International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020

Detection of Sub-Community Graph in N-Community Graphs using Graph Mining

Bapuji Rao, Sarojananda Mishra

Abstract: Detection of sub-graphs in community graphs is an important task and useful for characterizing community graphs. This characterization leads to classification as well as clusterings of community graphs. It also leads to finding differences among a set of community graphs as well as buildings of indices of community graphs. Finally, this characterization leads discovery of knowledge from sub-graphs. This proposed approach of detection of a sub-community graph from a group of community graphs using simple graph theory techniques. So, that knowledge could be discovered from the sub-community graph detected in a set of community graphs. The proposed algorithm has been implemented with two examples including one benchmark network and observed satisfactory results.

Keywords : community graph, community adjacency matrix, sub-community adjacency matrix, sub-community graph.

I. INTRODUCTION

Discovering a frequent sub-graph from a group of graphs is said to be a graph pattern. So that these sub-graphs are useful for building community graph indices, classification and clustering, and finding differences among a group of community graphs. The discovery of frequent sub-community graph problems leads discovery of frequent sub-community graphs in a group of community graphs. So that these frequent sub-community graphs are useful in data analysis and data mining for similarity search in databases of community graph, clustering, classification of community graphs, indexing of community graphs, etc. This proposed and revised algorithm is for the detection of a sub-community graph in 'n' number of community graphs using graph mining [13].

II. LITERATURE FINDINGS

AGM is a frequent sub-structure mining algorithm proposed by [6], which joins two frequent graphs of (Size-k) with the same Size number of vertices in a sub-graph.

Apriori-based frequent mining was proposed by [1], where the search starts from bottom to up. Discovering frequent subgraphs from a large graph database adopts the Apriori property proposed by [7]. The algorithm path-joining which joins paths was proposed by [14]. An algorithm for mining of frequent graphs from the isomorphic and symmetric graph was proposed by [2]. A novel algorithm, gSpan is used for frequent pattern mining from a graph datasets proposed by [15]. The identification of anomalous sub-graphs in an entity-relationship graph was proposed by [4]. An algorithm that detects of anomalies in all three types of possible changes in graphs i.e., labels modifications, insertions of vertex/edge, and deletions of vertex/edge was proposed by [3].

The proposed algorithm is the revised version of [13]. The idea of detection of sub-graph was adopted from [12] where it detects the common village sub-community graph.

The existing sub-structure mining methods [2, 7, 15] are quite different from the proposed technique [13]. In the proposed technique, the sub-community graph should be known at the beginning and treated as input on 'n' numbers of community graphs for detection. However, the technique is based on graph theory's matrix comparison.

III. PROPOSED ALGORITHM

The revised algorithm has three numbers of phases. The 1st phase reads and stores all the 'n' numbers of names of datasets in CommunityFileName[ ]. The datasets have the details of the communities' viz., the graph number, total numbers of communities, the community numbers, and the pair of community numbers for the edge. The 1st row has graph number, the 2nd row has the number of communities, the 3rd row has community numbers, and the 4th row onwards has pair of numbers for edge i.e., "from-community-number" to "to- community-number". Then it reads the sub-graphs dataset which contains the details of the communities. The 1st row contains the number of communities; the 2nd row contains community numbers, and from 3rd row onwards has pair of community numbers for edge i.e., "from-community-number" to "to-community-number" of the sub-community graph.

The 2nd phase starts verifying the sub-community graph in 'n' numbers of community graphs. The Detection( ) is called 'n' numbers of times for detection of sub-community graph. For every call of Detection( ), it passes a dataset from CommunityFileName[ ] and "SubGraph.Txt" for verification. In Detection( ), the community graph's dataset and sub-community graph's dataset are represented as matrices CM[ ][ ] and SM[ ][ ] respectively. Then the sub-procedure ColumnCompare ( ) which enables to detect SM[ ][ ] in CM[ ][ ]. If SM[ ][ ] is found in CM[ ][ ] then it returns a value 1; otherwise returns a value 0. So, the procedure ColumnCompare( ) returns a value 0 or 1 and assigns to the array, FlagValue[ ].

Revised Manuscript Received on February 15, 2020.
* Correspondence Author
Bapuji Rao*, Department of CSE, Biju Patnaik University of Technology, Rourkela, India. Email: bapuji.research@gmail.com
Sarojananda Mishra, Department of CSEA, Indira Gandhi Institute of Technology, Sarang, India. Email: sarose.mishra@gmail.com

Finally, FlagValue[ ] has 'n' values of 1 or 0 of 'n' community graphs.

The 3rd phase is to display the detected community graph's adjacency matrix. The array FlagValue[ ] is called 'n' numbers of times. When array FlagValue[ ] has 1 value then the procedure DetComGraph( ) is called for displaying the adjacency matrix of the community graph where the sub-community graph has been detected. The proposed and revised algorithm has complexity **O(n^2)**.

A. Algorithm SCGraphDetection( )

Algorithm Convention [5]

NC: To assign total numbers of community graphs.

CommunityFileName[NC]: Array to hold 'NC' numbers of names dataset files.

FlagValue[NC]: To store 0 or 1.

SubGraph.Txt: Sub-community graph's dataset file name.

CGN: To assign the community graph number.

NOC1: To assign the total numbers of communities of community graph.

NOC2: To assign the total numbers of communities of the sub-community graph.

```
{
   read(NC);
   for p:=1 to NC
     { read(CommunityFileName[p]); }
      // to detect sub-community graph
      for p:=1 to NC
        { FlagValue[p]:= Detection(CommunityFileName[p],
                                          "SubGraph.Txt"); }
//to display the adjacency matrix of detected community
//graph
  for p:=1 to NC
   { if(FlagValue[p]=1)
          DetComGraph(CommunityFileName[p]);  }
}
```

B. Procedure for displaying of detected community adjacency matrix

Procedure DetComGraph(DataFile)

DataFile: To store the name of file of community dataset.

fc: To store from-community-number.

tc: To store to-community-number.

CM[NC+1][NC+1]: To store adjacency matrix of community graph.

```
{
  open(DataFile);
  read(CGN);
  CM[1][1]:=CGN;
  read(NC);
  for p:=1 to NC
  {
    read(fc);
   CM[1][p+1]:=CM[p+1][1]:=fc;
   }
  while(Not End-Of-File)
  {
      read(fc);
      read(tc);
     for p:=1 to NC {
     // row side detection
        if (CM[p+1][1]=fc) break;   }
     for q:=1 to NC {
        // column side detection
        if (CM[1][q+1]=tc) break;   }
    CM[p+1][q+1]:=1;
  }
  // to display community graph's adjacency matrix
  for p:=1 to (NC+1) {
    for q:=1 to (NC+1) {
          display(CM[p][q]); }  }
  close(DataFile);
}
```

C. Procedure for detection of sub-community adjacency matrix

Procedure Detection(CFileName, SCFileName)

CM[NOC1+1][NOC1+1]: To assign community adjacency matrix.

