SCIENTIFIC CULTURE
www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 12, No. 4, (2026), pp. 1633-1654
Open Access. Online & Print

DOI: 10.5281/zenodo.19235126

COMMUNITY DETECTION IN COMPLEX NETWORKS: A REVIEW OF LOUVAIN, GIRVAN-NEWMAN, CNM, AND MAX-MIN ALGORITHMS

Srinivas Amedapu[1]*, Leela Velusamy R[2]

[1]Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India
[2]Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India

Received: 24/09/2025
Accepted: 28/01/2026

Corresponding Author: Srinivas Amedapu
(srinivasreddyamedapu@yahoo.com)

ABSTRACT

The rapid expansion of online social networks (OSNs) has increased the need for effective techniques to identify community structures, in which users exhibit strong interaction patterns. Detecting such communities is challenging due to large network sizes, noisy connections, and continuously evolving user behaviour. This study investigates and compares four widely used community detection algorithms, Louvain, Newman-Girvan, Clauset-Newman-Moore (CNM), and Max-Min, using the Dolphin social network as a real-world benchmark dataset. The algorithms are evaluated based on performance metrics, including modularity, which measures the strength of community separation, and Normalized Mutual Information (NMI), which assesses the agreement between detected communities and known ground-truth structures. Experimental results indicate that the Louvain algorithm achieves superior performance, obtaining the highest modularity value of 0.5188 and an NMI score of 0.911, demonstrating its effectiveness in uncovering well-defined community structures. In contrast, the Max-Min approach produces comparatively moderate results, with a modularity score of 0.4014 and an NMI of 0.580. Overall, the findings suggest that community detection methods centered on modularity optimization provide a more effective trade-off between capturing network topology and reflecting real-world interaction patterns in OSNs.

KEYWORDS: Community detection, Modularity, Normalized Mutual Information (NMI), Online Social Networks, Louvain, Newman-Girvan, Clauset-Newman-Moore, Max-Min.

1. INTRODUCTION

The analysis of complex networks, in general, faces a fundamental problem known as community detection, which seeks to determine groups of nodes that are more densely linked to one another than to the rest of the network. This kind of community is common across a broad range of real-life systems, including social networks on the Internet, citation networks, biological interaction networks, communication systems, and transportation systems [1]. Community identification helps show the organization that underlies the networks and provides significant information on functional, social, or semantic relationships between entities [2].

Community detection is especially important in the context of social networks (OSNs) on the web. The community of users is a natural phenomenon in OSNs, built around a common interest, social interaction, geographical location, or a common activity. These communities largely determine the spread of information, the changing of opinion, and the occurrence of collective behaviours within the network [3]. Consequently, this has made proper community identification an indispensable requirement in applications related to recommendation systems, maximizing influence, analysing information diffusion, and mitigating abnormal or malicious behaviour on social sites [4].

Community detection is a challenging problem, even though it is important given the nature of real-world networks. Contemporary networks are commonly vast, sparse, and noisy, with complex connectivity structures and lacking information [5]. Also, the community structure can be dynamic and overlapping, with nodes belonging to more than one community, and the community boundaries eventually change [6]. These demands necessitate the development of network detection algorithms that are scalable, robust, and computationally efficient, not only for detection within the community but also for accuracy.

The classical graph-theoretic methods served as the basis for the initial studies of community detection. The Girvan-Newman algorithm is one of the most powerful algorithms, in which community identification is achieved by systematically erasing edges with high betweenness centrality, thereby exposing the community borders as the network breaks down [7]. Although the method offers good interpretability and performance in small networks, its high computational cost makes it inapplicable to large-scale systems.

To overcome the scalability problem, methods based on modularity were proposed. Modularity measures the quality of a partition by comparing the density of edges within communities to that in a random network. The Clauset-Newman-Moore (CNM) algorithm uses a greedy hierarchical approach to maximize modularity and performs better than divisive algorithms [8]. Afterward, the Louvain algorithm became one of the most popular methods of community detection because of its multi-level optimization approach, which enables the efficient detection of high-quality communities in large networks [9]. It has been noted, however, that modularity-based methods suffer from the resolution limit, which is a limitation in which smaller communities can merge into larger ones as the network size increases [10].

