Chapter 3

Tree based Community Detection

3.1 Introduction

The continuous generation of data results in an evolving dynamic network. Nowadays most of the networks are dynamic which implies they do not have a fixed topology. Traditionally, a network is represented as a graph with nodes and their in-between links. But this definition of network needs some modification to incorporate the other important dimension, time. The temporal dimension facilitates improvised understanding of network by embedding valuable information to it. The work presented here deals with this modified network definition.

An evolving network continues to introduce more aspects requiring their consideration even while the process of deducing information from its former version is in progress. Data is ever growing and changing. It is not practical for these networks to be stored at one place or to be processed as a whole in one run. An efficient methodology always needs data to be sampled into small sizes. The available literature suggests two broad approaches for handling such evolving data. They are the temporal network approach and the snapshot network approach. The community structure of a network is updated on every change at next time unit in temporal network settings. On the contrary, in a snapshot setting the complete network graph present at current time is treated as a static network and the consecutive community structure is mapped to determine the progress. Temporal setting has been used in the paper.

There are some studies which discusses the surprising powers of social networks like connectedness and influence along with how they shape our lives [31]. With evolving dynamic network, we utilize both the properties connectedness and influence to identify the community structure. The small-world phenomena or six degrees of separation is based on the connectedness property and states that two unlikely connected peoples may be connected by six or fewer mutual acquaintances. Similarly, the three degree theory [31] states that influence of a people is limited to its local region up to three hope count. Therefore, connectedness and influence can be useful to identify closely related groups in the network. For this, we used a tree-like structure to accommodate both three degree theory and small-world phenomena.

This chapter address the community detection problem using connectedness and influence perspective. With these properties, we present a tree-based community detection algorithm on dynamic social network (TCD2). The main contributions of the work are summarized as follows:

- We utilized network level properties connectedness and influence to identify closely related clusters in the social network on dynamic setting. To accommodate these properties, the proposed algorithm utilize a tree-like structure. Also, an exemplary graph is discussed to show the working of TCD2 along with complexity analysis.
- The experiments are performed on both real-world and synthetic social networks to validate the proposed algorithm with several well-known evaluation metrics. Also, the comparative analysis against four state-of-the-art static and dynamic algorithms. The performance of TCD2 is analyzed under both static and dynamic settings. We also show the impact of level counts, window size, and mixing parameter on performance measures.
- The experiment results demonstrate that TCD2 is more adept to find true community groups compared to state-of-the-art algorithms. Moreover, the proposed algorithm is flexible in managing community quality and accuracy with no prior requirement of community set cardinality. The results of TCD2 conclude that it provides a trade-off between quality and accuracy.

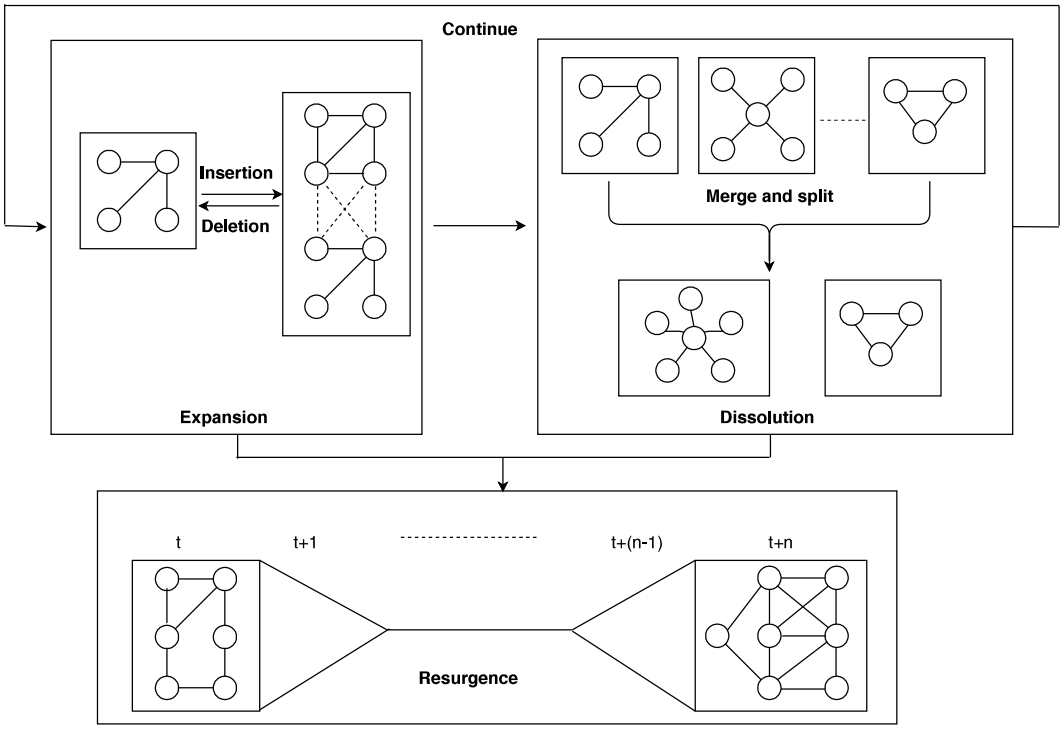


FIGURE 3.1: The proposed framework TCD2

3.2 Proposed Work

In this section, we discuss the proposed framework, as illustrated in Figure 3.1. The proposed algorithm works in three phases: initialization, expansion and dissolution. In initialization phase, the parameters and experimental setting are initialized. In expansion phase, network is expanded in dynamic setting and possible communities are identified. In dissolution phase, the identified communities are re-examined after a time window. This process continues until no network change is detected. These phases are discussed in more detail below.

1. **Initialization.** In this phase, the number of levels of tree-structure is assigned based on three degree theory and six-degree of separation. Also, the size of window is initialized for re-examining the tree-structure in dissolution phase. The cluster head of community structure is initially empty.

2. **Expansion.** Basically two sub-tasks are performed in this phase. First is to identify new community and second is to expand existing community. We have utilized

a tree-based structure to identify community structure of dynamic social network. Initially, the node set is empty at time-stamp $t = 0$, so as cluster head is also empty. Whenever a new edge (x, y) is identified in the network, it is inserted in the tree structure based on four rules presented in Table 3.1. The rule $R1$ states that when both the node x and y are newly created nodes, then make node x as cluster head and y as intermediate node. And add x and y at level 0 and 1 respectively. The rules $R2$ and $R3$ state that when one node y is newly created node and second node x is older node, then make y as child of x iff x is not a leaf node. If x is a leaf node then delete x from leaf and add to level 0 as cluster head with y as a child. The rule $R4$ states that if both the nodes are older nodes then only updates the intra-connection and inter-connection index of both the nodes. Level counts are defined based on three degree theory and small-world phenomena.

3. **Dissolution.** Identified communities from expansion phase are re-examined after a time-stamp window W again in this phase. The communities which are not detachable from rest of the network are merged with most suitable and stable communities. These unstable communities can be identified using detachability index [150]. The detachability index $D(C)$ of any community C is defined based on the context of connectedness, which is given as follows.

$$D(C) \leftarrow \frac{Intra - connections}{Intra - connections + Inter - connections} \tag{3.1}$$

If any identified community C in expansion phase have detachability index less than threshold θ, then C is considered as unstable community.

