5. COMMUNITY DETECTION BASED ON SUBGRAPH ANALYSIS

An in-depth analysis of detected communities is required further exploration about cohesive subgraphs in the network, their nodes, the interaction between the nodes and their in-degree and out-degree, in order to improve the quality of community detection. The problem of identifying sub-graphs in a graph structure and complex networks is challenging. The subgraph analysis approach is adopted here based on maximal k-core, k-plex and maximal clique for further investigation of sub-community detection. This chapter describes the implementation of three primary sub-graph based detection methods for sub-community detection and summarizes the findings.

5.1 INTRODUCTION

All graphs are finite, simple, and directed. A social network is a graph whose vertices represent a set of actors and whose edges indicate relationships between actors. In computational social networks, finding a large cohesive subgraph is an extensively studied topic with a large number of applications. The sociologic applications of cohesive subgroups include identification of work groups, sports teams, political party, religious cults, or hidden structures like criminal gangs and terrorist cells. A subgraph induces a clique if there exists an edge between every pair of vertices. Clique is one of the earliest and most commonly used models in the field of cohesive subgraphs detection. A clique is a graph with an edge between any pair of vertices, which can be regarded as the most cohesive graph. Cliques also provide an intuitive approach for detecting cohesion in social networks.

Cliques and graph theoretic clique relaxations are used to model clusters in graph-based data mining, where data is modeled by a graph in which an edge implies some relationship between the entities represented by its endpoints. Cohesive subgroups are subsets of actors amongst whom there are relatively strong, direct, intense, frequent, or positive ties. These subgroups are interesting because they facilitate the emergence of consensus among the actors. In other words, members within a cohesive subgroup tend to exhibit homogeneity. The properties of cliques like vertex degree, path length, and connectivity are captured to model cohesion in social networks. The maximum clique problem has applications in ad hoc wireless networks, data mining, combinatorial optimization, biochemistry, and genomics. In contrast to existing sub-graph extraction techniques which are based on a complete clustering of the graph nodes, the algorithm takes

into account the fact that not every participating node in the network needs to belong to a community. Another advantage is that the method does not require specifying the number of clusters. This motivates the study of clique relaxations and the current research has relaxed a variety of clique properties including familiarity, reachability, and robustness. Some techniques for identifying cohesive subgroups based on relaxed cliques are k-clique, k-core and k-plex [83].

The problem of identifying sub-graphs helps to analyze graph structures and complex networks and it is challenging. This work demonstrates an approach for identifying a set of sub-graphs of a given graph through a set of partitioning techniques using maximal k-clique, maximal k-core, and maximal k-plex algorithms.

5.2 SUBGRAPH ALGORITHMS

A subgraph is a subset of the nodes of a network, and the edges linking these nodes. Any group of nodes can form a subgraph. **S**ubgraphs or components are portions of the network that are disconnected from each other. In this research work, an in-depth investigation of communities has been performed using three subgraph algorithms such as maximal k-clique, maximal k-core, maximal- k-plex which are described in the following sections.

5.2.1 Maximal K-Clique Algorithm

The core elements of this algorithm are cliques and k-cliques. Before presenting the maximal k-clique algorithm, the mathematical formulations of cliques and k-cliques are stated below.

Clique: A clique is a subset of the vertices such that every pair of vertices in the subset is connected by an edge. Given a directed graph G= (V, E) where V denotes the set of vertices and E the set of edges, the graph G_1= (V_1, E_1) is called a sub-graph of G if $V_1 \subseteq V, E_1 \subseteq E$ and for every edge (v_i, vj)€ E_1 the vertices v_i,v_j € V_1. A sub-graph G_1 is said to be complete if there is an edge for each pair of vertices. A complete sub-graph is also called a clique as shown in Fig.5.1. A clique is maximal, if it is not contained in several other cliques. The clique number of a graph is equal to the cardinality of the largest clique of G and it is obtained by solving the maximum clique problem.

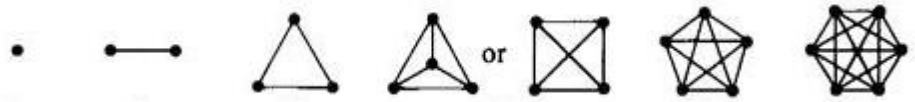

Fig. 5.1 Cliques with 1, 2, 3, 4, 5 and 6 Vertices

The clique structure, where there is an edge for each pair of vertices, shows many restrictions in real life modeling. So, alternative approaches are suggested in order to relax the clique concept, such as k-clique, k-core and k-plex.

K-clique: A clique of a graph G is a complete subgraph of G, and the clique of the largest possible size is referred to as a maximum clique. A maximal clique is a clique that cannot be extended by including one or more adjacent vertex, such that it is not a subset of a larger clique. A group of size k is called a k-clique. It is a maximal set of vertices that are at a distance not greater than k from each other such that 1-cliques correspond to vertices, 2-cliques to edges, and 3-cliques to cycles.

Alternately k-clique is the distance based model, where k is the maximum path length between each pair of vertices. A k-clique is a subset of vertices C such that, for every i, i€C, the distance d (i, j) $\leq$ k. The one-clique is identical to a clique because the distance between the vertices is one edge. The 2-clique is the maximal whole sub-graph with a path length of one or two edges. The path distance of two can be represented by the friend of a friend connection in social relationships. The increase of the value k communicates to a gradual relaxation of the criterion of clique membership.

Determining clique nodes is the first step to build a maximal-clique graph. A set of maximal cliques in the original graph *G* are adopted as clique nodes of the corresponding maximal-clique graph G^c. Since each maximal clique is one of the largest cliques having all the nodes of G, the determination process of clique nodes is transformed into finding all the largest cliques that each node in *G* belongs to, and the algorithm is developed.

First, the algorithm calculates the degree of each node in *G* and sorts the nodes in descending order of their degrees. Since the nodes with higher degrees are more likely to constitute larger maximal cliques, the clique nodes are determined in descending order according to their clique sizes. Suppose that the number of nodes of *G* is *N* and the largest node degree is k_{max}, then the size of cliques in *G* is not larger than k_{max}+1. The algorithm searches for the *k*-clique(s) of each node as *k* decreases from k_{max}+1 to 1. As every two nodes in a clique are adjacent, the degrees of all nodes in a *k*-clique must be larger than *k*-1. Since

only the largest clique(s) for each node are interested, the searching process stops seeking smaller cliques for a node if it has been assigned to the larger ones.

Algorithm

Determining the clique nodes of G^C
Input: Original graph G = (V, E);
Output: set of clique nodes V^C
$V^C \leftarrow \phi$;

Calculate the degree (v_i) of each node $v_i \in V$;
$K_{max} \leftarrow \max_{vi} e^v \; k(v_i)$;
Sort the nodes in descending order of the degree,
For $k=k_{max}+1$ to 1 do
    For each node $v_i \in V$ do
        If k (v_i)<k-1
            No more k-cliques exist and goto Outer loop;
        end if
        if v_i has been assigned to one clique node
            goto Inner loop;
        end if
        Neigh(v_i)$\leftarrow$\{v_j|(v_j is adjacent to vi and k(v_j)$\geq$k-1\};
        If |Neigh (v_i)|<k-1
            v_i cannot constitute k-cliques and goto Inner loop ;
        end if
        if the nodes in Neigh(v_i) can constitute q(k-1)-cliques(q$\geq$1)
            Vc$\leftarrow$Vc $\cup$\{v_i,((k-1)-clique)1\}$\cup\ldots\cup$(v_i,((k-1)q\};
        end if
        Inner loop;
    end for
    Outer loop;
end for

The process of searching for *k*-clique(s) of a node happens when the following three conditions are satisfied; (i) the node degree is not smaller than (*k*-1), (ii) the node has not been assigned to any cliques (iii) at least (*k*-1) adjacent nodes have a degree no smaller than (*k*-1). If all these conditions are satisfied, then the problem of finding all the *k*-cliques that contain this node is transformed into the task of searching for the (*k*-1)-clique(s) constituted by its neighbors. Then all the discovered *k*-cliques are added to the set of clique nodes [84].