SM[NOC2+1][NOC2+1]: To assign sub-community adjacency matrix.

```
{
 open(CFileName);
 read(CGN);
 read(NOC1);
 open(SCFileName);
 read(NOC2);
 if(NOC2 > NOC1)  return(0);
 else
 {
   // reading from CFileName
   for p:=1 to NOC1
   {
      read(a);
     CM[p+1][1]:=CM[1][p+1]:=a;
  }
   while(Not End-Of-File)
   {
    read(a);
    read( b);
    for p:=1 to NOC1
    {   // row side detection
       if (CM[p+1][1]=a) break;  }
    for q:=1 to NOC1
    {   // column side detection
       if (CM[1][q+1]=b) break;  }
```

```
CM[p+1][q+1]:=1;
       }
     close(CFileName);
    // reading from SCFileName
     for p:=1 to NOC2
     {
       read(a);
      SM[p+1][1]:=SM[1][p+1]:=a;
     }
    while(Not End-Of-File)
    {
       read(a);
       read(b);
     for p:=1 to NOC2 {
        // row side detection
        if (SM[p+1][1]=a) break;   }
     for q:=1 to NOC2 {
        // column side detection
        if (SM[1][q+1]=b) break;  }
     SM[p+1][q+1]:=1;
    }
    close(SCFileName);
    // to count NOC2
    total:=0;
    for p:=1 to NOC2 {
      for q:=1 to NOC1 {
         if(SM[1][p+1]=CM[1][q+1]) total:=total+1; }  }
    if(total=NOC2)
    {
     for p:=1 to NOC2 {
       for q:=1 to NOC1 {
          if(SM[p+1][1]=CM[q+1][1])
        flag:= ColumnCompare (SM, NOC2, p, CM, NOC1, q);
} }
// SM[ ][ ] and CM[ ][ ] have different community numbers
 return(flag);
}
 else
  // SM[ ][ ] and CM[ ][ ] have different community numbers
 return(0);
 }
}
```

D. Procedure to compare column community numbers with row community numbers

Procedure ColumnCompare (SM, NC2, row2, CM, NC1, row1)

row1: To assign row index of CM[ ][ ].

row2: To assign row index of SM[ ][ ].

```
{
  for c2:=2 to NOC2 {
     for c1:=2 to NOC1 {
    if(SM[1][c2]=CM[1][c1])
       if(SM[row2][c2]≠CM[row1][c1])
            return(0);  }
    }
      return(1);
}
```

IV. EXAMPLES AND EXPERIMENTAL RESULTS

A. Example-I (Village Community Graph)

Social graphs [9, 10, 11, 12] have studied by the authors, and considered as village community graphs in a panchayat. A village has different communities living together and has a relationship among them. Each community is treated as a node with a number as its identifier. Hence a village is said to be a graph with a set of community nodes. Sometimes it is necessary to detect a sub-community graph among the villages in a village community graph. An efficient algorithm has been proposed by the authors who are able to detect those villages' sub-community graphs in 'n' numbers of village community graphs.

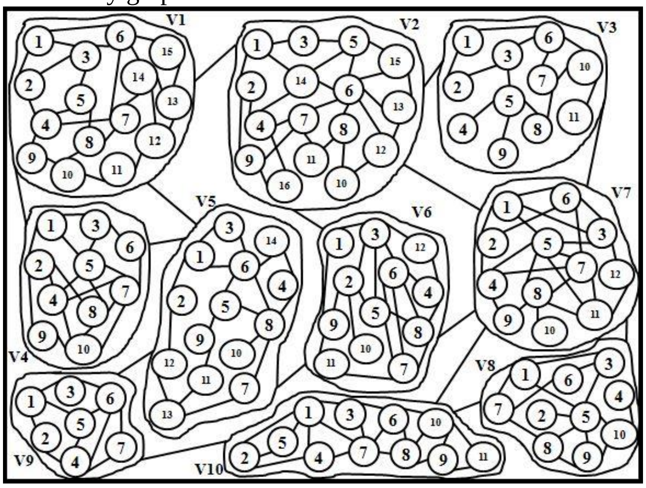
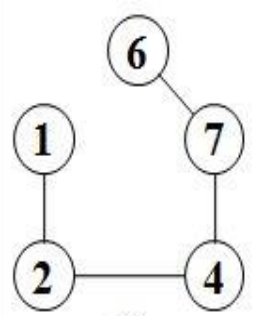


Fig. 1. Village community graph.

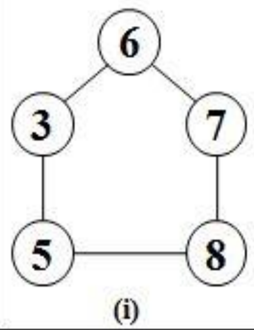
For this purpose, the authors have proposed a village community graph and shown in "Fig. 1", which has ten villages namely V1 to V10. Each village is considered as a community graph. V1 community graph has 1 to 15 communities; the V2 community has 1 to 16 communities, and so on.



	1	2	4	6	7
1	0	1	0	0	0
2	1	0	1	0	0
4	0	1	0	0	1
6	0	0	0	0	1
7	0	0	1	1	0

(ii)

Fig. 2. (i) 1st village sub-community graph. (ii) 1st village sub-community adjacency matrix.



	3	5	6	7	8
3	0	1	1	0	0
5	1	0	0	0	1
6	1	0	0	1	0
7	0	0	1	0	1
8	0	1	0	1	0

(ii)

Fig. 3. (i) 2nd village sub-community graph. (ii) 2nd village sub-community adjacency matrix.

The authors wish to detect two village sub-community graphs shown in "Fig. 2" and "Fig. 3" from "Fig. 1". The path of the 1st village sub-community graph is 1-2-4-7-6 which has no cycle i.e., the 1st and the last nodes are different. Having no cycle exists in the village sub-community graph; its existence in the village community graph can have only one cycle. Such a village community graph can be considered as the presence of the input village sub-community graph. The village sub-community graph having more than one cycle in a village community graph must not be considered. Due to the criteria, the 1st sub-community graph has successfully detected in village community graphs V1, V2, V4, V7, and V9, and shown in "Fig. 4". But the village community graph V10 which has the 1st village sub-community graph in it has three numbers of cycles such as 1-2-4-1, 1-2-4-7-1, and 1-4-7-1. Therefore the village community graph V10 has been rejected.

