IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 21 October 2022, accepted 24 November 2022, date of publication 2 December 2022,
date of current version 7 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3226511

RESEARCH ARTICLE

SLPA-IF1: Label Propagation Based Overlapping Community Detection

MONIKA AND VEENU MANGAT, (Member, IEEE)
Department of Information Technology, UIET, Panjab University, Chandigarh 160014, India
Corresponding author: Monika (monikahsp@gmail.com)

ABSTRACT Detection of overlapping communities over a network is imperative due to its applicability in multiple domains starting from geographical to online networks. This paper proposes an effective overlapping community detection method SLPA-IF1. Initially, nodes label initialization is done during pre-processing of data. Label updation and propagation is performed during the evolution phase which consists of selection of listener node, speaker rule and listener rule. Speaker rule is modified to consider the mean of occurring frequency of labels instead of random label selection. We have also proposed a new measure named label specificity for listener rule which is calculated as the mean of occurring frequency of labels minus probability of occurrence of that label. The proposed method leads to more accurate label selection during detection of communities over a network. The run time computation has shown the scalability of the proposed method with respect to increasing network size. For large scale networks, the computing time of the proposed method is less than other state-of-the-art methods.

INDEX TERMS LFR generation, label specificity, overlapping community detection.

I. INTRODUCTION

In the real world, a single individual can be part of different communities at the same time. In terms of network topology, this is known as overlapping communities. In current scenario, people have diversified interest areas, so one individual can belong to different communities [1]. Therefore, overlapping community detection has been considered as key research area over the last decade. We can visualize a social network where one node can belong to multiple communities. For example, one user is part of a ‘‘researcher group’’ as well as a ‘‘friend group’’ simultaneously. Consider Fig. 1 as an example, we assume community structure shown with red color denotes ‘‘researcher group’’ and community structure shown with green color denotes ‘‘friend group’’. Node 5 is shown as part of both the groups depicting overlapping community structure.

Various state-of-the-art methods has been used in literature to detect communities in a social network. Democratic Estimate of the Modular Organization of a Network (DEMON) is used to detect communities in a democratic way where each node vote for communities that are present in its surroundings [2]. Label propagation algorithm is used as base in DEMON core, leading to limited time complexity. In future it aims to detect communities for denser networks also. Cluster Affiliation Model for Big Networks (BigClam) is used to detect densely overlapping communities [3]. It has changed the perspective of sparse overlapping between communities to denser overlapping. C-Finder is a platform independent

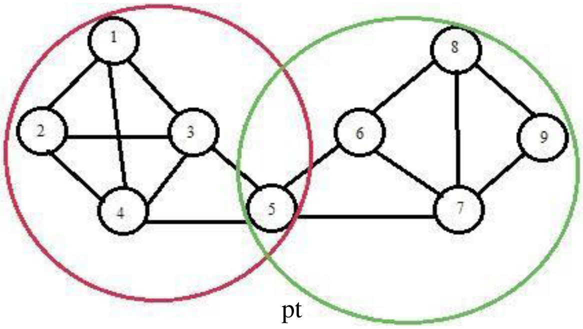


FIGURE 1. Visualizing overlapping communities.

The associate editor coordinating the review of this manuscript and approving it for publication was Rahim Rahmani.

application used to detect overlapping nodes for dense networks [4]. Clique Percolation Method (CPM) is used as a base of algorithm in C-Finder for locating and visualizing overlapping nodes. Community Overlap Propagation Algorithm (COPRA) was introduced by Gregory [4]. It aims to improve label propagation algorithm by assigning multiple community identifiers to a particular node. A node can be part of certain number of communities that is set by threshold parameter in COPRA.

Speaker listener- based propagation algorithm (SLPA) has been widely used for detection of overlapping communities over a network, is an extension of the label propagation algorithm (LPA) [5]. Detection of overlapping community structure in linear run time is the major reason behind extensive use of these methods. In the case of the label propagation algorithm, a single label is randomly assigned to the nodes. It follows an iterative approach for updating the label that has the maximum number in the neighbouring nodes. Nodes containing the same labels belong to the same community within a network. LPA does not require any priori information regarding the number of communities within a network. But this algorithm suffers from a major disadvantage that it produces no unique solution i.e., each run of LPA algorithm results in different community structures because of its random rule for selection and updation of labels.

SLPA method overcomes this random behaviour of LPA while detecting overlapping community structure in a more effective manner. In SLPA, a node can be considered as speaker or listener based on the fact that it is acting as information supplier or receiver. Also, in a given network structure a node can hold multiple labels. Initially nodes are assigned unique labels and one of the nodes is selected as a listener node. SLPA is based upon two rules i.e., speaker rule and listener rule. Speaker rule is used to select labels based on the probability of occurring frequency of those labels. Whereas listener rule is used to accept one most popular label from multiple labels. As SLPA is an iterative algorithm, it converges when a predefined number of iterations are reached.

In SLPA, the above process is used to describe the network structure but the detection of communities is performed using post processing criteria. In post processing, the threshold parameter r, where r $\in$ (0,1), is used. If the likelihood of presence of a label is less than the given threshold range, then the label is deleted, otherwise it is retained in the memory. After the thresholding process, nodes containing similar labels are grouped together to form a community. If a node is having multiple labels, then it is a part of more than one community and this form of node is termed as overlapping node.

In SLPA-MPI, message passing interface is incorporated with SLPA for detection of overlapping communities in a network [6]. Number of processors allocated to the network is equal to the number of partitions within a network. Each processor executes SLPA along with the number of nodes allocated to it. After the completion of each iteration, transfer of the label list from one processor to another processor takes place. In this way, the update of the label list is done. This method is suitable for denser networks because multiple processors help in faster execution of SLPA. Despite showing faster execution, SLPA-MPI suffers from irregular data dependencies due to involvement of multiple processors in computation.