In addition to modularity-oriented strategies, distance- and dispersion-oriented methods, such as Max-Min strategies, have been studied to enhance community segregation and boundary demarcation. These approaches aim a maximize the distance between community representatives and minimize intra-community distances, offering a different solution to density-based optimization [11]. These are especially effective when fine-grained or user-controlled community organizations are required.

In recent years, advances in artificial intelligence and deep learning have also influenced research in community detection. Models based on artificial intelligence have been effectively used to analyse large-scale interaction data, enabling the discovery of latent group structures and behavioural patterns in complex systems [12]. The analysis of content and interaction using deep learning on social platforms has shown the significance of organized groupings and community-conscious analysis to understand user behaviour and interaction [13]. These trends highlight the growing importance of integrating classical principles of community detection with modern AI-based analytics.

Recent studies emphasize the growing importance of intelligent, data-driven approaches for analysing systems characterized by high complexity and extensive interactions. In particular, AI-based methods have demonstrated significant potential in uncovering hidden interaction patterns at large scales in social media networks, enabling deeper insights into user behaviour and group-specific dynamics [14], [15]. These findings highlight the need for systematic grouping and community-aware analysis, as well as the importance of effective and interpretable community detection techniques as fundamental tools for investigating complex networks.

Based on these insights, the paper analyses and compares well-known community detection algorithms, including Louvain, Girvan-Newman, Clauset-Newman-Moore, and MaxMin. By analysing their values, strengths, and weaknesses, this paper aims to provide a clear understanding of their appropriateness for analysing community structures in intricate and online social networks.

2. LITERATURE REVIEW

Community detection has long been of interest in the study of online social networks (OSNs), with the boom of user-generated content and the dynamic nature of social interaction. As the size and complexity of OSNs have increased, meaningful communities have been found to play a role in interpreting collective behaviour and network structure [16]. The community within OSNs can also be defined as a subset of users with similar interests, communication influences, or social roles; thus, community detection is an essential method in social network analysis and data mining tasks employed in applied data mining domains.

Graph-theoretic techniques were the first to be developed for community detection and relied on network topological properties. Modularity-based methods have produced a significant result, as modularity provides a quantitative measure of the quality of network partitioning by considering intra-community connectedness relative to a random reference [17]. These methods served as the foundation for many subsequent algorithms and demonstrate inherent limitations, such as sensitivity to network size and an inability to identify small or overlapping communities.

To address these problems, researchers began discussing machine learning and deep learning approaches to community detection. De Santo et al. propose a semi-supervised model that integrates convolutional neural networks and network structure to train structural and contextual features, achieving improved results on sparse social networks [18]. This paper also highlighted the potential to combine learned representations with traditional graph features to overcome sparsity and noise in OSN data.

Following the intersection of learning-based models and network analysis, Ali et al. studied the application of deep recurrent learning to social media analysis by combining community detection with content classification to identify hate-driven user groups on Twitter [19]. Their approach combines LSTMGRU textual analysis models with graph-based community detection, demonstrating the potential of structural grouping to enhance downstream social media monitoring processes. These works reveal the importance of integrating content and structure analysis into existing OSNs.

As multimedia content has spread on social platforms, community detection methods have since surpassed text-based models, and they are on the rise. Ferraro et al. responded to this shift by proposing a model for multimedia-based social networks that discovers shared visual and semantic content using hypergraph representations and convolutional neural networks [20]. This approach demonstrated how heterogeneous data modalities can be useful for discovering semantically coherent communities.

Most recent advances in graph neural networks (GNNs) have further expanded the boundaries of the community detection literature. Rashnodi et al. have proposed a dual-embedding graph convolutional network that jointly leverages modularity constraints and Markov transition matrices to learn expressive node representations [21]. They achieved significant improvements over classical methods, particularly in networks with strong structural dependencies. This has occurred with other GNN-based models, where representation learning enables the learning of higher-order neighbourhood information [22].

