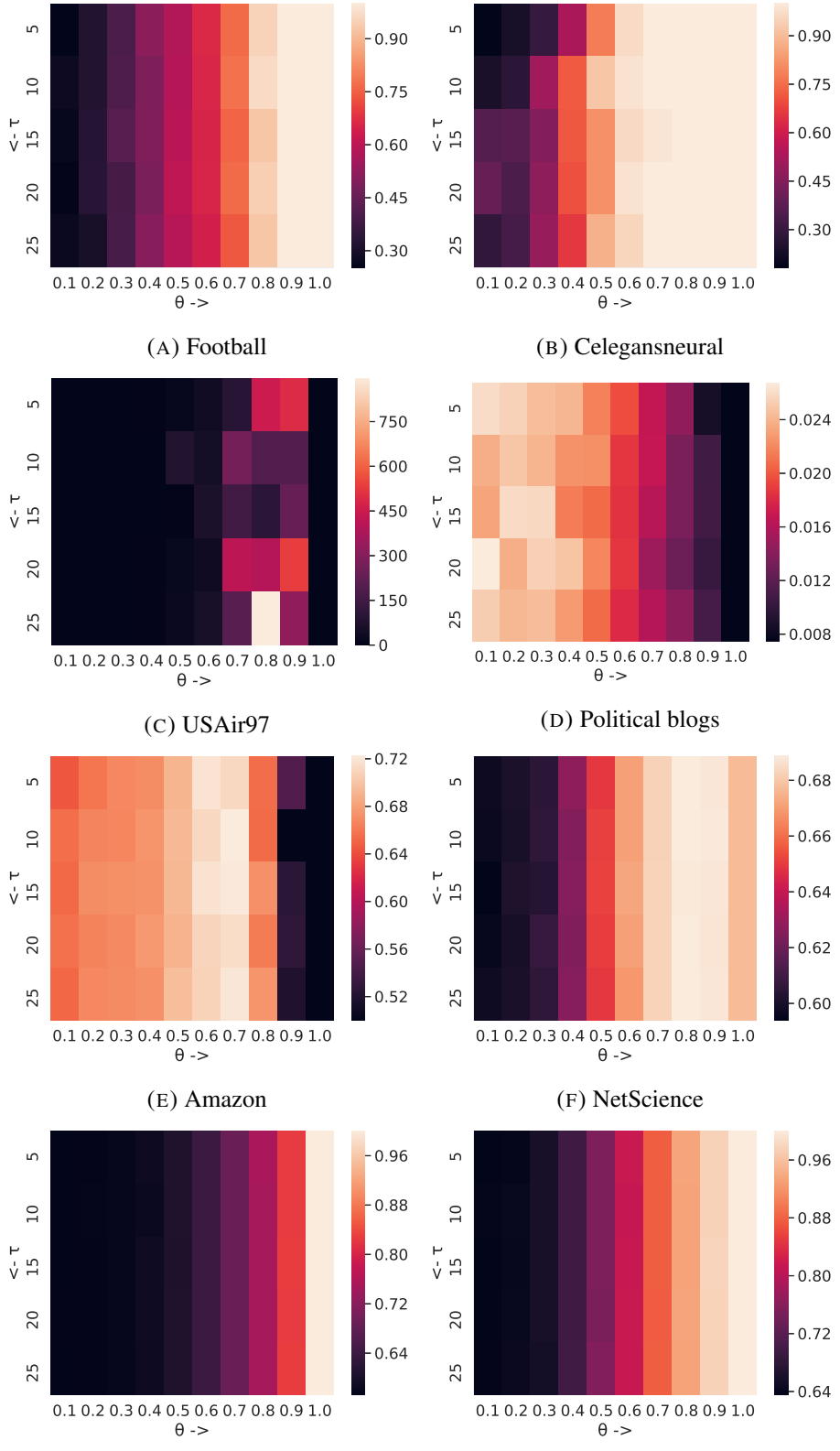
$$C(z) = \frac{t_z}{D_z \times (D_z - 1)} \tag{6.20}$$

where t_z denotes total number of triangles passing through the node z.

9 **Level-2 node Clustering Coefficient-based Link Prediction(CCLP2).** In 2019, Kumar et al. [185] extended the CCLP metric up to level 2 of a node. The CCLP2 score can be computed as follows.

$$S(x,y) = \sum_{z \in (N(x) \cap N(CN(x,y)))||(N(CN(x,y)) \cap N(y))} CC(z) \tag{6.21}$$

where $CC(z)$ refers to level-2 clustering coefficient score defined in [185].



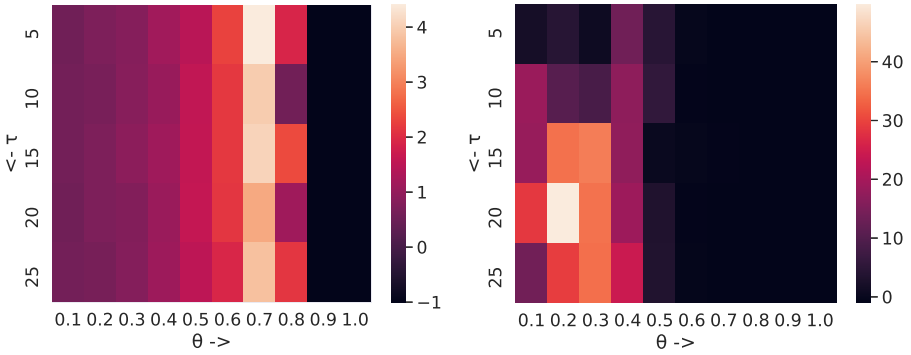
(A) Football (B) Celegansneural

(C) USAir97 (D) Political blogs

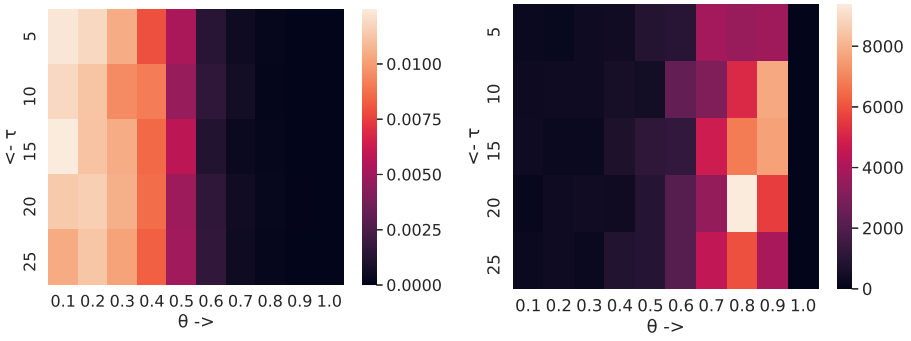
(E) Amazon (F) NetScience

(G) Power (H) GrQc

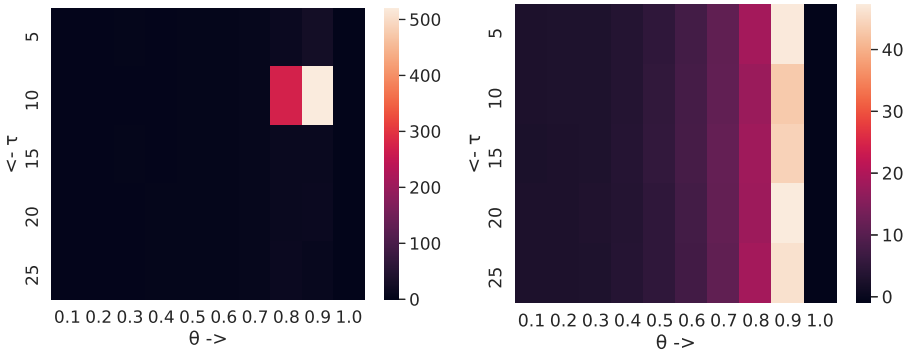
FIGURE 6.2: Safe Zone Predicted with Average Isolability Metric Corresponding to τ and θ Ranges over Mean Value of Ratios in Different Datasets



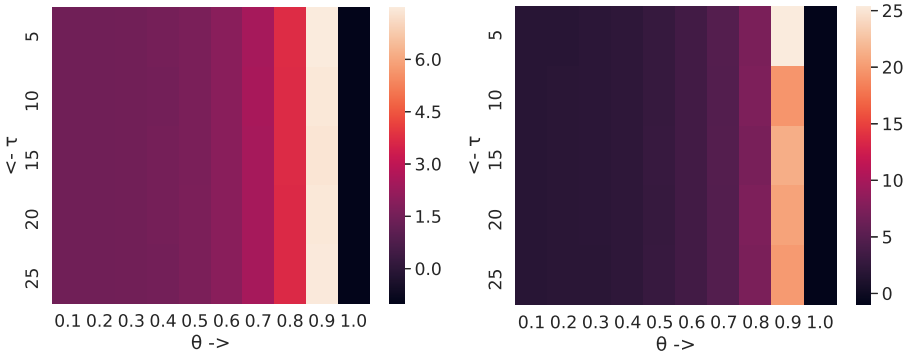
(A) Football (B) Celegansneural



(C) USAir97 (D) Political blogs



(E) Amazon (F) NetScience



(G) Power (H) GrQc

FIGURE 6.3: Safe Zone Predicted with External Density Metric Corresponding to τ and θ Ranges over Mean Value of Ratios in Different Datasets