TABLE 3.1: List of rules during INSERTION operations in expansion phase of TCD2 framework, when a edge (x, y) is identified in dynamic setting (0: indicates new node is identified in the network; 1: indicates node is already presents on the network)

Rule	x	y	Operation: INSERTION	Remarks
R1	0	0	Level 0: New cluster formation with x as cluster head Level 1: Make y as child of x along with Intra-connection index updation	Both the nodes are newly created nodes
R2	0	1	Level Intermediate: Make y as child of x with Intra-connection index	Only one node is newly created and other is older one
R3	1	0	Level Leaf: New cluster formation with x as cluster head	
R4	1	1	Updation of Intra-connections and Inter-connection index	Both the nodes are older nodes

Algorithm 1: TCD2: Tree-based Community Detection in Dynamic social networks

Input: Dynamic Graph: $G(V,E)$
Output: $Cluster.Heads.List$: Community Structure

1 $W_{temp} \leftarrow 0$ ▷ Initialization
2 $W \leftarrow$ Initialize window size for dissolution phase
3 $l \leftarrow$ Initialize the number of levels for tree structure
4 $\theta \leftarrow$ Initialize the stability threshold
5 $Cluster.Heads.List \leftarrow \phi$
6 **foreach** *new edge (x,y)* **do**
7 $Cluster.Heads.List \leftarrow Cluster.Heads.List \cup$ INSERTION(x,y) ▷ Expansion
8 $W_{temp} \leftarrow W_{temp} + 1$ ▷ Dissolution
9 **if** $(Is.Equal(W, W_{temp}))$ **then**
10 **for** *each* $C_z \in Cluster.Heads.List$ **do**
11 $\theta_z \leftarrow$ Calculate detachability $D(C_z)$ of community C_z
12 **if** $\theta_z < \theta$ **then**
13 $N_E \leftarrow$ Find each node $w \in N(C_z) \cap (V \setminus C_z)$
14 $T_c \leftarrow$ Find set of number of times each $w \in N_E$ appeares in $w \in N(C_z) \cap (V \setminus C_z)$
15 $C_{max} \leftarrow argmax_{w \in N_E}(T_c(w))$
16 $N_S(C_{max}) \leftarrow$ Set of all nodes those have C_{max}
17 $CS \leftarrow$ Set of all communities those have node $w \in N_S(C_{max})$
18 $M_{ID} \leftarrow -\infty$
19 **for** *each* $C_i \in CS \setminus C_z$ **do**
20 $T_{ID} \leftarrow D(C_z \cup C_i) - D(C_i)$
21 **if** $T_{ID} > M_{ID}$ **then**
22 $M_{ID} \leftarrow T_{ID}$
23 $t \leftarrow i$
24 $C_{merge} \leftarrow$ Merge commuinties of cluster head C_Z and C_t
25 Update cluster head of merge community nodes
26 Update C_{Inter} and C_{Intra} value of each node of merge communites
27 $Cluster.Heads.List \leftarrow Cluster.Heads.List \setminus C_Z$
28 $W_{temp} \leftarrow 0$
29 **Return** $Cluster.Heads.List$

3.2.1 Algorithm

Algorithm 1 takes an input, dynamic graph G and identifies the community structure based on connectedness and influence. The algorithm works in three phases: initialization (lines 1-5), expansion (line 7), and dissolution (lines 8–28). Lines 1 and 2 initialize the

Algorithm 2: INSERTION

Input: New Edge: (x,y), Dynamic Graph: $G(V,E)$
Output: $Cluster.Heads.List$: Community Structure

```
if (!Is.Selected.x) ∧ (!Is.Selected.y) then
    Cluster.Heads.List ← Cluster.Heads.List ∪ x
    C_H.x ← x
    level.x ← 0
    level.y ← 1
    C_H.y ← x
    C_Intra.y ← 1
    C_Intra.x ← 1
    Is.Selected.x ← 1
    Is.Selected.y ← 1
else if (!Is.Selected.x) ∨ (!Is.Selected.y) then
    if (!Is.Selected.x) then
        Interchange(x, y)
    if (level.x ≤ l − 1) then
        level.y ← level.x + 1
        C_Intra.y ← 1
        C_Intra.x ← C_Intra.x + 1
        C_H.y ← x
        Is.Selected.y ← 1
    else
        Cluster.Heads.List ← Cluster.Heads.List ∪ x
        C_H.x ← x
        level.x ← 0
        level.y ← 1
        C_H.y ← x
        C_Inter.x ← C_Intra.x
        C_Intra.y ← 1
        C_Intra.x ← 1
        Is.Selected.y ← 1
else
    if (Is.Same(C_H.x, C_H.y)) then
        C_Intra.y ← C_Intra.y + 1
        C_Intra.x ← C_Intra.x + 1
    else
        C_Inter.y ← C_Inter.y + 1
        C_Inter.x ← C_Inter.x + 1
Return Cluster.Heads.List
```

temporary W_{temp} and actual window size W with zero and an integer respectively. Line 3 assigns the maximum possible level in proposed tree-structure. Line 4 assigns the stability threshold θ for re-examining community structure. Line 5 initializes cluster head with an empty set. The **foreach** loop in lines 6-28 identifies the community structure of given dynamic network iteratively. Line 7 insert each newly identified edge into proposed tree-structure using **Algorithm 2**. Line 8 increases the temporary window W_{temp} by one. Line 9 checks whether W and W_{temp} are equal. If it is, then **for** loop in lines 10-27 re-examines the community structure by merging unstable communities with most suitable and stable communities. Line 28 again resets temporary window with zero. Finally, line 29 returns identified communities.

Algorithm 2 takes two inputs: a new identified edge (x,y) and an evolving network G at current time-stamp. Line 1 checks whether both the nodes x and y are newly created nodes. If it is, then it adds x to cluster head and make y child of x along with updating the level, inter-connections, inter-connections, and selection flag in lines 2-10. If it is not, then line 11 checks whether any one of the node y is newly created node. if it is, then node y is added to tree-structure as child of x and updates the level, inter-connections, inter-connections, and selection flag in lines 12-29. Otherwise both the nodes are newly created and added to suitable community in tree-structure as lines 30-36 accordingly. Finally, line 37 returns updated cluster head.

3.2.2 Applying the Algorithm

In order to explain the working of proposed algorithm, we consider a dynamic network at different time-stamps t ranges over 0-4 under three degree theory (level = 3). Table 3.2 shows the updation of tree-structure and different values like cluster head, parent and child nodes, number of intra and inter connection, etc., based on expansion phase. For example, at time-stamp $t = 0$, tree is empty and two new edges (A,B) and (A,C) are identified. Therefore, node A is considered as a cluster head and added to $Cluster.Heads.List$. Then both the edges (A,B) and (A,C) are added in tree-structure based on $R1$ and $R2$. Similarly, for time-stamp $t = 1-4$, the tree-structure and updation are shown in Figure 3.2 and Table 3.2 respectively. After a window size W (fixed number of time-stamps), TCD2 enters in dissolution phase and merge unstable communities with suitable and stable communities.

TABLE 3.2: The updation of tree-structure using Expansion Phase on exemplary graph

Time t	Edges (x,y)	Cluster Head Assignment		Intra-connection Index		Inter-connection Index		Level		Is.Selected		Cluster Head List
		$C_H.x$	$C_H.y$	$C_{Intera}.x$	$C_{Intera}.y$	$C_{Inter}.x$	$C_{Inter}.y$	$level.x$	$level.y$	x	y	
0	(A,B)	A	A	1	1	0	0	0	1	1	1	{A}
	(A,C)	A	A	2	1	0	0	0	1	1	1	{A}
1	(A,E)	A	A	3	1	0	0	0	1	1	1	{A}
	(B,D)	A	B	2	1	0	0	1	2	1	1	{A}
	(G,H)	G	G	1	1	0	0	0	1	1	1	{A,G}
2	(A,D)	A	B	4	2	0	0	0	2	1	1	{A,G}
	(D,L)	B	D	3	1	0	0	2	3	1	1	{A,G}
	(I,E)	E	A	1	2	0	0	2	1	1	1	{A,G}
	(C,M)	A	C	2	1	0	0	1	2	1	1	{A,G}
	(C,F)	A	C	3	1	0	0	1	2	1	1	{A,G}
3	(B,Q)	A	B	3	0	0	1	1	2	1	1	{A,G}
	(L,N)	L	L	1	1	1	0	0	1	1	1	{A,G,L}
	(H,I)	G	E	1	1	1	1	1	2	1	1	{A,G,L}
	(M,N)	C	L	1	1	1	1	2	1	1	1	{A,G,L}
	(L,K)	L	L	2	1	1	0	0	1	1	1	{A,G,L}
	(Q,J)	B	G	2	1	0	0	2	3	1	1	{A,G,L}
4	(H,N)	G	L	2	2	2	2	1	1	1	1	{A,G,L}
	(F,K)	C	L	2	2	1	1	2	1	1	1	{A,G,L}
	(H,T)	G	H	3	1	2	0	1	2	1	1	{A,G,L}
	(G,O)	G	G	2	1	0	0	0	1	1	1	{A,G,L}
	(N,P)	L	N	3	1	2	0	1	2	1	1	{A,G,L}

The algorithm again perform expansion after dissolution is over. This process is continue until no new nodes are identified.

3.2.3 Complexity Analysis

Now, we analyze the time complexity of the proposed **Algorithm 1**. Lines 1-5 perform initialization and take $O(1)$ time. The **for** loop in lines 6-28 runs $O(E)$ times and enter in both phases expansion and dissolution. The expansion phase in line 7 adds edges in tree-structure using **Algorithm 2** in $O(1)$ time. The dissolution phase in lines 8-28 merges unstable communities with most suitable and stable communities in $O((\frac{|E|}{W})^2 C_{avg} D_{avg})$ time, where C_{avg} and D_{avg} denote average community size and avg degree of network. Therefore, overall complexity of TCD2 is $O(EC_{avg}D_{avg})$.