The other important research direction is dynamic community detection, which captures the temporal dynamics of real-world social networks. To determine community changes over time in big-data networks, Asaidi et al. developed the Incremental Speaker-Listener Propagation Algorithm [23]. This method minimizes computational cost by updating community assignments one at a time, while maintaining time continuity. Community detection temporal extensions are particularly applicable to OSNs, where user interactions evolve rapidly, and even a fixed model cannot detect new patterns [24].

The significance of overlap community detection has also been provided with a substantial measure of consideration, as the users of OSN tend to belong to different social groups simultaneously. Sarswat et al. proposed a two-step approach in which modularity density, coupled with silhouette coefficients, and node degree-based classification identify overlapping communities [25]. They found more network overlaps to be accurate. Similarly, Huang et al. developed an optimization model using particle swarm optimization and line graph mutations to remove noise in overlapping community detection [26].

The alternative approach to identifying influential nodes and dense substructures in networks is the centrality-based method. To improve the performance of clique percolation methods, Kasoro et al. introduced eigenvector centrality, which enhanced the characterization of influential overlaps in dense social networks [27]. This research line examines the role of node importance in defining community boundaries, particularly in networks with non-uniform degree distributions.

Another urgent need for community detection algorithms is scalability, which is critical as OSNs scale to millions of users. Georgiou has addressed this issue by including local edge betweenness in the user content information to produce scalable, high-quality community detection results on real social networks [28]. This mixed approach has shown the capacity to exploit local structural information to obtain a computational saving in the reduction of the quality of identification.

Besides purely algorithmic developments, several surveys and comparative studies offer useful insights into the opportunities and weaknesses of existing approaches. The modes of community recognition via modularity maximization, statistical learning, propagation dynamics, and learning-based frameworks were enumerated in extensive surveys published in several reputed journals. These surveys show that no universal method is effective across all network types, which explains the need for an algorithm tailored to the circumstances.

Recent research has also highlighted the growing role of artificial intelligence in analysing social networks. Clustering and segmentation processes developed using AI have been applied to understand user behaviour and interactions on online platforms and have demonstrated clear similarities with the objectives of community detection [30]. Such techniques often invoke latent grouping algorithms that make the community structures in the interaction graph implicit.

Several research articles continue to show how smart-data-based models are implemented in systems that are interaction-intensive. It has also been shown that social media behaviour has been incorporated into deep learning, revealing the hidden dynamics of users and enhancing the role of structured grouping in the massive digital space [31]. The findings can be compared to the general trends in the OSN research literature, where community detection is a preliminary analysis step.

In addition to social media, AI-based clustering has been applied in educational and organizational settings to subdivide users and mimic engagement patterns [32]. Such studies are not claimed to be the problems of community detection; nevertheless, they are based on the grounds of grouping and similarity analysis, to which the interdisciplinary applicability of community detection methods can be ascribed.

Explainability and interpretability have become critical in recent community-detection research. Because outputs from detection are used to make decisions in sensitive areas, such as security and policy-making, there has been a growing interest in community assignments with transparent algorithms that provide interpretable outputs to the community [33]. This requirement has given rise to a new interest in classical and hybrid methods, which are trade-offs between accuracy and interpretability.

Overall, the literature suggests a clear trend toward hybrid, learning-based community detectors that would merge structure, content, and time. Despite this, classical graph partitioning algorithms, such as Louvain, Girvan-Newman, and CNM, are effective baselines due to their simplicity of understanding and interpretation and their strong performance [34]. Comparative analysis of these approaches is an essential feature not only of understanding how to apply them to networks in real life but also of guiding the development of more advanced approaches. Community detection can be applied to a wide range of applications, as illustrated in Figure 1.

