

Evaluating Community Detection Algorithms: A Focus on Effectiveness and Efficiency

Srilatha Tokala¹, Murali Krishna Enduri^{1,*}, T. Jaya Lakshmi^{1,2}, Koduru Hajarathaiah¹

¹Department of Computer Science and Engineering, Algorithms and Complexity Theory Lab, SRM University, Amaravati, Andhra Pradesh, INDIA.

²Department of Computing and Software Engineering, Sheffield Hallam University, Sheffield, England, UNITED KINGDOM.

ABSTRACT

Many practical problems and applications are characterized in the form of a network. If the network becomes huge and complex, it becomes very difficult to identify the partitions and the relationships among each of the network's nodes. As a result, the graph is divided into communities and several community detection methods are proposed to associate those communities. The formation of virtual clusters or communities often occurs in networks due to the likelihood of individuals with similar choices and desires associating with one another. Detecting these communities holds significant benefits across various applications, such as identifying shared research areas in collaboration networks, detecting protein interaction in biological networks and finding like-minded individuals for marketing and suggestions. Numerous community detection algorithms are applied in different domains. This paper gives a brief explanation of existing algorithms and approaches for community detection like Louvain, Kernighan-Lin, Girvan Neuman, Label Propagation and Leiden algorithms as well as discusses various applications of community detection. We have evaluated our comparison with six different datasets namely bioclegans, ca-netscience, usair97, webpolblogs, email-univ and powergrid for comparing the efficiency of the methods. The modularity and conductance scores are used to assess the caliber of the partitioned community. A special emphasis on the comparison of these community detection methods is concerned and how the quality resembles and the time taken for its evaluation. We have evaluated all these algorithms and concluded that Louvain and Leiden community detection algorithms are used for effective community division in terms of its structure and time.

Keywords: Network, Community, Community detection and Modularity score.

Correspondence:

Murali Krishna Enduri

Department of Computer Science and Engineering, Algorithms and Complexity Theory Lab, SRM University, Amaravati, Andhra Pradesh, INDIA.
Email: muralikrishna.e@srmmap.edu.in

Received: 26-07-2023;

Revised: 17-01-2024;

Accepted: 01-05-2024.

INTRODUCTION

Complex network research is an essential field of study. Real-world objects are modeled in mathematics as nodes in a graph and their interactions are modeled as edges.^[1] In recent times, complex network research has garnered substantial interest across multiple disciplines, spanning social sciences, biology, computer science and more. These intricate networks offer a robust framework for analyzing and comprehending the intricate patterns of connectivity that exist within a wide range of systems.^[2,3] From social media platforms and biological networks to transportation systems and the internet, complex networks serve as a valuable tool for investigating and deciphering the complex interconnections that underlie these diverse domains. The real-world entities have a natural tendency of existing in groups, causing communities to

form in the corresponding networks. Many analytical task such as link prediction, information propagation etc., can be performed on these networks.^[4] Most of these can be reflected within the community than between/among the communities. Therefore, detecting the communities in a network is a crucial task.^[5] These communities are located using a technique called community detection.

Community detection stands as a pivotal goal in the study of complex networks, aiming to unearth cohesive and densely connected subgroups or communities embedded within the expansive networks.^[6] These communities represent clusters of nodes that exhibit stronger interconnections among themselves than with nodes outside their respective communities. By successfully identifying and delineating these communities, we unlock valuable insights into the modular organization, functional relationships and emergent behaviors that characterize complex systems. The motivations behind community detection in complex networks are diverse and compelling.^[7] Firstly, community detection offers a valuable avenue for comprehending the structure and organization of complex systems across various



DOI: 10.5530/jscires.20250839

Copyright Information :

Copyright Author (s) 2025 Distributed under Creative Commons CC-BY 4.0

Publishing Partner : Manuscript Technomedia, [www.mstechnomedia.com]

scales. By identifying communities, we can unveil groups of nodes that collaborate or possess similar characteristics, enabling us to gain a deeper understanding of the system's functionality and dynamics with enhanced effectiveness.

Community detection is a key element in complex networks. Network nodes occasionally connect to nodes outside the community but have strong connections to other nodes within the community.^[8] A sample network with 11 nodes and 15 edges containing 3 communities is depicted in Figure 1.

Problem Statement

Newman *et al.* define the work of community detection as: Given a complex network, where $G=(V, E)$ has a collection of nodes and edges at time t , the task of community discovery is to produce k communities: V_1, V_2, \dots, V_k where V_i is a subset of V such that the edges in V_i are dense compared to edges between V_i, V_j where $1 \leq i, j \leq k$ and $i \neq j$.^[9]

In this paper we are evaluating different community detection algorithms based on the quality metric modularity score and conductance score for different datasets on different domains. The quality of the network is assessed by using the appropriate community structures.

Community detection is important in the study of biological networks because it is frequently employed to identify functional modules, or collections of genes or proteins, that display coordinated behavior or fulfill similar functions within a biological system. These modules frequently stand in for biologically significant elements and can offer important details about the structure, operation and connections of biological networks. The ability to recognize major groupings or communities of users within the network is one of the most crucial components of social network analysis. These communities typically consist of clusters of individuals who exhibit shared interests, social connections, or affiliations. The identification of these communities provides researchers with valuable insights into the fundamental structure, dynamics and relationships the social network has to offer.

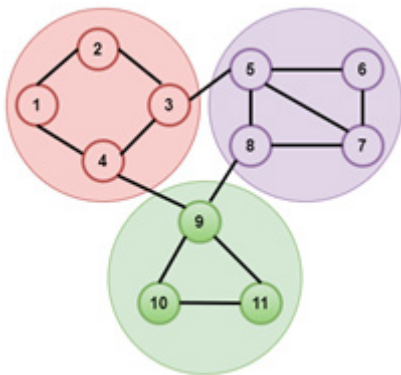


Figure 1: The node which is bounded by the same color belongs to the same community.

Community detection in marketing is employed to identify and comprehend consumer communities within a target market. These communities consist of individuals who have shared interests, preferences, behaviors, or demographics. By utilizing community detection techniques, marketers can acquire valuable insights into the distinct needs, motivations and preferences of these consumer communities. In recommendation systems, community detection is used to find user communities with shared tastes and helps in creating recommendations more accurate. Community detection involves the segregation of individuals who share similar interests or characteristics. While numerous studies in the literature have utilized community detection in recommendation systems, we will provide a few examples from to illustrate this approach.^[10,11]

Community detection in healthcare is valuable for identifying clusters of individuals with similar health conditions, optimizing resource allocation, enhancing personalized medicine, improving patient engagement and designing effective public health campaigns.^[12] It contributes to health research and policy development by considering community-level factors. Overall, community detection enhances healthcare outcomes, resource allocation, personalized medicine, patient engagement and public health strategies.^[13]

Community detection algorithms play a crucial role in categorizing co-authorship data for research paper publications. The bulk of community detection methods assign the same label to all the nodes in the same community. Considering the process of assigning the community labels, the algorithms are classified into two: non-overlapping and overlapping.