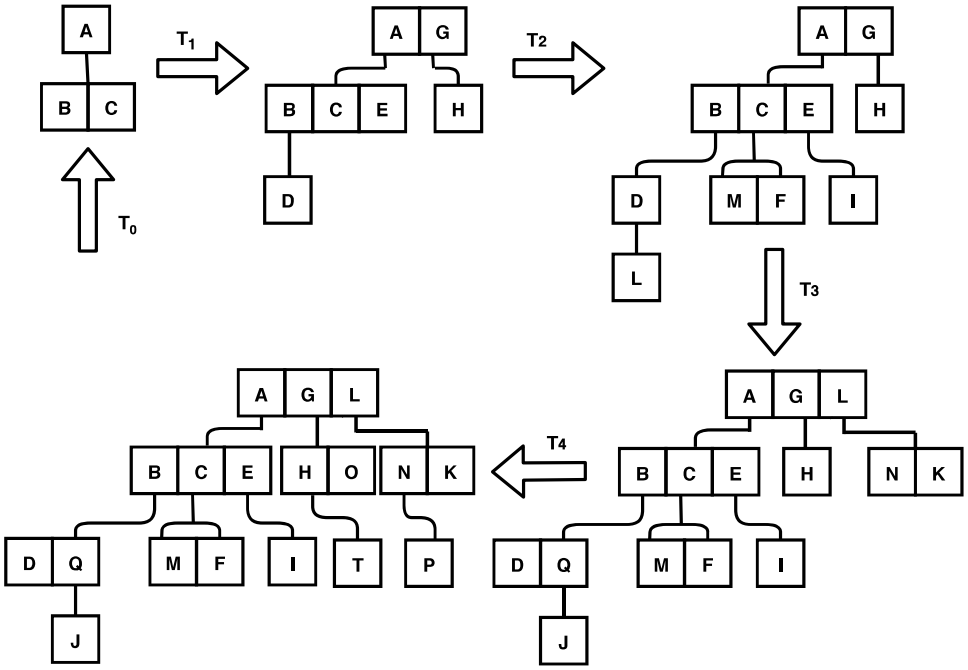


FIGURE 3.2: The identification of communities on example graph based on expansion phase of TCD2

3.3 Evaluation Setup

3.3.1 Performance Metrics

The field of community detection algorithms have a rich literature. Various novel metrics have been proposed for evaluation of algorithm performance. We have carefully selected a set of accuracy and quality metrics whose short description has been presented in the section 2.4.

3.3.2 Datasets

The performance of the algorithm is evaluated on a wide range of datasets. They have been briefly introduced in this section. Selections have been made on the basis of important characteristics to ensure better evaluation of the algorithm on a diverse ground. The datasets can be classified based of following aspects:

- Real world vs Synthetic datasets
- *Ground truth* : Known vs Unknown community structure
- *Size* : Small vs Large

3.3.2.1 Known Community Structures

Real World Datasets Certain widely used open datasets have been used for the evaluation of the community detection algorithm. An overview of these has been presented here.

Karate : [177] studied a university's karate club and constructed this social network dataset. It is originally collected for developing a model of an anthropological problem. The club was a group of 34 members. The members and their interactions are monitored for a period of 3 years. The club reflects group fission because of conflicts between the administrator and the instructor. We have used the [48] version of Karate dataset.

Strike : Management of a sawmill proposes a few changes in policies which was not being accepted by its labourers/workers and they initiated a strike. The pattern of communication among the labourers regarding the new policy has been analyzed by [111] for detection and study of issues in the management-worker communication.

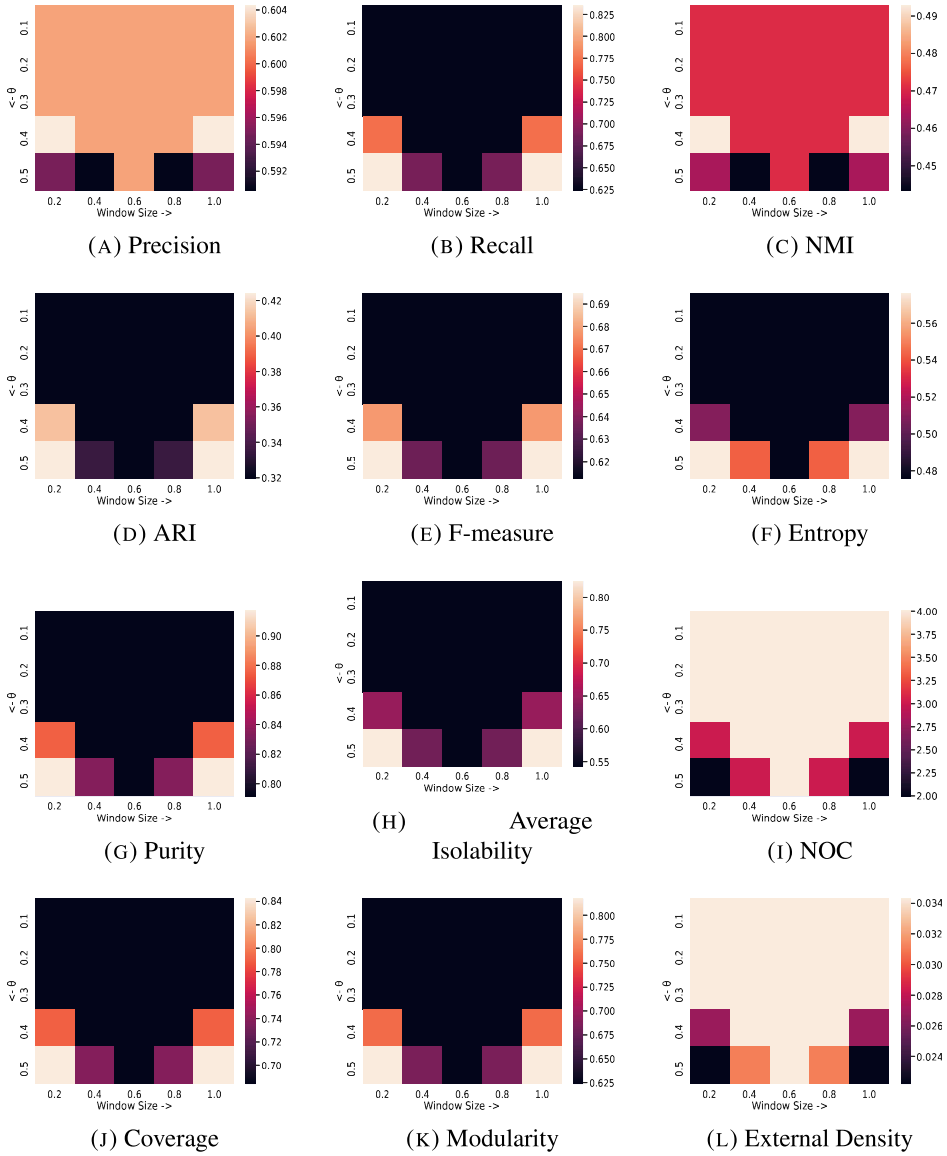
Football : It represents the American football games played between Division IA colleges. The graph consists of vertices which represent the college teams and edges which represent the season matches played between the participating teams. In a given season, all of these teams are divided into conferences with 8-12 teams per conference. Teams in the same conference are more likely to be competed than with other conferences.

Dolphin : Lusseau et al. [99] have compiled this dataset based on the 62 bottlenose dolphins which were residing at Doubtful Sound in New Zealand. Their behavior has been studied during the period ranging from 1994 to 2001.

Synthetic Datasets Lancichinetti-Fortunato-Radicchi benchmark (LFR) [81] is used for generating synthetic graphs that resemble the real world graphs in characteristics. The benchmark follows power law distribution for the node degrees as well as community sizes.