5.2.2 Maximal k-Core Algorithm

Degree based models of cohesion, which overcome the drawbacks inherent in the definitions of k-clique and k-club, were introduced in [79] and the concept of a k-core was introduced in [80], which is a subgraph with minimum degree at least k. In other words, $S \subseteq V$ is a k-core if $|N(v) \cap S| \geq k \; \forall \; v \; S$, where N (v) denotes the set of neighbors of a vertex $v \in$

V in G. K-cores were noted to only indicate dense regions of the graph and not necessarily identify a cohesive subgroup. This approach was only to produce global measures that captured the cohesive subgroups as well as regions surrounding them. Pick a vertex v of minimum degree δ (G), if δ (G) ≥ k then it has a k-core. If δ (G) < k, then that vertex cannot be in a k-core. Delete the corresponding vertex, G ← G − v and continue recursively until the vertex set of G is a maximum k-core or the set is empty. These structures are easy to find and they only point out dense regions of the graph where interesting subgroups can be found.

A k-core is the subgraph generated by recursively removing all nodes with degree smaller than k from a graph. As it uses the degree to induce the subgraphs, it is also called a degree core. Mathematically, let G= (V, E) be a simple graph. A graph G = (V, E) of |V | = n vertices and |E| = e edges; a k-core is defined as A subgraph H = (C, E|C) induced by the set C ⊆ V is a k-core or a core of order k iff ∀v ∈ C : degree H(v) ≥ k, and H is the maximum subgraph with this property. A k-core of G can be obtained by recursively removing all the vertices of degree less than k, until all vertices in the remaining graph have at least degree k. A vertex i have coreness c if it belongs to the c-core but not to (c + 1)-core. Let c_i the coreness of vertex i. A shell C_c is composed of all the vertices whose coreness is c. The maximum value c such that C_c is not empty is denoted c_{max}. The k-core is thus the union of all shells C_c with c ≥ k. Each connected set of vertices having the same coreness c is a cluster Q_c. Each shell C_c is thus composed by clusters Q_m^c such that $C_c = \cup_1 \leq m \leq q_{max}^c \; Q_m^c$ where q_{max}^c is the number of clusters in C_c.

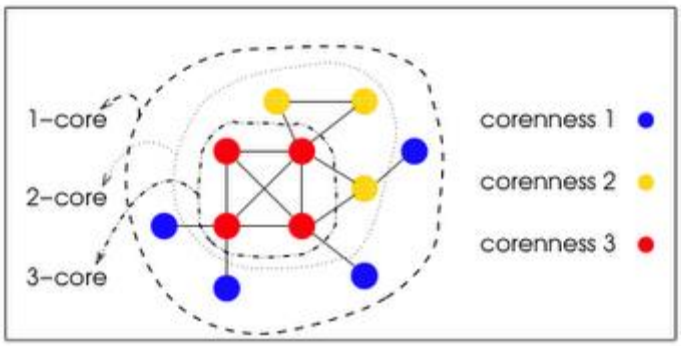

Fig. 5.2 k-Core for Small Graph

Fig.5.2 shows a simple illustration of k-core decomposition of a connected graph and its visual representation. Every vertex of a associated graph belongs to the 1-core. A dashed line encloses all the vertices in the 1-core i.e. the entire graph. Then, all vertices of degree d < 2 are recursively removed. The other vertices maintain a degree d ≥ 2 are not eliminated. The remaining vertices form the 2-core is enclosed by a dotted line. Further pruning allows

identifying the innermost set of vertices, the 3-core. All vertices having an internal degree at least 3 are highlighted by a dash-dotted line. Each closed line contains the set of vertices belonging to a given k-core, while different k-shells are distinguished.

The algorithm computes every non-empty degree core of a graph and identifies the connected components of these subgraphs. These components form a hierarchy where two components have a parent-child relationship when the latter has been immediately split from the former. Component A has split from a component B if A is a subgraph of B and A is a component of the k-core and B a component of the (k+1) core for some integer k. V (A) is taken to the set of vertices contained within the component A [85]. The maximal k-core algorithm is given below.

Algorithm

```
Compute the degrees of vertices;
order the set of vertices V in increasing order of their degrees;
 for each v ∈ V in the order do begin
        core[v] := degree[v];
        for each u ∈ Neighbors(v) do
                if degree[u] > degree[v] then begin
                        degree[u] := degree[u] − 1;
                        reorder V accordingly
        end
end;
```

5.2.3 Maximal K-Plex Algorithm

A subset of vertices S is said to be a k-plex if the degree of every vertex in the induced subgraph G[S] is at least |S| − k. That is, S ⊆ V is a k-plex if the following condition holds:

$$\deg_{G[S]}(v) = |N(v) \cap S| \geq |S| - k \;\forall\; v \in S \qquad (5.1)$$

A k-plex is said to be maximal if it is not strictly restricted in any other k-plex. It is also called as the cardinality of the largest k-plex in the graph and denoted by $\omega_k(G)$. The maximum k-plex problem is to find the largest k-plex of the given graph. The maximum k-clique and maximum k-core problems are reduced to the maximum clique problem when k = 1 and is a relaxation of the clique requirement for all other k > 1, allowing for at most k − 1 non-neighbors inside the set. This is illustrated in Fig.5.4. The set {1, 2, 3, 4} is a 1-plex (clique), sets {1, 2, 3, 4, 5} and {1, 2, 3, 4, 6} are 2-plexes and the entire graph is a 3-plex.

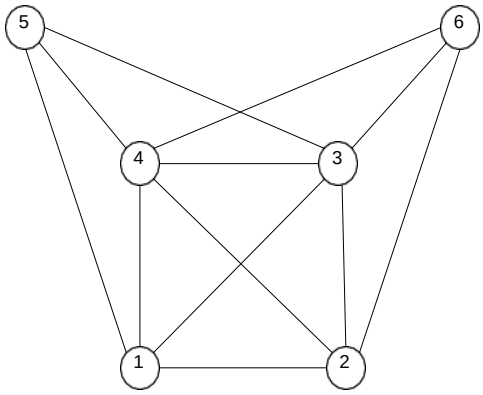

Fig. 5.3 Illustration of k-plexes for k = 1, 2, 3

The clique algorithm is generalized to find maximum k-plex. The clique algorithm examines every clique in G where G can contain an exponential number of cliques with respect to |V||. The algorithm attempts to avoid enumeration of an exponential number of subgraphs. The algorithm for maximal k-clique is given below.