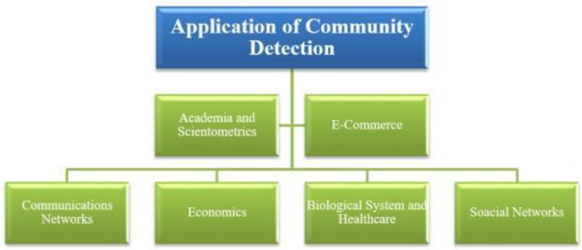


Figure 1: Application of community detection [6]

3. ALGORITHMS COMPARISON ON BENCHMARK DATASET

To ensure that the proposed research meets the criteria for quality and informative community identification, the overall research approach is a systematic, multi-step process that will be applied to the online social network system, as indicated in Figure 2. The methodology begins with the selection of a real-life benchmark dataset, i.e., the Dolphin social network, which captures the patterns of interactions among 62 dolphins and 159 social ties. Because it has a clear structure, this data set provides an opportunity to research social relations and identify coherent subgroups within a network.

Four community-detection algorithms, including Louvain, Girvan-Newman, Clauset-Newman-Moore (CNM), and Max-Min, were run and evaluated to produce a systematic evaluation. All algorithms were run on the Dolphin network to find communities, assess boundary separation, and evaluate partition quality using multiple measures, including modularity and computation time.

The dolphin dataset has been chosen because the structure of the community is a well-studied and reported topic that can serve as a good benchmark for comparing algorithms. The resulting partitions were compared by visual and quantitative analysis to enable evaluation of the algorithms' behaviour as a whole. This research paradigm strikes a fair balance between data set selection, algorithm implementation, and comparative analysis, thereby providing a solid foundation for analysing community formation in complex social networks.

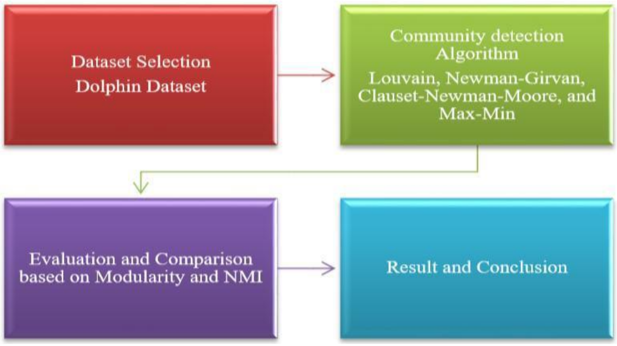


Figure 2: Block chart of proposed work

3.1. Dataset Used

The effectiveness of the selected community-detection algorithms is evaluated on the Dolphin social network dataset, as summarized in Table 1. It is a well-known standard dataset in network science and community detection, based on the interaction patterns of a population of dolphins observed off the coast of New Zealand.
This dataset captures frequent associations between 62 bottlenose dolphins.

- The data was collected by David Lusseau over several years of field observations.
- It forms an undirected and unweighted graph
- Nodes represent individual dolphins.
- Edges represent social bonds (i.e., observed frequent associations between dolphins).

Table1: Dataset of Dolphin Social Network

Dataset Name	Number of Nodes	Number of Edges	Expected Communities k	Dataset Description
Dolphins [31]	62	159	3	Dolphins Social Network

3.2. Algorithms for Community Detection

This section provides a summary of four popular community detection algorithms. Both techniques employ network structural properties based on information that is not directly connected to the nodes. In this case, higher-order information refers to the degree of graph elements (nodes and edges) that are looked at or traversed in the process of computation.

3.3. Louvain Algorithm

The Louvain method, also known as a multilevel community detection algorithm, is widely used for identifying communities in complex networks. The process starts by assigning each node to its own community. Nodes are then repeatedly reassigned to neighboring communities whenever such movements lead to an improvement in local connectivity, with the primary goal of maximizing modularity, a measure that reflects the quality of the network's division into communities.