SLPA-OMP also known as SLPA Open Multi-Processing, is aimed to improve system efficiency [7]. It parallelizes various processes in SLPA by using multicore architecture. Optimization of SLPA is done to improve the performance through better memory management. This optimized method suffers from a major drawback of loading the entire graph into the memory. Also, no information is provided regarding how much memory was consumed during parallelization of processes, which needs to be explored further.

A Speaker-listener push-pull propagation approach also known as SL3PA, for overlapping community detection is based on the speaker listener-based algorithm [8]. SL3PA works in three stages: graph splitting, label propagation and community detection. Input graph is divided into various subgraphs with k nodes. All the labels of semi hubs are parallelly propagated in the push label process whereas the pull label process is composed of two substages that are label propagation and updating strategy. In the community detection phase, similar features are extracted from each community and based on Jaccard similarity metric, various communities are merged together. In SL3PA method utilization of memory space is better while maintaining linear time complexity. But various challenges with respect to memory allocation still need to be addressed. Loading of the entire network graph into memory and thresholding process to initialize parameters, needs further attention. We further investigate SLPA while keeping in view following two aspects:

1. After performing node initialization, instead of selecting a label based on its popularity we have modified the listener rule that will select labels based on mean of occurring frequency of labels minus probability of occurrence of that label. It results in more accurate label selection.

2. Entire graph loading can be done at one time which is the basic issue of various optimized versions of label propagation method.

The main contributions of the proposed method in this paper are summarized as follows:

1. We have proposed an improved SLPA-IF1 method for overlapping community detection.

2. A new measure named label specificity for listener rule has been proposed.

3. We empirically evaluate proposed SLPA-IF1 method against state-of-the-art methods for overlapping community detection which shows the outstanding performance of the proposed method.

4. Proposed SLPA-IF1 method yields better results in terms of computational complexity, which further makes it scalable for denser networks as well.

This article is organized as follows: Section II describes the proposed SLPA-IF1 method in detail. Section III includes description of overlapping community detection methods and metrics considered for research work. Section IV comprises of results and discussions of proposed method. Section V concludes this article and provides future research directions.

II. PROPOSED SLPA-IF1 METHOD

A. PRELIMINARIES

A social network is commonly represented in the form of a graph composed of vertices and links. The vertices in the graph indicate social individuals and the links represent the relationship or ties between them. Various types of graphs are broadly classified as directed and undirected graph structure. In an undirected graph traversal between two nodes can be performed in any direction [9]. Directed graphs depict the communication between two nodes with the help of edges represented by arrows. Directed graphs are also known as digraphs [10].

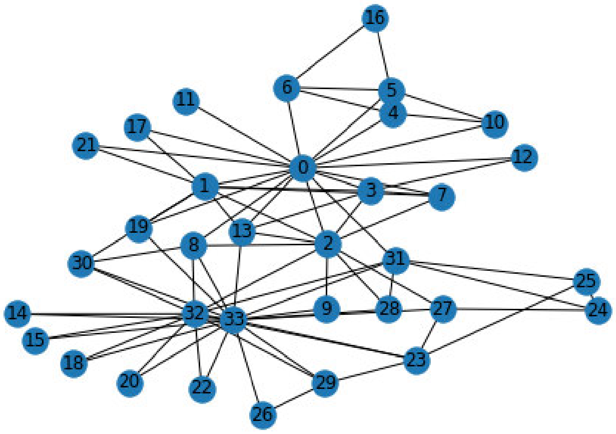


FIGURE 2. Visualization of Zachary's Karate Club network before split.

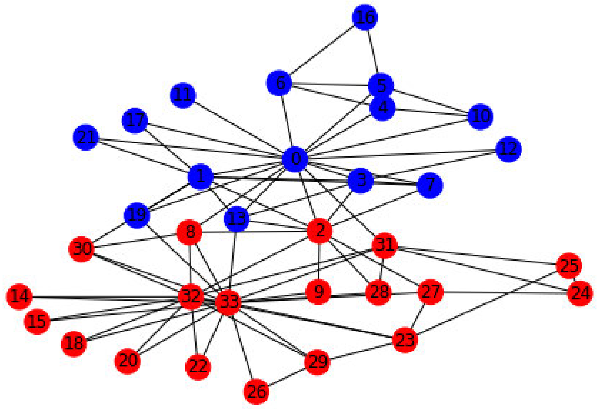


FIGURE 3. Visualization of communities after split in Zachary's Karate Club network.

Zachary's Karate Club network is considered for case study to visualize the concept of community detection. Wayne W. Zachary studied a social network of karate Club for a duration of three years from 1970 to 1972. The network represents the relationship between 34 members of club who interacted outside club. As shown in Fig. 2 each node represents an individual and an edge depicts the relationship between club members. A conflict was observed during study between the administrator "John A" and instructor "Mr. Hi" which split the club into two parts. In this visualization, nodes 0 and 33 are Mr. Hi and John A respectively which clearly show that nodes 0 and 33 are the most connected nodes of the network.

Community detection in network has numerous applications. In the context of the Zachary's karate Club, it predicts which members are with side of Mr. Hi and which member are with side of John A. Community detection algorithms were used for accuracy prediction of members joining Mr. Hi group or John A group. In Fig. 3 blue nodes are members of Mr. Hi and red nodes depicts members of John A.

In the similar manner we have considered LFR benchmark dataset on which proposed SLPA-IF1 community detection algorithm is applied. F1 score of proposed algorithm is evaluated and analyzed by varying number of nodes and mixing parameter which indicates how many nodes are overlapping in other communities.

Mathematically, a graph can be depicted as $G = (N, E)$ where N is the set of nodes and E is the set of edges showing communication between various nodes over a network [11]. Let G be an undirected graph where node set and edge set is represented as $|N(G)| = n$ and $|E(G)| = m$. Definitions of various parameters used in research work is given below:

Degree of Node: It is defined as the number of edges attached to the particular node [12].

Average Degree: Average number of links that each node possess is termed as average degree of the graph. Prerequisite for attainment of average degree is computation of degree of each node present in the graph [12].