Algorithm

```
while U≠Ø
        If |K| + |U| ≤ max
                Return
        End
        K= K∪ {v}; U = U\{v} for some v∈ U
        U' := {u ∈U; K U {u} is a k-plex
        basicPlex(U',K)
        end
        if |k|> max
end
return
```

The candidate set U for a k-plex k is defined as

$$U \coloneqq \{v \in V \backslash K : K \cup \{v\} is\ a\ k-plex\} \quad (5.2)$$

For u, v ∈ V, let d (u, v) be the length of the shortest path from u to v in G. The concept of neighborhood is based on the parity of shortest path lengths from some root node s. Given a root s ∈ V, define the following sets:

$$K_0 \coloneqq \{v \in V | d(s,v) even\} \; and \; K_1 \coloneqq \{v \in V | d(s,v) odd \quad (5.3)$$

For i ∈ {0, 1}, notice that $u, v \in K_i \; and \; uv \in E(\overline{H})$ together imply d (u, s) = d (v, s). Otherwise, d (u, s) and d (v, s) would have different parities. Therefore, for every v ∈ K_i,

$$N(v) \cap \{u \in K_i \{v\} : d(u,s) \neq d(v,s)\} = \emptyset \quad (5.4)$$

Let $K_i \notin \mathcal{I}_H$, there exists many subsets $K' \subseteq K_i$such that$K' \subseteq I_H$. In order to examine these subsets, elements in I_H are constructed from K_i by removing one end of every edge in $\bar{H}$ [K_i]. The edge uv ∈ E($\bar{H}$ [K_i]) can be removed using the following rules.

Rule 1. If deg$\bar{H}$ [K_i](v) ≤ deg$\bar{H}$ [K_i](u), remove u. Otherwise, remove v.

Rule 2. If deg$\bar{H}$ (v) ≤ deg$\bar{H}$ (u), remove u. Otherwise, remove v.

Rule 3. Always remove v.

Rule 4. Always remove u.

Let K_i^jbe the subset obtained from K_i be applying Rule j to every edge in E ($\bar{H}$ [K_i]). Rules 1 and 2 are greedy metrics. Rules 3 and 4 are included to diversify the search space. Then each set K_i^j is extended to a maximal k-plex in H. All k-plexes that are constructed from a set K_i in this way constitute a neighborhood. The search space is essentially a function of the root nodes, and specifying a set of neighborhoods is equivalent to specifying a set of root nodes R [86].

5.3 SUBGRAPH BASED COMMUNITY DETECTION MODEL

Here the community detection model is developed using subgraph analysis with maximal k-clique, maximal k-core, and maximal k-plex. The model is constructed with three components, input, process, and output. The input component uses twitter network data presented in chapter 3. The graph representation of the twitter data is used as input. The second component includes a community detection process wherein three different algorithms described in sections 5.2 are employed. In the first case, the maximal k-clique algorithm is implemented using the adjacency matrix representation of the given network. The community detection process is performed by determining the complete sub-graphs with maximum path length for each node. In the second case, the maximal k-core method is employed wherein the subgraphs are identified by recursively deleting all nodes of degree less than k from the given graph. The maximal k-plex algorithm is demonstrated in the third case wherein the subgraph is identified by the largest k-plex using nodelist. The third component is the output logic in which the effectiveness of this sub-graph based community detection algorithms is evaluated using centrality measures and also it analyses the detected communities based their membership distribution. The inferences are drawn based on the experiment results. The architecture of the model is shown in Fig.5.4.

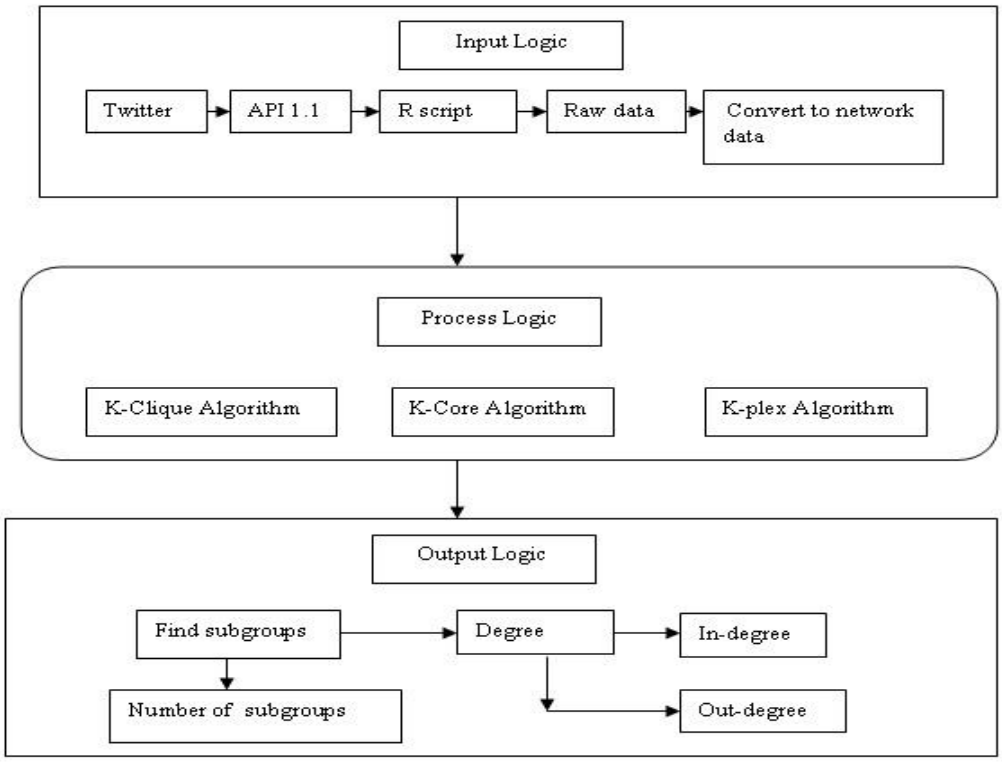

Fig. 5.4 Sub Graph Community Detection Framework

5.4 EXPERIMENTS AND RESULTS

Subgraph analysis through maximal k-clique, maximal k-core, and maximal k-plex has been carried out for the same twitter network data in Matlab 2016a and R platform. In each case, the investigation was done for various k values ranging from 1 to 12. But the commonly used value of k is 3 in most of the existing research. Various centrality measures are evaluated and the membership distribution of nodes in various subgraphs is analyzed for all the three cases.

Results of k-Clique Algorithm

The adjacency matrix of the sample twitter network is used here and it is transformed into a network matrix using the get_community_matrix features in closeness, degree, in-degree, out-degree, edge-betweenness centrality, subgraph centrality, eigen vector centrality. This community matrix is used for further processing. The community matrix of the adjacency matrix shown in Table V of chapter 3 is given in Table XIII. The maximal k-clique algorithm has discovered 135 subgroups from the given network when k=3. Fig.5.5 shows the maximal k-clique subgraphs of the sample network.