Non-overlapping communities have exclusive groupings of the network.^[14,15] Each node belongs to a single community. Whereas, overlapping communities exist when a node can become simultaneously a member of several communities. Overlapping communities enable nodes to have multiple community memberships.^[16,17] Examples of non-overlapping and overlapping community networks are displayed in Figure 1 and Figure 2 respectively. In Figure 1 we can observe that there are three different community structures where there is no overlap between the communities. Each and every node belongs to only one community. Whereas in Figure 2 we observe that there are two community structures and node 3 lies in both the communities which is having an overlapped structure.

The inherent diversity and flexibility of real-world networks, where nodes frequently take part in a variety of social or functional interactions, are captured in overlapping communities. Most community detection techniques currently in use find non-overlapping communities. A different approach is to be followed to find these overlapping memberships.^[18] The following section discusses the research on various community detection strategies.

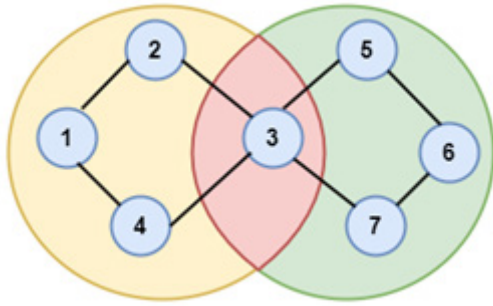


Figure 2: Sample network with overlapping communities, node 3 is member of both the communities.

Applications

Community detection algorithms are used in various fields of research.^[19] Some of the most widely used applications are in bioinformatics,^[20] social networks,^[21] finance,^[22] recommendation systems,^[23] health care,^[24] transportation,^[25] telecommunications,^[26] education,^[27] online platforms and e-commerce^[28] and others.

Community detection helps in **bioinformatics** to identify the group of proteins or genes which interacts closely to each other. By getting this information, the biological genes can be identified. The group of genes that are limited to specific diseases can be identified. In the network structures, the identification of interactions helps in the drug improvement. **Social Networks** are used for identifying clusters of people, which helps the social platforms to build strong connections. Based on the information spread in the social networks, the identification of nodes into groups helps to form an interaction with the influencers. By analyzing the influencers, the marketing strategies can be implemented. The more interactive **financial** markets are identified by community detection which is important for the market analysis. Helps to identify and analyze the risks among financial institutions, which requires monitoring the situations to the financial risks. By applying **recommender systems** to community detection algorithms, businesses may get a deeper understanding of the structure and behavior of their user communities, allowing them to provide more personalized and effective suggestions to users within them.

The spread of diseases within huge population can be identified using community detection for **health care**. Identifying the similar groups helps to get more accurate predictions of disease patterns. Community detection also plays a major role in identifying the mental health conditions within the network individuals. The irregularities and outliers present in the network can also be identified. The detection of communities in the **transportation networks** helps to identify the hubs and central points where the intersection of transport occurs. This helps to design the transport network in an effective manner. In case of

quick emergency access, the communities help to identify the critical situations for emergency planning.

By optimizing the network infrastructure, the communities are identified in **telecommunication** network. The identification of group of users or the devices with same communication patterns assists in identifying the traffic management and allot the resources to improve the network performance. The recommendation courses based on the academic performance or their study habits or their course preferences can be analyzed using communication in **education**. The study groups can be formed based on their similar learning needs. The co-curricular activity planning can also be made with specific communities. These promote meaningful discussions for knowledge sharing and collaboration solving skills.

The online users who share the same interest on the products can be grouped by using communities and recommend the product preferences in the **online platforms**. This enhances the shopping experiences of the users for successful transactions. Incentives or rewards can be offered based on the preferences and behaviors of user's clusters. The identification of fraudulent activities from the users or transactions can be identified using clusters.

Related Work

Over the years, a number of community detection techniques that can find overlapping and non-overlapping communities have been presented.^[29] The most popular existing community detection algorithms are given in Table 1.

The nodes in the communities are more tightly associated to one another than the other nodes in the graph produced by the community detection approaches.

These node communities or clusters frequently share characteristics or capabilities.^[30] By identifying the communities in the graph, community identification techniques are sufficient to support the network's underlying structure and connections between nodes.^[31] The works of Girvan Neuman,^[32] Louvain^[33] and Kernighan-lin^[34] and Leiden^[35] algorithms detect non-overlapping communities. Whereas label propagation,^[36] Infomap^[37] and Walktrap^[38] detect overlapping communities.

Edges with the highest betweenness centrality are regularly detached by the Girvan Neuman algorithm while identifying communities. This algorithm faces issues in scaling to large graphs.^[39] Louvain algorithm has the capability of handling enormous networks having a huge number of nodes and edges that has the quick and scalable capacity to optimize modularity. The algorithm can detect non-overlapping communities. The Kernighan-Lin algorithm is a straightforward and efficient method for finding communities in big graphs. It can detect communities in small to medium-sized graphs that are of high quality.^[40] The improvement of fixing the badly connected

communities that will never become disconnected will be done in Leiden algorithm.

Label propagation technique uses the distribution of node labels to detect communities. This algorithm has the benefit of ease of use and scalability. This algorithm handles big graphs with millions of nodes and is memory-efficient and quick. The initial labeling that is used does not accurately reflect the underlying community structure.^[41] As a way to identify communities, Rosvall *et al.* devised the Infomap technique, which mimics the information flow via a graph. The algorithm recognizes communities in directed and weighted graphs.

If the input data required for the algorithm is imperfect, it may lead to biased outcomes.^[42] The Walktrap algorithm locates communities by calculating the likelihood that random walks will stay in a given community given by Pascal Pons *et al.* and Matthieu Latapy *et al.* The algorithm handles huge graphs effectively of various sizes and forms. The algorithm produces ideal results for overlapping communities.^[43]

Fast Greedy technique is proposed by Aaron Clauset *et al.* in the year 2004 to identify clusters or communities in a network. Its primary objective is to maximize the modularity metric by iteratively merging communities in a greedy manner.^[44] By doing so, it seeks to achieve the highest possible modularity score, thereby revealing the underlying community structure of the network. Leading Eigen Vector algorithm is proposed by Mark E. J. Newman *et al.* in 2013.^[45] It is recognized for its straightforwardness and computational efficiency when it comes to community detection. By leveraging the eigenvector associated with the modularity matrix, the algorithm effectively identifies communities within networks, shedding light on the network's underlying community structure. It offers an insightful approach

to community detection by utilizing the leading eigenvector to uncover distinct community assignments within the network. A brief review of each community detection algorithm strength and weakness is depicted in Table 2.

Community detection algorithms

Using community detection methods, the network is organized into subgroups that share similar characteristics.^[46] These algorithms uncover the hidden structure of networks by identifying clusters of nodes that exhibit strong internal connections and weaker connections with other clusters. This knowledge reveals the network's organizational principles, hierarchical relationships and functional units, shedding light on its overall structure and dynamics.^[47] Five alternative community detection techniques are covered in this section.

Modularity is crucial in these techniques for assessing the quality of the communities formed. In this section, modularity is briefly explained first before the actual discussion of community detection algorithms.