Degree Distribution: This parameter is used to examine whether each node in a network has the same degree or some variation is there in links between various nodes. A network can possess a scenario where a huge number of nodes have lesser links and few nodes have more links. Degree distribution quantitatively shows deviation in number of links related to various nodes over a network [13].

Mixing Parameter: It is defined as a fraction of intra community links to each node. Its value ranges from 0 to 1 and it is denoted as mu or μ also [14].

Tau 1: It denotes the power law exponent for the degree distribution of the graph. Its value must be greater than 1 [14].

Tau 2: It denotes the power law exponent for the community size distribution of the graph. Its value is also considered to be greater than 1 always [14].

Max_community: It specifies the maximum size of communities in a graph. It is of variable range and if its value is not externally specified then it is set to n i.e., number of nodes in a graph [14].

B. DATA PRE-PROCESSING

Data preprocessing phase includes generation and linking of the dataset. LFR (Lancichinetti, Fortunato and Radicchi) benchmarks dataset has been used to perform ground truth testing [14]. LFR is named after Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi who created this benchmark dataset. To measure the performance of overlapping community detection methods LFR is the most widely used synthetic network. Earlier Girvan and Newman (GN) benchmark was generally used for community detection but due to its various constraints like consideration of small networks, nodes of the network must have the same degree, same community size leads to generation of realistic LFR benchmark dataset [15]. Heterogeneity in terms of community size and node degree makes LFR the most popular benchmark dataset for community detection in large networks. To generate distinct networks LFR provides a set of parameters like number of nodes, average degree, maximum size of communities, mixing parameter, degree distribution and community size distribution respectively. The parameter set used in research work is presented in table 1. Generation of LFR benchmark is shown in fig. 4.

TABLE 1. LFR benchmark dataset parameters.

Parameter Name / Type	Description	Range/Value
n(int)	Number of nodes	variable range
Average degree (float)	Average degree of nodes in created graph	[0, n]
Tau 1 float)	Degree distribution of the graph	[>1]
Tau 2 (float)	Community size distribution	[>1]
Mixing parameter	Fraction of intra community links to each node	[0,1]
Max_ community (int)	Maximum size of communities	Variable range

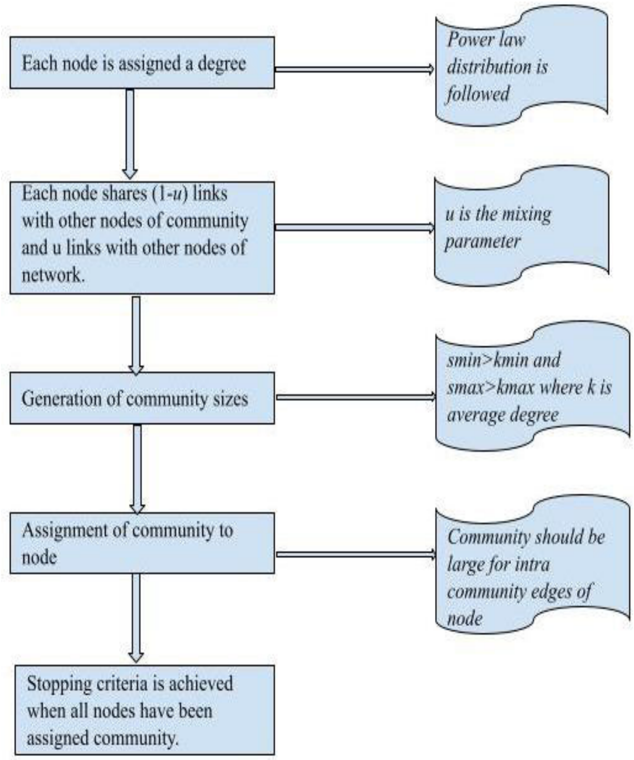

Each node is assigned a degree
Power law distribution is followed
Each node shares (1-u) links with other nodes of community and u links with other nodes of network.
u is the mixing parameter
Generation of community sizes
smin>kmin and smax>kmax where k is average degree
Assignment of community to node
Community should be large for intra community edges of node
Stopping criteria is achieved when all nodes have been assigned community.

FIGURE 4. LFR benchmark generation procedure.

Initially each node is assigned a degree based on power law distribution. It includes minimum degree, maximum degree and average degree of nodes where maximum degree must be equal to number of nodes. Mixing parameter u plays a significant role in benchmark generation indicating the fraction of intra-community edges incident to each node.

Value of the mixing parameter should always range between 0 to 1. Generation of community size is one of the crucial steps in which minimum and maximum community sizes are specified in accordance with power law distribution. Generation of community sizes continues till their sum is equal to n i.e., number of nodes. Next steps include assignment of the communities to the nodes in an iterative manner. Initially none of the nodes is assigned any community. Community assignment process starts by randomly assigning a community to a node. But it must satisfy the condition that community size should be greater than the internal degree of the node. In case a node does not meet above said criteria, it remains unassigned till the next iteration. This criterion is repeated in all iterations until all nodes have been assigned a community.

C. WORKING MODEL OF SLPA-IF1 METHOD

Based on a speaker listener-based algorithm an improved F1(SLPA-IF1) method has been proposed to improve the performance of overlapping community detection. Our proposed method overcomes the problem of loading the entire graph which was the basic issue in various optimized versions of SLPA methods discussed above. Also, improved F1 score has been achieved for better detection of overlapped communities by applying novel techniques. This section describes the working model of the SLPA-IF1 method.

Pre-processing Level: As shown in fig. 5, in the three-layer working model of SLPA-IF1, pre-processing is the first level in which a dataset is loaded into the memory. We have used the LFR benchmark dataset with the number of nodes varying from 1000 to 80,000 respectively. Next step is to initialize nodes by assigning a unique label to each node. Identification of each node can be done with the help of a label assigned to it.