Table XIII Community Matrix of Adjacency Matrix

0	0.00013	0	0.00014	0	0	0	0.00013	0	0.00012
0.00012	0.00013	0.00012	0	0.00012	0.00013	0.00012	0.00013	0.00012	0.00013
0.00012	0.00013	0.00012	0.00014	0.00012	0.00013	0.00012	0	0.00012	0.00013
0.00012	0.00013	0.00012	0.00012	0.00012	0.00012	0.00012	0	0.00012	0.00013
0	0	0	0.00013	0	0.00012	0	0.00013	0	0.00012
0.00012	0.00013	0.00012	0.00012	0.00012	0.00012	0.00012	0	0.00012	0.00013
0	0.00012	0	0	0	0.00013	0	0.00013	0	0
0.00012	0.00012	0.00012	0.00013	0.00012	0.00014	0.00012	0.00012	0.00012	0.00012
0.000123	0	0.00013	0	0.00013	0.00012	0	0.00014	0.00012	0.00013
0.00012	0.00013	0.00012	0.00012	0.00012	0.00012	0.00012	0	0.00012	0.00013

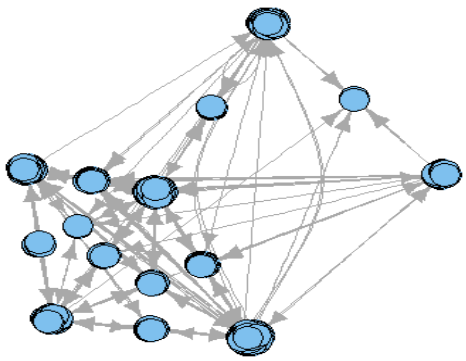

Fig. 5.5 Maximal Clique Subgroups of the Twitter Network

A sample of 10 sub graphs detected by maximal k- clique is shown below. The number indicates the node ids of the nodes in the communities.

{1 4 5 6 8 10 11 12 13 14 15 16 17 20 21 24 28 29 30 31 32 33 34 35 36 37 38 39 40 43 45 46 47 48 49 50 51}

{1 2 4 5 6 8 9 10 11 12 13 14 16 17 18 19 20 21 25 28 30 41 44 48 49 50 51 53 54 55 57 59 60 61 62 63 64 65}

{2 3 5 8 9 11 12 13 14 15 16 17 18 19 20 21 22 23 24 26 27 28 29 30 41 42 43 44 45 46 47 49 50 51 52 53 54}

{1 2 3 4 5 6 7 8 9 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 35 37 38 39 40 41 42}

{1 2 3 4 5 6 7 8 9 10 12 14 15 16 17 20 22 23 24 25 26 29 30 31 32 33 35 36 37 40 42 43 44 45 46 47 48 49 50}

{1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 32 33 35 36 37 38 39 40 41 42}

{2 3 5 6 7 8 9 10 13 14 16 21 23 24 25 26 27 28 29 30 32 35 36 37 38 39 40 47 48 51 54 56 58 59 60 62 64 66}

{2 3 4 5 6 7 8 9 10 13 14 15 16 17 18 19 20 21 24 25 26 27 29 30 32 33 35 36 38 39 40 42 43 44 46 48 49 50}

{3 4 5 6 7 8 10 12 15 16 17 18 19 20 22 24 25 26 27 28 29 30 31 32 33 35 36 38 39 40 41 43 45 46 47 48 49}

{3 4 5 6 7 8 11 12 14 15 16 17 18 19 20 21 22 23 25 26 27 28 29 30 31 34 36 38 39 40 43 45 46 47 48 50 51}

Modularity score is used to measure the strength of division of a network into modules. The modularity score achieved by maximal k-clique is 0.31. Membership distribution of friends and followers in communities is also analyzed. Out of 135 communities, 78 dense communities and 57 sparse communities are detected. Subgroups having less number of nodes ranging from 0 to 100 are considered as sparse subgroups and this indicate that interaction between the nodes is very less. The size of the largest community obtained is 1746 and the size of the smallest subgroup is 20. The metadata about communities are summarized in Table XIV.

Table XIV k-Clique Subgroups When k=3

k	Number of Communities	Largest Subgroup Size	Smallest Subgroup Size	Number of Dense subgroups	Number of Sparse Subgroups
3	135	1746	20	78	57

Various centrality measures of communities such as degree, in-degree, out-degree, closeness, edge-betweenness centrality, nodal centrality, sub-graph centrality, eigenvector centrality are derived from network matrix. From the results, it is found that the in-degree of 32 subgroups lies between 501 to 1800 whereas in-degree of 46 subgroups lies between 101 to 500. The in-degree of 57 sub-groups lies between 20 to 100 which indicate that friends and followers are less interactive with other nodes. The high out-degree of 22 subgroups lies between 101 to 250 and high out-degree of 27 subgroups lies between 61 to 100. High out-degree value of 24 subgroups suggests more interaction from outer node to the nodes in these subgroups. For other 84 subgroups, the out-degree lies in the range of 20 to 60. The degree measures of k-clique subgroups are evaluated using the k-clique algorithm and the results for 10 subgroups detected k=3 are presented in Table XV.

Table XV Centrality Measures of Communities

Measures	Com1	Com2	Com3	Com4	Com5	Com6	Com7	Com8	Com9	Com10
Closeness	0.0066	0.0058	0.0054	0.0057	0.0048	0.0055	0.0056	0.0066	0.0065	0.0064
Degree	172	130	150	137	72	119	152	160	127	143
In degree	82	62	62	71	40	60	89	71	39	59
Out degree	90	68	58	66	32	59	63	89	88	84
Betweenness	93	90	87	85	71	70	69	65	63	59
SC*	1.0652	1.0652	1.0427	1.0472	1.0239	1.0435	1.0456	1.0638	1.0616	1.0616
EC*	0.1089	0.0871	0.0759	0.0852	0.0399	0.0792	0.0825	0.1113	0.1047	0.1057
NC*	120.929	59.534	43.688	44.167	11.527	31.083	36.343	85.307	106.96	100.06

*SC-Sub-Graph Centrality, EC-Eigenvector Centrality, NC-Nodal Centrality

The subgroups recognized excessive in-degree, out-degree, closeness, sub-graph centrality for all sub-groups is in the range of 1.023-1.1/2 and Eigenvector centrality varies from 0.03 - 0.1. The highest value of nodal centrality is 120.93 for sub-community 1. The effects of maximal clique locating algorithm indicate the satisfactory consequences in all instances. This suggests that the sports person's community is dense and communication among nodes is excessive.

Results of k-Core Algorithm

A k-core is a maximal sub-graph that contains nodes of degree k or more. The coreness of a vertex is k if it belongs to the k-core. The main core is the core with the largest degree. The types of the core and the possible values like in-degree cores, out-degree cores, and total degree cores are computed. The experiment has been conducted in R statistical data mining platform with igraph packages to find the communities in sports person network. Based on the k-core size, subgraph communities are detected. Fig.5.6 shows the different sub-groups with core size 1 to 12.

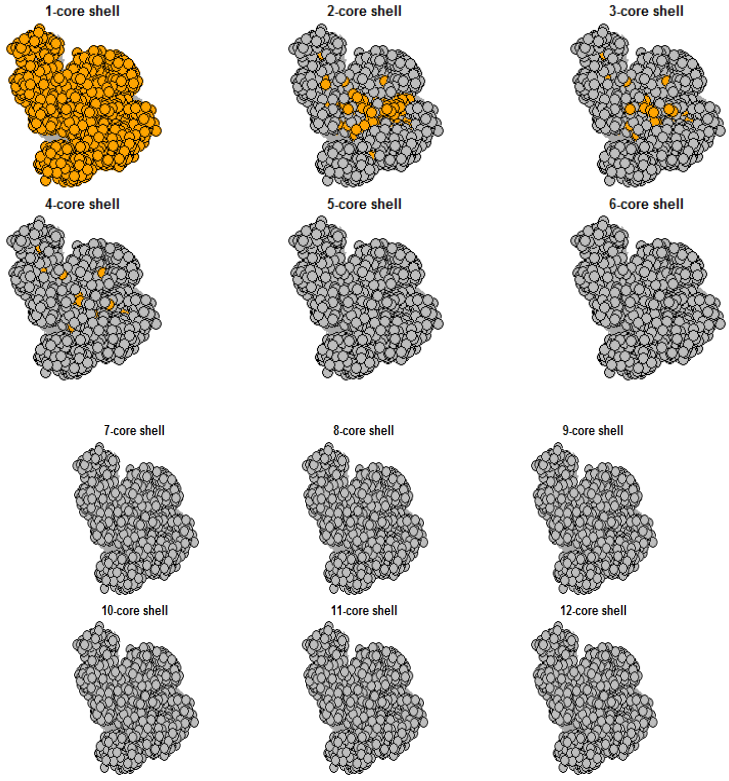

Fig. 5.6 (1-12) k-Core Sub Groups

When k value is 1, 2,3,4,5, the number of sub graph communities detected from the given network are 1590, 190, 150, 29, 20 respectively. Ten sample sub graphs identified from input network when k=3 with node ids are shown below.