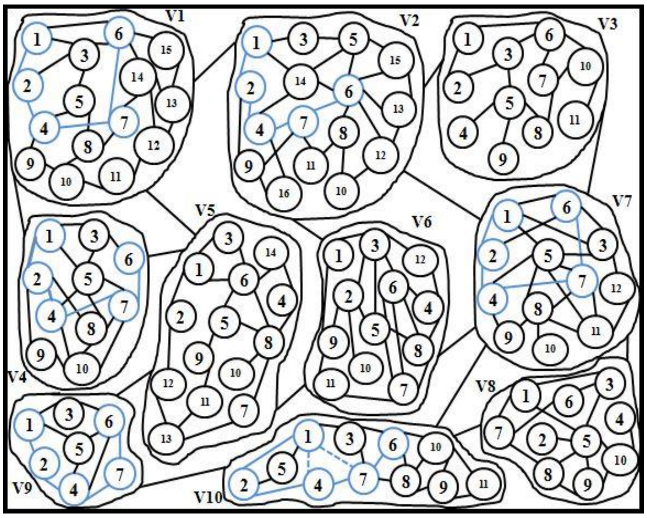


Fig. 4. 1st village sub-community graph detected in communities V1, V2, V4, V7, and V9.

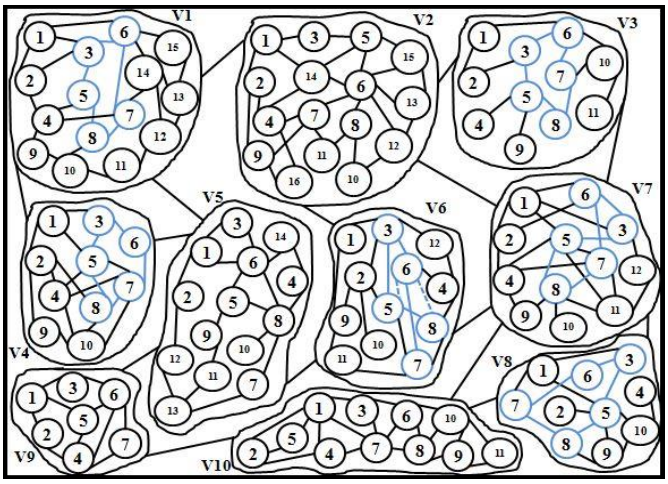


Fig. 5. 2nd village sub-community graph detected in communities V1, V3, V4, V7, and V8.

Similarly, the path of the 2nd village sub-community graph is 3-5-8-7-6-3 which has a cycle i.e., the 1st and the last nodes are the same. Having a cycle exists in the village sub-community graph; such a village sub-community graph's existence in a village community graph can contain only two numbers of cycles. Such a village community graph can be considered as the presence of the input village sub-community graph with a cycle. The village sub-community graph with more than two cycles in a village community graph must not be considered. Due to the criteria, the village sub-community graph 3-5-8-7-6-3 has successfully detected in village community graphs V1, V3, V4, V7, and V8 respectively and shown in "Fig. 5". But the village community graph V6 contains the 2nd village sub-community graph. Though, the 2nd village sub-community graph has five cycles such as 3-5-6-3, 3-5-8-6-3, 6-7-8-6, 6-7-8-5-6 and 5-6-8-5. Hence V6 has been rejected.

B. Datasets

Ten community graph dataset files "V1.TXT" to "V10.TXT", and two datasets for sub-community graphs "SUB-1.TXT" and "SUB-2.TXT" were created by the authors. The dataset files "V1.TXT", "SUB-1.TXT", and "SUB-2.TXT" are shown from "Fig. 6" to "Fig. 8".

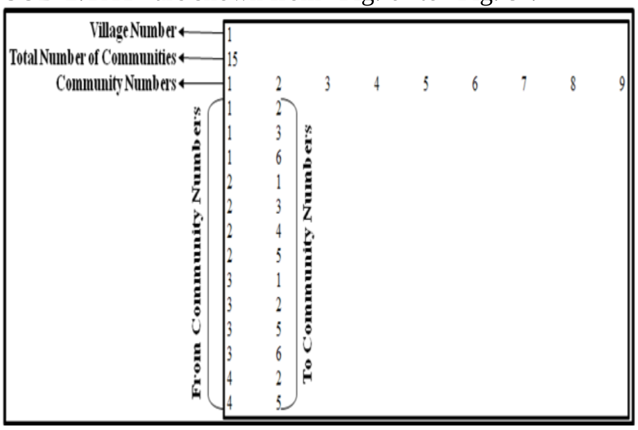


Fig. 6. Dataset of village community graph V1.

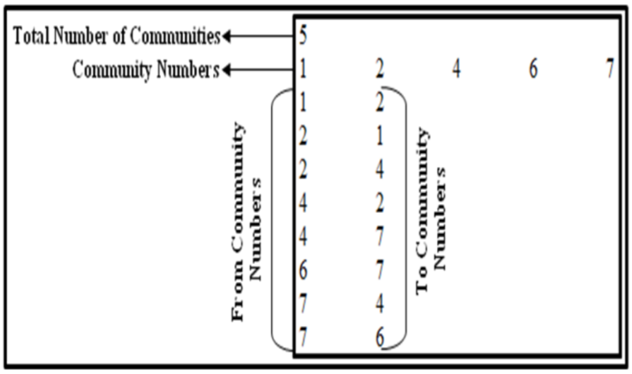


Fig. 7. Dataset of 1st village sub-community graph.