Evolution level: In the second phase label updation and propagation process is performed in an iterative manner. Initially a random node is selected as a listener node. All the neighbors of the selected node will send labels based on specific speaking rule. Instead of selecting a label randomly,

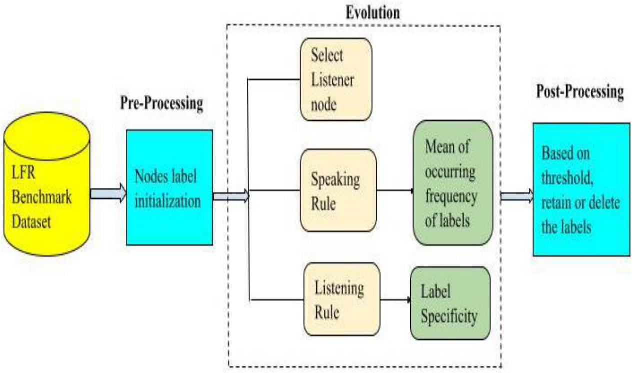


FIGURE 5. Proposed speaker listener propagation algorithm-improved F1 model.

we have considered the mean of occurring frequency of labels as a speaking rule. Listener node will accept labels from various labels on the basis of listener rule. Label specificity is the new measure we proposed instead of selecting a label based on its popularity. It is calculated as the mean of occurring frequency of labels minus probability of occurrence of that label. It helps in more accurate label selection among various nodes connected to it. This process is repeated for a certain number of iterations which was 20 in classic SLPA method. Irrespective of network size, SLPA used to converge after performing these number of iterations.

Post-Processing: In the post-processing phase, a decision regarding retaining a label or deleting a node label is made. It is based on the threshold range, r∈(0,1). A label is deleted if the probability of its occurrence is lesser than specified threshold range otherwise it is retained. After the process of thresholding, remaining nodes are combined to form a community. A node can have more than one label, so in this case it can belong to multiple communities resulting in overlapping nodes. So, the process of community detection is performed when post processing of information is performed. The aforementioned proposed process of SLPA-IF1 is presented in algorithm1 for detecting overlapping communities. Visualization of LFR benchmark with 1000 nodes and after applying SLPA-IF1 is shown in fig.6. Node color coding is used to differentiate among various communities.

III. METHODS AND METRICS USED

A. METHODS USED

The baseline for the experimentation that has been done is explained in this section. We have done the comparative study with works from the literature such as Ego-Network, walktrap and label propagation methods to evaluate and compare the performance of overlapping community detection methods. Among various methods for overlapping community detection, Ego-Network is selected as it identifies significant overlapping in a network and has been widely used for the analysis of online social networks like Facebook, Twitter etc. Label propagation algorithm is the classic algorithm and being the base of SLPA we have selected it for empirically analyzing the performance of our proposed SLPA-IF1 method. Another overlapping community detection method that we have considered for research work is walktrap as it provides

Algorithm 1 Proposed SLPA-IH approach

```
Input: G' {preprocessed graph}; t {number of iterations};
r {threshold (0,1)}
Output; Co (detected overlapped communities}
[nodes _ni] = load_network();
# phase 1: nodes label initialization
for(i=1→ n) do
    node_memory=[ ]
    memory.add(ni);
    memory[ni] memory;
 # phase 2: iterative process
for(t=1→T)do
    Nodes_List←Nodes.shuffle();
end for
# Speaker Rule
    Accepted label=mean(ocf_lb)
for(i= 1 →n) do
    labellist(lb)=speakers().speakersRule();
# Listener Rule
    Accepted label = mean(ocf_lb}-p(oc_lb);
    lb = listener.listenerRule(Label list);
# Update Listener Memory
    Listener.memory add (lb);
end for
# phase 3: post-processing
for (I=1—>n) do
    if (frequency_lb ≤ r) then
        delete_label(ni. lb);
        erase_memory(ni.memory);
    end
end for
return (CO);
```

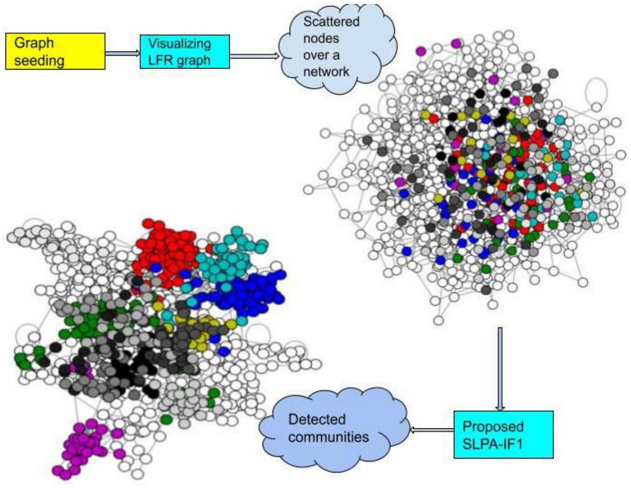


FIGURE 6. Visualization of detected communities with SLPA-IF1.

the best tradeoff between quality and run time even for large networks.

Ego-Network: Ego-Network consists of two main terminologies where ego means individual node named as central node [16]. All the other nodes connected to this central node are known as alters. Ego can be a person, group or an organization. Based on an increased level of interaction a series of alters is generally arranged in a sequence where the innermost circle contains those alters that have a very strong relation with ego. The outermost circle contains alters with almost inactive ties. Major reason behind the use of ego-network is that only ties between ego and alters need to be studied instead of entire ties between individuals over a network.