{2 3 5 8 9 11 12 13 14 15 16 17 18 19 20 21 22 23 24 91 22 95 22 94 22 87 22 86 33 40 41 43 45 46 51 66 92}

{3 5 6 7 8 9 10 13 14 16 21 23 24 25 26 27 28 29 30 32 35 36 40 41 43 45 46 73 104 19 60 19 59 19 56 19 55 195}

{13 14 15 16 17 18 19 20 21 22 23 24 25 26 2728 129 30 32 3440 41 43 45 46 53 19 51 29 50 194 9 195 71 94}

{19 44 20 22 23 24 25 26 29 30 31 32 33 35 36 37 40 42 43 44 45 46 47 48 49 50 56 58 60 65 66 80 81 61 45}

{1 2 3 4 5 6 7 8 9 10 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 2728 29 30 32 34 36 37 38 39 40 41 42 45}

{1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 24 27 28 29 30 32 34 36 37 38 40 41 42 44 46 48 50 52 54}

{1 2 3 4 5 6 7 8 9 10 12 13 14 18 19 21 22 24 25 26 29 34 35 36 37 38 39 40 45 46 48 50 51 53 54 55 57 58 60}

{2 3 4 5 6 7 8 9 10 13 14 15 16 17 18 19 20 21 24 25 26 27 29 30 32 33 35 36 38 39 40 42 43 44 46 48 49 54 58}

{3 4 5 6 7 8 10 12 15 16 17 18 19 20 22 24 25 26 49 79 14 13 66 92 18 17 18 14 18 31 66 91 92 81 71 81}

{1 2 4 5 6 8 9 10 11 12 13 14 16 17 18 19 20 21 25 28 30 33 35 41 44 56 59 91132 180 190 272 303 311 829}

In each case number of dense and sparse communities is identified. The metadata of k-core communities for k = 1 to 5 are depicted in Table XVI. Membership distribution of friends and followers in communities is also analyzed. For example, when k=3, number of communities detected are 135, in which 82 are dense communities and 68 are sparse communities. The size of the largest community obtained is 1684 and the size of the smallest subgroup is 16. Fig.5.7 depicts node distribution of k-core sub groups of the community network and Fig 5.8 to Fig.5.12 shows sample subgroups of the sports person corresponding to k=1 to 5.

Table XVI Size of k-Core Subgroups for k= 1 To 5

k-Core Size	Number of Community	Largest Subgroup Size	Smallest Subgroup Size	Number of Dense Subgroups	Number of Sparse Subgroups
1	1590	2162	14	873	717
2	190	1954	17	119	71
3	150	1684	16	82	68
4	29	1984	23	20	9
5	20	713	15	16	4