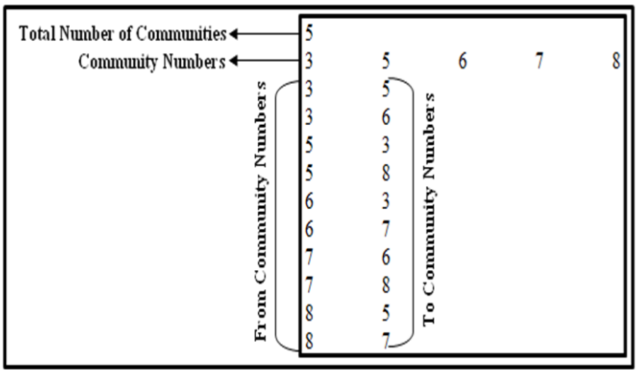
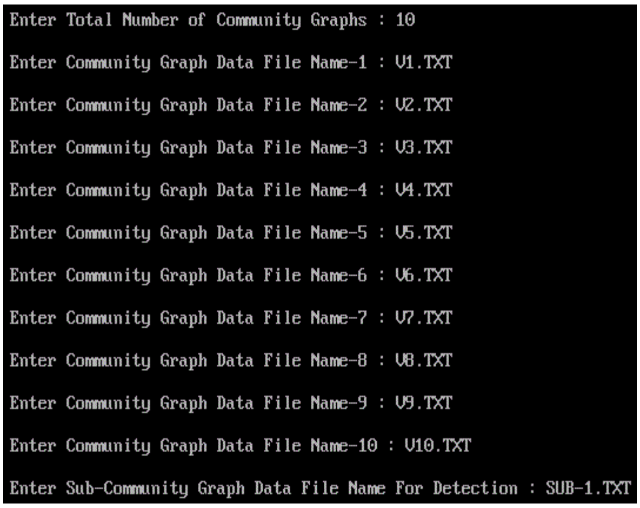


Fig. 8. Dataset of 2nd village sub-community graph.

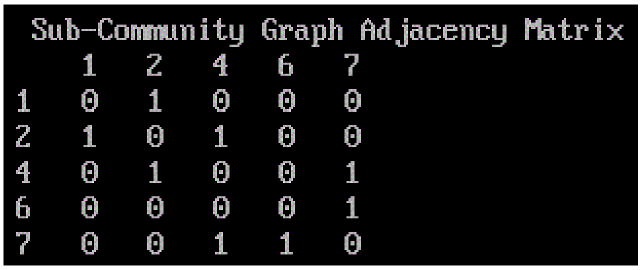
C. Result-I

To detect the 1st village sub-community graph, input the dataset "SUB-1.TXT" and 10 village community graphs datasets from "V1.TXT" to "V10.TXT" to the algorithm which is shown in "Fig. 9". Village community graphs V1, V2, V4, V7, and V9 have successfully detected the 1st village sub-community graph which is shown from "Fig. 10" to "Fig. 15" respectively.



```
Enter Total Number of Community Graphs : 10
Enter Community Graph Data File Name-1 : V1.TXT
Enter Community Graph Data File Name-2 : V2.TXT
Enter Community Graph Data File Name-3 : V3.TXT
Enter Community Graph Data File Name-4 : V4.TXT
Enter Community Graph Data File Name-5 : V5.TXT
Enter Community Graph Data File Name-6 : V6.TXT
Enter Community Graph Data File Name-7 : V7.TXT
Enter Community Graph Data File Name-8 : V8.TXT
Enter Community Graph Data File Name-9 : V9.TXT
Enter Community Graph Data File Name-10 : V10.TXT
Enter Sub-Community Graph Data File Name For Detection : SUB-1.TXT
```

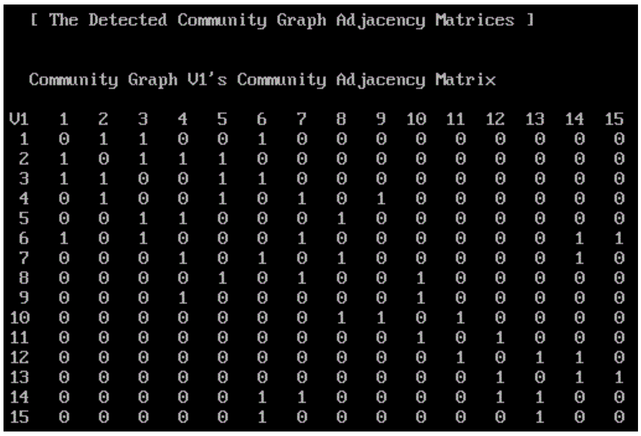
Fig. 9. The input of datasets 10 village community graphs and 1st village sub-community graph.



Sub-Community Graph Adjacency Matrix

	1	2	4	6	7
1	0	1	0	0	0
2	1	0	1	0	0
4	0	1	0	0	1
6	0	0	0	0	1
7	0	0	1	1	0

Fig. 10. 1st village sub-community adjacency matrix.

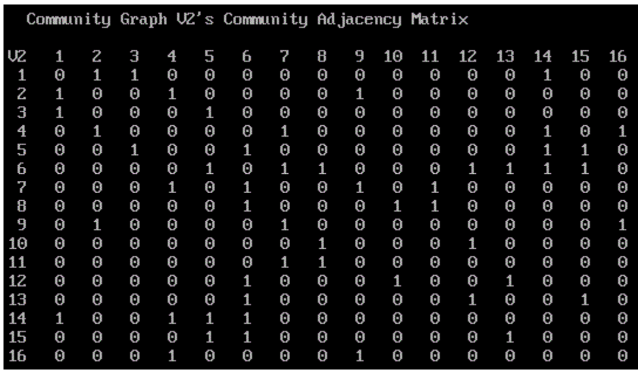


[The Detected Community Graph Adjacency Matrices]

Community Graph V1's Community Adjacency Matrix

V1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
2	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0
3	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0
4	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0
5	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
6	1	0	1	0	0	0	1	0	0	0	0	0	0	1	1
7	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0
8	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0
9	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
10	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0
11	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
13	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
14	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0
15	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0

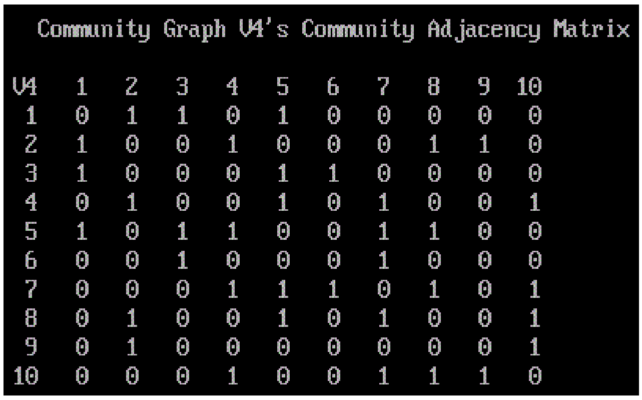
Fig. 11. Detected community graph V1's adjacency matrix.