TABLE 3.3: Safe range of window size and θ for quality and accuracy metrics corresponding to datasets

.	*Karate*	*Strike*	*Football*	*Dolphin*	$GR-QC$	$HEP-TH$	$Wiki-Voter$
Precision	$[(0.2,1.0),(0.4,0.1)]$, $[(0.4,0.8),0.5]$	$[0.2,0.4]$, $[1.0,0.4]$	$[(0.2,1.0),0.1]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$--$	$--$	$--$
Recall	$[0.2,0.4]$, $[1.0,0.5]$	$[0.2,0.4]$, $[1.0,0.5]$	$[o.4,0.5]$, $[0.8,0.5]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$--$	$--$	$--$
NMI	$[(0.2,1.0),(0.4,0.1)]$, $[(0.4,0.8),0.5]$	$[0.2,0.4]$, $[1.0,0.4]$	$[(0.4,0.8),1.0]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$--$	$--$	$--$
ARI	$[(0.2,1.0),(0.4,0.1)]$, $[(0.4,0.8),0.5]$	$[0.2,0.4]$, $[1.0,0.5]$	$[(0.2,1.0),0.1]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$--$	$--$	$--$
$F-measure$	$[0.2,0.4]$, $[1.0,0.5]$	$[0.2,0.4]$, $[1.0,0.5]$	$[(0.2,1.0),0.1]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$--$	$--$	$--$
Entropy	$[(0.2,1.0),(0.4,0.1)]$, $[(0.4,0.8),0.5]$	$[(0.2,1.0),(0.3,0.1)]$, $[(0.4,0.8),0.4]$, $[0.6,0.5]$	$[(0.2,1.0),0.1]$	$[1.0,(0.4,0.5)]$	$--$	$--$	$--$
Purity	$[0.2,0.4]$, $[1.0,0.5]$	$[0.2,0.4]$, $[1.0,0.5]$	$[o.4,0.5]$, $[0.8,0.5]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$--$	$--$	$--$
AvgIsolability	$[(0.2,1.0),(0.4,0.1)]$, $[(0.4,0.8),0.5]$	$[0.2,0.5]$, $[1.0,0.5]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$[0.2,0.5]$, $[0.8,0.5]$	$[1.0,(0.5,0.4)]$	$[0.6,0.5]$, $[0.8,0.4]$
Coverage	$[0.2,0.4]$, $[1.0,0.5]$	$[0.2,0.4]$, $[1.0,0.5]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$[0.2,0.5]$, $[1.0,(0.5,0.4)]$	$[(0.2,0.4),0.5]$, $[1.0,(0.5,0.4)]$
Modularity	$[0.2,0.4]$, $[1.0,0.5]$	$[0.2,0.4]$, $[1.0,0.5]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$[0.2,0.5]$, $[1.0,(0.5,0.4)]$	$[(0.2,0.4),0.5]$, $[1.0,(0.5,0.4)]$
ExternalDensity	$[0.2,0.4]$, $[1.0,0.5]$	$[0.2,0.4]$, $[1.0,0.5]$	$[0.2,0.5]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$[(0.2,0.4),0.5]$, $[0.8,0.5]$	$[0.2,0.5]$, $[1.0,(0.5,0.4)]$	$[(0.2,0.4),0.5]$, $[1.0,(0.5,0.4)]$



(A) Precision (B) Recall (C) NMI

(D) ARI (E) F-measure (F) Entropy

(G) Purity (H) Average Isolability (I) NOC

(J) Coverage (K) Modularity (L) External Density

FIGURE 3.3: Safe zone predicted with different metric values corresponding to w and θ ranges over $0.2 - 1.0$ and $0.1 - 0.5$ respectively based on three-degree theory (level=3) in strike dataset

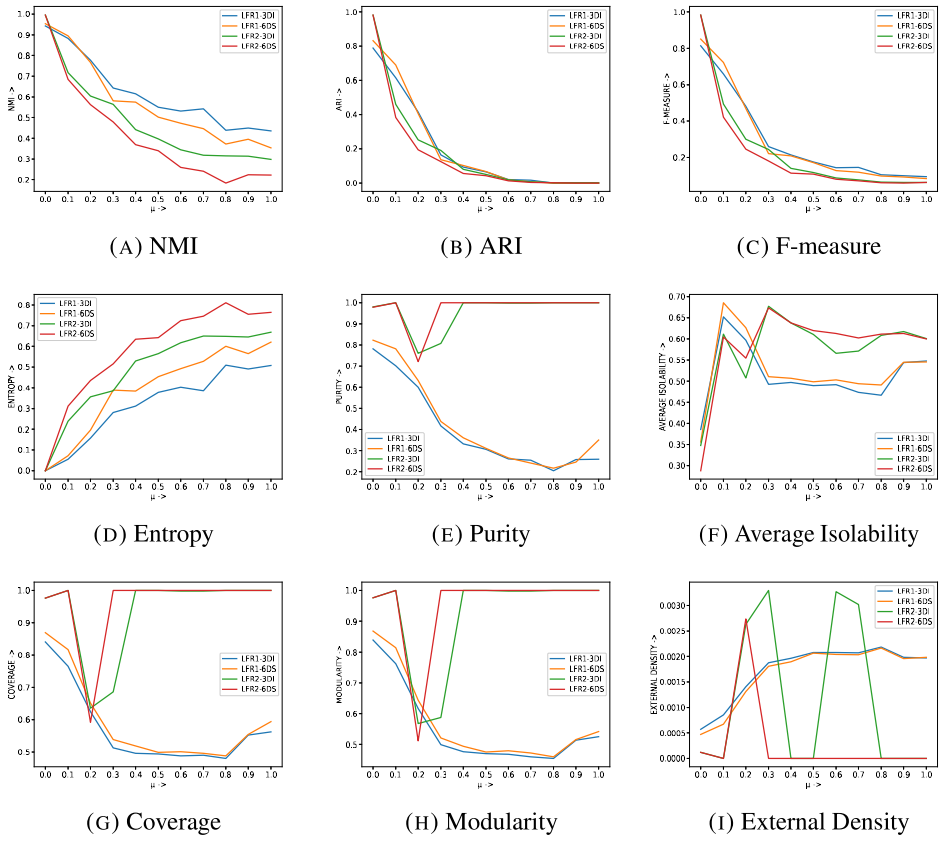
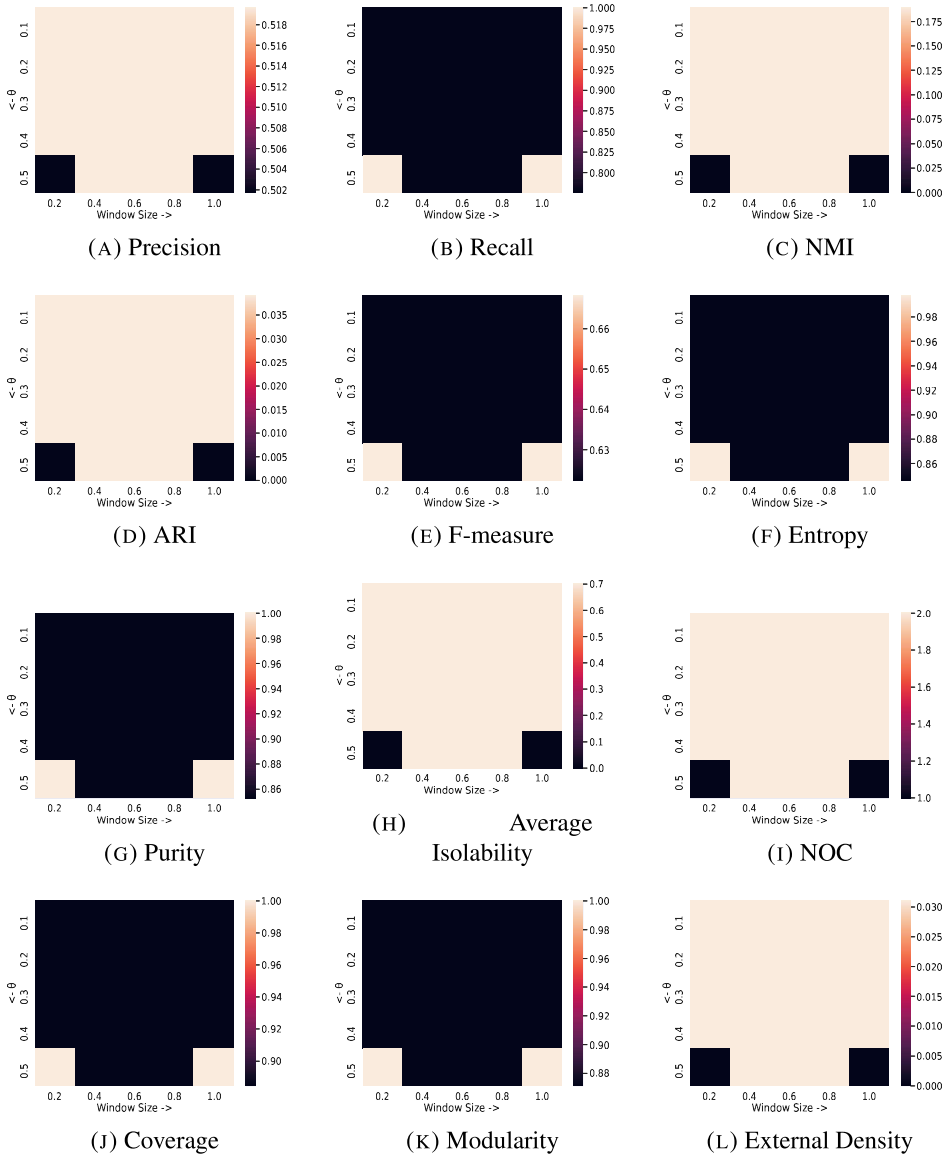


FIGURE 3.4: The comparison of different metric values corresponding to mixing parameter μ based on three degree theory (level=3) and six degree of separation (level=6) in synthetic dataset

Various graphs comprising of 500 and 1000 nodes are being generated for different values of μ in range $[0, 1]$.