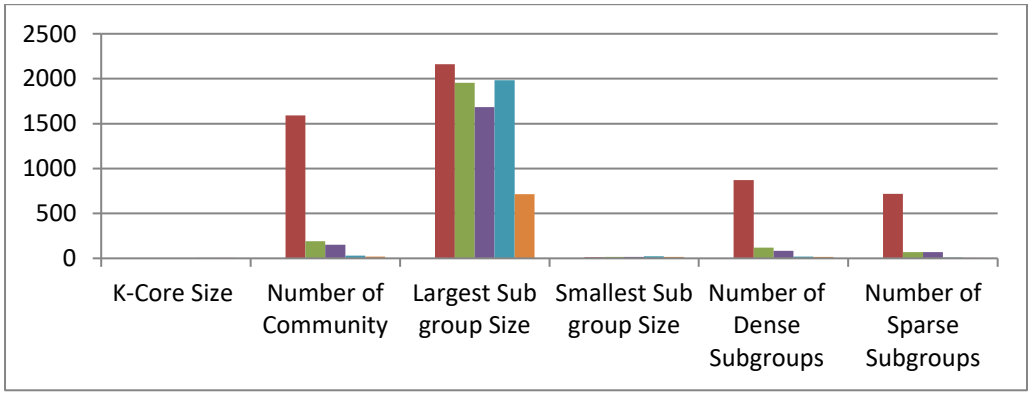

Fig. 5.7 Community Distribution of k-Core Subgraphs

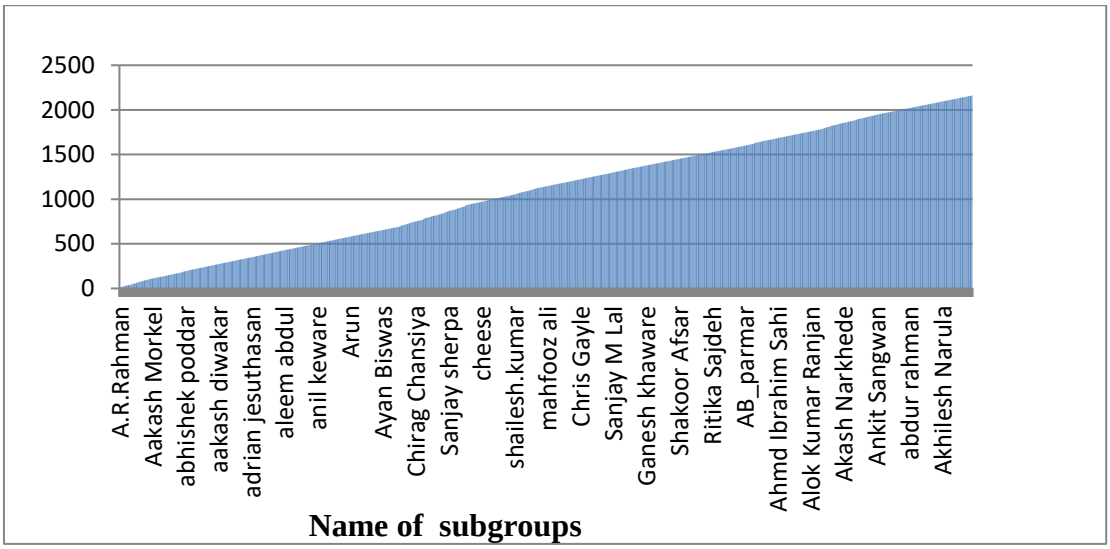

Fig. 5.8 Maximal k- Core Subgraphs when k=1

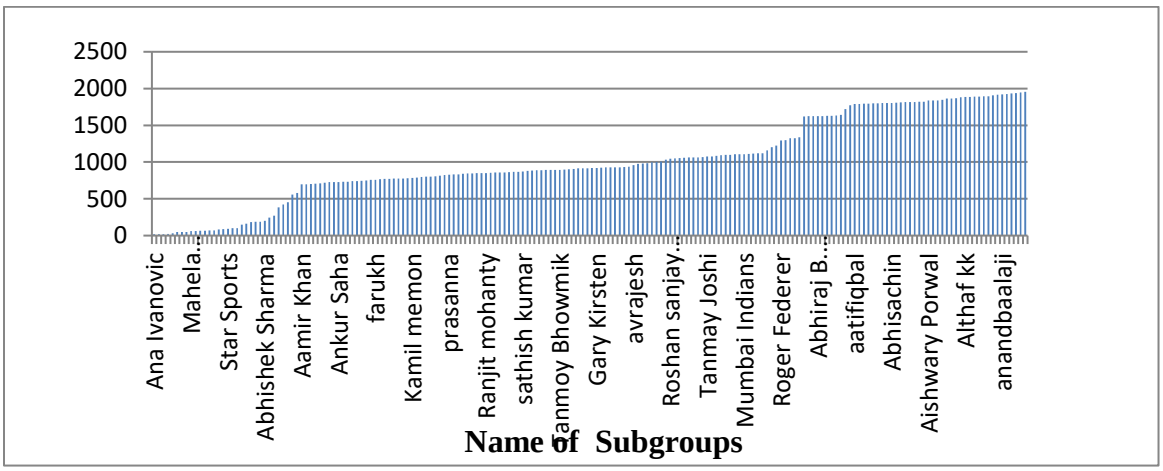

Fig. 5.9 Maximal k- Core Subgraphs when k=2

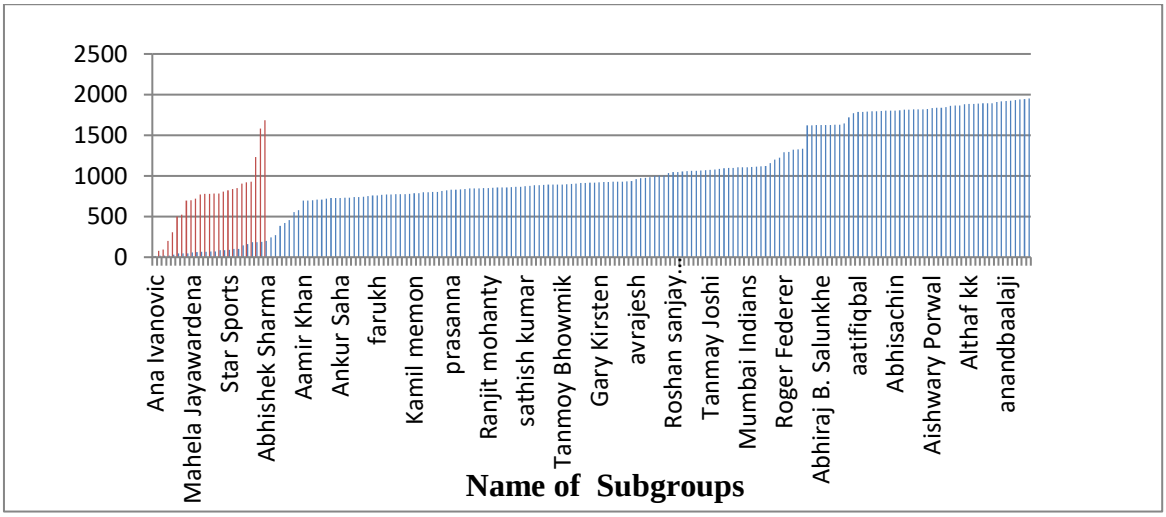

Fig. 5.10 Maximal k- Core Subgraphs when k=3

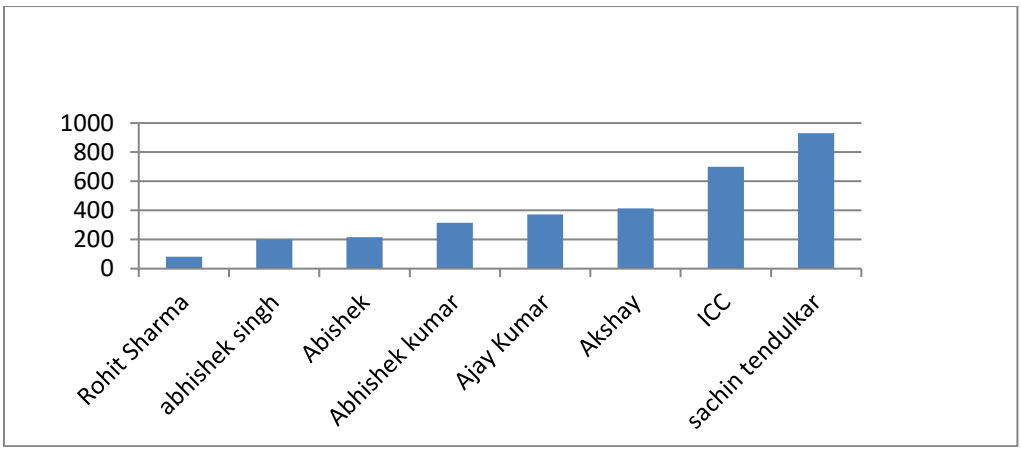

Fig. 5.11 Maximal k- Core Subgraphs when k=4

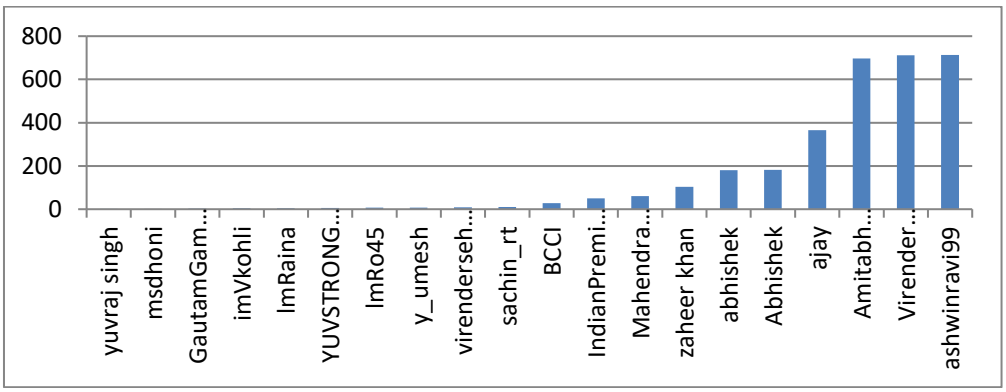

Fig. 5.12 Maximal k- Core Subgraphs when k=5

The Maximal k-core modularity score of 0.45 is achieved. This reveals dense connection exists between the nodes within clusters but sparse connections between nodes in different clusters. The degree measures of k-core subgroups are evaluated using the k-core algorithm and the basic measures of communities such as in-degree, out-degree of subgraphs, are derived for k=1 to 5 and analysis was done. Table XVII portrays the smallest and highest degree of the k-core subgraphs of the given input network. In this, the minimum degree is 11 maximum degrees is 2162 for one k-core subgraph while the minimum degree is 17 and the maximum degree is 1954 for two k-core sub-communities. The minimum and maximum degrees are 13, 81, 10, and 1684, 931, 173 for 3, 4 and 5 k-core communities.