Community Graph V2's Community Adjacency Matrix

V2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0
2	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
3	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1
5	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0
6	0	0	0	0	1	0	1	1	0	0	0	1	1	1	1	0
7	0	0	0	1	0	1	0	0	1	0	1	0	0	0	0	0
8	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0
9	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
11	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
12	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0
13	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0
14	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0
16	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0

Fig. 12. Detected community graph V2's adjacency matrix.



Community Graph V4's Community Adjacency Matrix

V4	1	2	3	4	5	6	7	8	9	10
1	0	1	1	0	1	0	0	0	0	0
2	1	0	0	1	0	0	0	1	1	0
3	1	0	0	0	1	1	0	0	0	0
4	0	1	0	0	1	0	1	0	0	1
5	1	0	1	1	0	0	1	1	0	0
6	0	0	1	0	0	0	1	0	0	0
7	0	0	0	1	1	1	0	1	0	1
8	0	1	0	0	1	0	1	0	0	1
9	0	1	0	0	0	0	0	0	0	1
10	0	0	0	1	0	0	1	1	1	0

Fig. 13. Detected community graph V4's adjacency matrix.

Community Graph V7's Community Adjacency Matrix

V7	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	1	0	1	1	0	0	0	0	0	0
2	1	0	0	1	0	1	0	0	0	0	0	0
3	1	0	0	0	1	1	0	0	0	0	0	1
4	0	1	0	0	1	0	1	0	1	0	0	0
5	1	0	1	1	0	0	1	1	0	0	1	0
6	1	1	1	0	0	0	1	0	0	0	0	0
7	0	0	0	1	1	1	0	1	0	0	1	0
8	0	0	0	0	1	0	1	0	1	1	1	0
9	0	0	0	1	0	0	0	1	0	0	0	0
10	0	0	0	0	0	0	0	1	0	0	0	0
11	0	0	0	0	1	0	1	1	0	0	0	1
12	0	0	1	0	0	0	0	0	0	0	1	0

Fig. 14. Detected community graph V7's adjacency matrix.

Community Graph V9's Community Adjacency Matrix

V9	1	2	3	4	5	6	7
1	0	1	1	0	1	0	0
2	1	0	0	1	0	0	0
3	1	0	0	0	0	1	0
4	0	1	0	0	1	1	1
5	1	0	0	1	0	1	0
6	0	0	1	1	1	0	1
7	0	0	0	1	0	1	0

Fig. 15. Detected community graph V9's adjacency matrix.

D. Result-II

```
Enter Total Number of Community Graphs : 10
Enter Community Graph Data File Name-1 : V1.TXT
Enter Community Graph Data File Name-2 : V2.TXT
Enter Community Graph Data File Name-3 : V3.TXT
Enter Community Graph Data File Name-4 : V4.TXT
Enter Community Graph Data File Name-5 : V5.TXT
Enter Community Graph Data File Name-6 : V6.TXT
Enter Community Graph Data File Name-7 : V7.TXT
Enter Community Graph Data File Name-8 : V8.TXT
Enter Community Graph Data File Name-9 : V9.TXT
Enter Community Graph Data File Name-10 : V10.TXT
Enter Sub-Community Graph Data File Name For Detection : SUB-2.TXT
```

Fig. 16. The input of datasets of 10 village community graphs and 2nd village sub-community graph.

Similarly, to detect 2nd village sub-community graph, then input 2nd sub-community graph dataset "SUB-2.TXT" and 10 village community graphs datasets from "V1.TXT" to "V10.TXT" to the algorithm and shown in "Fig. 16". The village community graphs V1, V3, V4, V7, and V8 have successfully detected the 2nd sub-community graph which is shown from "Fig. 17" to "Fig. 22" respectively.

Sub-Community Graph Adjacency Matrix

	3	5	6	7	8
3	0	1	1	0	0
5	1	0	0	0	1
6	1	0	0	1	0
7	0	0	1	0	1
8	0	1	0	1	0

Fig. 17. 2nd village sub-community adjacency matrix.

[The Detected Community Graph Adjacency Matrices]

Community Graph V1's Community Adjacency Matrix

V1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
2	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0
3	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0
4	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0
5	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
6	1	0	1	0	0	0	1	0	0	0	0	0	0	1	1
7	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0
8	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0
9	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
10	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0
11	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
13	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
14	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0
15	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0

Fig. 18. Detected community graph V1's adjacency matrix.

Community Graph V3's Community Adjacency Matrix

V3	1	2	3	4	5	6	7	8	9	10	11
1	0	1	0	0	0	1	0	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0	0
3	0	1	0	0	1	1	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0
5	0	0	1	1	0	0	0	1	1	0	0
6	1	0	1	0	0	0	1	0	0	1	0
7	0	0	0	0	0	1	0	1	0	1	0
8	0	0	0	0	1	0	1	0	0	0	0
9	0	0	0	0	1	0	0	0	0	0	0
10	0	0	0	0	0	1	1	0	0	0	1
11	0	0	0	0	0	0	0	0	0	1	0

Fig. 19. Detected community graph V3's adjacency matrix.

Community Graph V4's Community Adjacency Matrix

V4	1	2	3	4	5	6	7	8	9	10
1	0	1	1	0	1	0	0	0	0	0
2	1	0	0	1	0	0	0	1	1	0
3	1	0	0	0	1	1	0	0	0	0
4	0	1	0	0	1	0	1	0	0	1
5	1	0	1	1	0	0	1	1	0	0
6	0	0	1	0	0	0	1	0	0	0
7	0	0	0	1	1	1	0	1	0	1
8	0	1	0	0	1	0	1	0	0	1
9	0	1	0	0	0	0	0	0	0	1
10	0	0	0	1	0	0	1	1	1	0

Fig. 20. Detected community graph V4's adjacency matrix.

Community Graph V7's Community Adjacency Matrix

V7	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	1	0	1	1	0	0	0	0	0	0
2	1	0	0	1	0	1	0	0	0	0	0	0
3	1	0	0	0	1	1	0	0	0	0	0	1
4	0	1	0	0	1	0	1	0	1	0	0	0
5	1	0	1	1	0	0	1	1	0	0	1	0
6	1	1	1	0	0	0	1	0	0	0	0	0
7	0	0	0	1	1	1	0	1	0	0	1	0
8	0	0	0	0	1	0	1	0	1	1	1	0
9	0	0	0	1	0	0	0	1	0	0	0	0
10	0	0	0	0	0	0	0	1	0	0	0	0
11	0	0	0	0	1	0	1	1	0	0	0	1
12	0	0	1	0	0	0	0	0	0	0	1	0

Fig. 21. Detected community graph V7's adjacency matrix.