Walktrap: Based on random walks similarity between vertices is measured. It is measured in terms of distance as it can be computed efficiently [17]. If the two vertices belong to different communities then distance is large otherwise it is small if both the vertices belong to same communities. Probability information, $\mathrm{P}^{\mathrm{t}}_{\mathrm{xy}}$ is used to go from vertex x to another vertex y in t steps. While comparing two vertices following points are considered:

- The probability $\mathrm{P}^{\mathrm{t}}_{\mathrm{xy}}$ will be high if both nodes x and y belong to the same community.
- Probability $\mathrm{P}^{\mathrm{t}}_{\mathrm{xy}}$ is influenced by degree of particular node because high degree vertices are generally accessed by the walker more.
- Vertices belonging to the same community tend to see all the other vertices in a similar manner. Assuming x and y in same community we will have $\mathrm{P}^{\mathrm{t}}_{\mathrm{xk}} \simeq \mathrm{P}^{\mathrm{t}}_{\mathrm{yk}}$.

The hierarchical community structure can be obtained by merging the vertices into communities. This concept can be further used while applying hierarchical clustering algorithm [18]. Community structure obtained in hierarchical manner is represented in the form of dendrogram.

Label Propagation: It is one of the most popular methods while detecting communities over a network due to linear time complexity of the algorithm. Label propagation does not require any a priori information regarding size or number communities. The algorithm starts by initializing each node with a unique label. It follows an iterative approach and at each iteration only that label is adopted by a node which is possessed by most of its neighboring nodes. Finally, nodes that are densely connected and having the same label are assumed to form a community. The algorithm converges when all the communities over a network are identified. As it identifies only disjoint communities an extension of label propagation is proposed by Gregory [19] to detect overlap communities also. It is based on the fact that the same label can be part of different communities also.

B. METRICS USED

To evaluate the quality of overlapping community detection methods is a difficult task as each method optimizes a different metric. In literature normalized mutual index (NMI) has been generally used for this purpose [20]. But it suffers from a major drawback of quadratic computational complexity in the number of communities. Due to this reason for empirical evaluation of the proposed SLPA-IF1 method we have used F1 score and run time as metrics. We have applied the above-mentioned metrics to check the performance of SLPA-IF1 for community detection over a network.

F1 score: To evaluate the quality of clusters detected by various overlapping community detection methods F1 score is widely used [21]. It is based on precision and recall parameters and a perfect overlapping is detected when both precision and recall are 1. Mathematically it can be calculated as follows:

$$F1 = \frac{2.precision.recall}{precision + recall} \tag{1}$$

Precision is the fraction of correctly detected clusters out of the total number of detected clusters. Mathematically precision is defined as [22]:

$$precision = \frac{TP}{TP + FP} \tag{2}$$

where TP (true positive) is the number of detected clusters that are correct and FP i.e., false positive is the total number of detected clusters minus true positive. Recall can be defined as the fraction of correctly detected clusters out of the true number of clusters present. Mathematically recall is defined as [22]:

$$recall = \frac{TP}{TP + FN} \tag{3}$$

where FN (false negative) is defined as the number of known clusters that are not matched with predicted clusters.

Run time: To evaluate the scalability of the proposed SLPA-IF1 method we have used run time as another evaluation metric. It shows how long a particular method will take to process some input. It has been used to check the effect of increase in network size on the computational time of various community detection methods. We have calculated run time using performance counter function follows:

$$timeTaken.append(time.perf\_counter() - startTime) \tag{4}$$

Performance of an algorithm with respect to magnitude of its operations can be easily quantified with the help of run time [23]. Lower computational complexity of various overlapping community detection methods is the main focus in literature because still large numbers of methods are possessing quadratic and exponential complexity.

IV. RESULTS AND DISSCUSSION

To implement the proposed method we have used Google Colab, the most popular open-source data science platform which provides a wide variety of setup environments [24]. We have used python 3.7 as a programming language. The main reason for choosing these technologies for research work is due to the fact that most of the libraries and features to quantify the results are supported by these open-source technology stack. Matplotlib module and pyplot are also used to enhanced representation of results [25][26]. The complete detail of the required resources is shown in table 2.

TABLE 2. Technology stack used.

Technology Stack	Feature
Google Colab	Open-source platform
Python 3.7	It provides huge collection of libraries to quantify results
Networkx package	To create and analyze graphs
Matplotlib module	To draw various plots
Pyplot	To enhance representation of plots

A. EXPERIMENTAL DATASET

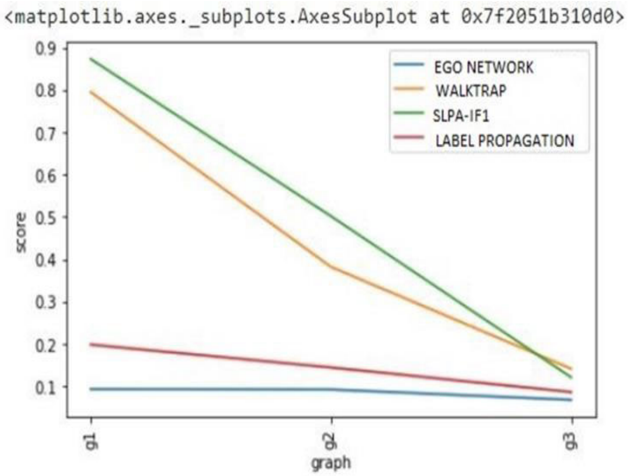
To test the performance of the proposed algorithm LFR benchmark dataset with ground truth communities is used. LFR generator produce different network graphs with a set of parameters that includes number of nodes, average degree, Tau1 and Tau2 power law exponents, maximum community size, minimum community size and mixing parameter respectively. The statistical information of the dataset used is given in table 3. We have used undirected graph with the number of nodes ranging from 1000 to 80000. Five groups of networks are generated with mixing parameter ranging from 0.2 to 0.8. The reason behind choosing this value is the characteristic of this parameter which results in less clear community structure as the parameter value increases. Majority of the community detection methods have used values of $\mu <$ 0.8 for better performance during study of complex networks. It is due to the reason that when $\mu = 0$ all links of nodes are within community links and if $\mu = 1$ all edges are between nodes that belong to different communities. Tau1 and Tau2 variables specify degree distribution and community size distribution of graph respectively and power law distribution is followed to set their values. Minimum community size is not explicitly specified as it is set to the minimum degree of nodes in the graph.