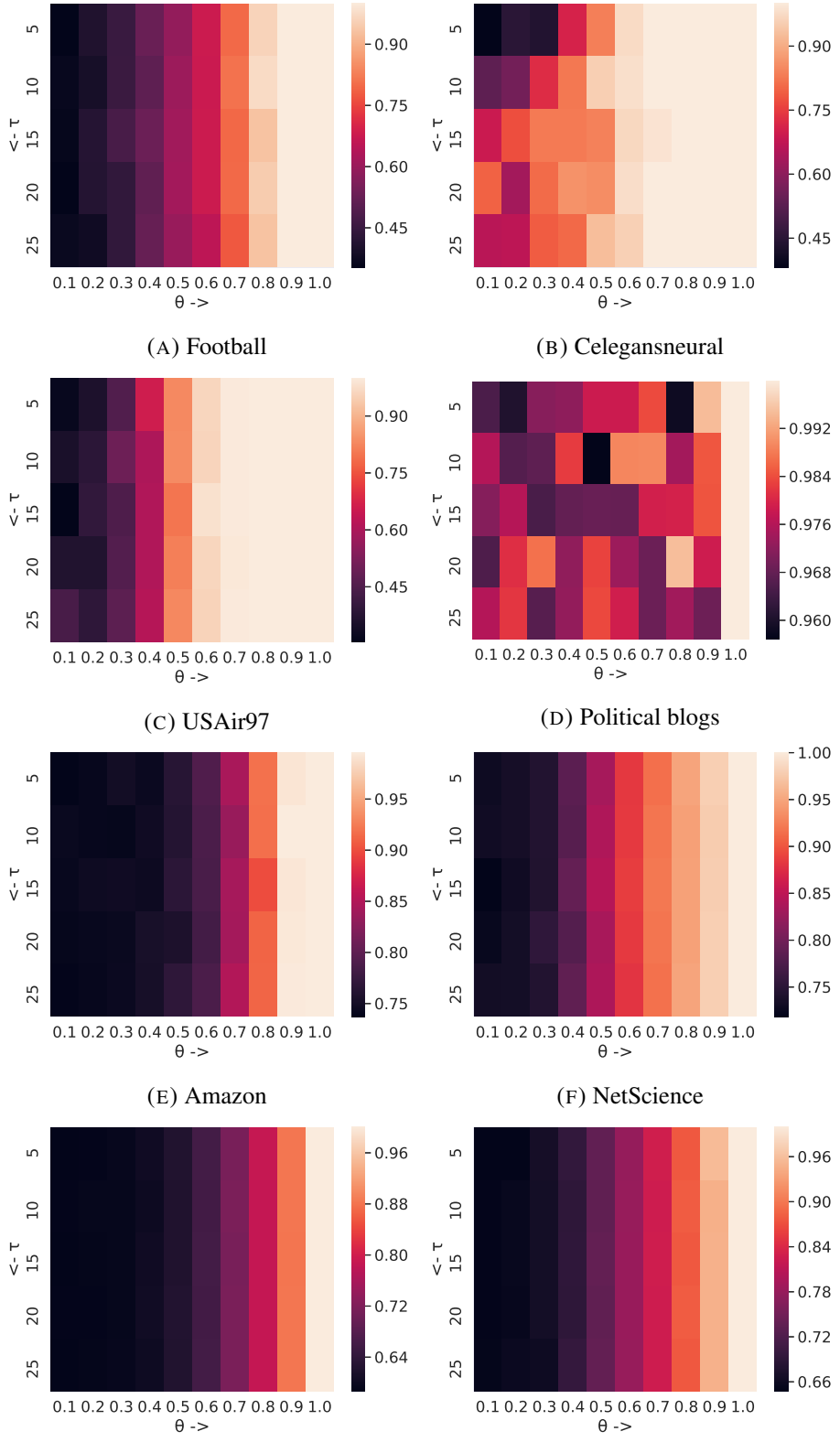
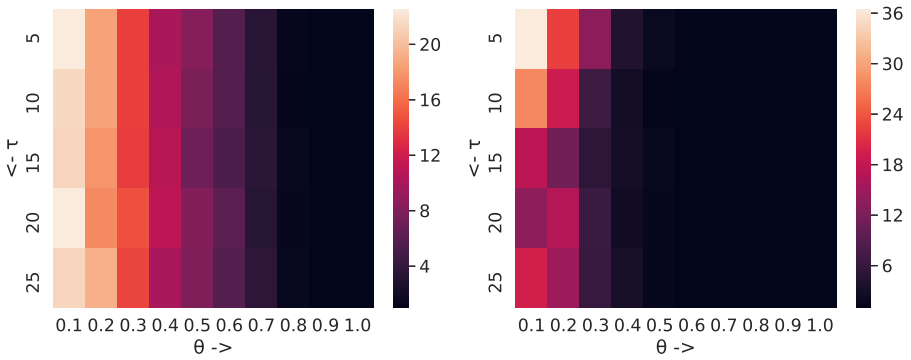
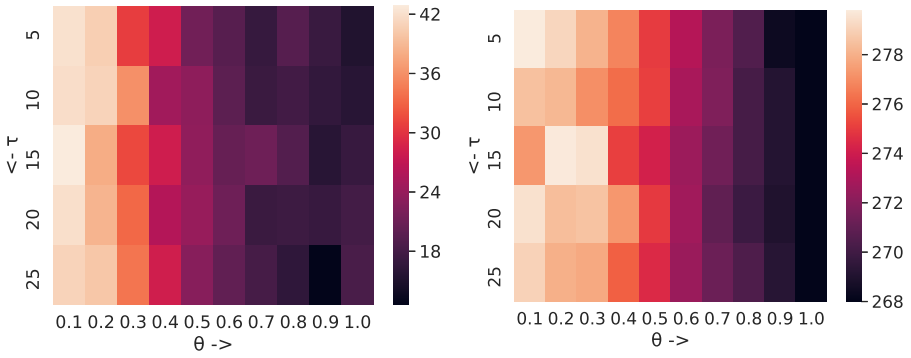


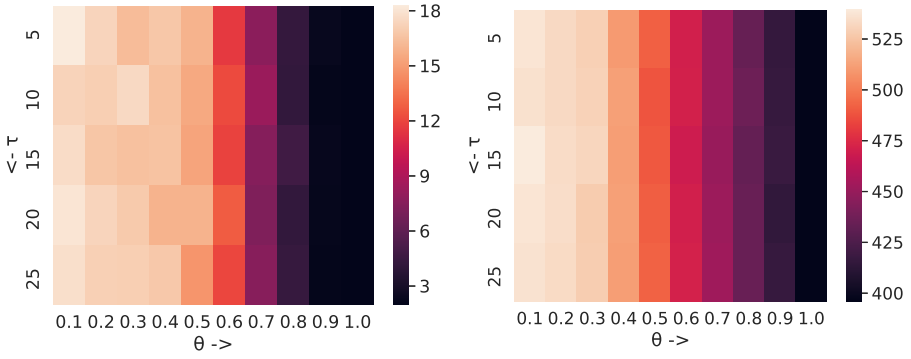
FIGURE 6.4: Safe Zone Predicted with Coverage Metric Corresponding to τ and θ Ranges over Mean Value of Ratios in Different Datasets



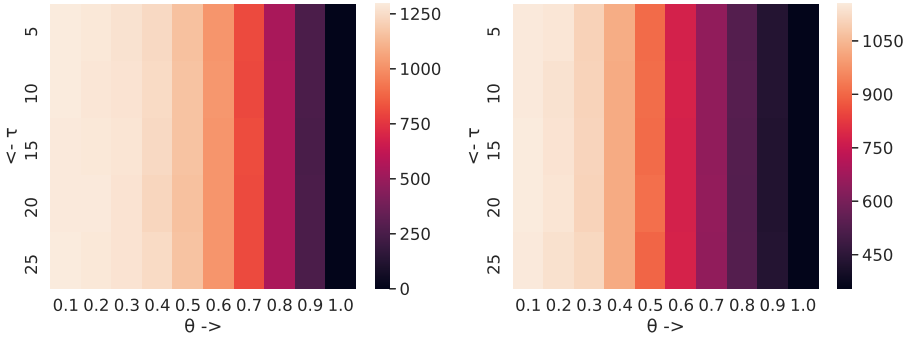
(A) Football (B) Celegansneural



(C) USAir97 (D) Political blogs

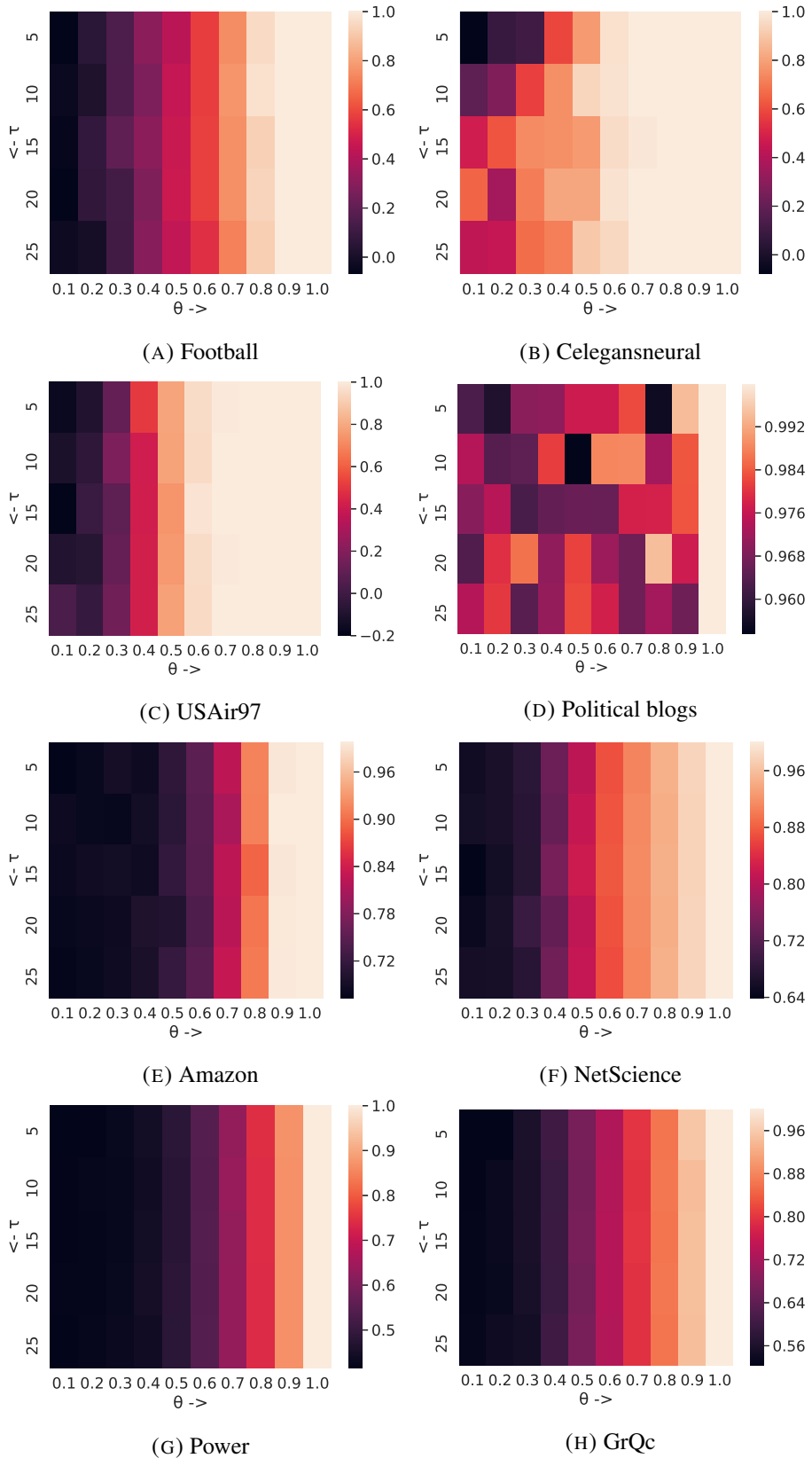


(E) Amazon (F) NetScience



(G) Power (H) GrQc

FIGURE 6.5: Safe Zone Predicted with Cluster Count Metric Corresponding to τ and θ Ranges over Mean Value of Ratios in Different Datasets



(A) Football (B) Celegansneural

(C) USAir97 (D) Political blogs

(E) Amazon (F) NetScience

(G) Power (H) GrQc

FIGURE 6.6: Safe Zone Predicted with Modularity Metric Corresponding to τ and θ Ranges over Mean Value of Ratios in Different Datasets

6.4.3 Parameter Analysis with Community Detection Performance Metrics

In this section, we analyze the parameters τ and θ in the performance metrics of community detection. For parameter analysis, we use quality metrics with unknown ground truth community structure, namely, average isolability, external density, coverage, cluster count, and modularity [177]. To evaluate the quality metrics corresponding to the parameters τ and θ, we consider eight real-world networks. The results obtained on each network corresponding to all 50 pairs of τ and θ values are shown in figs. 6.2 to 6.6.

6.4.3.1 Average isolability

Before discussing the average isolability corresponding to different values of τ and θ, we define the isolability metric for measuring the quality of the community structure. The isolability of a community C_i is defined as the ratio of intra-links and total links of the community C_i. As in the case of the relative density measure, isolability is defined as follows:

$$Isolability(C_i) = \frac{Links_{intra}(C_i)}{Links_{intra}(C_i) + Links_{inter}(C_i)} \tag{6.22}$$

Let $|C_S|$ denote the number of communities in the community structure C_S; then, the average isolability defined as follows:

$$Avg\,Isolability(C_S) = \frac{\sum_{C_i \in C_S} Isolability(C_i)}{|C_S|} \tag{6.23}$$

The results obtained for the average isolability metric for different datasets are shown in FIGURE 6.2. Clearly, the average isolability appears to reach its maximum value for τ and θ values in the ranges 5–25 and 0.8–1.0, respectively, in the Football dataset. Other value pairs of τ and θ generate quite small average isolability values. Hence, the safe zone for the Football dataset is obtained with $\tau = 5$–25 and $\theta = 0.8$–1.0. Similarly, the safe zone for the Celegansneural dataset is obtained with $\tau = 5$–25 and $\theta = 0.6$–1.0. The safe zone for the USAir97 dataset is obtained with $\tau = 25$ and $\theta = 0.8$. The safe zones for both the Political Blogs and Amazon datasets are obtained with $\tau = 5$–25; however, the corresponding values for θ are different: 0.1–0.4 and 0.5–0.7, respectively. Similarly, the safe zone for the NetScience dataset is obtained with $\tau = 5$–25 and θ in the range 0.6–1.0. The safe zone for the Power dataset is obtained with $\tau = 5$–25 and $\theta = 1.0$. Similarly, the safe zone for the GrQc dataset is obtained with $\tau = 5$–25 and θ in the range 0.8–1.0.