The Louvain algorithm seeks to build communities that are densely connected internally and sparsely connected externally. It relies on modularity as the primary metric, which quantifies how well a network is partitioned into communities. The modularity score m ranges from -1 to +1 and is calculated using the following formula:

$$m = \frac{1}{2|E|} \sum_{(i,j)} \left(w_{i,j} - \frac{degree(v_i) * degree(v_j)}{2|E|} \right) \quad (1)$$
$$v_i \epsilon c_i,\ v_j \epsilon c_j$$

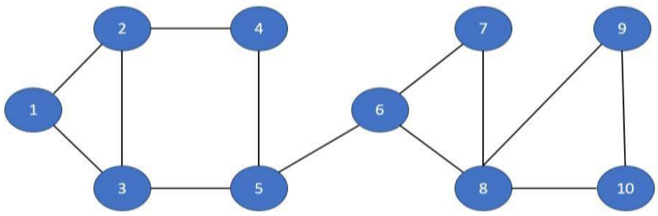
The weight of $(i,\ j)$ is denoted by $w_{i,j}$ in equation (1), and c_i and c_j stand for the communities to which v_i and v_j belong. The result is always a weighted graph where the weights are determined by the local densities of the edges, even if Louvain's way of analysis works for non-weighted graphs. Graphs that started with a weight of 1 are called non-weighted graphs. There are two separate steps to carry out the optimization of modularity. In the beginning, each v_i is connected to all of its neighbours in a cluster C , and the change in modularity Δm is calculated as the modification to the latest type minus the old [36]. Ultimately, v_i is assigned to c_j, leading to an increased Δm. In the second step, researchers construct a new network that consolidates the vertices associated with the common grouping into a single vertex. In addition, every vertex that connects the two sets of data forms a single vertex whose weight is the sum of all the weights.

3.4. Algorithm Steps of the Louvain

1. Initialization: Each node in the network is assigned to its own community. So, for a

network with N nodes, there are initially N communities.

2. Modularity Optimization For each node: Consider moving it to the community of each of its neighbors. Calculate the modularity gain from each possible move. Move the node to the community where the increase in modularity is maximum (only if the gain is positive). Repeat this process for all nodes until no further improvement in modularity is possible.
3. Community Aggregation - A new network is created: Each node in the new network represents a community from the previous phase. Edges between these new nodes represent the total weight of edges between nodes in the corresponding communities. Self-loops are included if there are internal edges within the same community.
4. Repeat: Apply modularity optimization again on the new (aggregated) network. Continue repeating steps 2 and 3 until there is no further increase in modularity.



Graph-2

Table-2

Node	Degree	Node	Degree
1	2	6	3
2	3	7	2
3	3	8	4
4	2	9	2
5	3	10	2

Graph-2 illustrates the example network used to explain the Louvain community detection methodology.

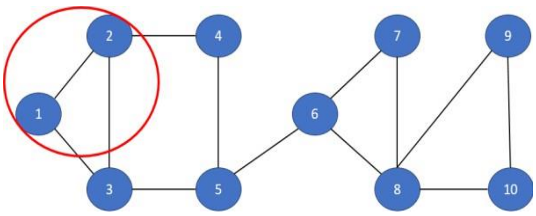
Table-2 presents the degree of each node in Graph-2.

Table-3(a)

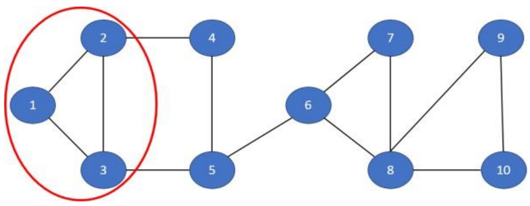
C_1	C_2	m	A_{ij}	K_i	K_j	Delta Q
1	2	13	1	2	3	**0.0592**
1	3	13	1	2	3	0.0592

Table-3(b)

C_1	C_2	m	A_{ij}	K_i	K_j	Delta Q
(1,2)	3	13	2	5	3	**0.1095**
(1,2)	4	13	1	5	2	0.0473



Graph-3(a)



Graph-3(b)

In Table-3(a), the modularity calculation begins with node v1, considering its possible merges with v2 and v3. It is observed that the Delta Q value is the same (0.0592) for both pairs v1-v2 and v1-v3. Since v1 -v2 appears first, nodes v1and v2 are merged into a single community, as illustrated in Graph-3(a). In Table-3(b), the Delta Q values are calculated for the community {v1, v2} when merged with v3 and with v4. The highest Delta Q value (0.1095) is obtained for the merge of {v1, v2} with v3. Consequently, v3 is added to the community, and the updated structure is shown in Graph-3(b).