Girvan and Neuman define modularity as^[39] as shown in Equation 1.

$$G = \frac{1}{2E} \sum_{(m,n)} [U_{mn} - \frac{d_m d_n}{2E}] \delta(c_m, c_n) \quad (1)$$

Where, E is the total number of edges in the graph, U_{mn} is the edge weight between nodes m and n. d_m, d_n are the degree of nodes m and n respectively,

c_m, c_n are the communities to which nodes m and n belongs and, $\delta(c_m, c_n)$ is the Kronecker delta function, equal to 1 if $c_m = c_n$ or else 0. The term $[U_{mn} - \frac{d_m d_n}{2E}]$ reflects the difference between the number of edges that are actually seen between the nodes m and n and the number of edges that would be anticipated in the

Table 1: Exploring the Landscape: An Overview of Community Detection Algorithms.

Community Type	Algorithm	Year	Contribution
Non-Overlapping	Girvan Neuman	2002	Computes communities based on edge-betweenness.
	Louvain	2008	Optimizes the modularity by improving the community structure within neighborhoods of nodes.
	Kernighan-Lin	1970 (existed from 2009).	Iterative partitioning approach that minimizes the total edge cut between communities.
	Leiden	2019	Modularity optimization is accomplished by eliminating local maximum traps and assuring more robust community assignments, which provide computational efficiency and scalability for large networks.
Overlapping	Label Propagation	2007	Label updating process, which propagates node labels based on the most common labels among their neighbors.
	Infomap	2008	Use information theory and random walks to identify communities in a network.
	Walktrap	2011	Identifies communities by detecting dense regions of the network where random walks are likely to remain in a given neighborhood.

Table 2: Analyzing Community detection techniques: Strengths vs. Weaknesses.

Community Detection Algorithm	Strengths	Weaknesses
Girvan Neuman Community detection.	The network is decomposed into communities by iteratively removing high-betweenness edges. This algorithm is applicable for weighted and unweighted as well as directed and undirected graphs.	The performance is very sensitive for its size and for the density of the network. The calculation of betweenness centrality for the edges is very expensive.
Louvain Community detection.	Within the short-limited time, Louvain algorithm can handle large networks. Based on the structure of the network the communities are identified without any prior knowledge.	The communities are more dependent on the initial partition. Due to the non-deterministic nature of the network, different results will be produced for different iterations.
Kernighan-Lin Community detection.	If the partition is not for two groups this algorithm is better, as well as get well-defined communities in the network. It improves the quality of the partition by iteratively exchanging the nodes.	The algorithm struggles in defining the community structures for large networks. As it is of only bipartition, this algorithm is not preferred in most of the cases.
Leiden Community detection.	It can handle overlapping community structures which help in analyzing complex communities. The algorithm produces more deterministic results for the given initial conditions.	It mainly focuses on modularity optimization, which is not preferred for all network structures. The memory and computational requirements has become a main drawback which results in multiple resolutions.
Label Propagation Community detection.	This algorithm is simple, scalable and efficient which is suitable for large networks. It can adopt to various community structures.	The algorithm feels vulnerable to assign label at initial stages. Incase if node belongs to more community structures, the algorithm struggles in detecting the communities.
Infomap Community detection.	The algorithm has the capability of identifying local and global community structures with different scales. If a node belongs to more communities, this algorithm can detect the overlapping communities.	It is computationally demand for large networks. Identifying the similar communities are more difficult for the larger structures if their sizes are comparable.
Walktrap Community detection.	The emphasis of local neighborhood is considered for random walks of short lengths. The run time is manageable as the algorithm is scalable and efficient for analyzing large networks.	It is sensitive to noise which leads to detecting irrelevant nodes in the community structures. The choice of selecting parameters requires some practical analysis.

null model, where edges are distributed at random. The values of modularity (G) varies between -1 to $+1$.^[48] If the value is $+1$, the observed number of edges within communities is larger than it would be under the null model and if it is -1 , it is less than it would be under the null model.

Consider an example graph G , with 6 nodes and 7 edges as shown in Figure 3 and a sample community structure of G with two communities.

Community 1: {1, 2, 3}

Community 2: {4, 5, 6}

The modularity score of the given community partition for Figure 3 is computed using Equation 1 as follows:

E =The graph's edge count is 7.

$U_{mn} = 1$ if the nodes m and n have an edge, else it is 0.

The degree of each node is given as $d_1=2, d_2=2, d_3=3, d_4=3, d_5=2$ and $d_6=2$.

If the two nodes m and n are members of the same community, the Kronecker delta function value is 1, otherwise 0.

Now, substituting all these values in Equation 1, the modularity score of 0.510 is computed for the given community partition. The positive modularity score indicates that the connections between the nodes in the same community are tighter than would be predicted by chance.

Conductance is one the quality metric used to assess the quality of the community structures that are evolved by the community detection algorithms by using the quality score.^[49] Conductance is a way to measure how well a group of connected nodes, or community, sticks together. It looks at both the connections within the group and those outside it. The lower the conductance score, the better the community sticks together. The conductance

score ranges between 0 and 1. Conductance is calculated as shown in Equation 2.^[50]

$$\text{Conductance } \phi(c) = \frac{e_b}{2e_c + e_b} \quad (2)$$

Where e_b is defined as the number of edges on the boundary and e_c is defined as the number of edges in c . Let us compute the conductance value for the graph as shown in Figure 3. From the figure, it is observed that there are two communities. The conductance score for community 1 is calculated as follows:

e_b value is 1, as there is only one edge that is passing from the boundary. e_c value is 3, as there are three edges present in community 1. By substituting the values in the Equation 2, the conductance value is computed as 0.142. The similar conductance score will also be observed for community 2.

The following sections discuss five popular community detection algorithms.

Louvain Community Detection Algorithm

The Louvain algorithm is proposed by Blondel *et al.*, in 2008.^[30] In large networks, the Louvain algorithm is frequently used for community detection due to its quick running time. The algorithm used to find communities is displayed in Algorithm 1.

Algorithm 1 Louvain Community Detection Algorithm

Input: A graph $G=(V, E)$, where V represents the vertex set and E denotes the edge set.

Output: A set of communities C , where $C = \cup_i C_i$ and $C_i \subseteq V$. C_i denotes the i^{th} community of G .

Step 1: Form n communities of size 1, each containing a single node from V , where the number of nodes in G is n .

Step 2: Determine the modularity score of C using the Equation 3:

$$\text{Modularity } (C) = \sum_c \left[\frac{E_c}{E} - \left(\frac{d_c}{2E} \right)^2 \right] \quad (3)$$

Where d_c is the degree of community c . The modularity score assesses the effectiveness of the network clustering caused by C .

Step 3: while True do

Shift each node to the nearby community to form new communities C' .

Compute new modularity = modularity(C') using Equation 3.

if new modularity \neq modularity **then**

modularity = new modularity

else

break

end if

end while

Kernighan-Lin Community Detection Algorithm

The other algorithm that is used in detecting the communities is the Kernighan-Lin community detection method invented by Brian Kernighan and Shen Lin in 1970.^[35] This algorithm is mostly preferred for small to medium-sized networks. The procedure followed in detecting communities is shown in Algorithm 2.

Algorithm 2 Kernighan-Lin Community Detection Algorithm

Input: A graph $G=(V, E)$, where V represents the vertex set and E denotes the edge set.