6.4.3.2 External density

The external density of a community structure is defined as the ratio of the number of edges that link distinct communities to the maximum number of possible such edges, that is, the ratio of intra- and inter-community edges. The external density $ExtrDens$ of C_S is calculated as follows:

$$ExtrDens(C_S) = \frac{\sum_{C_i \in C_S} Links_{intra}(C_i)}{\sum_{C_i \in C_S} Links_{inter}(C_i)} \tag{6.24}$$

Small values of external density indicate better-quality communities. External density becomes insignificant for clusters between which no edges exist. For a graph with only one cluster, the external density can be assumed to be -1.

The results obtained for the external density quality metric are shown in FIGURE 6.3. For the Football dataset, the external density reached its smallest value with θ and τ values in the ranges 0.1–0.5 and 5–25, respectively. Hence, the safe zone for the Football dataset is obtained with $\theta = 0.1$–0.5 and $\tau = 5$–25. Similarly, the safe zone for the Celegansneural dataset is obtained with $\tau = 5$–25 and θ in the range 0.5–1.0. The safe zone for the USAir97 dataset is obtained with τ values in the range 5–25 and $\theta = 0.6$–1.0. The safe zone for the Political Blogs dataset is obtained with $\tau = 5$–25 and θ values in the range 0.1–0.5. Similarly, the safe zones for both the Amazon and NetScience datasets are obtained with τ values in the range 5–25 and θ values in the range 0.1–0.7. The safe zone for the Power dataset is obtained with $\tau = 5$–25 and $\theta = 1.0$. Similarly, the safe zone for the GrQc dataset is obtained with $\tau = 5$–25 and θ values in the range 0.1–0.7 and 1.0.

6.4.3.3 Coverage

The coverage of the community structure C_S is defined as the fraction of all community intra-connections to the total number of connections in the

network. It can be estimated as follows:

$$Coverage(C_S) = \frac{\sum_{C_i \in C_S} Links_{intra}(C_i)}{|E|} \tag{6.25}$$

Higher values of coverage indicate better-quality community structure. Coverage becomes 0 if all the communities have a single node, that is, $|C_S| = |V|$. Similarly, coverage becomes 1 if all the nodes belong to the same community, that is, $|C_S| = 1$.

The results obtained for coverage corresponding to a range of τ and θ values are shown in FIGURE 6.4. For the Football dataset, coverage attained its highest value for τ and θ values in the ranges 5–25 and 0.9–1.0, respectively. In these ranges, coverage attained a value of at least 0.90. However, it varied with θ. Hence, the safe zone for the Football dataset is reached at least at 0.75 with τ and θ values in the ranges 5–25 and 0.8–1.0, respectively. Similarly, the safe zones for both the Celegansneural and USAir97 datasets are reached at least at 0.80 with τ and θ values in the ranges 5–25 and 0.6–1.0, respectively. For the Political Blogs dataset, coverage reached its highest value for τ values in the range 5–25 and $\theta = 1.0$. The safe zone for the Political Blogs dataset is reached at least at 0.95 with τ and θ values in the ranges 5–25 and 0.5–1.0, respectively. Similarly, the safe zones for both the Amazon and NetScience datasets are reached at least at 0.90 with τ and θ values in the ranges 5–25 and 0.8–1.0, respectively. The safe zone for the Power dataset is obtained with $\tau = 5$–25 and $\theta = 1.0$. Similarly, the safe zone for

GrQc dataset is obtained with $\tau = 5$–25 and θ values in the range 0.9–1.0.

6.4.3.4 Cluster count

The cluster count of a community structure C_S is defined as the number of communities present in C_S. The results obtained for cluster count corresponding to a range of τ and θ values are shown in FIGURE 6.5. For the Football dataset, cluster count attained values in the range 5–15 for τ and θ values in the ranges 5–25 and 0.3–0.7, respectively. Similarly, the safe zones for both the Celegansneural and USAir97 datasets are reached for values in the range 15–20 with τ and θ values in the ranges 5–25 and 0.6–0.9, respectively. The safe zone for the Political Blogs dataset is obtained with τ values in the range 5–25 and $\theta = 1.0$. Similarly, the safe zone for the Amazon dataset is obtained with τ values in the range 10–15 and $\theta = 0.9$. The safe zone for the NetScience dataset is obtained with τ and θ values in the ranges 5–25 and 0.9–1.0, respectively. The safe zone for the Power dataset is obtained with $\tau = 5$–25 and $\theta = 0.9$–1.0. Similarly, the safe zone for the GrQc dataset is obtained with $\tau = 5$–25 and θ values in the range 0.8–1.0.

6.4.3.5 Modularity

Modularity is the most widely accepted quality metric to measure the quality of predicted communities. The modularity of a community

structure C_S is defined as the fraction of intra-connections in the network by subtracting the corresponding expected values. The value of modularity approaches 0 if the number of intra-connections in the communities is small. Similarly, the value of modularity approaches 1 if the number of intra-connections in the communities is higher. Modularity can be estimated as follows:

$$Modularity(C_S) = \sum_{C_i \in C_S} \left\{ \frac{Links_{intra}(C_i)}{|E|} - \left(\frac{Links_{inter}(C_i)}{|E|} \right)^2 \right\} \quad (6.26)$$

The results obtained for modularity corresponding to a range of τ and θ values are shown in FIGURE 6.6. For the Football dataset, modularity reached its highest value for τ and θ values in the ranges 5–25 and 0.9–1.0, respectively. Within this range, modularity attained a value of at least 1.0. Modularity attained a value of at least 0.80 with τ values in the range 5–25, but it varied with θ. Hence, the safe zone for the Football dataset is reached at least at 0.80 with τ and θ values in the ranges 5–25 and 0.8–1.0, respectively. Similarly, the safe zone for both the Celegansneural and USAir97 datasets reached its highest value with τ and θ values in the ranges 5–25 and 0.7–1.0, respectively. For the Political Blogs dataset, modularity reached its highest value for τ values in the range 5–25 and $\theta = 1.0$. The safe zone for the Political Blogs dataset is reached at least at 0.95 with τ and θ values in the ranges 5–25 and 0.4–1.0, respectively. Similarly, the safe zones for both the Amazon and

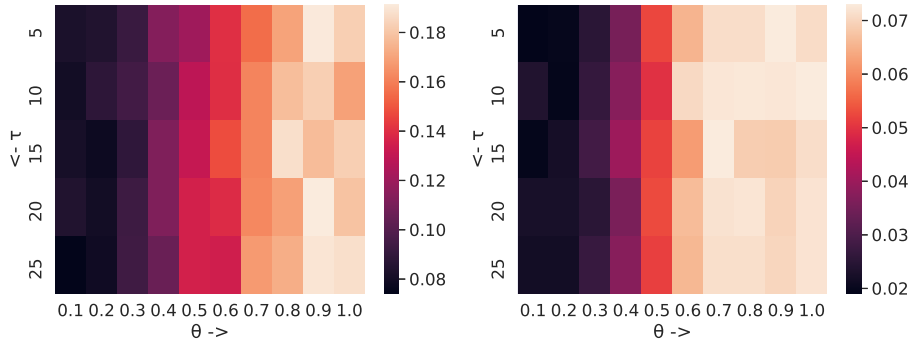
NetScience datasets are reached at least at 0.96 with τ and θ values in the ranges 5–25 and 0.9–1.0, respectively. The safe zone for the Power dataset is obtained with $\tau = 5$–25 and $\theta = 1.0$. Similarly, the safe zone for the GrQc dataset is obtained with $\tau = 5$–25 and θ values in the range 0.9–1.0.

6.4.4 Parameter Analysis with Link Prediction Performance Metrics

As in the previous section, we analyze the parameters τ and θ and the corresponding performance metrics of link prediction. Here, we use four quality metrics: AUPR, recall, AUC, and precision to predict the safe zones for the parameters. To evaluate the quality metrics corresponding to the parameters τ and θ, we consider eight real-world networks. The results obtained on each of the eight networks corresponding to all 50 pairs of τ and θ values are shown in figs. 6.7 to 6.10.

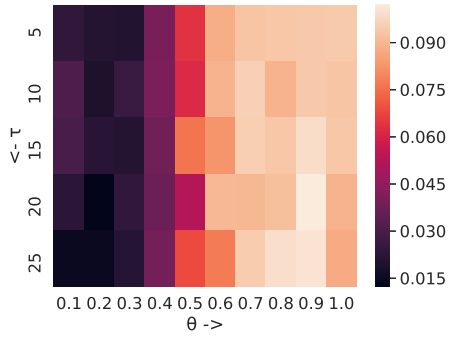
6.4.4.1 AUPR

The results obtained for AUPR corresponding to a range of τ and θ values are shown in FIGURE 6.7. For the Football dataset, AUPR attained a value of at least 0.28 for τ and θ values in the ranges 5–25 and 0.8–1.0, respectively. Hence, the safe zone for the Football dataset is obtained with τ and θ values in the ranges 5–25 and 0.8–1.0, respectively. The safe zones for both the Celegansneural and USAir97 datasets are obtained with