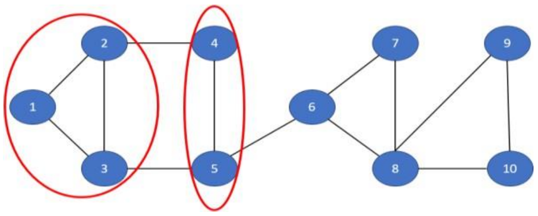
In Table 3(c)**,** the modularity gain Delta Q is calculated for the community {v1, v2, v3} with nodes v4 and v5 as well as for the pair (v4, v5). The results show that the Delta Q values are 0.0296 for {v1, v2, v3}-v4, 0.0059 for {v1, v2, v3}-v5, and 0.0592 for the pair (v4, v5). Since the highest Delta Q value is obtained for (v4, v5), these nodes are merged into a single community. The resulting structure is shown in Graph 3(c)**.**

Table-3(c)

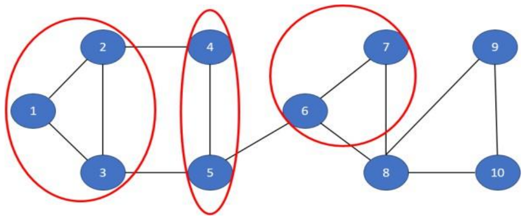
C_1	C_2	m	A_{ij}	K_i	K_j	Delta Q
(1,2,3)	4	13	1	8	2	0.0296
(1,2,3)	5	13	1	8	3	0.0059
4	5	13	1	2	3	**0.0592**

Table-3(d)

C_1	C_2	m	A_{ij}	K_i	K_j	Delta Q
(4,5)	6	13	1	5	3	0.0325
6	7	13	1	3	2	**0.0592**



Graph-3(c)



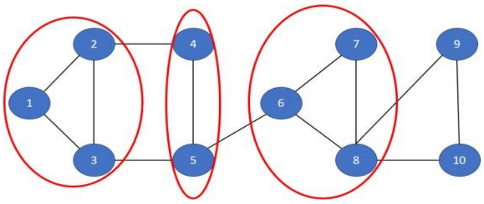
Graph-3(d)

In Table 3(d), the modularity gain is further evaluated by considering the community {v4, v5} with node v6, and the pair (v6, v7). The Delta Q values are 0.0325 for {v4, v5}-v6 and 0.0592 for (v6, v7). As the highest value is achieved for (v6, v7), these nodes are merged to form a new community. The updated community structure is presented in Graph 3(d).

In Table-3(e), the modularity gain Delta Q is computed for two possible merges: the community (v6, v7) with node v8, and node v8 with node v9. The results show that the Delta Q values are 0.0947 for (v6, v7) -v8 and 0.0533 for v8-v9. Since the highest modularity gain occurs for (v6, v7) -v8, node v8 joins the community {v6, v7}. The updated community structure is illustrated in Graph-3(e).

Table-3(e)

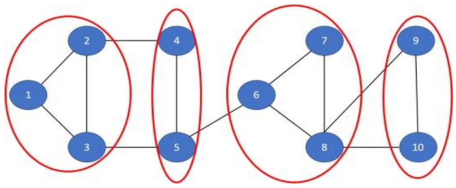
C_1	C_2	m	A_{ij}	K_i	K_j	Delta Q
(6,7)	8	13	2	5	4	**0.0947**
8	9	13	1	4	2	0.0533



Graph-3(e)

Table-3(f)

C_1	C_2	m	A_{ij}	K_i	K_j	Delta Q
(6,7,8)	9	13	1	9	2	0.0237
(6,7,8)	10	13	1	9	2	0.0237
9	10	13	1	2	2	**0.0651**