Table XVII Degree Measures of k-Core Subgraphs

k-core size	Minimum Degree	Maximum Degree	Highest In-Degree	Lowest In-degree	Highest Out-degree	Lowest Out-Degree
1	11	2162	1252	40	452	145
2	17	1954	1324	43	541	221
3	13	1684	1572	62	126	303
4	81	931	923	32	489	263
5	10	713	844	12	393	125

Also, when k=3, in-degree of 39 subgroups lies between 501 to 1800 whereas in-degree of 42 sub-groups lies between 101 to 500. The in-degree of 69 subgroups lies between 20 to 100, which indicate that the nodes in these communities are less interactive with other nodes. The high out-degree of 28 subgroups lies between 101 to 250 and high out-degree of 43 subgroups lies between 61 to 100. The high out-degree value of 53 subgroups suggests more interaction from the outer node to these nodes. For other 79 subgroups, the out-degree lies in the range of 20 to 60. The results for 15 subgroups are presented in Table XVIII and Table XIX. The degree measures of 15 k-core subgraphs of the sample input network are illustrated in Fig.5.13.

Table XVIII Size of k-Core Sub-Groups when k=3

Subgroup	Size of Sub-Groups
abhinav	305
anil	501
Ankit Sharma	522
Harbhajan Singh	698
Ishant Sharma	700
AMAN BERLIA	722
imran ansari	770
jegan	779
jitendra tomar	781
K Nityananda Reddy	784
k vijaykumar	785
manu	807
pankaj arora	823
puran dev	839
ramakarthik	850
vinita jain	907
ESPNcricinfo	920
praveen kumar	928
Gautam Gambhir	1234
Aakash Jain	1584
Ajay	1684

Table XIX In-Degree and Out-Degree of k-Core Subgroups when k=3

Subgroups	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
In-degree	1101	761	137	431	791	823	62	78	1661	83	168	710	86	65	121
Out-degree	55	45	63	100	37	48	49	120	47	94	76	71	47	53	79

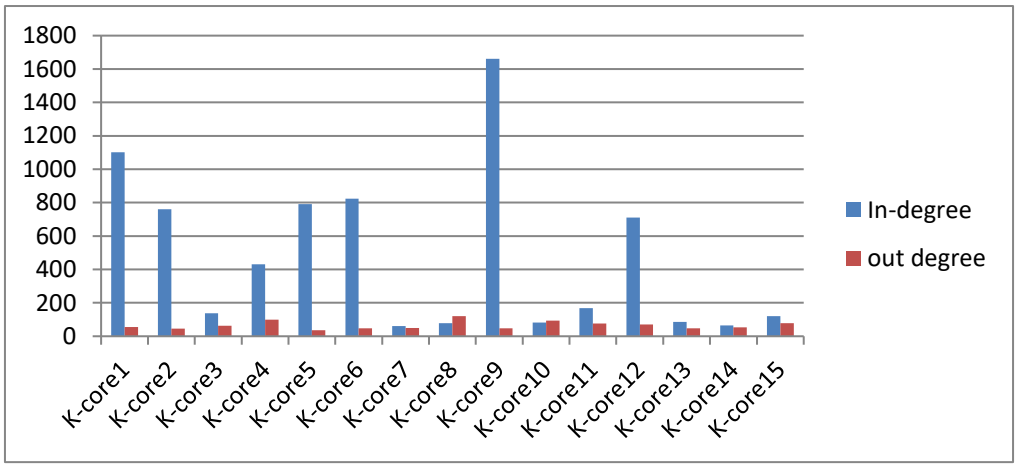

Fig. 5.13 In-Degree and Out-Degree of Maximal k- Core Sub Graphs when k=3

Results of k-Plex Algorithm

In this work, k-plex sub graph algorithm has been implemented using edgelist of the sample network and it has discovered 53 sub groups. The modularity score obtained is 0.23. Fig. 5.14a and Fig. 5.14b shows the k-plex subgraphs detected for the given nework. Two sample sub graphs of input network when k=3 with node ids are shown below.

{112, 14, 25, 62, 348, 10, 11, 12, 13, 141, 15, 160, 17, 20 ,21 ,241, 281, 129, 130, 131, 322, 331, 342, 352, 362, 372, 383, 394, 40, 43, 45, 46, 147 ,248 249,1 50, 251,12, 45, 68, 910, 110, 120 113, 142, 136, 147, 168, 199, 201, 210, 251, 258, 300, 401, 404, 408, 409, 500, 501, 530, 544, 554, 557, 509, 60, 61, 620, 633, 642, 651,23,589,611,712, 813, 914, 715, 916, 517, 418, 196, 203, 261, 222, 233, 244, 286, 272,829, 304,142, 433 444,546, 474,950, 515,253, 541,123, 456, 789, 111,213, 141,516, 171,819, 202,122, 239, 247, 252,627, 280, 293, 313,233, 353,738, 394, 414,212, 345, 678, 910, 120, 131,415, 161,718, 1920, 213, 224, 52,627,282,930, 323,436, 373,839, 404,142, 45,123, 456, 789, 1011, 121,314, 151,6 17, 181,920, 242,728, 293, 323,436, 379, 384, 414,244}

{464,850, 5254,1234, 567, 8910, 121,314, 181,921, 220, 240, 250, 261, 290, 343,536, 373,839 404,546, 485, 515,354, 555,758, 601,223 4, 568,789, 101,214, 155, 166 177, 220, 262, 232,4 25, 269, 2930, 313,233, 353,637, 4042, 4344, 4546, 4748, 4950,31 23, 14 56, 1789, 1011, 1213, 1415 ,1617, 1819, 2021, 2245, 2324, 2526, 2728, 2932, 335, 363,738,539,540,641,742,235, ,950, 515,253, 541,123, 678, 910, 131,416, 212,324, 252,627, 281, 2930, 324, 353, 363,738, 3940, 4748, 515,477, 565,8 59, 606,2 646,623, 456, 789, 1013, 141,516, 1718, 1920,

2124, 2526, 2729, 3032, 3335, 3638, 3940, 4243, 4446, 4849, 5034, 5678, 1012, 1516, 1718, 1920, 2224, 2526, 2728, 2930, 3132, 3335, 3638, 3940, 4143, 4546, 4748, 493, 4567, 811, 1214, 1516, 1718, 1920 2122, 2325, 26, 27, 2829, 3031, 3436, 3839, 4043, 4546, 4748, 5051}

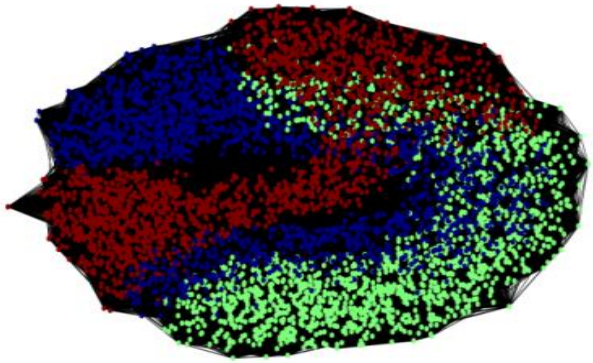

Fig. 5.14a Total k-plex Subgroup Network

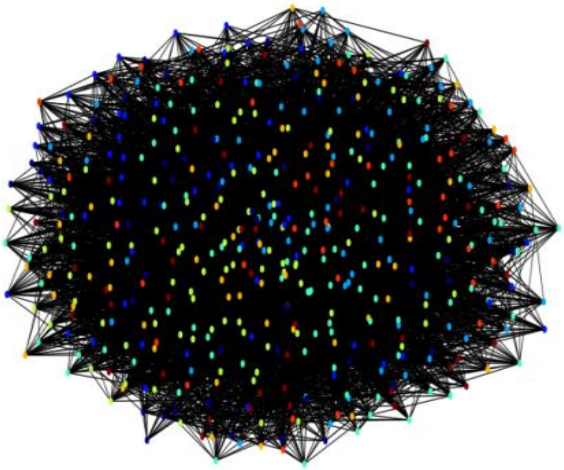

Fig. 5.14b k-Plex Sub Graph Total Network

The size of the subgroup is determined for each sub graph in the network when k=3. The large subgroups size denotes that the friends and followers are more interactive with sub groups of the network. When the size of the subgroup is small, the friends and followers are less interactive within sub groups. Out of 53, there are 33 dense communities and 20 sparse communities detected. The size of the largest community obtained is 1744 and the size of the smallest subgroup is 22.