3.3.2.2 Unknown Community Structures

The presented algorithm has been tested for 3 large datasets whose community structures are not known. These networks help in ascertaining how well the proposed algorithm is responding to the qualitative features of the underlying community structures. Short descriptions of the datasets have been listed below:



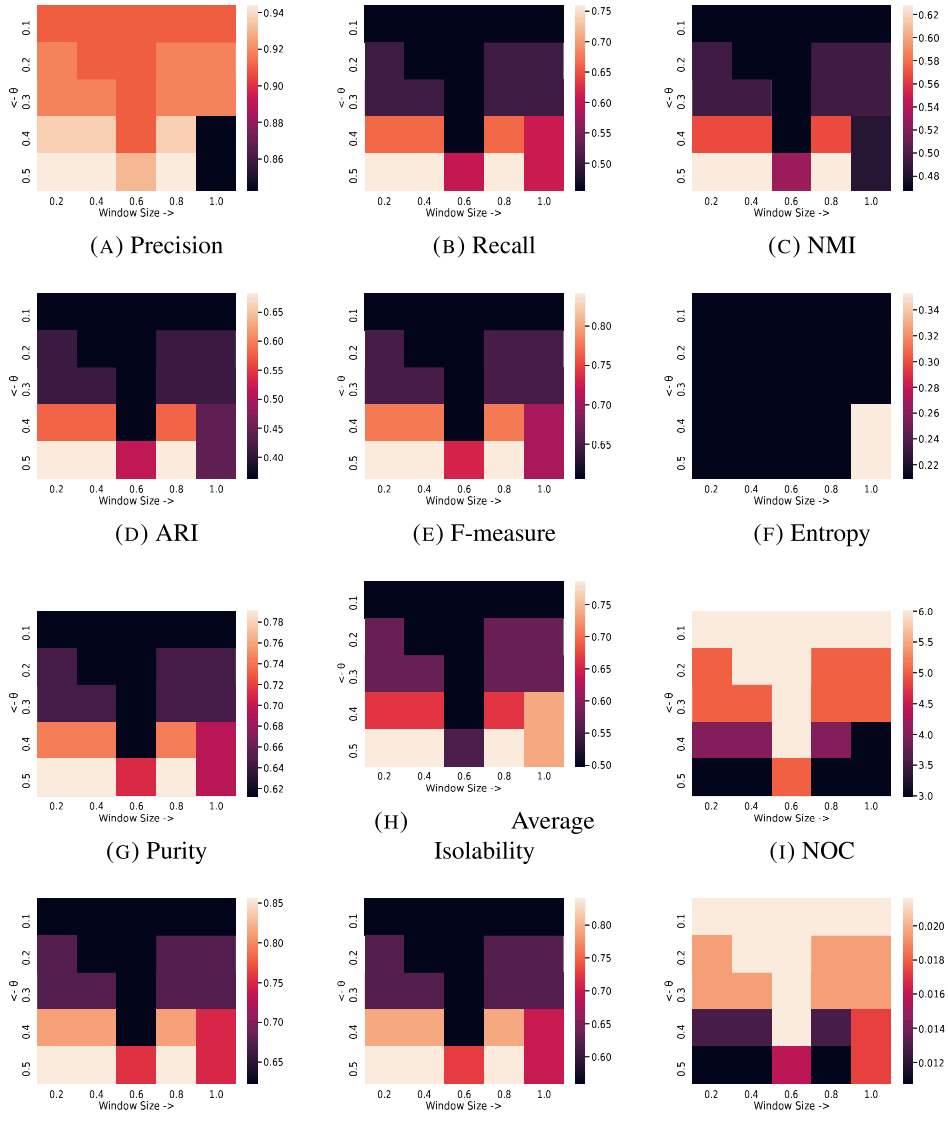
(A) Precision (B) Recall (C) NMI

(D) ARI (E) F-measure (F) Entropy

(G) Purity (H) Average Isolability (I) NOC

(J) Coverage (K) Modularity (L) External Density

FIGURE 3.5: Safe zone predicted with different metric values corresponding to w and θ ranges over $0.2 - 1.0$ and $0.1 - 0.5$ respectively based on three degree theory (level=3) in karate dataset



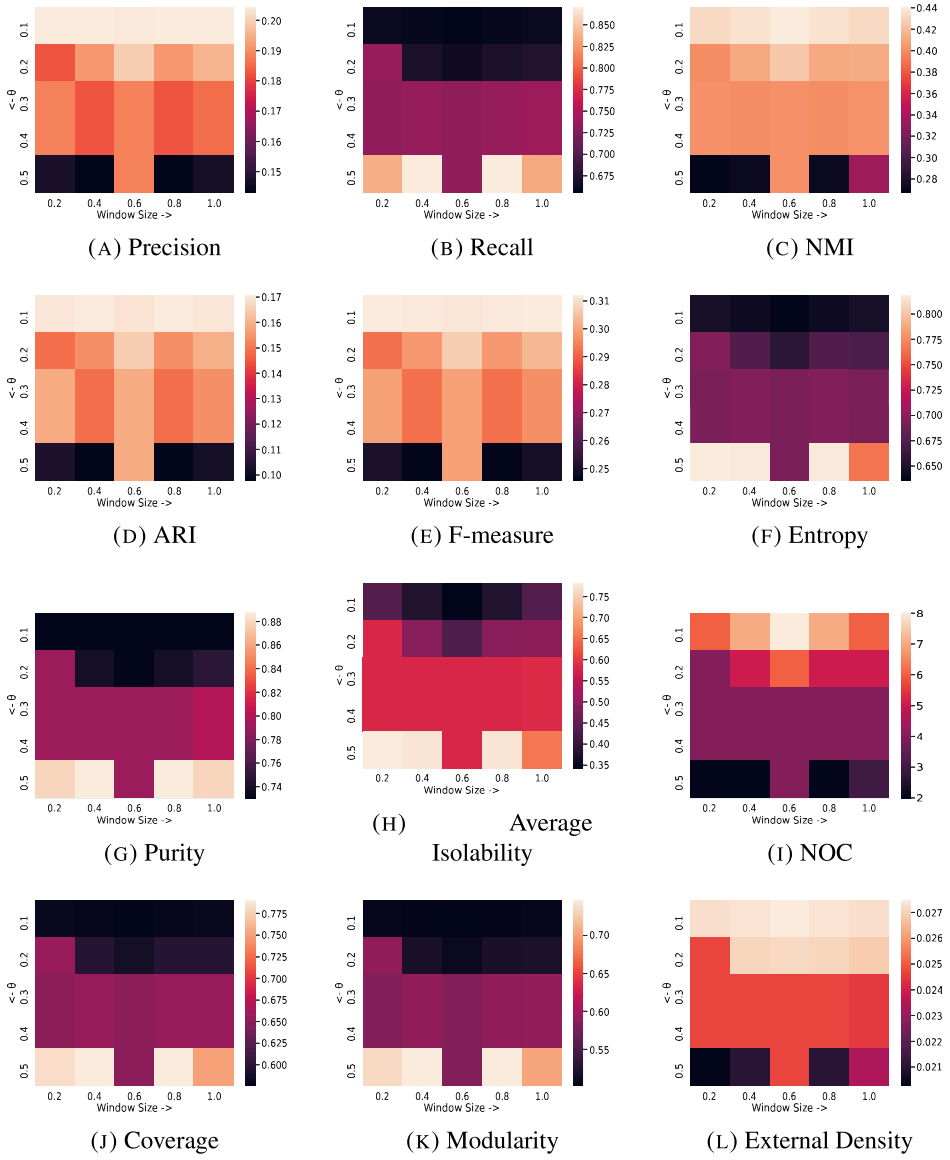
(A) Precision (B) Recall (C) NMI

(D) ARI (E) F-measure (F) Entropy

(G) Purity (H) Average Isolability (I) NOC

(J) Coverage (K) Modularity (L) External Density

FIGURE 3.6: Safe zone predicted with different metric values corresponding to w and θ ranges over $0.2 - 1.0$ and $0.1 - 0.5$ respectively based on three degree theory (level=3) in dolphin dataset