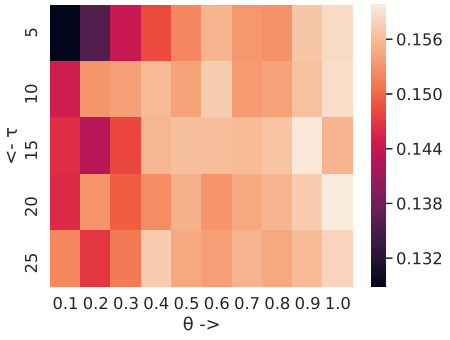


(A) Football

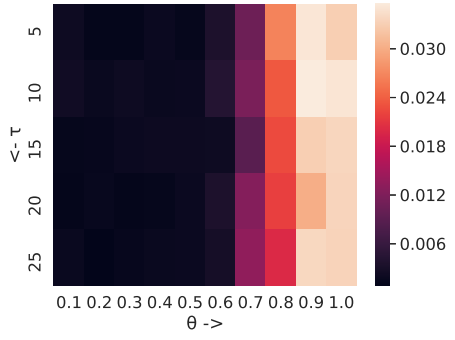
(B) Celegansneural



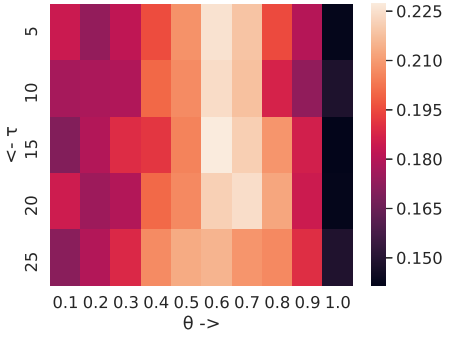
(C) USAir97



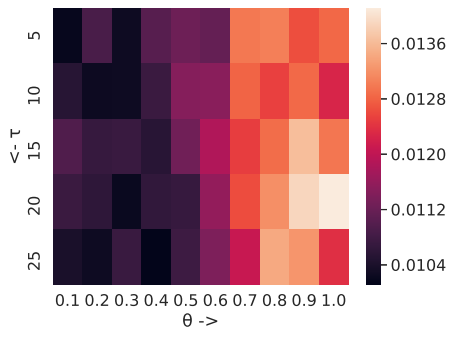
(D) Political blogs



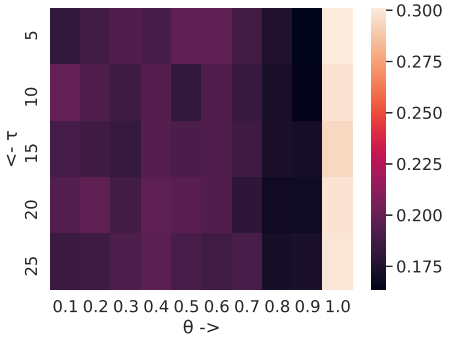
(E) Amazon



(F) NetScience

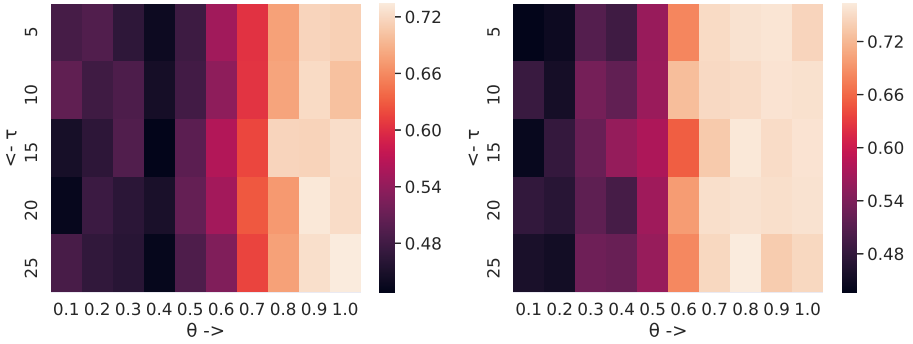


(G) Power



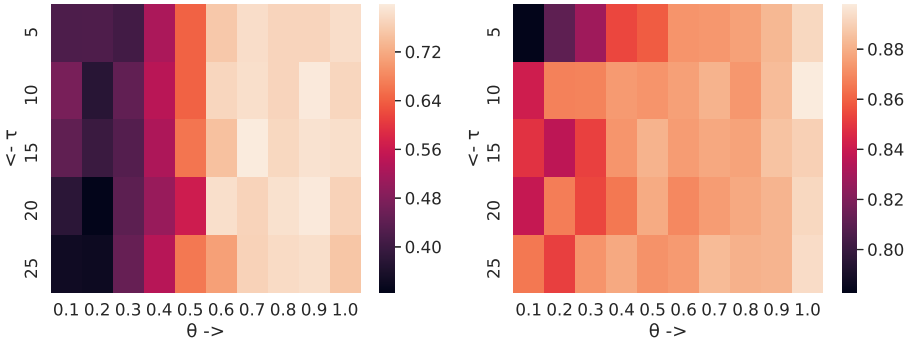
(H) GrQc

FIGURE 6.7: Safe Zone Predicted with AUPR Metric Corresponding to τ and θ Ranges over Mean Value of Ratios in Different Datasets



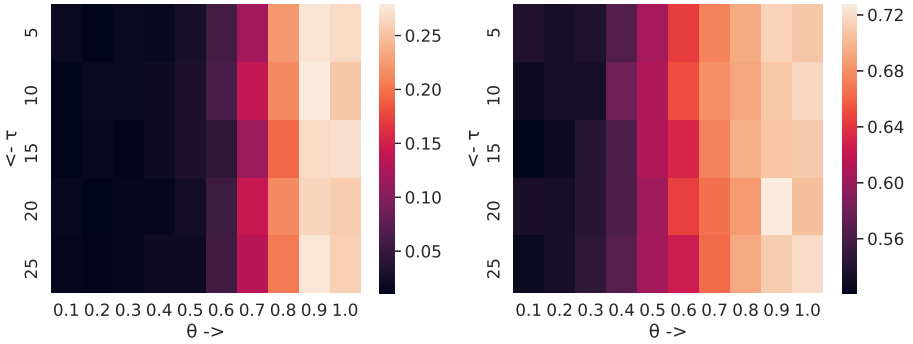
(A) Football

(B) Celegansneural



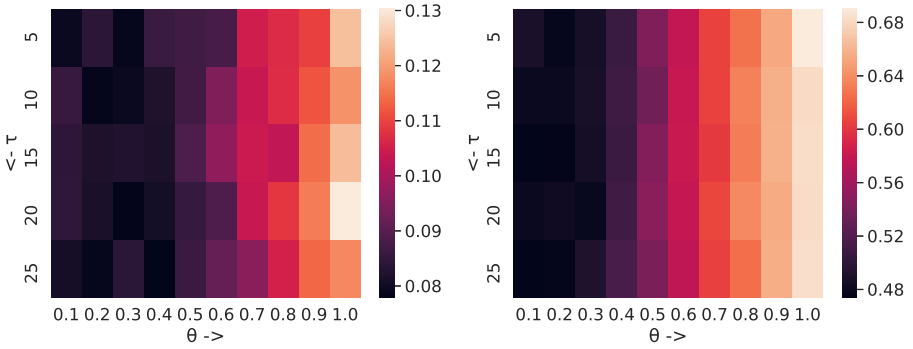
(C) USAir97

(D) Political blogs



(E) Amazon

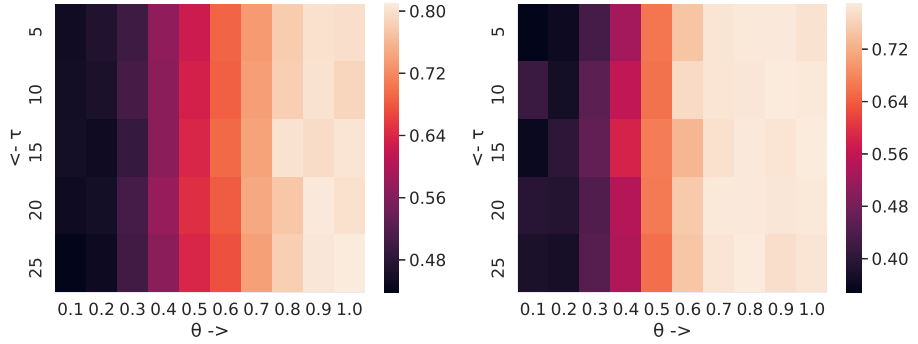
(F) NetScience



(G) Power

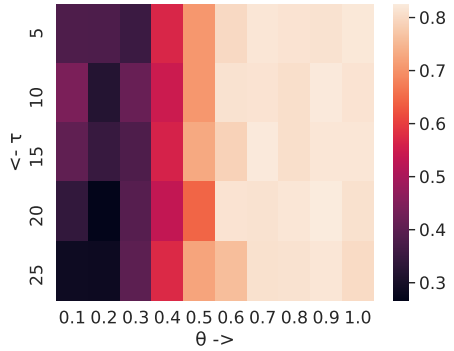
(H) GrQc

FIGURE 6.8: Safe Zone Predicted with Recall Metric Corresponding to τ and θ Ranges over Mean Value of Ratios in Different Datasets

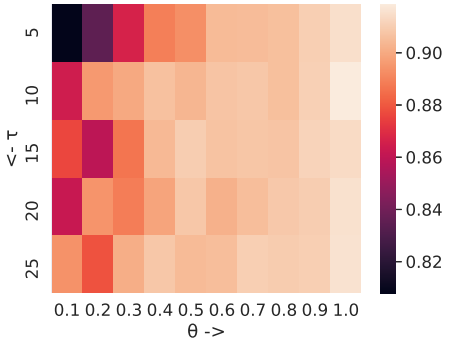


(A) Football

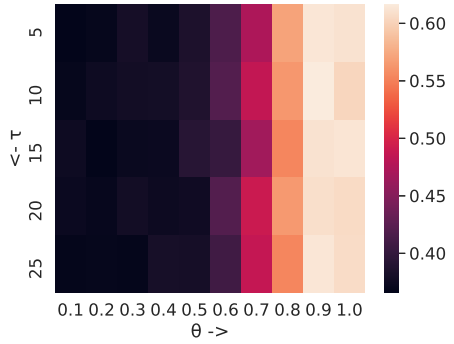
(B) Celegansneural



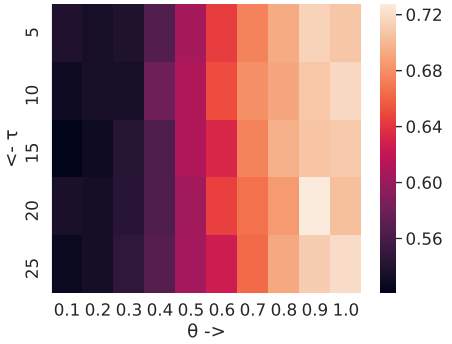
(C) USAir97



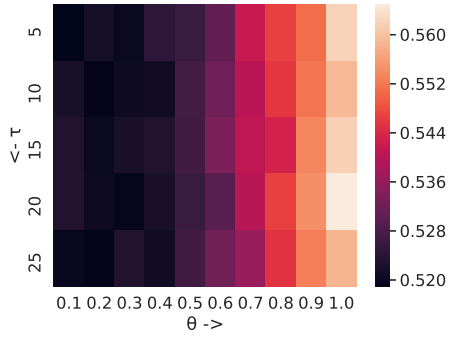
(D) Political blogs



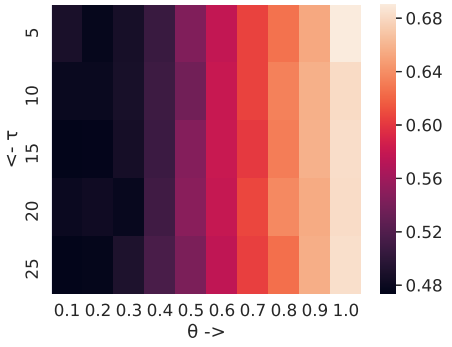
(E) Amazon



(F) NetScience

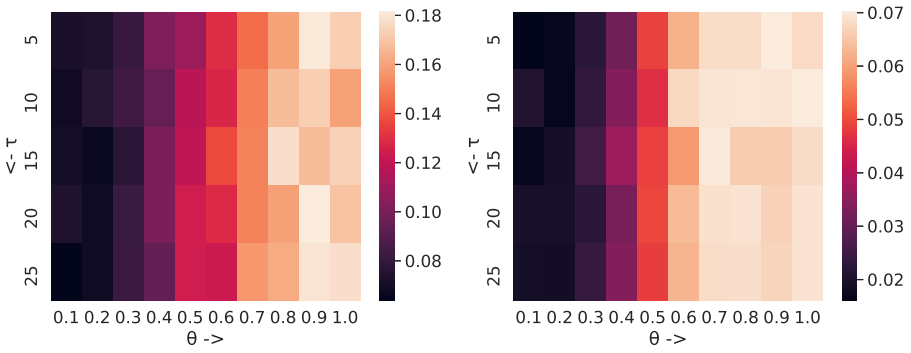


(G) Power



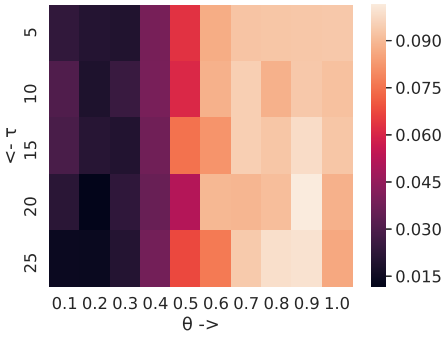
(H) GrQc

FIGURE 6.9: Safe Zone Predicted with AUC Metric Corresponding to τ and θ Ranges over Mean Value of Ratios in Different Datasets