Also, in-degree of 16 subgroups lies between 501 to 1800 and the in-degree of 17 sub-groups lies between 101 to 500 which indicate that friends and followers are more interactive with other nodes. The in-degree of 21sub-groups lies between 20 to 100, which show less

interaction with other nodes because it is a very popular node in the network. The high out-degree of 12 subgroups lies between 101 to 250. High out- degree value of 14 subgroups suggests more interaction from the outer node to these nodes. For other 27 subgroups, the out-degree lies in the range of 20 to 60. The results for 15 k-plex subgroups are presented in Table XX. The degree measures of k-plex subgroups are evaluated using the k-plex algorithm and the results for a sample of 20 subgroups are presented in Table XXI. The in degree and out the degree of all 53 k-plex subgraphs of the sample input network is illustrated in Fig.5.15.

Table XX Size of k-Plex Sub-Groups when k=3

Subgroup	Size of Sub-Groups
Amitabh Bachchan	405
Aneesh Gautam	421
Blades of Glory	522
Bunty Sajdeh	698
Cristiano Ronaldo	700
ESPNcricinfo	72
Gary Kirsten	770
Harbhajan Singh	779
Neha Dhupia	1210
praveen kumar	784
pankaj arora	78
Ishant Sharma	807
Ramakarthik	823
vinita jain	1683
ESPNcricinfo	850
Cristiano Ronaldo	1584
praveen kumar	1307
Gautam Gambhir	920
Neha Dhupia	928
praveen kumar	1234

Table XXI In-Degree and Out-Degree of the k-Plex Subgroups

Subgroups	In-degree	Out-degree
k-plex1	199	20
k-plex2	12	76
k-plex3	63	37
k-plex4	11	43
k-plex5	171	73
k-plex6	18	72
k-plex7	49	52
k-plex8	120	73
k-plex9	17	38
k-plex10	14	80
k-plex11	126	28
k-plex12	27	71
k-plex13	188	40
k-plex14	16	44
k-plex15	124	21
k-plex16	37	63
k-plex17	150	42
k-plex18	26	68
k-plex19	23	78
k-plex20	115	48

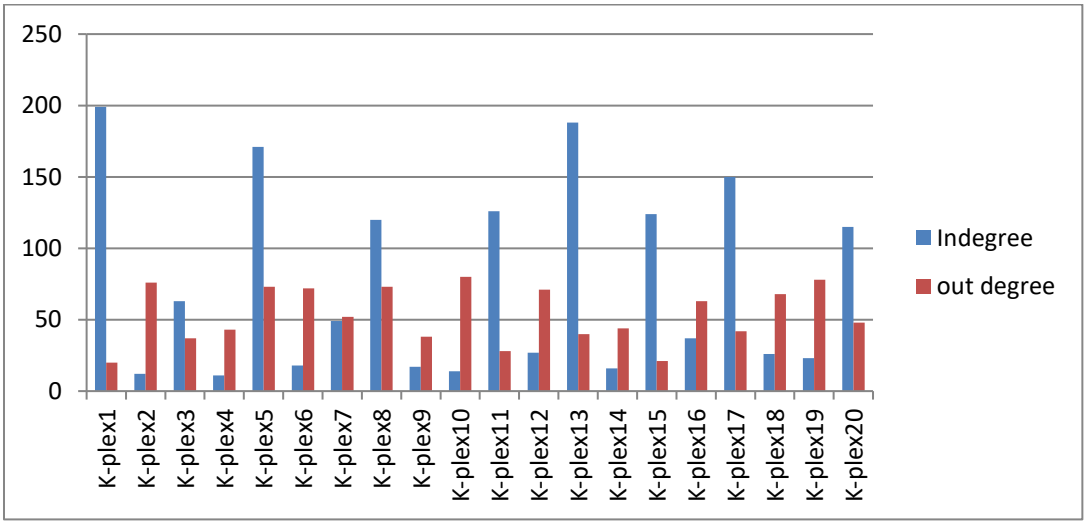

Fig. 5.15 In-Degree and Out-Degree of Maximal k- Plex Sub Graphs

Comparison of Maximal k-clique, k-core, and k-plex

The comparative analysis of these three subgraph based community detection algorithms was done with respect to various factors like a number of subgroups, membership

distribution, size of the communities, modularity, in-degree, out-degree. The comparative results are shown in Table XXII.

Table XXII Subgraph Analysis for k-clique, k-core and k-plex

Algorithm (k=3)	**Number of Subgroups**	**Dense Subgroups**	**Sparse Subgroups**	**Highest In-Degree**	**Highest Out-degree**	**Size of the Largest Community**	**Size of the Smallest Community**	**Modularity Score**
k-Clique	135	78	57	1438	189	1746	20	0.31
k-Core	150	82	68	1572	126	1684	16	0.45
k-Plex	53	33	20	1364	80	1744	22	0.23

From this subgroup analysis of sports person network, it is found that the k-clique algorithm discovered 135 sub-communities in the network in which 78 are dense and 57 are sparse subgroups. The sub-groups of a community detected based on the k-core value in the maximal k-core algorithm were 150 out of which 82 are dense and 68 are sparse subgroups whereas the maximal k-plex algorithm detected 53 different sizes of sub-community. The maximal k-core modularity score of 0.45 is much higher than the scores achieved by maximal k-clique and maximal k-plex. The maximal k-plex algorithm has yielded very less modularity score.

Also, the highest in-degree 1438 and highest out-degree 189 were obtained by maximal k-clique whereas maximum in-degree 1572 and out-degree 126 were observed for k-core when k=3. Similarly, the highest in-degree of 1364 and highest out-degree of 80 have been reported by the k-plex algorithm. The results of computational experiments indicate the effectiveness of the subgroups and the framework used.

Findings

From the exhaustive empirical analysis, the following interpretations are drawn.

- The maximal k-core algorithm detected subgroups based on the k-core value from the twitter network. The k-core size of 3 delivered more number of dense communities and sparse communities. Also the size of the sparse community is less in case of k-core. Therefore the k-core algorithm depicts higher communication between the nodes.
- The k-plex establishes the intractability of the communities for every fixed k as it is a graph-theoretic relaxation of cliques and confirms higher interaction between friends and followers.
- The maximal k-clique shows more number of strong communities as the degree of the communities detected by k-clique is higher than maximal k-core and k-plex.

- The maximal k-core algorithm yielded high modularity score which ascertains the better community detection quality.
- Among all three subgraph algorithms, k-core establishes superiority community detection measures.

SUMMARY

The application of subgroup analysis based on maximal k-clique, k-core and k-plex algorithm for detecting sub-communities from networks has been demonstrated in this chapter. The methodology of the approach and the corresponding experiments carried out on the real-time twitter network of sports person have been elucidated with tables and figures. Various measures evaluated and interpretations drawn from the examination were also discussed in this chapter. The next chapter is intended for overlapping community detection based on clique percolation algorithm on twitter data.