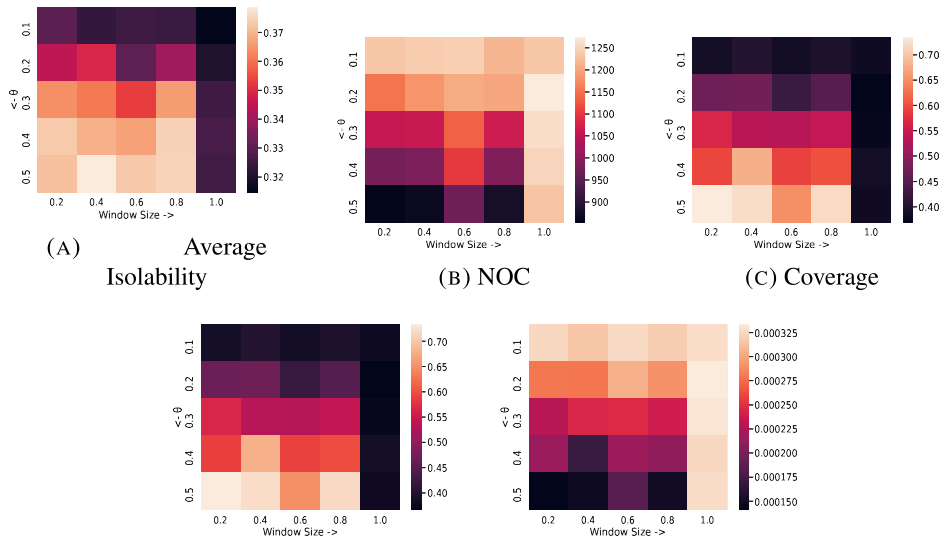
(A) Precision (B) Recall (C) NMI

(D) ARI (E) F-measure (F) Entropy

(G) Purity (H) Average Isolability (I) NOC

(J) Coverage (K) Modularity (L) External Density

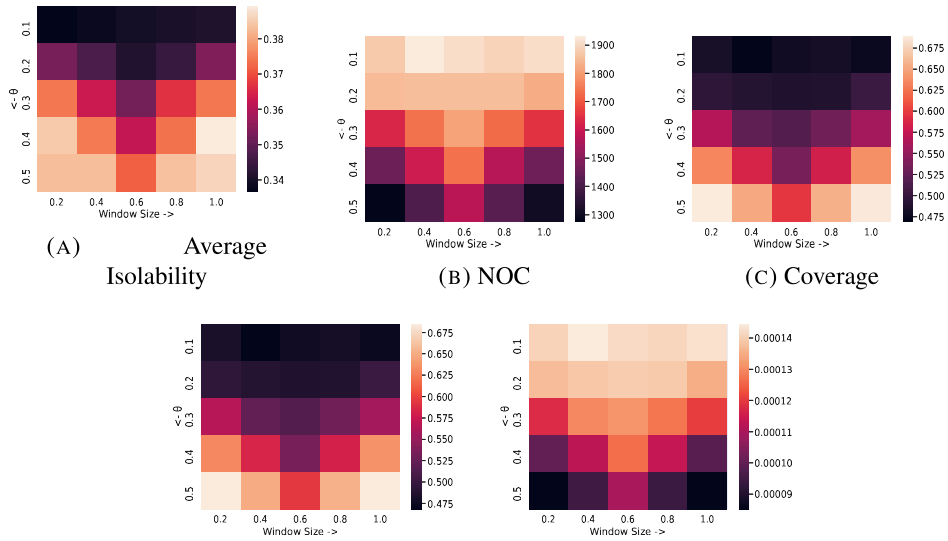
FIGURE 3.7: Safe zone predicted with different metric values corresponding to w and θ ranges over $0.2 - 1.0$ and $0.1 - 0.5$ respectively based on three degree theory (level=3) in football dataset



(A) Average Isolability (B) NOC (C) Coverage

(D) Modularity (E) External Density

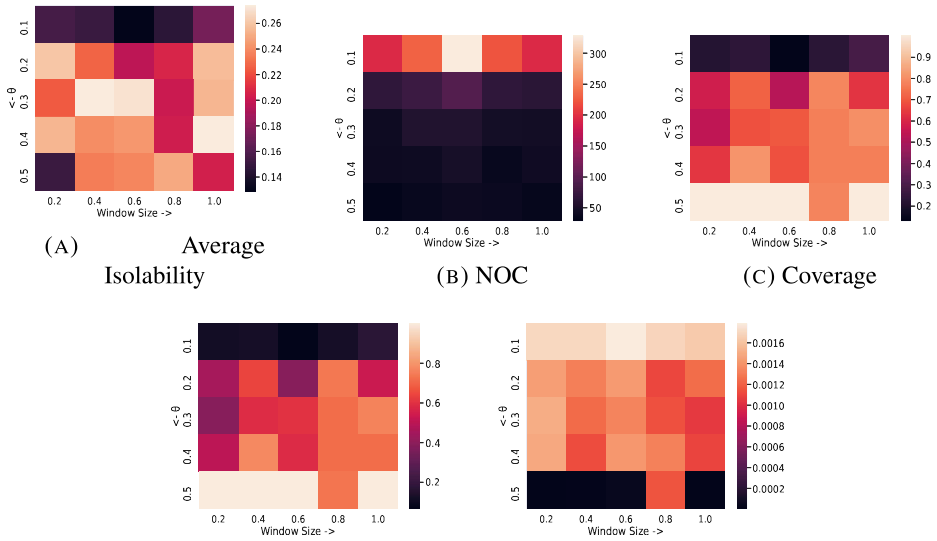
FIGURE 3.8: Safe zone predicted with different metric values corresponding to w and θ ranges over $0.2-1.0$ and $0.1-0.5$ respectively based on small world phenomena (level=6) in Gr-QC dataset whose ground truth communities are unknown



(A) Average Isolability (B) NOC (C) Coverage

(D) Modularity (E) External Density

FIGURE 3.9: Safe zone predicted with different metric values corresponding to w and θ ranges over $0.2-1.0$ and $0.1-0.5$ respectively based on small world phenomena (level=6) in HEP-TH dataset whose ground truth communities are unknown



(A) Average Isolability (B) NOC (C) Coverage

(D) Modularity (E) External Density

FIGURE 3.10: Safe zone predicted with different metric values corresponding to w and θ ranges over $0.2-1.0$ and $0.1-0.5$ respectively based on small world phenomena (level=6) in Wiki-Vote dataset whose ground truth communities are unknown

TABLE 3.4: The Comparison of Accuracy Metric Values of TCD2 Algorithm against the state-of-the-art algorithms in various datasets whose ground truth communities known

Dataset	Algorithm	Accuracy Metrices Values					
		F-measure	NMI	Purity	Entropy	ARI	NOC
Strike	LICOD	0.0385	0.4680	0.2502	0.3043	0.3712	6
	RandW	0.2000	0.7510	0.5000	0.0724	0.6333	5
	LeadF	0.0905	0.5653	0.9167	0.1706	0.2541	7
	TILES	0.4429	0.6074	0.4583	0.0000	0.1918	12
	TCD2	0.6947	0.4927	0.9167	0.4760	0.4240	4
Karate	LICOD	0.0801	0.1604	0.5588	0.3200	0.0514	6
	RandW	0.9704	0.8365	0.9706	0.1614	0.8823	2
	LeadF	0.1310	0.4522	0.9412	0.1765	0.3362	8
	TILES	0.4492	0.4894	0.4412	0.0000	0.2466	11
	TCD2	0.6682	0.1890	1.0000	0.8463	0.0391	2
Dolphin	LICOD	0.1534	0.3517	0.1935	0.5031	0.1232	11
	RandW	0.2222	0.4157	0.2419	0.1112	0.1846	9
	LeadF	0.0361	0.3226	0.2097	0.1613	0.0680	17
	TILES	0.1770	0.3343	0.2097	0.0000	0.0605	32
	TCD2	0.8412	0.6275	0.7903	0.2094	0.6822	6
Football	LICOD	0.0655	0.5088	0.1217	0.2846	0.0924	58
	RandW	0.3045	0.7292	0.6435	0.2808	0.5151	10
	LeadF	0.2544	0.6411	0.6000	0.4001	0.4306	9
	TILES	0.7259	0.8398	0.7391	0.0920	0.6690	22
	TCD2	0.3119	0.4401	0.8870	0.6358	0.1706	8