Output: A set of two communities C_1 and C_2 , where $C = \cup_i C_i$ and $C_1 \subseteq V$ and $C_2 \subseteq V$.

Step 1: Randomly divide the nodes of G into communities C_1 and C_2 .

Step 2: Determine the modularity score for each community in the network.

repeat

for each node V in community C_1 **do**

Step 3: Consider shifting node V from community C_1 to C_2 .

Determine the new partition's modularity score.

end for

for each node U in community C_2 **do**

Step 3: Consider shifting node U from community C_2 to C_1 .

Determine the new partition's modularity score.

end for

until no improvement in modularity score is observed.

Girvan Neuman Community Detection Algorithm

The Girvan Neuman (GN) algorithm is also known as the edge_betweenness algorithm to identify communities within

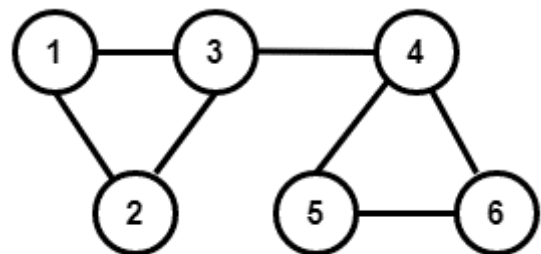


Figure 3: Example graph with 6 nodes and 7 edges to compute modularity and conductance scores.

a network developed by Girvan and Mark Newman in 2002.^[39] In small to medium-sized networks, the Girvan Newman method is frequently used for community detection despite being computationally expensive.^[51] GN algorithm is given in Algorithm 3.

Algorithm 3 Girvan-Neuman Community Detection Algorithm

Input: A graph $G=(V, E)$, where V represents the vertex set and E denotes the edge set.

Output: A set of communities C , where $C= \cup_i C_i$ and $C_i \subseteq V$. C_i denotes the i^{th} community of G .

while G is not fully disconnected **do**

Step 1: Calculate the centrality of each edge in E in terms of edge betweenness using the following Equation 4:

$$B(e)=\sum_{u,v \in V(G)} \frac{\sigma_{uv}(e)}{\sigma_{uv}} \quad (4)$$

Step 2: Remove the edge with high betweenness. This removal divides the network into two or more communities. Let the communities formed be C , where $C= \cup_i C_i$.

end

Note that, the GN algorithm is a top-down approach that iteratively removes an edge with high edge betweenness to divide the network into communities. This procedure can be stopped by computing modularity at each step and stopped when the modularity is high.

Label Propagation Community Detection Algorithm

The graph created by Raghavan, Albert and Kumara in 2007 is utilized to identify the communities using the label propagation approach.^[35] Because of its rapid running time, the label propagation technique is commonly employed for community detection in big networks. The procedure followed in detecting communities is shown in Algorithm 4.

Algorithm 4 Label Propagation Community Detection Algorithm

Input: A graph $G=(V, E)$, where V represents the vertex set and E denotes the edge set.

Output: A set of communities C , where $C= \cup_i C_i$ and $C_i \subseteq V$. C_i denotes the i^{th} community of G .

Step 1: Assign a unique label to each node in G .

Step 2:

repeat

changed \leftarrow false

for each node v in G **do**

neighborLabels \leftarrow empty list

for each neighbor u of v **do**

Append the label of u to neighbor Labels

end for

if neighborLabels is not empty **then**

mostCommonLabel \leftarrow FindMostCommonLabel(neighborLabels)

if mostCommonLabel \neq label of v **then**

Set the label of v to mostCommonLabel

Set changed to true

end if

end if

end for

until changed is false

Step 3: Return the node labels for each node in G .

Step 4: The set of appropriate communities with the same labels is determined.

Leiden Community Detection Algorithm

The Leiden community discovery technique, developed by Traag et al. in 2019, solves a fundamental problem in the famous Louvain algorithm: the tendency to yield weakly linked or even detached communities, especially when run repeatedly.^[35] The Leiden method outperforms Louvain in terms of runtime performance due to its use of a fast local move strategy. The procedure followed in detecting communities is shown in Algorithm 5.

Algorithm 5 Leiden Community Detection Algorithm

Input: A graph $G=(V, E)$, where V represents the vertex set and E denotes the edge set.

Output: A set of communities C , where $C= \cup_i C_i$ and $C_i \subseteq V$. C_i denotes the i^{th} community of G .

Step 1: Initialize the graph G and set the initial partition as each node is its own community.

Step 2:

repeat

for each node V in random order **do**

Step 3: remove V from the current community C_i and calculate the modularity gain for each neighboring community C_j .

Assign V to the community C_j .

end for

until no change in modularity

Step 4:

if modularity gain is positive:

Merge adjacent communities with same label and update the partition.

Calculate modularity of the updated partition.

end if

We have taken all the datasets from the network repository. We have performed several steps to preprocess the data collected from the datasets. (<https://networkrepository.com/networks.php>). The tasks include removing the outliers, handling missing values and transforming the variables. The simulations are run at central processing unit of 11th Gen Intel (R) Core (TM) i9-11900, with CPU running at 2.50 GHz with 64GB RAM of system type 64-bit operating system. Anaconda is used to compute the community structures and create graphical models. The generation of graphs is used by NetworkX package. All the visualization plots are drawn using Origin Pro software.

Some of the limitations that we have undergone during our experimental analysis are the consumption of memory and time are very high compared to the other programming languages with more efficient memory management. The integration with the external tools or libraries is not much compatible with python language. The implementation of python may not fully exploit the underlying optimizations in large networks which impacts the efficiency of the network. Despite of these limitations, python is a powerful language for community detection for small and medium-sized networks. It is essential to note that various algorithms may sometimes converge to different communities, as divergence can occur due to data characteristics, algorithm startup, or ambiguous community boundaries. It is almost 50% of the intersection is present in communities detected by different community detection algorithms.

Time complexity analysis

The analysis of time complexity in community detection methods is crucial for assessing scalability, comparing performance, selecting algorithms, optimizing design and implementing them practically. It enables researchers and practitioners to gauge an algorithm's ability to handle large-scale networks efficiently without overwhelming computational resources. By comparing time complexities, computational efficiency can be evaluated to identify the most appropriate algorithm. Additionally, time complexity analysis guides algorithm designers in optimizing efficiency and estimating computational resource requirements for practical implementation. The following 3 provides information about time complexities of different community detection algorithms.

Implementation

The four techniques for community detection that were discussed in the preceding section are empirically evaluated on four real-world benchmark datasets. This section gives a description of the datasets used for our experimental analysis.

Datasets

We employ six datasets from various areas to experimentally evaluate the community's performance. The statistics for the datasets are shown in Table 3. Each dataset is taken from a different domain with a specific number of nodes and edges in each dataset.

Researchers can examine the structural and functional characteristics of an actual, biological brain network using the biocelegans dataset. Scientists who investigate network theory and empirically study social, biological and technological networks are represented in the Ca-netscience dataset as co-authors. The UsAir97 dataset aids in comprehending the connectivity between airports, locating important hubs and determining the effects of network modifications or disruptions. The Webpolblogs dataset offers insights into the ideological polarisation that exists within the political blogosphere and how opinions are formed.