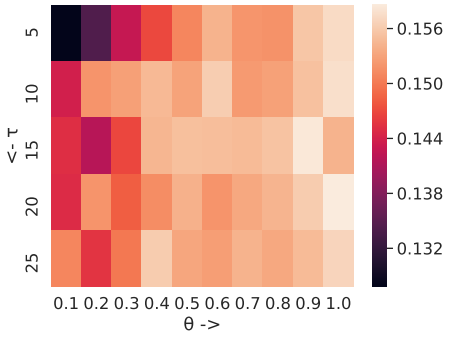


(A) Football

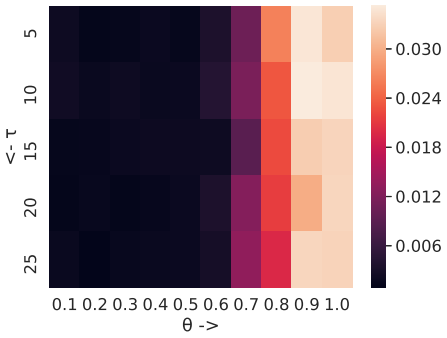
(B) Celegansneural



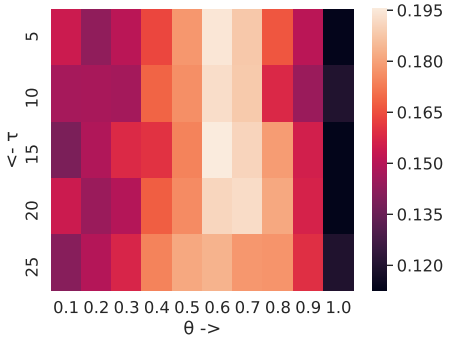
(C) USAir97



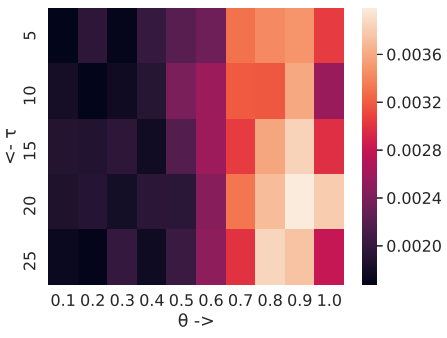
(D) Political blogs



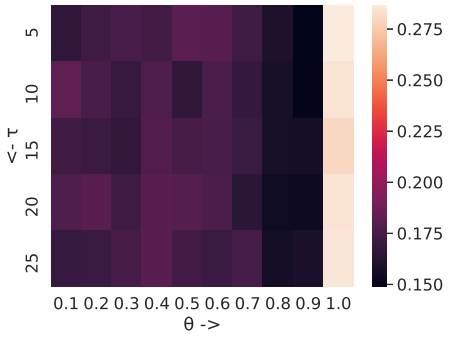
(E) Amazon



(F) NetScience



(G) Power



(H) GrQc

FIGURE 6.10: Safe Zone Predicted with Precision Metric Corresponding to τ and θ Ranges over Mean Value of Ratios in Different Datasets

τ and θ values in the ranges 5–25 and 0.5–1.0. For the Political Blogs dataset, AUPR attained a value of at least 0.060 for τ and θ values in the ranges 5–25 and 0.3–1.0, respectively. Hence, the safe zone for the Political Blogs dataset is obtained with τ and θ values in the ranges 5–25 and 0.3–1.0, respectively. Similarly, the safe zone is predicted for the Amazon dataset with τ and θ values in the ranges 10–20 and 0.4–0.6, respectively. The safe zone is predicted for the NetScience dataset with τ and θ values in the ranges 5–25 and 0.6–1.0, respectively. The safe zone for the Power dataset is obtained with $\tau = 5$–25 and $\theta = 0.7$–1.0. Similarly, the safe zone for the GrQc dataset is obtained with $\tau = 5$–25 and $\theta = 1.0$.

6.4.4.2 Recall

The results obtained for the recall corresponding to a range of τ and θ values are shown in FIGURE 6.8. For the Football dataset, the recall attained a value of at least 0.69 for τ and θ values in the ranges 5–25 and 0.8–1.0, respectively. Hence, the safe zone for the Football dataset is obtained with τ and θ values in the ranges 5–25 and 0.8–1.0, respectively. Similarly, the safe zones for both the Celegansneural and USAir97 datasets are obtained with τ and θ values in the ranges 5–25 and 0.6–1.0, respectively. For the Political Blogs dataset, the recall attained a value of at least 0.80 for τ and θ values in the ranges 5–25 and 0.5–1.0, respectively. Hence, the safe zone for the Political blogs dataset is

obtained with τ and θ values in the ranges 5–25 and 0.5–1.0, respectively. Similarly, the safe zone is predicted for the Amazon dataset with τ and θ values in the ranges 15–20 and 0.5, respectively. The safe zone is predicted for the NetScience dataset with τ and θ values in the ranges 5–25 and 0.1–0.4, respectively. The safe zone for the Power dataset is obtained with $\tau = 5$–25 and $\theta = 0.9$–1.0. Similarly, the safe zone for the GrQc dataset is obtained with $\tau = 5$–25 and $\theta = 0.8$–1.0.

6.4.4.3 AUC

Results obtained for AUC corresponding to a range of τ and θ values are shown in FIGURE 6.9. For the Football dataset, AUC attained a value of at least 0.75 for τ and θ values in the ranges 5–25 and 0.7–1.0, respectively. Hence, the safe zone for the Football dataset is obtained with τ and θ values in the ranges 5–25 and 0.7–1.0, respectively. Similarly, the safe zones for both the Celegansneural and USAir97 datasets are obtained with τ and θ values in the ranges 5–25 and 0.6–1.0, respectively. For the Political Blogs dataset, AUC attained a value of at least 0.85 for τ and θ values in the ranges 5–25 and 0.4–1.0, respectively. Hence, the safe zone for the Political Blogs dataset is obtained with τ and θ values in the ranges 5–25 and 0.4–1.0. Similarly, the safe zone is predicted for the Amazon dataset with τ and θ values in the ranges 5–25 and 0.8–1.0, respectively. The safe zone is predicted for the NetScience dataset with τ and θ values in the ranges 5–25 and 0.1–0.4, respectively. The safe zone

for the Power dataset is obtained with $\tau = 5$–25 and $\theta = 0.9$–1.0. Similarly, the safe zone for the GrQc dataset is obtained with $\tau = 5$–25 and $\theta = 0.7$–1.0.

6.4.4.4 Precision

The results obtained for the precision corresponding to a range of τ and θ values are shown in FIGURE 6.10. For the Football dataset, the precision attained a value of at least 0.28 for τ and θ values in the ranges 5–25 and 0.8–1.0, respectively. Hence, the safe zone for the Football dataset is obtained with τ and θ values in these ranges. Similarly, the safe zones for both the Celegansneural and USAir97 datasets are obtained with τ and θ values in the ranges 5–25 and 0.6–1.0, respectively. For the Political Blogs dataset, the precision attained its highest value for τ and θ values in the ranges 5–25 and $0.6-0-0.9$, respectively. Hence, the safe zone for the Political Blogs dataset is obtained with τ and θ values in these ranges. Similarly, the safe zone is predicted for the Amazon dataset with $\tau = 25$ and θ values in the range 0.1–0.8. The safe zone is predicted for the NetScience dataset with τ and θ values in the ranges 5–25 and 0.7–1.0, respectively. The safe zone for the Power dataset is obtained with $\tau = 5$–25 and $\theta = 0.8$–0.9. Similarly, the safe zone for the GrQc dataset is obtained with $\tau = 5$–25 and $\theta = 1.0$.

TABLE 6.3: Comparison of CLP-ID with the State-of-the-art Algorithms in terms of Accuracy Quantified by AUPR

Dataset	Ratio	Algorithm									
		N2V	CCLP	CCLP2	CN	RA	PA	CAR	NLC	LNBCN	CLP-ID
Football	0.1	0.11648	0.09221	0.09114	0.08339	0.20690	0.00630	0.11150	0.11104	0.09576	0.13589
	0.2	0.17146	0.15257	0.15288	0.18006	0.32175	0.01324	0.13857	0.16503	0.13978	0.19900
	0.3	0.21408	0.18937	0.20176	0.18599	0.27099	0.02006	0.17643	0.17291	0.20937	0.22419
	0.4	0.22692	0.17153	0.21998	0.21168	0.31235	0.03061	0.12834	0.21139	0.18535	0.23212
	0.5	0.22218	0.18006	0.20831	0.21514	0.27106	0.03683	0.02457	0.19971	0.17658	0.22690
Celegansneural	0.1	0.02115	0.03701	0.04286	0.03545	0.06273	0.02376	0.02770	0.05565	0.03484	0.04574
	0.2	0.03655	0.08605	0.06372	0.06573	0.09210	0.04613	0.06054	0.08742	0.07102	0.06758
	0.3	0.04831	0.09358	0.07410	0.07971	0.12581	0.06653	0.06732	0.11603	0.08839	0.08715
	0.4	0.05831	0.11038	0.08512	0.09695	0.12481	0.07647	0.07909	0.12128	0.10377	0.09794
	0.5	0.06608	0.11224	0.10499	0.10144	0.15844	0.09421	0.12026	0.12282	0.11295	0.10108
USAir97	0.1	0.04773	0.24960	0.22894	0.24963	0.47136	0.24797	0.23332	0.25916	0.27705	0.27899
	0.2	0.08654	0.32974	0.29227	0.33254	0.52955	0.33712	0.28597	0.35803	0.35068	0.34962
	0.3	0.10088	0.41507	0.33532	0.37018	0.54411	0.39694	0.29547	0.43216	0.38686	0.38250
	0.4	0.12430	0.42728	0.36680	0.39477	0.52481	0.42803	0.32622	0.44177	0.43565	0.39953
	0.5	0.12583	0.41710	0.37339	0.41568	0.52200	0.44449	0.28613	0.44198	0.40725	0.41050
Political blogs	0.1	0.01505	0.07578	0.08632	0.07601	0.07631	0.03159	0.06614	0.09520	0.07092	0.08456
	0.2	0.02544	0.12308	0.12774	0.13383	0.13019	0.06035	0.10921	0.16111	0.13027	0.13753
	0.3	0.03346	0.16583	0.16837	0.16738	0.17318	0.08655	0.13508	0.19320	0.17144	0.17641
	0.4	0.03970	0.19618	0.19054	0.19578	0.19567	0.10890	0.14121	0.22493	0.20066	0.20218
	0.5	0.04422	0.21503	0.20232	0.20809	0.21393	0.13081	0.15659	0.23854	0.21561	0.21826
Amazon	0.1	0.00523	0.01431	0.10677	0.01412	0.01472	0.10639	0.04291	0.07819	0.01615	0.02870
	0.2	0.00624	0.02659	0.10691	0.03276	0.02669	0.11216	0.05426	0.06901	0.03079	0.03882
	0.3	0.00718	0.03196	0.09231	0.04250	0.03586	0.13212	0.07493	0.08971	0.03563	0.04084
	0.4	0.00986	0.04226	0.08164	0.04316	0.03772	0.12817	0.04825	0.07327	0.03907	0.04389
	0.5	0.00802	0.04502	0.06865	0.04236	0.04318	0.13722	0.10780	0.08703	0.04687	0.04778
NetScience	0.1	0.09631	0.19348	0.16204	0.18831	0.69874	0.00375	0.18133	0.15769	0.23989	0.14942
	0.2	0.15169	0.23835	0.22757	0.26070	0.67891	0.00627	0.24240	0.23014	0.30209	0.21059
	0.3	0.18437	0.27084	0.24601	0.28896	0.65239	0.00838	0.25818	0.26225	0.32452	0.25496
	0.4	0.20763	0.27631	0.24516	0.29353	0.59811	0.00834	0.25950	0.25939	0.32248	0.29086
	0.5	0.20378	0.28501	0.28182	0.29096	0.55626	0.01067	0.24630	0.28844	0.31685	0.31682
Power	0.1	0.00237	0.00810	0.00911	0.00434	0.01293	0.00008	0.03623	0.00759	0.00763	0.00628
	0.2	0.00379	0.01652	0.01700	0.00930	0.01311	0.00013	0.07569	0.01565	0.01691	0.01100
	0.3	0.00434	0.01747	0.02686	0.01179	0.02128	0.00020	0.00008	0.02096	0.01988	0.01524
	0.4	0.00431	0.02966	0.03433	0.01537	0.02468	0.00024	0.00011	0.03776	0.02913	0.01963
	0.5	0.00384	0.02819	0.04014	0.01984	0.03219	0.00032	0.00014	0.04735	0.04581	0.02288
GrQc	0.1	0.04773	0.24960	0.22894	0.24963	0.47136	0.24797	0.23332	0.25916	0.27705	0.27899
	0.2	0.08654	0.32974	0.29227	0.33254	0.52955	0.33712	0.28597	0.35803	0.35068	0.34962
	0.3	0.10088	0.41507	0.33532	0.37018	0.54411	0.39694	0.29547	0.43216	0.38686	0.38250
	0.4	0.12430	0.42728	0.36680	0.39477	0.52481	0.42803	0.32622	0.44177	0.43565	0.39953
	0.5	0.12583	0.41710	0.37339	0.41568	0.52200	0.44449	0.28613	0.44198	0.40725	0.41050