TABLE 3.5: The Comparison of Quality Metric Values of TCD2 Algorithm against the State-of-the-art Algorithms in Various Datasets

Community Structure	Dataset	Algorithm	Quality Metrices Values				
			Modularity	Coverage	External Density	Average Isolability	NOC
Known	Strike	LICOD	0.5391	0.6316	0.0769	0.2589	6
		RandW	0.6589	0.7895	0.0369	0.5848	5
		LeadF	0.5170	0.6053	0.0610	0.4580	7
		TILES	0.4044	0.4737	0.0389	0.2873	12
		TCD2	0.8172	0.8421	0.0222	0.8246	4
	Karate	LICOD	0.6053	0.8462	0.0738	0.1865	6
		RandW	0.6341	0.8718	0.0351	0.7684	2
		LeadF	0.3540	0.4615	0.0911	0.2314	8
		TILES	0.1289	0.2949	0.0557	0.1818	11
		TCD2	1.0000	1.0000	0.0000	0.7029	2
	Dolphin	LICOD	0.3015	0.3396	0.0643	0.1362	11
		RandW	0.4313	0.4906	0.0488	0.2825	9
		LeadF	0.3777	0.4214	0.0523	0.2196	17
		TILES	0.2969	0.3522	0.0283	0.1147	32
		TCD2	0.8396	0.8553	0.0108	0.7859	6
	Football	LICOD	0.0798	0.0881	0.0891	0.0227	58
		RandW	0.5629	0.6215	0.0401	0.4000	10
		LeadF	0.4993	0.5514	0.0476	0.3420	9
		TILES	0.4772	0.5041	0.0246	0.3299	22
		TCD2	0.7453	0.7896	0.0203	0.7814	8
Unknown	GR-QC	LICOD	0.6503	0.7654	0.0002	0.7509	615
		RandW	0.5480	0.5532	0.0005	0.6553	862
		LeadF	0.5698	0.5769	0.0005	0.3865	1123
		TILES	0.5182	0.5212	0.0003	0.2673	1457
		TCD2	0.7435	0.7450	0.0001	0.3808	1268
	HEP-TH	LICOD	0.6889	0.6918	0.0001	0.6584	570
		RandW	0.4808	0.4816	0.0003	0.5356	1504
		LeadF	0.4294	0.4303	0.0003	0.2991	2445
		TILES	0.4243	0.4285	0.0002	0.3153	1942
		TCD2	0.6832	0.6873	0.0001	0.3883	1942
	Wiki-Vote	LICOD	0.9047	1.0000	0.0000	0.9771	25
		RandW	0.2712	0.3058	0.0035	0.0993	377
		LeadF	0.0666	0.0677	0.0037	0.0536	1382
		TILES	0.1309	0.3155	0.0016	0.0017	903
		TCD2	0.9999	0.9999	0.0000	0.2648	329

TABLE 3.6: The Comparison of Accuracy Metric Values of TCD2 Algorithm under Dynamic Settings (at Different Timestamps)

Tree Structure	Dataset	Timestamps	Accuracy Metric Values					
Level			F-measure	NMI	Purity	Entropy	ARI	NOC
Three Degree Theory (3)	Strike	1	1.00000	0.41359	0.29167	0.00000	1.00000	1
		2	0.72152	0.37170	0.41667	0.21031	0.62471	3
		3	0.65625	0.41807	0.62500	0.38196	0.50869	3
		4	0.60804	0.45697	0.75000	0.45658	0.34537	4
		5	0.61261	0.47059	0.79167	0.47598	0.31969	4
	Karate	1	0.88417	0.36388	0.50000	0.26128	0.83692	1
		2	0.72296	0.12076	0.64706	0.58390	0.58537	1
		3	0.67141	0.02720	0.85294	0.84635	0.34470	1
		4	0.62241	0.18895	0.85294	0.84635	0.03915	2
		5	0.62241	0.18895	0.85294	0.84635	0.03915	2
	Dolphin	1	0.54916	0.30457	0.29032	0.14998	0.45918	5
		2	0.57100	0.37693	0.45161	0.14998	0.42663	6
		3	0.57520	0.40448	0.53226	0.20241	0.38792	6
		4	0.60678	0.46756	0.61290	0.20936	0.36371	6
		5	0.60678	0.46756	0.61290	0.20936	0.36371	6
	Football	1	0.30588	0.26391	0.61739	0.51259	0.21036	3
		2	0.30760	0.41131	0.70435	0.61913	0.17394	7
		3	0.31159	0.43541	0.73043	0.64248	0.17060	7
		4	0.31189	0.43193	0.73043	0.64733	0.16971	6
		5	0.31189	0.43193	0.73043	0.64733	0.16971	6
Small World Phenomena (6)	Strike	1	1.00000	0.41359	0.29167	0.00000	1.00000	1
		2	0.68992	0.19508	0.54167	0.40948	0.55153	1
		3	0.54709	0.08308	0.75000	0.70768	0.31455	1
		4	0.51402	0.11232	0.87500	0.83360	0.04622	2
		5	0.51558	0.10607	0.91667	0.86881	0.03846	2
	Karate	1	0.88417	0.36388	0.50000	0.26128	0.83692	1
		2	0.72296	0.12076	0.64706	0.58390	0.58537	1
		3	0.67141	0.02720	0.85294	0.84635	0.34470	1
		4	0.62241	0.18895	0.85294	0.84635	0.03915	2
		5	0.62241	0.18895	0.85294	0.84635	0.03915	2
	Dolphin	1	0.54916	0.30457	0.29032	0.14998	0.45918	5
		2	0.57100	0.37693	0.45161	0.14998	0.42663	6
		3	0.83442	0.53763	0.70968	0.20241	0.72090	3
		4	0.84117	0.62747	0.79032	0.20936	0.68219	3
		5	0.84117	0.62747	0.79032	0.20936	0.68219	3
	Football	1	0.30410	0.24921	0.63478	0.52584	0.21044	2
		2	0.21921	0.20254	0.83478	0.81087	0.07910	2
		3	0.21287	0.20583	0.88696	0.86226	0.05844	2
		4	0.21287	0.20583	0.88696	0.86226	0.05844	2
		5	0.21287	0.20583	0.88696	0.86226	0.05844	2

TABLE 3.7: The Comparison of Quality Metric Values of TCD2 Algorithm under Dynamic Settings (at Different Timestamps)

Tree Structure	Dataset	Timestamps	Quality Metric Values				
Level			Modularity	Coverage	External Density	Average Isolability	NOC
Three Degree Theory (3)	Strike	1	1.00000	1.00000	0.00000	0.00000	1
		2	0.49074	0.61111	0.07000	0.57071	3
		3	0.63923	0.70370	0.04211	0.61746	3
		4	0.59954	0.66667	0.03659	0.53750	4
		5	0.62396	0.68421	0.03429	0.54318	4
	Karate	1	1.00000	1.00000	0.00000	0.00000	1
		2	1.00000	1.00000	0.00000	0.00000	1
		3	1.00000	1.00000	0.00000	0.00000	1
		4	0.86756	0.88158	0.03103	0.70196	2
		5	0.87130	0.88462	0.03103	0.70292	2
	Dolphin	1	0.86259	0.87179	0.00616	0.68102	5
		2	0.76036	0.78205	0.00954	0.58398	6
		3	0.62598	0.67521	0.01713	0.57824	6
		4	0.56501	0.62821	0.02085	0.50870	6
		5	0.55864	0.62264	0.02157	0.49887	6
	Football	1	0.85736	0.86928	0.00663	0.64589	3
		2	0.64678	0.68954	0.01169	0.45554	7
		3	0.53688	0.60566	0.01914	0.40738	7
		4	0.50437	0.58007	0.02720	0.43749	6
		5	0.50530	0.58075	0.02720	0.43767	6
Small World Phenomena (6)	Strike	1	1.00000	1.00000	0.00000	0.00000	1
		2	1.00000	1.00000	0.00000	0.00000	1
		3	1.00000	1.00000	0.00000	0.00000	1
		4	0.87654	0.88889	0.04762	0.63636	2
		5	0.88366	0.89474	0.04545	0.63810	2
	Karate	1	1.00000	1.00000	0.00000	0.00000	1
		2	1.00000	1.00000	0.00000	0.00000	1
		3	1.00000	1.00000	0.00000	0.00000	1
		4	0.86756	0.88158	0.03103	0.70196	2
		5	0.87130	0.88462	0.03103	0.70292	2
	Dolphin	1	0.86259	0.87179	0.00616	0.68102	5
		2	0.76036	0.78205	0.00954	0.58398	6
		3	0.85945	0.87179	0.00903	0.82281	3
		4	0.84402	0.85897	0.01029	0.79481	3
		5	0.83964	0.85535	0.01076	0.78594	3
	Football	1	0.90781	0.91503	0.00463	0.89701	2
		2	0.82281	0.84641	0.00992	0.81444	2
		3	0.75019	0.79303	0.01863	0.74452	2
		4	0.73310	0.78105	0.02627	0.71651	2
		5	0.73362	0.78140	0.02627	0.71667	2

GR-QC : It is an affiliation network of authors and papers submitted to arXiv repository under category General Relativity and Quantum Cosmology. The dataset used in the work is adapted from [19] which has been sampled from the original data collected in [85].