TABLE 3. Statistical information of dataset.

Networ k	No. of Node s	Averag e Degree	Tau 1	Tau 2	Max Comm -unity	Mixing Param -eter
LFR 1	1000	5	2	1	10	0.2-0.4
LFR2	10000	10	2	1	30	0.2-0.4
LFR3	50000	20	3	1.5	60	0.2-0.4
LFR4	80000	25	4	2	100	0.2-0.4
LFR5	80000	25	4	2	100	0.5-0.8

B. RESULTS AND DISCUSSIONS

Empirical evaluation of SLPA-IF1 method on LFR benchmark dataset is done using F1 score and run time as evaluation metrics. From LFR N1-LFR N4, three graphs g1, g2 and g3 were generated. Mixing parameter μ for g1 is 0.2, for g2 μ = 0.3 and for g3 μ = 0.4. In fig.7 for graph g1, proposed SLPA-IF1 method has shown highest F1 score value followed by Walktrap method. For both SLPA-IF1 and Walktrap methods, F1-score drops significantly for graph g2 and g3 due to increasing value of mixing parameter. But F1-score for generated graph g2 is still higher than all other methods which finally converge for graph g3. It indicates accuracy of overlapping community detection methods decreases when more intra community edges incident to each node.

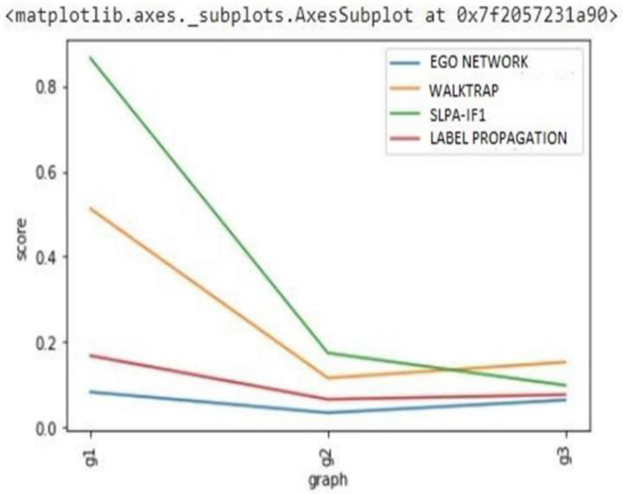


```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2051b310d0>
```

FIGURE 7. F1- score plot for LFR N1 with 1000 nodes.

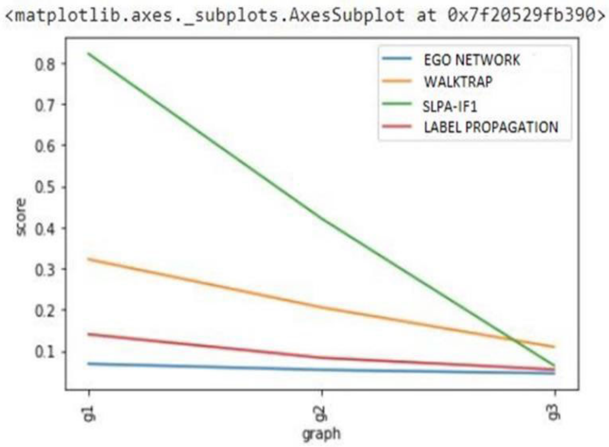
LFR benchmark graph with 10000 nodes in fig.8 depicts how increasing the number of nodes results in lower F1-score as compared to a benchmark graph with 1000 nodes. In LFR N2 three graphs g1, g2 and g3 are generated with mixing parameters 0.2, 0.3 and 0.4 respectively. For g1, performance of the walktrap method drops significantly due to more nodes whereas SLPA-IF1 has still shown significant results in terms of F1-score. With increase in μ the complexity of the network increases which makes it difficult to reveal community structure. Due to this F1-score value of all methods drops for graphs g2 and g3. In fig.9 a denser LFR N3 graph with 50000 nodes shows how proposed SLPA-IF1 outperforms all the other methods significantly. Though the accuracy in terms of F1-score decreases and tends to converge for all methods in graph g3 due to the increase in the value of mixing parameters.

The F1 score achieved for LFR N4 with 80000 nodes is shown in fig.10. Graphs g1, g2 and g3 are generated by taking mixing parameters 0.2, 0.3 and 0.4. As the link fraction that connects to other communities increases, performance of all the community detection methods decreases. It is due to the fact that community structure becomes less clear with increase in value of mixing parameter. Though for graph g3 with μ = 0.4, SLPA-IF1 is still relatively showing better results in terms of F1 score. In fig.11, LFR N5 consisting of 80000 nodes with mixing parameter ranging from 0.5-0.8 is shown. Values of the mixing parameter taken are $\mu = 0.5$ for



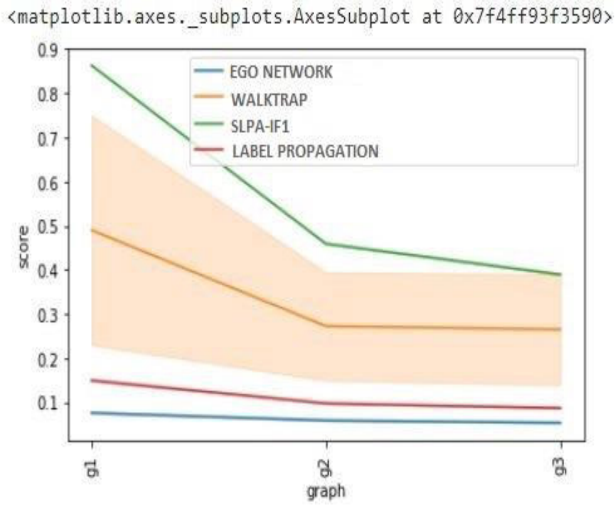
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2057231a90>
```

FIGURE 8. F1- score plot for LFR N2 with 10000 nodes.



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f20529fb390>
```