TABLE 6.4: Comparison of CLP-ID with the State-of-the-art Algorithms in terms of Accuracy Quantified by Recall

Dataset	Ratio	Algorithm									
		N2V	CCLP	CCLP2	CN	RA	PA	CAR	NLC	LNBCN	CLP-ID
Football	0.1	0.79677	0.80645	0.71774	0.69355	0.80645	0.17742	0.29032	0.85484	0.82258	0.89758
	0.2	0.76152	0.80488	0.67886	0.81301	0.78049	0.31707	0.16260	0.73984	0.78049	0.87764
	0.3	0.73080	0.75544	0.67663	0.76087	0.77174	0.26087	0.79348	0.60870	0.76630	0.76658
	0.4	0.69648	0.63821	0.64431	0.68699	0.72764	0.34959	0.80894	0.58537	0.66260	0.70589
	0.5	0.63627	0.55700	0.42020	0.57003	0.59609	0.31922	0.90554	0.43974	0.49186	0.60195
Celegansneural	0.1	0.67535	0.84651	0.76512	0.92093	0.88372	0.65581	0.15349	0.83256	0.83721	0.88628
	0.2	0.63659	0.85116	0.67558	0.87907	0.81861	0.70465	0.10233	0.77209	0.83721	0.85116
	0.3	0.59121	0.73488	0.62713	0.82946	0.77209	0.68372	0.04031	0.68372	0.78760	0.79868
	0.4	0.55496	0.70814	0.54593	0.72442	0.69767	0.65930	0.84535	0.58954	0.71047	0.72221
	0.5	0.50385	0.59218	0.46136	0.61173	0.61080	0.62384	0.92086	0.50000	0.54935	0.63152
USAir97	0.1	0.85603	0.92958	0.81690	0.97183	0.94836	0.77934	0.75587	0.89671	0.92488	0.92911
	0.2	0.81706	0.88967	0.79930	0.95305	0.92723	0.82160	0.64319	0.87089	0.90845	0.91455
	0.3	0.79509	0.90125	0.75000	0.92790	0.90282	0.84796	0.54075	0.85267	0.90909	0.89828
	0.4	0.77438	0.87427	0.74971	0.89659	0.84841	0.82491	0.41833	0.81199	0.89307	0.88320
	0.5	0.74431	0.81844	0.72013	0.85231	0.86171	0.81279	0.20602	0.77046	0.83161	0.84073
Political blogs	0.1	0.78441	0.92105	0.84510	0.94557	0.90817	0.90518	0.61483	0.87081	0.92492	0.93505
	0.2	0.77207	0.90162	0.81938	0.93540	0.90205	0.89846	0.50621	0.86270	0.92522	0.93219
	0.3	0.75366	0.89282	0.81001	0.90809	0.88417	0.90431	0.38557	0.84548	0.89952	0.91443
	0.4	0.73265	0.87925	0.78275	0.87843	0.85727	0.89967	0.25398	0.81249	0.88126	0.88276
	0.5	0.70902	0.81975	0.73684	0.83525	0.82089	0.89729	0.79626	0.76926	0.83239	0.83823
Amazon	0.1	0.17408	0.29744	0.28974	0.32564	0.25674	0.35339	0.19409	0.30090	0.30937	0.34363
	0.2	0.17610	0.28801	0.26346	0.26026	0.27728	0.31471	0.14249	0.29955	0.31454	0.30886
	0.3	0.16689	0.27474	0.24091	0.29855	0.26046	0.30881	0.10950	0.27887	0.28089	0.27490
	0.4	0.15984	0.22768	0.21784	0.24623	0.25096	0.27095	0.02729	0.25465	0.25931	0.28654
	0.5	0.14556	0.21808	0.20365	0.21603	0.22382	0.26033	0.04467	0.22068	0.22958	0.23383
NetScience	0.1	0.85042	0.78909	0.71455	0.88364	0.85818	0.50182	0.27273	0.72000	0.72000	0.87818
	0.2	0.81117	0.68670	0.62659	0.81967	0.80874	0.59745	0.18215	0.69035	0.73953	0.83761
	0.3	0.76468	0.62090	0.51154	0.71810	0.76063	0.58202	0.10693	0.62576	0.60875	0.76288
	0.4	0.71176	0.53145	0.40611	0.64357	0.66181	0.51686	0.05561	0.51413	0.53327	0.68282
	0.5	0.62801	0.42378	0.29723	0.55361	0.55288	0.52079	0.90591	0.39971	0.40190	0.57429
Power	0.1	0.47455	0.10152	0.06894	0.15909	0.18030	0.36212	0.97727	0.08182	0.10152	0.19849
	0.2	0.37892	0.09022	0.04359	0.14632	0.14860	0.32146	0.98332	0.06520	0.08946	0.16164
	0.3	0.29480	0.05407	0.03335	0.11268	0.12481	0.33350	0.98787	0.04295	0.05356	0.12683
	0.4	0.21387	0.04094	0.02218	0.08946	0.08567	0.32563	0.99811	0.03412	0.03222	0.09856
	0.5	0.15881	0.01850	0.01046	0.07128	0.06976	0.37216	0.99666	0.01729	0.02881	0.07340
GrQc	0.1	0.85603	0.92958	0.81690	0.97183	0.94836	0.77934	0.75587	0.89671	0.92488	0.92911
	0.2	0.81706	0.88967	0.79930	0.95305	0.92723	0.82160	0.64319	0.87089	0.90845	0.91455
	0.3	0.79509	0.90125	0.75000	0.92790	0.90282	0.84796	0.54075	0.85267	0.90909	0.89828
	0.4	0.77438	0.87427	0.74971	0.89659	0.84841	0.82491	0.41833	0.81199	0.89307	0.88320
	0.5	0.74431	0.81844	0.72013	0.85231	0.86171	0.81279	0.20602	0.77046	0.83161	0.84073

TABLE 6.5: Comparison of CLP-ID with the State-of-the-art Algorithms in terms of Accuracy Quantified by AUC

Dataset	Ratio	Algorithm									
		N2V	CCLP	CCLP2	CN	RA	PA	CAR	NLC	LNBCN	CLP-ID
Football	0.1	0.86833	0.82246	0.80970	0.76389	0.84980	0.27477	0.53198	0.87228	0.82790	0.89412
	0.2	0.85318	0.83577	0.79617	0.84536	0.83063	0.32456	0.48344	0.82080	0.82236	0.88625
	0.3	0.84237	0.81083	0.79742	0.80876	0.82171	0.33468	0.43983	0.75932	0.82686	0.82095
	0.4	0.82824	0.75493	0.77819	0.77981	0.80662	0.40215	0.41748	0.76681	0.76631	0.79599
	0.5	0.80663	0.72252	0.67895	0.72850	0.74593	0.39556	0.45543	0.69626	0.69767	0.74935
Celegansneural	0.1	0.80692	0.84315	0.80909	0.84690	0.88441	0.75993	0.46214	0.87077	0.83633	0.87072
	0.2	0.78891	0.85640	0.78896	0.82423	0.84667	0.76871	0.45379	0.84561	0.83639	0.83331
	0.3	0.77215	0.80534	0.76284	0.79720	0.82033	0.75735	0.44247	0.81380	0.81175	0.80409
	0.4	0.75738	0.78702	0.71992	0.76243	0.77832	0.74175	0.44110	0.78001	0.77380	0.76735
	0.5	0.73607	0.71906	0.68108	0.71752	0.74378	0.73234	0.46996	0.71684	0.72015	0.73192
USAir97	0.1	0.90341	0.94441	0.91189	0.94656	0.95856	0.85418	0.83679	0.94692	0.93792	0.94189
	0.2	0.89586	0.92448	0.91005	0.93743	0.94779	0.88177	0.77591	0.93014	0.92010	0.92927
	0.3	0.88105	0.92915	0.87525	0.92474	0.93315	0.89113	0.71124	0.91947	0.92465	0.92414
	0.4	0.87247	0.90848	0.87653	0.90994	0.90314	0.88210	0.65069	0.89980	0.91364	0.90673
	0.5	0.85290	0.87656	0.85067	0.88527	0.90275	0.86967	0.50419	0.88120	0.88058	0.88915
Political blogs	0.1	0.88627	0.93964	0.93111	0.93569	0.93939	0.92962	0.74983	0.93964	0.93453	0.94070
	0.2	0.87965	0.92841	0.91895	0.93126	0.93510	0.92801	0.68689	0.93483	0.93158	0.93612
	0.3	0.87336	0.92308	0.91080	0.91737	0.92461	0.93095	0.61260	0.92539	0.92159	0.92409
	0.4	0.86642	0.91232	0.89436	0.90451	0.90914	0.92808	0.53562	0.91078	0.90832	0.90856
	0.5	0.85940	0.88486	0.86956	0.88465	0.88289	0.92734	0.45953	0.88432	0.88669	0.88885
Amazon	0.1	0.82974	0.61381	0.64068	0.62638	0.59122	0.40741	0.59506	0.63822	0.62088	0.63821
	0.2	0.76307	0.61565	0.62621	0.60030	0.61057	0.42819	0.55564	0.63911	0.62953	0.62568
	0.3	0.71089	0.61528	0.61615	0.62608	0.60740	0.47222	0.52759	0.63169	0.61838	0.61515
	0.4	0.66832	0.59658	0.60594	0.60539	0.60846	0.49726	0.47377	0.62146	0.61267	0.62652
	0.5	0.62202	0.59691	0.59997	0.59513	0.60003	0.53200	0.49468	0.60654	0.60323	0.60371
NetScience	0.1	0.89808	0.89382	0.85631	0.94105	0.92880	0.64480	0.52557	0.85927	0.85935	0.94382
	0.2	0.86408	0.84265	0.81256	0.90901	0.90404	0.68256	0.51012	0.84452	0.86910	0.91464
	0.3	0.83608	0.80976	0.75518	0.85824	0.88001	0.66808	0.46000	0.81229	0.80379	0.88337
	0.4	0.80369	0.76517	0.70257	0.82109	0.83059	0.65307	0.46497	0.75655	0.76611	0.83530
	0.5	0.75408	0.71148	0.64834	0.77620	0.77619	0.64562	0.46250	0.69953	0.70057	0.78731
Power	0.1	0.68117	0.55052	0.53435	0.57893	0.58964	0.45167	0.48941	0.54073	0.55052	0.59863
	0.2	0.61773	0.54494	0.52172	0.57264	0.57388	0.44897	0.49205	0.53248	0.54456	0.58099
	0.3	0.55961	0.52690	0.51662	0.55592	0.56209	0.45175	0.49394	0.52139	0.52666	0.56299
	0.4	0.51799	0.52040	0.51105	0.54440	0.54260	0.46043	0.49906	0.51701	0.51605	0.54895
	0.5	0.49660	0.50921	0.50522	0.53540	0.53471	0.47317	0.49834	0.50862	0.51437	0.53646
GrQc	0.1	0.90341	0.94441	0.91189	0.94656	0.95856	0.85418	0.83679	0.94692	0.93792	0.94189
	0.2	0.89586	0.92448	0.91005	0.93743	0.94779	0.88177	0.77591	0.93014	0.92010	0.92927
	0.3	0.88105	0.92915	0.87525	0.92474	0.93315	0.89113	0.71124	0.91947	0.92465	0.92414
	0.4	0.87247	0.90848	0.87653	0.90994	0.90314	0.88210	0.65069	0.89980	0.91364	0.90673
	0.5	0.85290	0.87656	0.85067	0.88527	0.90275	0.86967	0.50419	0.88120	0.88058	0.88915