HEP-TH : It is a citation network of authors and papers submitted to arXiv repository under the category of High Energy Physics Theory. The dataset used in the work is adapted from [19] which has been sampled from the original data collected in [85].

Wiki-Voter : [84] has used the edit history of Wikipedia to generate a graph for the selection of administrator. It is a directed graph representing users as nodes and the directions of the edges show the votes of the users. An edge from u to v means, u has voted for v. We have translated the directed graph to an undirected one and then used it.

3.3.3 Methods for Comparison

The results obtained have been compared with four state-of-the-art works found in the literature including both static and dynamic settings of the community detection algorithms. LICOD, RandW and LeadF considers network as a static graph while TILES and TCD2 works with dynamic graphs. Their brief descriptions have been listed below.

1. LICOD : [70] have proposed a leader based approach for community detection. It classifies nodes as leaders and followers. The algorithm first identifies the potential leaders by using topological metrics in an iterative manner. The memberships of the nodes are updated till the community of the nodes attain a stable assignment.

2. RandW : Authors in [152] have stated the importance of the metadata available in network. They proposed an improvised version of the random walk algorithm. Instead of randomly selecting the next node in the algorithm, they proposed to weigh this decision with the attributes of the nodes and the edges.

3. LeadF : This algorithm also considers the leader-follower based community structure in a given network [146]. The authors have proposed that the internal community properties are more reliable and flexible for detecting communities. The algorithm works in two phases, the leader identification phase and the community formation phase.

4. TILES : The authors in [138] have proposed two types of memberships, core and peripheral, for nodes of a given network. As the algorithm works with streaming interactions of graph, label propagation among the core nodes is initialized for every change in network.

5. TCD2 : It is our proposed algorithm for community detection in social networks which models the community structure of the network to a tree structure.

3.4 Performance Analysis

A brief analysis and results are presented in this section to validate the performance of proposed algorithm against the state-of-the-art algorithms. Also, parameter analysis are presented to identify the safe zone for different parameter on various datasets.

3.4.1 Parameter Analysis

3.4.1.1 Window size W and dissolution threshold θ

This section present parameter analysis of w and θ in quality and accuracy metrics over a range of seven real world datasets. Safe zone predicted in Strike dataset for these parameters is shown in form of confusion matrix in figure 3.3. Confusion matrix for remaining datasets used in the experiment are presented in Figures 3.5 to 3.10. Safe zone is the range of parameters which facilitates the desirable metric value. In other words, its a trade-off between parameters for desirable metric value. Table 3.3 lists safe zone ranges for a dataset in a given metric. Each cell of the table comprises range/s in form $[(a,b),(c,d)]$ where (a,b) and (c,d) represent inclusive range of w and θ respectively.

3.4.1.2 Mixing parameter μ

The impact of mixing parameter μ on different accuracy and quality measures are shown in Figure 3.4. Figure 3.4a show that LFR1 has better accuracy in terms of NMI than LFR2 under both three degree theory and small-world phenomena. Also, user's influence better suited than connectedness regarding NMI accuracy measure. Similarly, the impact of μ on different accuracy measures are shown in figures 3.4b to 3.4e. Figure 3.4f show that LFR2 has better quality in terms of average isolability than LFR1 under both three degree theory and small-world phenomena. Also, connectedness property better suited than influence regarding average isolability quality measure. Similarly, the impact of μ on different quality measures are shown in figures 3.4g to 3.4i.

3.4.2 Accuracy Analysis

In order to analyze the performance of our algorithm in terms of accuracy measures against the state-of-the-art methods, four datasets with known community-structures are utilized. The results obtained from these datasets are shown in Table 3.4. From the results, it is clearly visible that the proposed algorithm outperform all methods in terms of Purity and under-performs in terms of Entropy. For Strike dataset, TCD2 shows higher F-measure and ARI than compared methods. Whereas, TCD2 have better NMI than LICOD but lagging behind other compared methods. For Karate dataset, TCD2 outperforms other competitors except RandW in terms of F-measure. TCD2 have comparable NMI and ARI than LICOD but slightly lagging behind other compared methods. For Dolphin dataset, our algorithm outperform state-of-the-art methods in terms of F-measure, NMI, and ARI. For Football dataset, TCD2 outperforms compared methods except TILES in terms of F-measure and slightly lagging behind others in NMI and ARI. Overall, TCD2 have trade-off among the state-of-the-art algorithms in terms of accuracy.

3.4.3 Quality Analysis

In order to analyze the performance of proposed algorithm in terms of quality measures against the state-of-the-art methods, four datasets with known community-structures and three dataset with unknown community-structures are utilized. The results obtained from these datasets are shown in Table 3.5. TCD2 performs better than all compared methods in all metrics on datasets with known ground truth except for Karate dataset where it slightly lagging behind than RandW in Average Isolability. For unknown ground truth dataset GR-QC, TCD2 perform better than compared methods in Modularity, Coverage, and External Density, whereas slightly lagging behind in Average Isolability. For HEP-TH dataset, the proposed algorithm outperforms compared methods except LICOD in Modularity, Coverage, and External Density whereas comparable with LICOD. TCD2 comparable with LeadF and TILES in terms of Average Isolability, whereas slightly lagging behind LICOD and RandW. For Wiki-Vote dataset, TCD2 perform better than compared methods in Modularity, Coverage, and External Density, whereas slightly lagging behind in Average Isolability. Overall, TCD2 perform almost better than compared methods with known ground truth and comparable on unknown ground truth datasets. Therefore, it is

clear that TCD2 is inclined towards quality and achieves a trade-off between accuracy and quality.

3.4.4 Performance Measuring under Dynamic Settings

The accuracy and quality analysis of TCD2 under dynamic settings is presented in Table 3.6 and Table 3.7 respectively. Algorithm's performance values are noted on five consecutive timestamps based on three degree theory and small-world phenomena on Strike, Karate, Dolphin and Football datasets. Table 3.6 demonstrates that accuracy measures F-measure, NMI, Purity and ARI have positive trend with increasing timestamps whereas Entropy increases with time which is because of the increasing number of nodes. Similar observation can be drawn from Table 3.7, owing to increase in number of nodes, External Density is increasing with time whereas Modularity, Coverage and Average Isolability shows positive trend with time.

3.5 Conclusion and Future Work

In this chapter, we have addressed two node level properties connectedness and influence to identify closely related groups. To incorporate these properties, two well-known concepts small-world phenomena and three degree theory are utilized. Building on these concepts, a tree based community detection in dynamic social networks (TCD2) algorithm is presented. This work also discusses the procedure of initialization, expansion, and dissolution of TCD2. Besides, we have performed experiments on real-world networks along with synthetic networks to validate the performance of our algorithms. The algorithm's performance is supported by well-known quality and accuracy metrics. The experimental results showed that the algorithm achieves a trade-off between quality and accuracy against compared algorithms. Precisely, the algorithm is slightly inclined towards quality. These results also validate the advantage of tree-structure based on small-world phenomena and three degree theory.

There is a considerable scope for undertaking further studies based on the proposed work in the area of community detection in dynamic social network. For example, due to large

number of users and high-speed data transfer in social networks, a lot of uncertainness are present to estimate users influences. Therefore, an adoptive influence estimation strategy can be useful under a specific diffusion model to tackle these uncertainness. Similarly, to explore connectedness, ego strength of users corresponding to ego networks can be utilized. There are some interesting problems in social network analysis such as link prediction, content spread enhancement, and influence maximization can be addressed using community detection.