The email-univ dataset represents the email communication between the members in an institution or a university. It serves as a resource for providing the security and predicting the communication patterns. The PowerGrid dataset is critical for infrastructure planning, allowing engineers to examine grid structure and vulnerabilities to ensure optimal design and maintenance. It enables them to understand how electricity networks are designed and how they may fail, allowing them to strengthen and improve their reliability.

RESULTS AND DISCUSSION

The networks are divided into communities using various methods for community detection. For example, we have shown in Figure 4, how Louvain community detection algorithm has divided into communities for community value 3 for six different datasets. Three different colors are used in the visualizations for

Table 3: Time Complexity analysis of different community detection algorithms.

Algorithm	Time Complexity
Girvan Neuman ^[52]	$O(nm^2)$
Louvain ^[53]	$O(n \log n)$
Kernighan-Lin ^[54]	$O(n^2)$
Label Propagation ^[52]	$O(m)$
Infomap ^[52]	$O(n \log n)$
Walktrap ^[55]	$O(n^2 \log n)$
Leiden ^[56]	$O(n + m)$

Here, n is the number of nodes and m is the number of edges in the graph G.

three different communities. We have used red, blue and green colors for the visualizations. After communities are formed by dividing the networks, the modularity score is computed to determine the quality of those communities. Varying numbers of communities are compared with the modularity score and the time needed to compute the community detection algorithm to display the findings. For all different community detection methods 25 communities are taken, modularity score and time taken for evaluation are compared.

Figure 5 shows the evaluation of modularity score with five different community detection namely Louvain, Girvan Neuman, Kernighan-Lin, Label propagation and Leiden methods on biocelegans, ca-netscience, UsAir97, webpolblogs, email-univ and powergrid datasets. As randomness is involved in all the algorithms, for about 25 iterations the algorithm is processed and the modularity score is computed. It is observed in the figure for the biocelegans dataset, the Girvan Neuman method is performing well and has a drastic increase in the modularity

score from 0.04 to 0.7 which indicates there are more appropriate metabolic reactions between the substrates forms communities. For Louvain and Kernighan-Lin methods, simultaneously the modularity score increases and after some time constant score will be maintained. For this dataset, the co-authorship relations are maintained constant after iterating certain communities. As the increase in the communities, the capacity of the passenger information and the dynamics related to travel are maintained constant. For the Label propagation method, the modularity score behaves differently compared to the remaining methods. There are up and falls in the modularity score for this method and the modularity score is also less compared to all the methods for the biocelegans dataset. It has less interactions provides between the hyperlinks and the webpages that are provided as not suitable community division occurs. For the Leiden method, a constant modularity score is maintained for every iteration of the community, which indicates the method gives the similar score for any community. Girvan Neuman's method is giving

Table 3: Dataset Statistics.

Dataset	Domain	Nodes	Edges	No. of nodes	No. of edges
Biocelegans	Biological	Substrates	Metabolic Reactions	453	2025
Ca-netscience	Collaborative	Authors	Scientific Collaborations	379	914
UsAir97	Transportation	Airports	Flights	332	2126
Webpolblogs	Webgraphs	Webpages	Hyperlinks	643	2280
Email-univ	Communication	Email addresses	Communication between email addresses.	1134	5451
PowerGrid	Technological	Power stations	Transmission lines	10670	22002

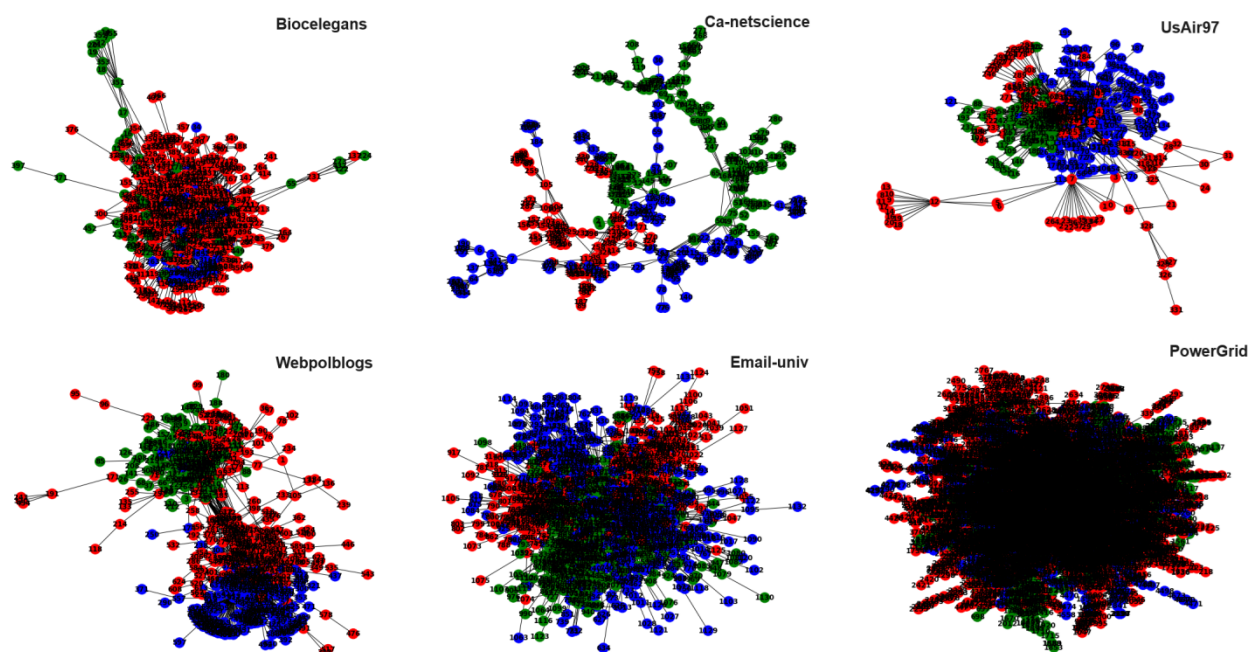


Figure 4: Visualizations showing the Louvain community detection algorithm division for three communities on six different datasets namely Biocelegans, Ca-netscience, UsAir97, Webpolblogs, Email-univ and PowerGrid.