TABLE 6.6: Comparison of CLP-ID with the State-of-the-art Algorithms in terms of Accuracy Quantified by Precision

Dataset	Ratio	Algorithm									
		N2V	CCLP	CCLP2	CN	RA	PA	CAR	NLC	LNBCN	CLP-ID
Football	0.1	0.12010	0.09431	0.09071	0.08588	0.21269	0.00703	0.06515	0.11411	0.09750	0.13661
	0.2	0.17405	0.15265	0.14732	0.18219	0.32283	0.01409	0.05315	0.16527	0.13821	0.19740
	0.3	0.21617	0.18415	0.18859	0.16552	0.26390	0.02095	0.04668	0.16503	0.20448	0.21788
	0.4	0.22856	0.15709	0.19287	0.18368	0.29326	0.03170	0.03700	0.19548	0.17063	0.21783
	0.5	0.22353	0.15015	0.15185	0.17316	0.23633	0.03786	0.04482	0.16060	0.14654	0.20017
Celegansneural	0.1	0.02150	0.03744	0.04333	0.03281	0.06414	0.02446	0.01566	0.05646	0.03543	0.04619
	0.2	0.03683	0.08629	0.06311	0.05786	0.09217	0.04658	0.02270	0.08742	0.07084	0.06718
	0.3	0.04855	0.09205	0.07184	0.06768	0.12478	0.06691	0.01799	0.11463	0.08684	0.08561
	0.4	0.05852	0.10629	0.07912	0.07841	0.12075	0.07676	0.01932	0.11733	0.09868	0.09363
	0.5	0.06629	0.10346	0.08983	0.07824	0.14856	0.09399	0.02448	0.11177	0.10091	0.09265
USAir97	0.1	0.04860	0.25220	0.23027	0.24495	0.47234	0.24879	0.23107	0.26201	0.27994	0.28034
	0.2	0.08733	0.33071	0.29243	0.32237	0.52940	0.33698	0.27007	0.35903	0.35156	0.34998
	0.3	0.10142	0.41501	0.33358	0.35382	0.54298	0.39614	0.25781	0.43183	0.38668	0.38211
	0.4	0.12480	0.42509	0.36259	0.37015	0.52161	0.42653	0.24938	0.43933	0.43389	0.39748
	0.5	0.12621	0.41077	0.36376	0.38010	0.51580	0.44250	0.14097	0.43649	0.40187	0.40534
Political blogs	0.1	0.01511	0.07593	0.08643	0.07387	0.07647	0.03174	0.06289	0.09549	0.07116	0.08461
	0.2	0.02548	0.12292	0.12741	0.12870	0.13005	0.06051	0.09832	0.16105	0.13009	0.13729
	0.3	0.03349	0.16521	0.16739	0.15896	0.17264	0.08663	0.10873	0.19258	0.17080	0.17570
	0.4	0.03973	0.19473	0.18835	0.18257	0.19418	0.10892	0.09021	0.22340	0.19913	0.20059
	0.5	0.04425	0.21163	0.19759	0.18839	0.21053	0.13078	0.05294	0.23490	0.21229	0.21488
Amazon	0.1	0.00561	0.01502	0.10607	0.01394	0.01511	0.10786	0.01686	0.07918	0.01706	0.02892
	0.2	0.00647	0.02698	0.10497	0.03202	0.02687	0.11279	0.01362	0.06917	0.03141	0.03879
	0.3	0.00737	0.03197	0.08914	0.03816	0.03570	0.13223	0.02112	0.08882	0.03560	0.04061
	0.4	0.01004	0.04159	0.07666	0.03770	0.03694	0.12843	0.00636	0.07137	0.03824	0.04334
	0.5	0.00816	0.04325	0.06030	0.03605	0.04170	0.13691	0.01446	0.08329	0.04522	0.04671
NetScience	0.1	0.09712	0.19065	0.15592	0.16725	0.68746	0.00418	0.10794	0.15340	0.23593	0.14700
	0.2	0.15230	0.22252	0.20567	0.22471	0.65805	0.00657	0.09821	0.21433	0.28971	0.20071
	0.3	0.18485	0.24195	0.20114	0.23452	0.61250	0.00853	0.06130	0.23000	0.29185	0.22955
	0.4	0.20803	0.22288	0.17363	0.22154	0.53594	0.00817	0.02985	0.20279	0.26881	0.24522
	0.5	0.20411	0.19699	0.15914	0.20398	0.45272	0.01053	0.00960	0.18168	0.22320	0.23934
Power	0.1	0.00240	0.00343	0.00201	0.00131	0.00835	0.00008	0.00018	0.00220	0.00287	0.00303
	0.2	0.00384	0.00456	0.00233	0.00257	0.00460	0.00013	0.00022	0.00255	0.00467	0.00430
	0.3	0.00436	0.00277	0.00277	0.00252	0.00678	0.00020	0.00016	0.00266	0.00377	0.00517
	0.4	0.00433	0.00258	0.00205	0.00268	0.00590	0.00024	0.00022	0.00295	0.00331	0.00562
	0.5	0.00386	0.00148	0.00113	0.00287	0.00615	0.00032	0.00027	0.00195	0.00331	0.00527
GrQc	0.1	0.04860	0.25220	0.23027	0.24495	0.47234	0.24879	0.23107	0.26201	0.27994	0.28034
	0.2	0.08733	0.33071	0.29243	0.32237	0.52940	0.33698	0.27007	0.35903	0.35156	0.34998
	0.3	0.10142	0.41501	0.33358	0.35382	0.54298	0.39614	0.25781	0.43183	0.38668	0.38211
	0.4	0.12480	0.42509	0.36259	0.37015	0.52161	0.42653	0.24938	0.43933	0.43389	0.39748
	0.5	0.12621	0.41077	0.36376	0.38010	0.51580	0.44250	0.14097	0.43649	0.40187	0.40534

6.4.5 Performance Analysis

In this section, we investigate the performance of the proposed method against state-of-the-art algorithms in terms of four accuracy metrics: AUPR, recall, AUC, and precision. In the experiments, we use five different ratios or fraction sets of observed links as the test set $(10, 20, 30, 40, 50\%)$. This is because a sparsification level exceeding 50% may disconnect the network. The results on different datasets for various fractions of observed links against state-of-the-art algorithms are presented in tables 6.3 to 6.6. Each experiment was conducted on eight real-world network datasets.

6.4.5.1 AUPR

Owing to the sparsity of real-world social networks, the number of non-existing links (target links) is far greater than the number of existing links. Some studies suggest that the AUPR metric is more informative than AUC for imbalanced networks. Therefore, AUPR is considered one of the prominent quality metrics. TABLE 6.3 presents the results of AUPR on different datasets for different sets of observed links.

From TABLE 6.3, it can be observed that RA is the best performing method for the Football and Celegansneural datasets, and CLP-ID outperforms all the other state-of-the-art methods on both datasets. For datasets with smaller clustering coefficient, the proposed algorithm is the

best performing method after RA. On the USAir97 dataset, CLP-ID outperforms all the state-of-the-art methods. Similarly, CLP-ID outperforms all the state-of-the-art methods except CCLP and CCLP2 on the Political blogs dataset. The proposed algorithm has comparable AUPR values with those of CCLP and CCLP2 for different ratios (10–50%). For datasets with a larger clustering coefficient ($C \geq 0.8$), CLP-ID has comparable AUPR with that of the state-of-the-art methods. Hence, CLP-ID has comparable AUPR on the Amazon and NetScience datasets.

Similarly, for the remaining datasets, CLP-ID outperforms all the state-of-the-art methods except RA, CCLP2, PA, and NLC. The AUPR of the proposed algorithm is almost the same as that of the RA, CCLP2, PA, and NLC methods. To summarize the accuracy of CLP-ID in terms of AUPR, it can be concluded that the algorithm outperforms all the compared methods except RA and is comparable to CCLP2, PA, and NLC.

6.4.5.2 Recall

TABLE 6.4 presents the results of recall on each dataset for different sets of observed links. It can be seen that the proposed algorithm outperforms almost all (except CN and PA) of the compared methods in terms of recall on each dataset except the Power dataset. CN has the best performance. CLP-ID have comparable recall with that of CN and PA. For the Power dataset, CAR is the best method. After CAR, CLP-ID is the second best

method and outperforms all the state-of-the-art methods. To summarize the accuracy of CLP-ID in terms of recall, it can be concluded that the algorithm outperforms all the other methods on all datasets except Power under different sets of observed links ($10-50\%$). The algorithm has comparable performance with that of CN and PA in terms of recall.

6.4.5.3 AUC

TABLE 6.5 shows the results for AUC on each dataset for different sets of observed links. On the Football dataset, the proposed algorithm outperforms all the other methods. On the Celegansneural, USAir97, and GrQc datasets, the proposed algorithm is comparable with RA and outperforms the other methods. Similarly, on the remaining datasets, CLP-ID outperforms all the state-of-the-art methods. To summarize the accuracy of CLP-ID in terms of AUC, it can be concluded that the algorithm outperforms all the other methods except RA, with which is in fact comparable.

6.4.5.4 Precision

TABLE 6.6 presents the results for precision on each dataset for different sets of observed links. It can be observed that the proposed algorithm outperforms all the other methods except RA on datasets with smaller clustering coefficient (Football, Celegansneural, and USAir97) for each

set of observed links. For the Political blogs dataset, CLP-ID is the best method after NLC. Similarly, for the remaining datasets, CLP-ID has comparable precision values with those of the state-of-the-art methods. To summarize the accuracy of CLP-ID in terms of precision, it can be concluded that the algorithm outperforms all the other methods except RA on all datasets under different sets of observed links (10–50%).

6.4.6 Statistical Test

In this section, we present the results of some statistical tests to analyze the significant differences between the proposed algorithm and the state-of-the-art algorithms in terms of AUPR, Recall, AUC, and Precision. First, we applied Friedman test [159, 160] to analyze whether there are significant differences in link prediction quality metrics. If there are significant differences, then we applied the Friedman Conover test [160] as post hoc procedure to find the degree of rejection of each hypothesis. The post hoc procedure considers CLP-ID as control algorithm for each quality metric. We consider the level of confidence as $\alpha_c = 0.05$ and the degree of freedom as $D_f = 9$ in all the cases.