Community Graph V8's Community Adjacency Matrix

V8	1	2	3	4	5	6	7	8	9	10
1	0	0	1	0	1	0	1	0	0	0
2	0	0	0	0	1	0	0	0	0	0
3	1	0	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0	0	1
5	1	1	1	0	0	0	0	1	1	1
6	0	0	1	0	0	0	1	0	0	0
7	1	0	0	0	0	1	0	1	0	0
8	0	0	0	0	1	0	1	0	1	0
9	0	0	0	0	1	0	0	1	0	1
10	0	0	0	1	1	0	0	0	1	0

Fig. 22. Detected community graph V8's adjacency matrix.

E. Example-II (Dolphin Network)

The dolphin social network has 62 dolphins and there is a frequent association among them, compiled by Lusseau et al. [8]. The 62 dolphins are divided into four communities such as C1, C2, C3, and C4. The communities C1, C2, C3, and C4 have dolphins' codes from 1 to 20, 1 to 7, 1 to 15, and 1 to 20 respectively and shown in "Fig. 23".

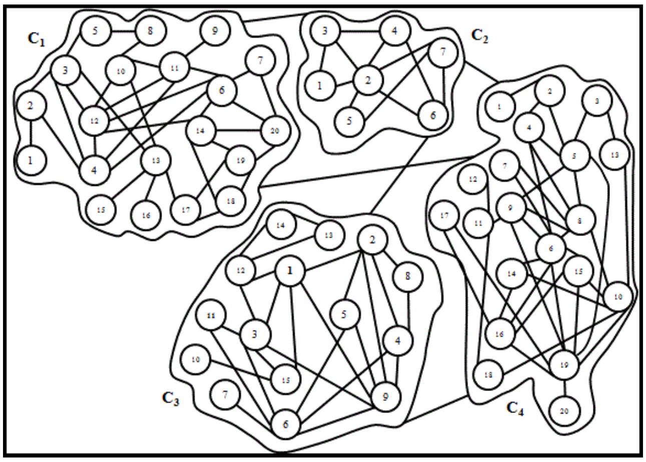
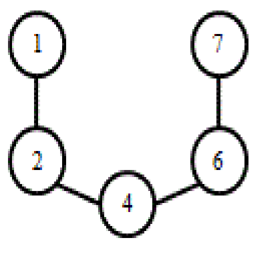


Fig. 23. Dolphin community graph.



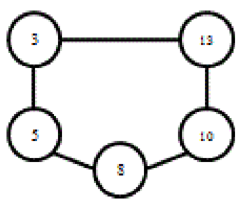
(i)

	1	2	4	6	7
1	0	1	0	0	0
2	1	0	1	0	0
4	0	1	0	1	0
6	0	0	1	0	1
7	0	0	0	1	0

(ii)

Fig. 24. (i) 1st dolphin sub-community graph. (ii) 1st dolphin sub-community adjacency matrix.

The authors wish to detect two dolphin sub-community graphs in the dolphin network shown in "Fig. 24" and "Fig. 25" respectively. The path of the 1st dolphin sub-community graph is 1-2-4-6-7 with no cycle i.e., the 1st and last dolphin nodes are different. Having no cycle exists in the dolphin sub-community graph; its existence in a dolphin community graph can have only one cycle. Such dolphin community graphs can be considered as the presence of the input dolphin sub-community graph. So the dolphin community graph which contains the dolphin sub-community graph having more than one cycle must not be considered. Hence, the 1st dolphin sub-community graph has successfully detected in dolphin community graphs shown in "Fig. 26". But the 1st dolphin sub-community graph with path 1-2-4-6-7 in dolphin community graph C2 has three cycles i.e., 2-4-6-2, 2-4-6-7-2, and 2-6-7-2 respectively. Hence dolphin community graph C2 has been rejected.



(i)

	3	5	8	10	13
3	0	1	0	0	1
5	1	0	1	0	0
8	0	1	0	1	0
10	0	0	1	0	1
13	1	0	0	1	0

(ii)

Fig. 25. (i) 2nd dolphin sub-community graph. (ii) 2nd dolphin sub-community adjacency matrix.

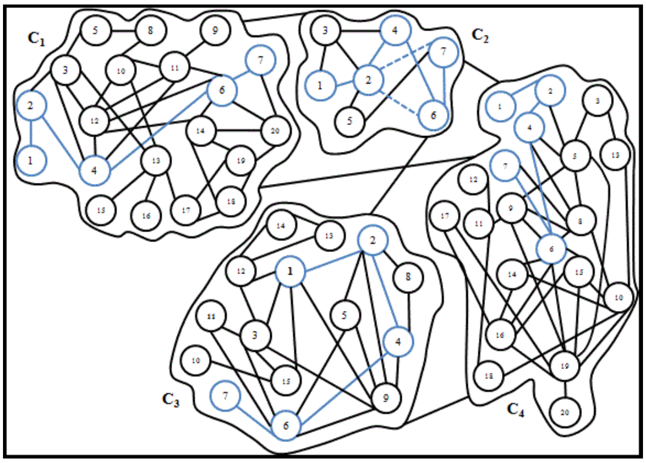


Fig. 26. 1st dolphin sub-community graph detected in communities C1, C3, and C4.

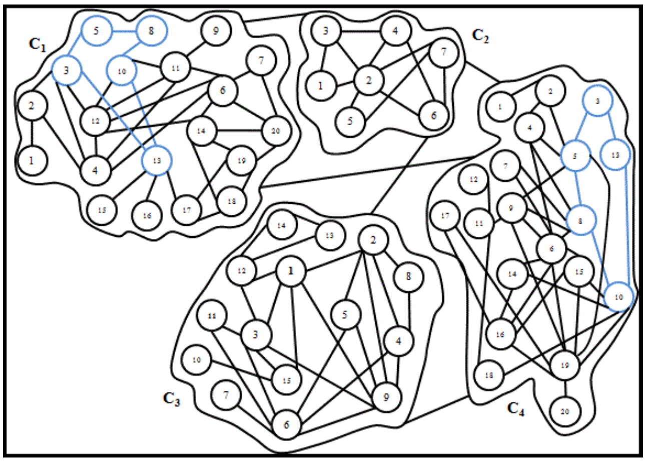


Fig. 27. 2nd dolphin sub-community graph detected in communities C1and C4.