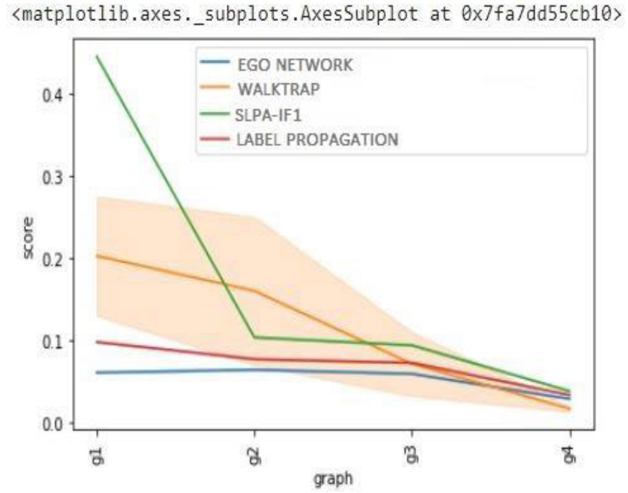
FIGURE 9. F1- score plot for LFR N2 with 50000 nodes.



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ff93f3590>
```

FIGURE 10. F1- score plot for LFR N2 with 80000 nodes.

g1, $\mu = 0.6$ for g2, $\mu = 0.7$ for g3 and $\mu = 0.8$ for g4 respectively. For detecting communities with more accuracy in denser networks SLPA-IF1 has performed better in terms of F1 score even when mixing parameter value is increased to $\mu = 0.5$. Although walktrap has a slightly higher F1 score when $\mu = 0.6$, SLPA-IF1 has shown stable and better results for $\mu \geq 0.7$ as compared to ego network, walktrap and label propagation methods.