6.4.6.1 Friedman test: non-parametric analysis

The Friedman test indicates that there is a significant difference in various quality metrics for different ratios, as shown in TABLE 6.7. The null

TABLE 6.7: The Friedman Test on AUPR

Ratio	Dataset	AUPR-value										F_f	State Result
		N2V	CCLP	CCLP2	CN	RA	PA	CAR	NLC	LNBCN	CLP-ID		Is $F_f > \chi^2$?
0.1	Celegansneural	0.02115	0.03701	0.04286	0.03545	0.06273	0.02376	0.02770	0.05565	0.03484	0.04574	53.47	Null
	Football	0.11648	0.09221	0.09114	0.08339	0.20690	0.00630	0.11150	0.11104	0.09576	0.13589		Hypothesis
	Political blogs	0.01505	0.07578	0.08632	0.07601	0.07631	0.03159	0.06614	0.09520	0.07092	0.08456		Rejected
	NetScience	0.09631	0.19348	0.16204	0.18831	0.69874	0.00375	0.18133	0.15769	0.23989	0.14942		
	Amazon	0.00523	0.01431	0.10677	0.01412	0.01472	0.10639	0.04291	0.07819	0.01615	0.02870		
	GrQc	0.04878	0.24390	0.18547	0.22424	0.57664	0.01769	0.28148	0.22989	0.24870	0.22937		
	Power	0.00237	0.00810	0.00911	0.00434	0.01293	0.00008	0.03623	0.00759	0.00763	0.00628		
	USAir97	0.04773	0.24960	0.22894	0.24963	0.47136	0.24797	0.23332	0.25916	0.27705	0.27899		
0.2	Celegansneural	0.03655	0.08605	0.06372	0.06573	0.09210	0.04613	0.06054	0.08742	0.07102	0.06758	54.57	Null
	Football	0.17146	0.15257	0.15288	0.18006	0.32175	0.01324	0.13857	0.16503	0.13978	0.19900		Hypothesis
	Political blogs	0.02544	0.12308	0.12774	0.13383	0.13019	0.06035	0.10921	0.16111	0.13027	0.13753		Rejected
	NetScience	0.15169	0.23835	0.22757	0.26070	0.67891	0.00627	0.24240	0.23014	0.30209	0.21059		
	Amazon	0.00624	0.02659	0.10691	0.03276	0.02669	0.11216	0.05426	0.06901	0.03079	0.03882		
	GrQc	0.08321	0.29686	0.24187	0.29775	0.56168	0.02978	0.37109	0.29909	0.31670	0.30550		
	Power	0.00379	0.01652	0.01700	0.00930	0.01311	0.00013	0.07569	0.01565	0.01691	0.01100		
	USAir97	0.08654	0.32974	0.29227	0.33254	0.52955	0.33712	0.28597	0.35803	0.35068	0.34962		
0.3	Celegansneural	0.04831	0.09358	0.07410	0.07971	0.12581	0.06653	0.06732	0.11603	0.08839	0.08715	55.00	Null
	Football	0.21408	0.18937	0.20176	0.18599	0.27099	0.02006	0.17643	0.17291	0.20937	0.22419		Hypothesis
	Political blogs	0.03346	0.16583	0.16837	0.16738	0.17318	0.08655	0.13508	0.19320	0.17144	0.17641		Rejected
	NetScience	0.18437	0.27084	0.24601	0.28896	0.65239	0.00838	0.25818	0.26225	0.32452	0.25496		
	Amazon	0.00718	0.03196	0.09231	0.04250	0.03586	0.13212	0.07493	0.08971	0.03563	0.04084		
	GrQc	0.11073	0.32951	0.27426	0.33291	0.54803	0.03100	0.42027	0.32445	0.33173	0.32800		
	Power	0.00434	0.01747	0.02686	0.01179	0.02128	0.00020	0.00008	0.02096	0.01988	0.01524		
	USAir97	0.10088	0.41507	0.33532	0.37018	0.54411	0.39694	0.29547	0.43216	0.38686	0.38250		
0.4	Celegansneural	0.05831	0.11038	0.08512	0.09695	0.12481	0.07647	0.07909	0.12128	0.10377	0.09794	55.17	Null
	Football	0.22692	0.17153	0.21998	0.21168	0.31235	0.03061	0.12834	0.21139	0.18535	0.23212		Hypothesis
	Political blogs	0.03970	0.19618	0.19054	0.19578	0.19567	0.10890	0.14121	0.22493	0.20066	0.20218		Rejected
	NetScience	0.20763	0.27631	0.24516	0.29353	0.59811	0.00834	0.25950	0.25939	0.32248	0.29086		
	Amazon	0.00986	0.04226	0.08164	0.04316	0.03772	0.12817	0.04825	0.07327	0.03907	0.04389		
	GrQc	0.12921	0.33441	0.29614	0.35538	0.49461	0.03548	0.45793	0.33845	0.35289	0.32887		
	Power	0.00431	0.02966	0.03433	0.01537	0.02468	0.00024	0.00011	0.03776	0.02913	0.01963		
	USAir97	0.12430	0.42728	0.36680	0.39477	0.52481	0.42803	0.32622	0.44177	0.43565	0.39953		
0.5	Celegansneural	0.06608	0.11224	0.10499	0.10144	0.15844	0.09421	0.12026	0.12282	0.11295	0.10108	54.07	Null
	Football	0.22218	0.18006	0.20831	0.21514	0.27106	0.03683	0.02457	0.19971	0.17658	0.22690		Hypothesis
	Political blogs	0.04422	0.21503	0.20232	0.20809	0.21393	0.13081	0.15659	0.23854	0.21561	0.21826		Rejected
	NetScience	0.20378	0.28501	0.28182	0.29096	0.55626	0.01067	0.24630	0.28844	0.31685	0.31682		
	Amazon	0.00802	0.04502	0.06865	0.04236	0.04318	0.13722	0.10780	0.08703	0.04687	0.04778		
	GrQc	0.13770	0.33935	0.32289	0.35186	0.45409	0.03952	0.43085	0.32148	0.35887	0.33929		
	Power	0.00384	0.02819	0.04014	0.01984	0.03219	0.00032	0.00014	0.04735	0.04581	0.02288		
	USAir97	0.12583	0.41710	0.37339	0.41568	0.52200	0.44449	0.28613	0.44198	0.40725	0.41050		

hypothesis (H_0) states that the methods compared are statistically equivalent, with no significant difference. The Friedman test rejects the hypothesis H_0 if the test statistic value F_f is higher than the $\chi^2(\alpha_c, D_f)$, i.e., $F_f > 16.919$. TABLE 6.7 perform the Friedman test for AUPR and

TABLE 6.8: The Posthoc Friedman Conover Test (Control Method = CLP-ID) Corresponding Different Accuracy Metrics

Metric	Ratio	p-value								
		N2V	CCLP	CCLP2	CN	RA	PA	CAR	NLC	LNBCN
AUPR	0.1	**0.009464**	0.583510	0.695170	0.185576	0.161371	**0.017486**	0.753864	0.814052	0.814052
	0.2	**0.009464**	0.348393	0.481207	0.753864	0.309993	**0.037455**	0.531079	0.531079	0.814052
	0.3	**0.025805**	0.875387	0.753864	0.753864	0.074943	0.088158	0.242064	0.531079	0.695170
	0.4	**0.014306**	0.753864	0.481207	0.695170	0.348393	0.053445	0.139683	0.434061	0.875387
	0.5	**0.017486**	0.695170	0.481207	0.531079	0.212434	0.103241	0.274562	0.481207	0.814052
Recall	0.1	**0.037238**	0.139223	**0.001544**	0.813837	0.411000	**0.007585**	**0.000347**	**0.033974**	0.160883
	0.2	**0.044756**	0.139529	**0.000748**	0.813981	0.368521	**0.021237**	**0.000208**	**0.017443**	0.257764
	0.3	0.103108	0.161208	**0.001223**	0.875338	0.638158	0.129587	**0.006146**	**0.034110**	0.389577
	0.4	**0.088158**	**0.044838**	**0.000352**	0.481207	0.309993	0.161371	**0.017486**	**0.004948**	0.212434
	0.5	0.139683	**0.037455**	**0.000209**	0.434061	0.481207	0.583510	0.185576	**0.003153**	0.088158
AUC	0.1	0.120360	**0.063427**	**0.006165**	0.212434	0.481207	**0.000041**	**0.000024**	0.274562	**0.037455**
	0.2	0.185576	0.120360	**0.003957**	0.274562	0.695170	**0.000094**	**0.000031**	0.161371	0.161371
	0.3	0.753864	0.531079	**0.017486**	0.638290	0.531079	**0.007653**	**0.000455**	0.481207	0.531079
	0.4	0.274562	0.242064	**0.004948**	0.348393	0.753864	**0.002503**	**0.000054**	0.185576	0.212434
	0.5	0.185576	**0.044838**	**0.000585**	0.185576	0.583510	**0.003153**	**0.000018**	**0.025805**	0.120360
Precision	0.1	**0.004948**	0.481207	0.212434	**0.031155**	0.389762	**0.006165**	**0.006165**	0.814052	0.638290
	0.2	**0.009464**	0.434061	0.139683	0.389762	0.348393	**0.014306**	**0.002503**	0.875387	0.875387
	0.3	**0.014306**	0.481207	0.242064	0.103241	0.242064	**0.017486**	**0.001562**	0.814052	0.875387
	0.4	**0.031155**	0.434061	0.120360	0.103241	0.389762	**0.021286**	**0.000962**	0.814052	0.937497
	0.5	**0.025805**	0.274562	**0.044838**	0.088158	0.481207	**0.037455**	**0.000352**	0.695170	0.531079

indicates that the null hypothesis is rejected. Similarly, the null hypothesis is rejected for remaining metrics Recall, AUC, and Precision on distinct ratio. Therefore, the Friedman Conover test is applied to measure the actual differences between the algorithms.

6.4.6.2 Friedman Conover test: Post hoc analysis

The Friedman Conover procedure rejects hypothesis H_1 to H_{i-1} if the p-value is greater than the adjusted level of confidence value, i.e., $p_i > \alpha_c/((D_f+1)-i)$, where i is the smallest integer. We consider dependent test statistics to perform Friedman Conover post hoc procedure. Let $p_1, p_2, \ldots, p_{D_f}$ are the ordered p-values (smallest to largest) and

$H_1, H_2, \ldots, H_{D_f}$ are the corresponding hypothesis. The Friedman Conover procedure starts with most significant p-value. TABLE 6.8 shows the p-value results for AUPR, Recall, AUC, and Precision metrics. In the tables, the highlighted value indicate the rejected hypothesis, respectively.

The statistical tests on different accuracy metrics (AUPR, Recall, AUC, and Precision) demonstrate that the proposed algorithm is significantly different from the state-of-the-art algorithms. From TABLE 6.8, we can observe the level of significant differences between our proposed algorithm and other standard algorithms. The highlighted value shows the significant difference ($\leq$ 0.05) between CLP-ID and remaining algorithms. In TABLE 6.8. the combined ratio indicates that the statistical test is performed simultaneously for different sets of observed links.

6.5 Conclusion

In this chapter, a community-based link prediction algorithm (CLP-ID) is presented, which uses an information diffusion perspective to predict target links. Classical node-based similarity methods for link prediction primarily focus on the prediction accuracy rate. By contrast, CLP-ID considers both the accuracy rate and the information spread in the social network. Comparative analysis is performed on various real-world networks. The analysis revealed following facts.

- Incorporation of information diffusion in community detection accounts influences between nodes, rather than only the connection between nodes, such that the influence degree of nodes within a community can be as close as that in the whole network.

- Incorporation of community structure in link prediction accounts positive influence for intra-community future links and negative influence for inter-community links.

- The empirical results show that the proposed algorithm performs better than compared methods in terms of AUPR metric for datasets with higher clustering coefficient value. It is also observe that the proposed algorithm outperforms almost all (except CN and PA) of the compared methods in terms of Recall on each dataset except Power dataset. CLP-ID outperforms compared methods except RA in terms of AUC. It also performs better than compared methods except RA in terms of Precision on datasets with lower clustering coefficient values.

- The statistical tests are performed to analyze the significant differences in AUPR, Recall, AUC, and Precision between the proposed algorithm and compared algorithms. The post hoc analysis states the significant difference between CLP-ID and state-of-the-art algorithms.