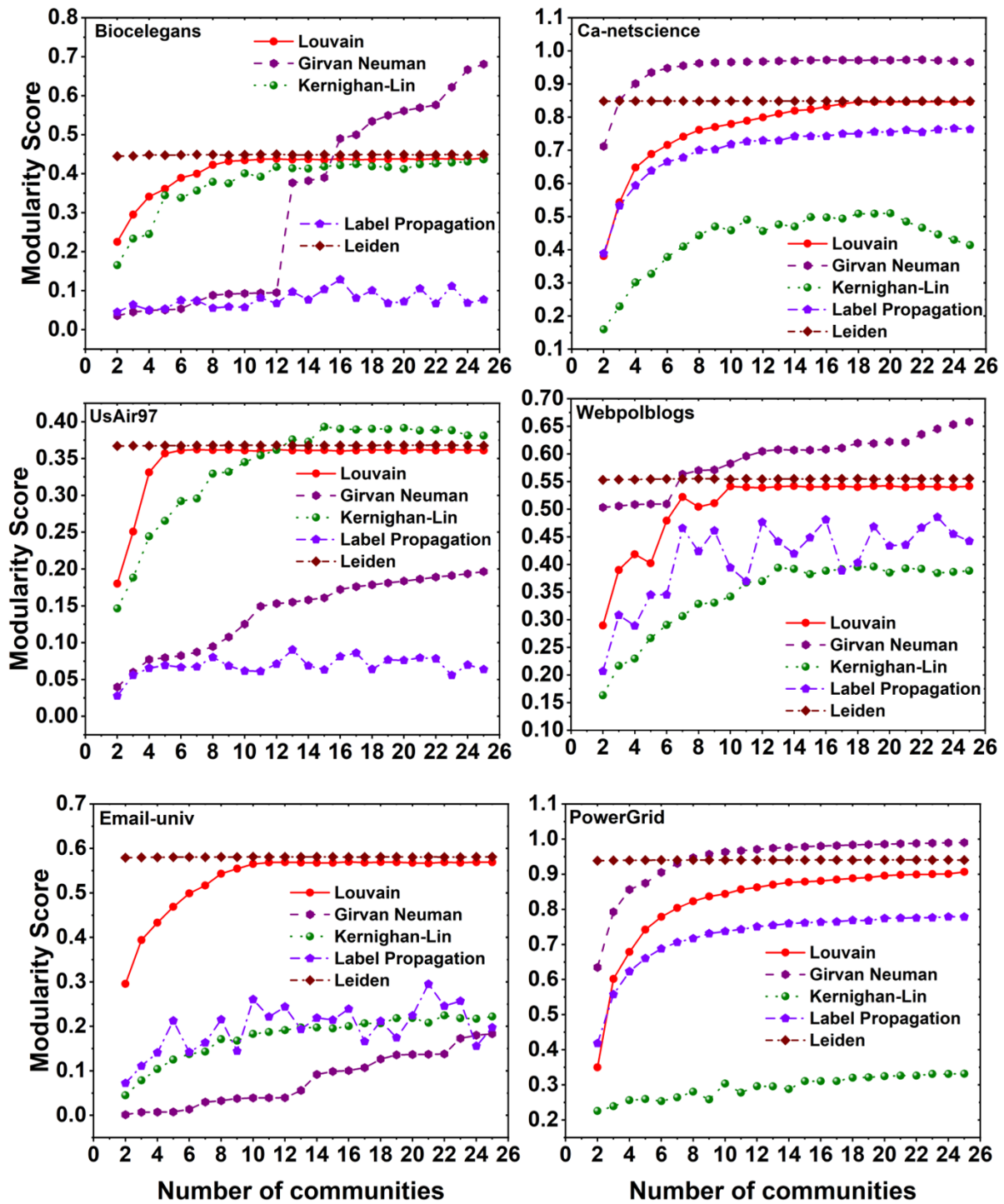


Figure 5: Modularity Score computed for five different community detection methods namely Louvain, Girvan Neuman, Kernighan-Lin, Label Propagation and Leiden for 25 different communities on Biocelegans, Ca-netscience, UsAir97, Webpolblogs, Email-univ and PowerGrid datasets.

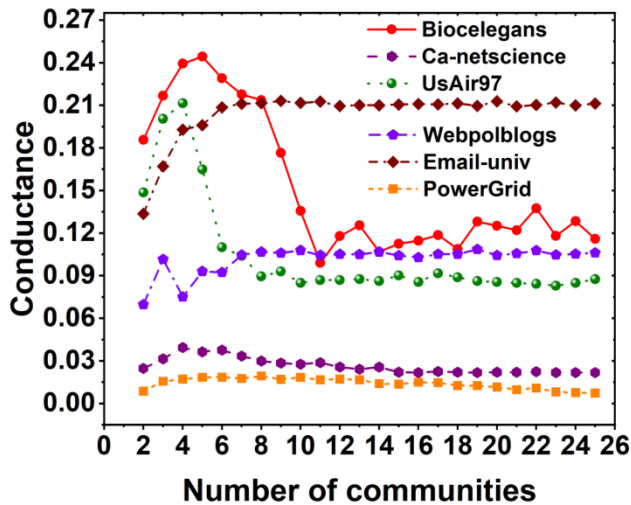
high modularity score with a drastic increase, Louvain and Kernighan-Lin's methods are behaving same for the biocelegans dataset. Label Propagation method is giving less modularity score. Compared to all the methods, Leiden method is giving similar modularity score for every community.

For the Ca-netscience dataset, for all the methods, by adding more communities, the modularity score rises. It is seen that

Girvan Neuman's method is giving a higher modularity score and Kernighan-lin's method is giving a less modularity score. There is only a little bit of change in the modularity score for the Louvain and Label propagation methods by increasing the communities. The leiden method is maintaining the constant modularity score for every community and better modularity compared to the other methods. It is observed in the dataset that all the methods

Table 4: Comparative analysis for the time taken to compute modularity score in seconds for five community detection algorithms on six datasets.

Dataset (↓) / Algorithm (→)	Louvain	Kernighan-Lin	Girvan Neuman	Label Propagation	Leiden
Bioclegans	571.72	14753.55	736.27	1.86	8.21
Ca-netscience	931.95	5143.57	40.23	1.41	5.63
UsAir97	85.09	8781.67	195.77	1.09	6.22
Webpolblogs	467.26	29281.57	719.75	2.08	10.08
Email-univ	560.64	105156.86	2318.95	3.62	13.19
PowerGrid	313387.68	1224371.29	14445.76	24.77	60.27

**Figure 6:** Conductance value computed for Louvain community detection method for 25 different communities on Bioclegans, Ca-netscience, UsAir97, Webpolblogs, Email-univ and PowerGrid datasets.

except Leiden will maintain a constant in the modularity score after a period of increase in the communities. In the ca-netscience dataset by using Girvan Neuman algorithm, more suitable communities occur between the scientific collaborators due to the increase in the modularity score as suitable metabolic collaborations between the authors occur.

In Us Air97 dataset, a higher modularity score is maintained by Leiden method and a lower modularity score is maintained by the label propagation method. A similar modularity score is maintained by the Leiden method for every community. A drastic increase is seen in the Kernighan-lin method as the modularity score increases from 0.14 to 0.37. A similar kind of increase in modularity score is also seen in the Louvain method, but after 6 community partitions the modularity score is maintained constant. There is also an increase in the modularity score behavior for the Girvan Neuman algorithm from 0.03 to 0.17. The label propagation method's modularity score has significant ups and downs and it has not significantly increased. The flights between the airports division is maintained appropriate by using Leiden algorithm in UsAir97 dataset.

For the webpolblogs dataset, for all the methods as we increase the number of communities, the modularity score will rise

Table 5: Time taken to compute conductance score of the Louvain community detection algorithm calculated in seconds for six different datasets.

Dataset	Conductance time taken (in sec)
Bioclegans	288.26
Ca-netscience	1522.46
UsAir97	79.08
Webpolblogs	446.24
Email-univ	1068.61
PowerGrid	382420.37

except for the Leiden method. The Leiden community method is maintaining a similar modularity score while iterating each community with high modularity score. Girvan Neuman's method is also performing well having a better modularity score and the Kernighan-Lin method is having a less modularity score. After 10 community partitions, the Louvain method is having constant behavior as the communities are increased. Along with the increase in the modularity score, there are up and falls for the label propagation method.