Similarly, to detect the 2nd dolphin sub-community graph has path 3-5-8-10-13-3 with a cycle i.e., the 1st and the last dolphin nodes are the same. Having a cycle exists in the dolphin sub-community graph; such a dolphin sub-community graph's existence in a dolphin community graph can have 2 cycles. Such a dolphin community graph can be considered as the presence of the input dolphin sub-community graph with a cycle. The dolphin sub-community graph having more than 2 cycles in a dolphin community graph must not be considered. So, the 1st dolphin sub-community graph 3-5-8-10-13-3 has successfully detected in dolphin community graphs C1 and C4 and shown in "Fig. 27".

F. Datasets

Four dolphin community dataset files from "DC1.TXT" to "DC4.TXT", and two dataset files of dolphin sub-community "DSUB-1.TXT" and "DSUB-2.TXT" were created by the authors. The dataset files "DC1.TXT", "DSUB-1.TXT", and "DSUB-2.TXT" are shown from "Fig. 28" to "Fig. 30".

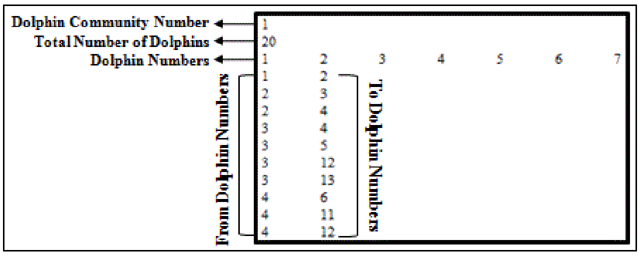


Fig. 28. Dataset of dolphin community C1.

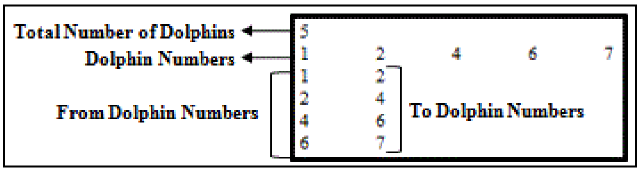


Fig. 29. Dataset of 1st dolphin sub-community graph.

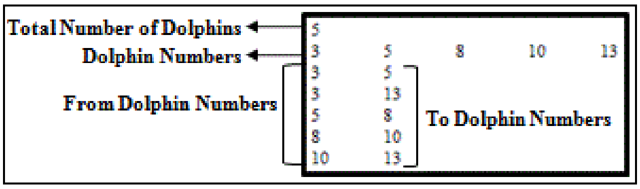


Fig. 30. Dataset of 2nd dolphin sub-community graph.

G. Result-I

```
Enter Total Number of Dolphin Community Graphs : 4

Enter Dolphin Community Data File Name-1 : DC1.TXT

Enter Dolphin Community Data File Name-2 : DC2.TXT

Enter Dolphin Community Data File Name-3 : DC3.TXT

Enter Dolphin Community Data File Name-4 : DC4.TXT

Enter Dolphin Sub-Community Data File Name For Detection : DSUB-1.TXT
```

Fig. 31. The input of datasets of 4 dolphin community graphs and 1st dolphin sub-community graph.

To detect the 1st dolphin sub-community graph in the dolphin community graph, then input the 1st dolphin sub-community graph dataset "DSUB-1.TXT" and 4 dolphin community graphs datasets from "DC1.TXT" to "DC4.TXT" to the algorithm which is shown in "Fig. 31". The adjacency matrix of the 1st dolphin sub-community graph's adjacency matrix has been detected successfully in the adjacency matrices of dolphin community graphs C1, C3, and C4 and shown from "Fig. 32" to "Fig. 35".

Dolphin Sub-Community Adjacency Matrix

	1	2	4	6	7
1	0	1	0	0	0
2	1	0	1	0	0
4	0	1	0	1	0
6	0	0	1	0	1
7	0	0	0	1	0

Fig. 32. The adjacency matrix of the 1st dolphin sub-community graph.

Dolphin C1's Community Adjacency Matrix

C1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0
4	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
5	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
6	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	0	0	1
7	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
10	0	0	1	0	1	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0
11	0	0	0	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0
12	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	0
13	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0
14	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1
20	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0

Fig. 33. Detected dolphin community graph C1's adjacency matrix.

Dolphin C3's Community Adjacency Matrix

C3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	1	0	0	0	0	0	1	0	0	1	0	0	1
2	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0
3	1	0	0	0	0	1	0	0	1	0	1	1	0	0	1
4	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0
5	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0
6	0	0	1	1	1	0	1	0	1	0	1	0	0	0	0
7	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
8	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
9	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
11	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
12	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0
13	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
14	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
15	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0

Fig. 34. Detected dolphin community graph C3's adjacency matrix.

International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020

Dolphin C4's Community Adjacency Matrix

C4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0
6	0	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	0	0	1	0
7	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0
10	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	1	1	0
11	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
13	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0
15	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0
16	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	1	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
18	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
19	0	0	0	0	1	1	0	0	1	1	0	0	0	0	1	1	1	0	0	1
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fig. 35. Detected dolphin community graph C4's adjacency matrix.

H. Result-II

```
Enter Total Number of Dolphin Community Graphs : 4

Enter Dolphin Community Data File Name-1 : DC1.TXT

Enter Dolphin Community Data File Name-2 : DC2.TXT

Enter Dolphin Community Data File Name-3 : DC3.TXT

Enter Dolphin Community Data File Name-4 : DC4.TXT

Enter Dolphin Sub-Community Data File Name For Detection : DSUB-2.TXT
```

Fig. 36. The input of datasets of 4 dolphin community graphs and 2nd dolphin sub-community graph.

Dolphin Sub-Community Adjacency Matrix

	3	5	8	10	13
3	0	1	0	0	1
5	1	0	1	0	0
8	0	1	0	1	0
10	0	0	1	0	1
13	1	0	0	1	0

Fig. 37. The adjacency matrix of the 2nd dolphin sub-community graph.

Dolphin C1's Community Adjacency Matrix

C1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0
4	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
5	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
6	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	0	0	1
7	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
10	0	0	1	0	1	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0
11	0	0	0	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0
12	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	0
13	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0
14	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1
20	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0

Fig. 38. Detected dolphin community graph C1's adjacency matrix.

Dolphin C4's Community Adjacency Matrix

C4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0
6	0	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	0	0	1	0
7	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0
10	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	1	1	0
11	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
13	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0
15	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0
16	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	1	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
18	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
19	0	0	0	0	1	1	0	0	1	1	0	0	0	0	1	1	1	0	0	1
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fig. 39. Detected dolphin community graph C4's adjacency matrix.