```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa7dd55cb10>
```

FIGURE 11. F1- score plot for LFR N5 with 80000 nodes.

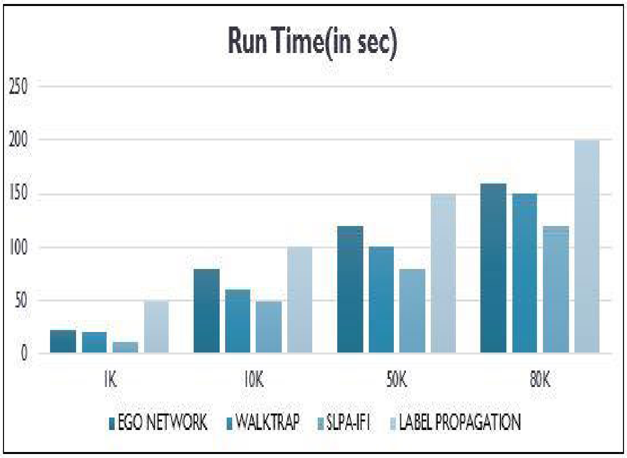


FIGURE 12. Run Time for four methods with nodes 1K- 80K.

Running time of the proposed SLPA-IF1 is evaluated over LFR benchmarks consisting of 1000-80000 nodes. For different datasets we have compared the running time of SLPA-IF1 method with ego-network, walktrap and label propagation method. It can be observed from fig.12 that SLPA-IF1 can be used for larger datasets as it is faster than other considered methods. Our proposed method took less than 150 seconds to process thousands of nodes. With the growth of network size, the running time is increased. However, it can be observed that the proposed algorithm is faster than the other methods. The reason for this is the labels to be propagated are quickly determined, allowing computing time to be shortened. It shows the scalability of the proposed method for denser networks also.

V. CONCLUSION AND FUTURE SCOPE

The proposed SLPA-IF1 method has outperformed other state-of-the-art methods selected for overlapping community detection. The core idea of changing the evolution phase by proposing new measures for listener rule results in more accurate performance of the proposed method while retaining linear computational time. F1 score and linear run time quantifies the better performance of the proposed SLPA-IF1 method for the detection of communities over a network. Even for denser networks, F1 score approaching 0.9 signifies promising results by the proposed method. The statistical significance of the mixing parameter mentioned in results shows how it affects the performance of a particular method even for an equal number of nodes. It indicates that overlapping community detection methods should be used coetaneous with network properties. The run time computation has shown the scalability of the proposed method with respect to increasing network size. For large scale networks the computing time of the proposed method is less than other state-of-the-art methods.

A. FUTURE RESEARCH DIRECTIONS:

As the interactions and information sharing over social networks is an ongoing process, there are still few related open challenges which can be carried out by researchers in future. These future research directions are described as follows:

Efficacy of method with reference to mixing parameter

Mixing parameter is the fraction of intra-community links to each node over a network. As the value of the mixing parameter increases the F1 score of all the methods tends to decrease even for the same number of nodes in a network. Keeping in view the statistical significance of the mixing parameter, the proposed method can be enhanced to provide stable results even for $\mu \geq 0.8$.

Integration with multi-scale visualization tool

Extensive experiments have shown that the proposed method provides good accuracy in terms of F1 score for various graph sizes. While representing detection of overlapping communities for very large and complex networks, the proposed method could be integrated with a multi-scale visualization tool. It would help in visualizing results in a more effective manner.

Implementation of the proposed method for different network structures

Current implementation of the proposed method is used to detect overlapping communities. As we know a social network comprises numerous community structures. In the future researchers can extend this method to detect bipartite communities, fuzzy communities. Also, in edge clustering methods where generated communities are composed of edges rather than nodes extension of this method can be proposed.

DECLARATION: The authors declare that they have no relevant financial or non-financial/ competing interests to disclose in any material discussed in this article.

REFERENCES

[1] C. Rashmi and M. M. Kodabagi, "A review on overlapping community detection methodologies," in *Proc. Int. Conf. Smart Technol. Smart Nation (SmartTechCon)*, Aug. 2017, pp. 1296–1300.
[2] J. J. Whang, D. F. Gleich, and I. S. Dhillon, "Overlapping community detection using seed set expansion," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2013, pp. 2099–2108.
[3] J. Yang and J. Leskovec, "Overlapping community detection at scale: A nonnegative matrix factorization approach," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 587–596.
[4] M. V. Monika, "Diversified overlapping community detection methods in social networks: A survey," *Adv. Eng. Sci. J.*, vol. 54, no. 5, pp. 1943–1960, 2022.
[5] J. Xie, B. K. Szymanski, and X. Liu, "SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process," in *Proc. IEEE 11th Int. Conf. Data Mining Workshops*, Dec. 2011, pp. 344–349.
[6] K. Kuzmin, S. Y. Shah, and B. K. Szymanski, "Parallel overlapping community detection with SLPA," in *Proc. Int. Conf. Social Comput.*, Sep. 2013, pp. 204–212.
[7] Y. Qiao, H. Wang, and D. Wang, "Parallelizing and optimizing overlapping community detection with speaker-listener label propagation algorithm on multi-core architecture," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2017, pp. 439–443.
[8] A. Mahabadi and M. Hosseini, "SLPA-based parallel overlapping community detection approach in large complex social networks," *Multimedia Tools Appl.*, vol. 80, no. 5, pp. 6567–6598, Oct. 2020.
[9] J. Zhang, S. Ding, and N. Zhang, "An overview on probability undirected graphs and their applications in image processing," *Neurocomputing*, vol. 321, pp. 156–168, Dec. 2018.
[10] C. W. Wu, "Algebraic connectivity of directed graphs," *Linear Multilinear Algebra*, vol. 53, no. 3, pp. 203–223, 2005.
[11] V. D. F. Vieira, C. R. Xavier, and A. G. Evsukoff, "A comparative study of overlapping community detection methods from the perspective of the structural properties," *Appl. Netw. Sci.*, vol. 5, no. 1, pp. 1–42, Dec. 2020.
[12] G. Ayyappan, C. Nalini, and A. Kumaravel, "A study on SNA: Measure average degree and average weighted degree of knowledge diffusion in GEPHI," *Indian J. Comput. Sci. Eng.*, vol. 7, no. 6, pp. 230–237, 2017.
[13] E. Müller and R. Peres, "The effect of social networks structure on innovation performance: A review and directions for research," *Int. J. Res. Marketing*, vol. 36, no. 1, pp. 3–19, Mar. 2019.
[14] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New J. Phys.*, vol. 11, no. 3, Mar. 2009, Art. no. 033015.
[15] G. Rossetti, "Graph benchmark handling community dynamics," *J. Complex Netw.*, vol. 5, no. 6, pp. 893–912, 2017.
[16] J. Mcauley and J. Leskovec, "Discovering social circles in ego networks," *ACM Trans. Knowl. Discovery Data*, vol. 8, no. 1, pp. 1–28, Feb. 2014.
[17] P. Pascal and M. Latapy, "Computing communities in large networks using random walks," in *Proc. Int. Symp. Comput. Inf. Sci.* Berlin, Germany: Springer, 2005, pp. 284–293.
[18] H. A. M. Malik, "Analysis of social media complex system using community detection algorithms," *Int. J. Comput. Digit. Syst.*, vol. 11, no. 1, pp. 664–672, 2022.
[19] S. Gregory, "Finding overlapping communities in networks by label propagation," *New J. Phys.*, vol. 12, no. 10, Oct. 2010, Art. no. 103018.
[20] A. Amelio and C. Pizzuti, "Correction for closeness: Adjusting normalized mutual information measure for clustering comparison," *Comput. Intell.*, vol. 33, no. 3, pp. 579–601, Aug. 2017.
[21] Z. Sun, B. Wang, J. Sheng, Z. Yu, and J. Shao, "Overlapping community detection based on information dynamics," *IEEE Access*, vol. 6, pp. 70919–70934, 2018.
[22] Q. Qi, Y. Luo, Z. Xu, S. Ji, and T. Yang, "Stochastic optimization of areas under precision-recall curves with provable convergence," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 1752–1765.

[23] A. D. Popescu, V. Ercegovac, A. Balmin, M. Branco, and A. Ailamaki, ‘‘Same queries, different data: Can we predict runtime performance?’’ in *Proc. IEEE 28th Int. Conf. Data Eng. Workshops*, Apr. 2012, pp. 275–280.

[24] *Google Colab*. (2021). [Online]. Available: https://colab.research.google.com/

[25] *Python*. (2021). [Online]. Available: https://pypi.org/project/networkx/

[26] *Matplotlib*. (2021). [Online]. Available: https://matplotlib.org/

MONIKA received the Master of Engineering degree in information technology from Panjab University, Chandigarh. Currently, she is working as an Assistant Professor in information technology at UIET, Panjab University. She has a teaching experience of more than 12 years. Her research interests include social networks analysis, data mining, and software engineering.

VEENU MANGAT (Member, IEEE) received the Master of Engineering degree in computer science and engineering from the Punjab Engineering College (PEC), in 2004, and the Ph.D. degree in engineering and technology (computer science) from Panjab University, India, in 2016. She is currently working as an Associate Professor in information technology at UIET, Panjab University. She has a teaching experience of more than 17 years. Her research interests include data mining, machine learning, privacy, and security. She is currently a Project Co-ordinator of AICTE SPICES grant for developing model student programming club. She is also a Co-Principal Investigator in Research Project on ‘‘Monitoring of Active Fire Locations and Precision in Allied Agricultural Activities using Communication Technologies’’ funded by the Ministry of Electronics & IT of Government of India worth Rs. 75.75 lakhs. She has also worked on research project titled ‘‘Pedestrian Detection from Thermal Imaging’’ funded by Design Innovation Centre of Ministry of HRD and consultancy project in the area of machine learning. She has edited two international volumes and authored one book in the area of data mining and machine learning. She has successfully guided 22 Master of Engineering dissertations and is currently guiding seven Ph.D. scholars.