In the email-univ dataset, for all the methods we observe that, as the number of communities are increased there will be an increase in the modularity score. Girvan Neuman and Kernighan-lin algorithms are having a gradual increase in the modularity score as the communities increase. There are several ups and falls for the label propagation algorithm in the modularity score as the communities increases. There is a drastic increase in the modularity score for the Louvain algorithm and after community 10 a constant modularity score is maintained. From community 2, a constant modularity score is maintained by Leiden algorithm. Compared to all the algorithms in the email-univ, it is observed that the Leiden algorithm and the Louvain algorithms performance is better. For the email-univ dataset, better communication between the email-addresses is maintained by the Louvain and the Leiden methods.

In the powergrid dataset, for all the methods we observe that, as the communities' increases, the modularity score values also increases. A constant level of higher modularity is seen by Leiden method and the lowest modularity is maintained by the Kernighan-lin method. Except the Kernighan-lin method, all

the other community methods are maintaining a similar kind of modularity as there is an increase in the number of communities. The Girvan Neuman method has an increase in the modularity score from 0.62 to 1.0 as maintaining a constant behavior from 16 communities. The Louvain method has the modularity value from 0.35 to 0.9 having a drastic change in the modularity. The label propagation method has the change in modularity from 0.42 to 0.75 as the number of communities increase. The transmission lines between the power stations can be properly aligned by using the Girvan Newman and the Leiden methods in the powergrid dataset.

It is observed in Figure 5 that compared to all the methods, Leiden and Louvain are having a similar kind of modularity score after iterating the communities in for all the datasets. The Girvan Neuman method is giving high modularity score in all the datasets except for UsAir97. The lower modularity score is maintained by either Kernighan-Lin or Label Propagation in all the datasets. Hence, better community division is provided by Leiden and Louvain community detection algorithms.

Table 4 displays the computation's time in seconds and the modularity score using the five community detection methods. The table shows that the label propagation method is taking less time to compute the modularity score whereas the Kernighan-lin method is taking more time to compute the modularity score. As per the time taken calculation, the label propagation method is taking very less time to compute the modularity score and as the plots are observed it is giving less modularity score for Biocelegans and UsAir97 datasets. The Kernighan-lin method is taking more time to compute and is also giving a less modularity score for Ca-netscience, Webpolblogs and PowerGrid datasets. Girvan Neuman method is giving high modularity score for the four datasets but takes more time than the Louvain method to compute. The Louvain method is taking more time than label propagation and less time than the remaining community detection methods and gives better modularity score computation. It is clearly observed in the table that Leiden community method is giving better modularity score in less time compared to all the other methods in every dataset. If the modularity score is high only, it indicates that the communities are partitioned correctly having a definite network structure. So, it indicates that the Louvain and Leiden method's performance in detecting community structures is well-defined.

Figure 6 shows the evaluation of conductance value for the Louvain community detection algorithm.

On six different datasets namely Biocelegans, Ca-netscience, UsAir97, Webpolblogs, Email-univ and PowerGrid datasets. As the Louvain community detection algorithm involve randomness, the algorithm is iterated for 25 times on 25 different communities. It is observed in the figure that lower conductance value is

produced by the PowerGrid dataset and the higher conductance value is given by the email-univ dataset.

The time taken to compute the conductance score is given in Table 5. We observed in the table that less time for computation is taken by the UsAir97 dataset and more time for computation is taken by the PowerGrid dataset. It is also seen that less complex network structures take less time for the computation and dense complex network takes more time for the computation of the conductance score.

CONCLUSION AND FUTUREWORK

In this paper, we provide a comprehensive overview of various community detection algorithms that have been developed over time. Our focus is on evaluating

the performance of these algorithms using the modularity and conductance scores as a quality measure. Additionally, we measure the execution time of each algorithm to assess its computational efficiency. Our findings reveal that the label propagation algorithm exhibits faster computation times compared to different algorithms and the Kernighan-Lin algorithm takes more time for its computation. Furthermore, we observe that the Louvain and Leiden methods perform well in identifying community structures with high modularity scores and less time for its computation. By observing in terms of modularity score and the time taken for the computation, we say that the Louvain community detection algorithm performs well. In terms of conductance score assessed for the Louvain community detection algorithm, PowerGrid dataset takes less conductance score and more time taken for computation.

Looking ahead, the future of community detection algorithms lies in addressing scalability challenges and effectively handling dynamic and complex network structures. Furthermore, future research can focus on areas such as multi-objective optimization, benchmarking and evaluation, application-specific community detection, interpretability, visualization and community evolution analysis. Advancements in these areas can significantly enhance the capabilities of community detection algorithms, enabling them to better meet the requirements of diverse networks. By addressing these aspects, community detection algorithms can provide deeper insights into the structures and dynamics of networks, empowering researchers and practitioners to make more informed decisions and gain a comprehensive understanding of complex systems.

Getting datasets when the time evolve is also a challenging task and in future we want to investigate for such kind of datasets and analyse hoe the communities within the network change over time.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