Similarly, for the detection of the 2nd dolphin sub-community graph in the dolphin community graph, the authors have input 2nd dolphin sub-community graph dataset "DSUB-2.TXT" and 4 dolphin community graphs datasets from "DC1.TXT" to "DC4.TXT" to the algorithm and shown in "Fig. 36". Then the adjacency matrix of the 2nd dolphin sub-community graph has been detected successfully in the adjacency matrix of dolphin community graph C1 and C4 and shown from "Fig. 37" to "Fig. 39".

The algorithm was written in the C++ programming language and compiled in TurboC++ compiler. The experiment was run on MS-Windows 7 OS with Intel Core I5-3230M CPU + 2.60 GHz Laptop.

V. CONCLUSION

The authors have extended the earlier proposed algorithm for the detection of a sub-community graph in 'n' numbers of community graphs. Its literature survey, example, and the algorithm can be found in [13]. A village community graph and a benchmark dolphin network have been considered as examples for the detection of sub-community graphs. In both cases, the results have been found satisfactory.

REFERENCES

1. Agrawal R., and Srikant R., "Fast algorithms for mining association rules," In Proceedings of 1994 International Conference Very Large Data Bases (VLDB'94), pp. 487–499, Santiago, Chile, Sept, 1994.
2. Borgelt C., "On Canonical Forms for Frequent Graph Mining," In Proceedings of 30th Annual Conference of the Gesellschaft fürKlassifikation e.v., Freie Universität Berlin, 2006, published in Springer's Advances in Data Analysis, Pp. 337-349.C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.
3. Eberle W., and Holder L., "Anomaly detection in data represented as graphs", in *Intelligent Data Analysis* 11 (2007) 663–689, IOS Press.
4. Gupta M., Mallya A., Roy S., Cho J H D., and Han J., "Local Learning for Mining Outlier Subgraphs from Network Datasets", In Proc. of the 2014 SIAM Intl. Conf. on Data Mining (SDM), pp. 73-81, Philadelphia, PA, 2014.
5. Horowitz, Sahni, and Rajasekaran (1998), "Fundamentals of Computer Algorithms," 5, Ansari Road, Darya Ganj, New Delhi-110 002 @ 1998 by W. H. Freeman and Company: Galgotia Publications Pvt. Ltd.

6. Inokuchi, Washio T., and Motoda H., "An apriori-based algorithm for mining frequent substructures from graph data," In Proceedings of 2000 European Symposium Principle of Data Mining and Knowledge Discovery (PKDD'00), pp. 13–23, Lyon, France, Sept 2000.
7. Kuramochi M., and Karypis G., "Frequent subgraph discovery," In Proceedings of 2001 International Conference on Data Mining (ICDM'01), pp. 313–320, San Jose, CA, Nov 2001.
8. Lusseau D., Schneider K., Boisseau O. J., Haase P., Slooten E., and Dawson S. M., "The bottlenose dolphin community of Doubtful Sound features - a large proportion of long-lasting associations", *Behavioral Ecology and Sociobiology* 54, 396-405, 2003.
9. Rao B., Mitra A., and Narayana U., "An approach to Study Properties and Behaviour of Social Network Using Graph Mining Techniques," DIGNATE 2014: ETEECT 2014, pp. 1-7, India, 2014.
10. Rao B., and Mitra A., "A New Approach for Detection of Common Communities in a Social Network Using Graph Mining Techniques," International Conference on High Performance Computing & Application (ICHPCA-2014 IEEE), Bhubaneswar, India, pp. 1-6, Dec 2014, DOI: 10.1109/ICHPCA.2014.7045335
11. Rao B., and Mitra A., "An Approach to Merging of two Community Sub-Graphs to form a Community Graph Using Graph Mining Techniques," 2014 IEEE - ICCIC, Coimbatore, India, pp. 1-7, Dec 2014, DOI: 10.1109/ICCIC.2014.7238392
12. Rao B., Mitra A., and Padhi P., "An Approach to Detect Common Community Sub-Graph between two Community Graphs Using Graph Mining Techniques," ITC - 2015 and CNC - 2015, March 2015, Chennai, India. Published in Advances in Information Technology and Power Electronics, pp. 176 – 185, McGraw-Hill Education (India) Private Limited, ISBN(13): 978-93-392-2161-4 [Print Edition] and GRENZE DIGITAL LIBRARY, DOI:02.ITC.2015.6.9.
13. Rao B., Maharana H S., and Mishra S N., "An Approach to Detect Sub-Community Graph in n-Community Graphs Using Graph Mining Techniques", IEEE Explore Digital Library, 1-6, May, 2017. DOI: 10.1109/ICCIC.2016.7919676
14. Vanetik N., Gudes E., and Shimony S E., "Computing frequent graph patterns from semistructured data," In Proceedings of 2002 International Conference on Data Mining (ICDM'02), pp. 458–465, Maebashi, Japan, Dec, 2002.
15. Yan X., and Han J., "gSpan: Graph Based Substructure Pattern Mining," In proceedings of 2002 IEEE International Conference on Data Mining (ICDM '02), Page 721, IEEE Computer Society Washington, DC, USA, 2002.

AUTHORS PROFILE

Bapuji Rao is currently pursuing a Ph.D. (CSE) from BPUT, Rourkela, Odisha, India. He has received M.Tech (CS) from Berhampur University, Berhampur, Odisha, India. He has published fourteen papers in International Journals of repute, seventeen international conference papers, one national conference paper, five chapters in IGI-Global, USA. He has published a research textbook entitled "Representation of Call-Duration and Social Graph as Multi-Layer Graphs – A Graph Mining Techniques", LAP LAMBERT Academic Publishing, Mauritius. His research area focuses on Graph Mining, Social Network, Data Mining, Opinion Mining, Attributed Graph, and Multi-Layer Graph.

Prof. (Dr.) Sarojananda Mishra is currently working as Professor and Head of the Department of C SEA at Indira Gandhi Institute of Technology (IGIT), Sarang, Dhenkanal, Odisha, India. He has published more than 100 papers in International Journals and National Journals of repute. His research area focuses on Fractal Graphics, Fractal Geometry, Internet Data Analysis, and Web Mining. He has more than 25 years of teaching and research experiences. Five numbers of students obtained PhD degree and more than ten numbers of students are continuing their PhD and M. Tech research work under his guidance.