REFERENCES

- Amaral LA, Scala A, Barthelemy M, Stanley HE. Classes of small-world networks. *Proceedings of the national academy of sciences*. 2000;97(21):11149-52.
- Lancichinetti A, Fortunato S. Community detection algorithms: a comparative analysis. *Physical review E*. 2009;80(5):056117.
- Leskovec J, Lang KJ, Mahoney M. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web 2010* (pp. 631-40).
- Leung JX, Hui P, Lio P, Crowcroft J. Towards real-time community detection in large networks. *Physical Review E*. 2009;79(6):066107.
- Khan BS, Niazi MA. Network community detection: A review and visual survey. *arXiv preprint arXiv:1708.00977*. 2017
- Liu X, Cheng HM, Zhang ZY. Evaluation of community detection methods. *IEEE Transactions on Knowledge and Data Engineering*. 2019;32(9):1736-46.
- Malliaros FD, Vazirgiannis M. Clustering and community detection in directed networks: A survey. *Physics reports*. 2013;533(4):95-142.
- Fortunato S. Community detection in graphs. *Physics reports*. 2010;486(3-5):75-174.
- Newman ME, Girvan M. Finding and evaluating community structure in networks. *Physical review E*. 2004;69(2):026113.
- Abdrabbah SB, Ayachi R, Amor NB. Collaborative filtering based on dynamic community detection. *Dynamic Networks and Knowledge Discovery*. 2014;85.
- Lalwani D, Somayajulu DV, Krishna PR. A community driven social recommendation system. In *2015 IEEE International conference on big data (big data) 2015* (pp. 821-6). IEEE.
- Gangopadhyay A, Chen S. Health care fraud detection with community detection algorithms. In *2016 IEEE International Conference on Smart Computing (SMARTCOMP) 2016* (pp. 1-5). IEEE.
- Pourhabibi T, Ong KL, Kam BH, Boo YL. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*. 2020;133:113303.
- Hajiabadi M, Zare H, Bobarshad H. IEDC: An integrated approach for overlapping and non-overlapping community detection. *Knowledge-Based Systems*. 2017;123:188-99.
- Negara ES andryani R. A review on overlapping and non-overlapping community detection algorithms for social network analytics. *Far East Journal of Electronics and Communications*. 2018;18(1):1-27.
- Mohamed EM, Agouti T, Tikniouine A, El Adnani M. A comprehensive literature review on community detection: Approaches and applications. *Procedia Computer Science*. 2019;151:295-302.
- Doluca O, Oğuz K. APAL: Adjacency Propagation Algorithm for overlapping community detection in biological networks. *Information Sciences*. 2021;579:574-90.
- Du N, Wu B, Pei X, Wang B, Xu L. Community detection in large-scale social networks. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis 2007* (pp. 16-25).
- Valmohammadi C, Taraz R, Mehdi khani R. The effects of brand community identification on consumer behavior in online brand communities. *Journal of Internet Commerce*. 2023;22(1):74-96.
- Gaspiretti F, Sansonetti G, Micarelli A. Community detection in social recommender systems: a survey. *Applied Intelligence*. 2021;51:3975-95.
- Karataş A, Şahin S. Application areas of community detection: A review. In *2018 International congress on big data, deep learning and fighting cyber terrorism (IBIGDELFT) 2018* (pp. 65-70). IEEE.
- Rostami M, Oussalah M, Berahmand K, Farrahi V. Community Detection Algorithms in Healthcare Applications: A Systematic Review. *IEEE Access*. 2023 Mar 22.
- Varun, E., and Ravikumar, P. (2016). Telecommunication community detection by decomposing network into n-cliques. In *2016 International Conference on Emerging Technological Trends (ICETT)* (pp. 1-5). IEEE.
- Yassine, S., Kadry, S., and Sicilia, M. A. Application of community detection algorithms on learning networks. The case of Khan Academy repository. *Computer Applications in Engineering Education*, 2021;29(2):411-24.
- Chattopadhyay, S., Basu, T., Das, A. K., Ghosh, K., and Murthy, L. C. Towards effective discovery of natural communities in complex networks and implications in e-commerce. *Electronic Commerce Research*, 2021;21:917-54.
- Lu M, Zhang Z, Qu Z, Kang Y. LPANNI: Overlapping community detection using label propagation in large-scale complex networks. *IEEE Transactions on Knowledge and Data Engineering*. 2018;31(9):1736-49.
- Xie J, Kelley S, Szymanski BK. Overlapping community detection in networks: The state-of-the-art and comparative study. *Acm computing surveys (csur)*. 2013;45(4):1-35.
- Yang J, Leskovec J. Community-affiliation graph model for overlapping network community detection. In *2012 IEEE 12th international conference on data mining 2012* (pp. 1170-5). IEEE.
- Gupta S, Singh DP. Recent trends on community detection algorithms: A survey. *Modern Physics Letters B*. 2020;34(35):2050408.
- Ramzan N, van_Zwol R, Lee JS, Clüver K, Hua XS, editors. *Social media retrieval*. Springer Science and Business Media; 2012.
- Bedi P, Sharma C. Community detection in social networks. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*. 2016;6(3):115-35.
- Girvan M, Newman ME. Community structure in social and biological networks. *Proceedings of the national academy of sciences*. 2002;99(12):7821-6.
- Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*. 2008; 2008(10):P10008.
- Kemighan BW. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical Journal*, 1970. 1970;49:291-307.
- Traag, V. A., Waltman, L., and Van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports*, 2019;9(1):5233.
- Raghavan UN, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*. 2007;76(3):036106.
- Rosvall M, Bergstrom CT. Maps of random walks on complex networks reveal community structure. *Proceedings of the national academy of sciences*. 2008;105(4):1118-23.
- Pons P, Latapy M. Post-processing hierarchical community structures: Quality improvements and multi-scale view. *Theoretical Computer Science*. 2011;412(8-10):892-900.
- Newman ME. Analysis of weighted networks. *Physical review E*. 2004;70(5):056131.
- Jiang H, Liu Z, Liu C, Su Y, Zhang X. Community detection in complex networks with an ambiguous structure using central node based link prediction. *Knowledge-Based Systems*. 2020;195:105626.
- Yang G, Zheng W, Che C, Wang W. Graph-based label propagation algorithm for community detection. *International Journal of Machine Learning and Cybernetics*. 2020;11:1319-29.
- Smith NR, Zivich PN, Frerichs LM, Moody J, Aiello AE. A guide for choosing community detection algorithms in social network studies: The question alignment approach. *American journal of preventive medicine*. 2020;59(4):597-605.
- Pons P, Latapy M. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005: 20th International Symposium, Istanbul, Turkey, 2005. Proceedings 20 2005* (pp. 284-293). Springer Berlin Heidelberg.
- Clauset A, Newman ME, Moore C. Finding community structure in very large networks. *Physical review E*. 2004;70(6):66111.
- Newman ME. Spectral methods for community detection and graph partitioning. *Physical Review E*. 2013;88(4):42822.
- Kumar P, Chawla P, Rana A. A review on community detection algorithms in social networks. In *2018 4th International Conference on Applied and Theoretical Computing and Communication Technology (iCATcT) 2018* (pp. 304-9). IEEE.
- Xia C, Luo Y, Wang L, Li HJ. A fast community detection algorithm based on reconstructing signed networks. *IEEE Systems Journal*. 2021;16(1):614-25.
- Javed MA, Younis MS, Latif S, Qadir J, Baig A. Community detection in networks: A multidisciplinary review. *Journal of Network and Computer Applications*. 2018;108:87-111.
- Leskovec, J., Lang, K. J., and Mahoney, M. (2010, April). Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web* (pp. 631-40).
- Wagenseller, P., Wang, F., and Wu, W. Size matters: A comparative analysis of community detection algorithms. *IEEE Transactions on Computational Social Systems*, 2018;5(4):951-60.
- Newman ME. Fast algorithm for detecting community structure in networks. *Physical review E*. 2004;69(6):066133.
- Varsha, K., and Patil, K. K. (2020, February). An overview of community detection algorithms in social networks. In *2020 International Conference on Inventive Computation Technologies (ICICT)* (pp. 121-6). IEEE.
- Seifkar, M., Farzi, S., and Barati, M. C-blondel: an efficient louvain-based dynamic community detection algorithm. *IEEE Transactions on Computational Social Systems*, 2020;7(2):308-18.
- Veira, V. D. F., Xavier, C. R., Ebecken, N. F. F., and Evsukoff, A. G. (2014). Performance evaluation of modularity based community detection algorithms in large scale networks. *Mathematical Problems in Engineering*, 2014.
- Nerurkar, P., Chandane, M., and Bhirud, S. (2019). A comparative analysis of community detection algorithms on social networks. In *Computational Intelligence: Theories, Applications and Future Directions-Volume I: ICCI-2017* (pp. 287-98). Springer Singapore.
- Sahu, S. (2024). Addressing Internally-Disconnected Communities in Leiden and Louvain Community Detection Algorithms. *arXiv preprint arXiv:2402.11454*.

Cite this article: Tokala S, Enduri MK, Lakshmi TJ, Hajarathiah K. Evaluating Community Detection Algorithms: A Focus on Effectiveness and Efficiency. *J Scientometric Res*. 2025;14(1):62-74.