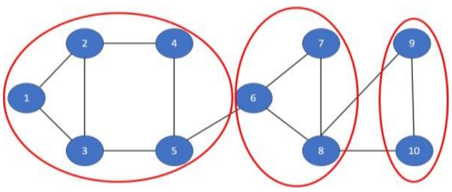
Graph-3(f)

In Table-3(f), the Delta Q values are calculated for merging the community {v6, v7, v8} with node v9, the same community with node v10, and the pair (v9, v10). The respective values are 0.0237, 0.0237, and 0.0651. As the maximum value is obtained for the pair (v9, v10), these two nodes are merged into a new community. The resulting partition of the graph is shown in Graph-3(f).

In Table-3(g), the modularity gain Delta Q is calculated for merging the two communities {v1,v2,v3} and {v4,v5}. The resulting Delta Q value is 0.0355, which indicates an improvement in modularity. Hence, these two communities are merged to form a larger group {v1, v2, v3, v4, v5}. The updated community structure is illustrated in Graph-3(g), where we observe three communities: {v1, v2, v3, v4, v5}, {v6, v7, v8}, and {v9, v10}.

Table-3(g)

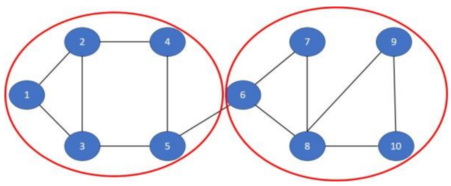
C_1	C_2	m	A_{ij}	K_i	K_j	Delta Q
(1,2,3)	(4,5)	13	2	8	5	**0.0355**



Graph-3(g)

Table-3(h)

C_a	C_b	E_{ab}	m	k_a	k_b	Delta Q
(1,2,3,4,5)	(6,7,8)	13	1	13	9	-0.2692
(6,7,8)	(9,10)	13	2	9	4	**0.0473**



Graph-3(h)

In Table-3(h), the modularity change is evaluated for two possible merges: {v1, v2, v3, v4, v5} with {v6, v7, v8}and {v6, v7, v8} with {v9, v10}. In the first case, the calculated Delta Q value is -0.2692, showing that merging these two communities reduces modularity and is therefore not beneficial. In the second case, the Delta Q value is 0.0473. Since this is positive and the highest among the options, the merge is accepted, resulting in a new community {v6, v7, v8, v9, v10}. The updated structure is shown in Graph-3(h), where the network now has two well-defined communities: {v1, v2, v3, v4, v5} and {v6, v7, v8, v9, v10}.

3.5. Newman-Girvan (NG) Algorithm

A notable method of finding out the community structures in social networks is the Girvan-Newman algorithm, introduced by Michelle Girvan and Mark Newman. The algorithm is based on the principle of edge betweenness centrality, which quantifies the significance of the edge by counting the number of shortest paths between pairs of nodes that pass through the edge. The edges that have a higher score on betweenness tend to be those that connect different groups into communities since they serve as one of the main routes of the interaction between two or more groups.

The edge betweenness centrality associated with an edge e is formally defined as follows:

$$C_B(e) = \sum_{s\neq t} \frac{\sigma_{st}(e)}{\sigma_{st}}$$

Where:

- C_B(e) denotes the betweenness centrality of edge e,
- σ_{st} represents the total number of shortest paths between nodes s and t
- $\sigma_{st}(e)$ is the number of those paths that pass-through edge e

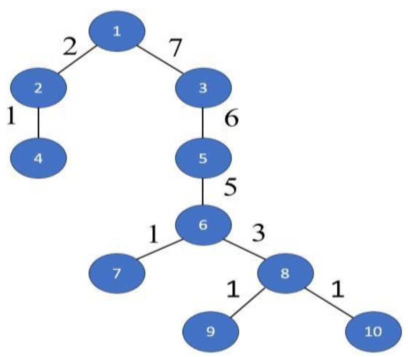
The process is iterative in the sense that the edge that has the highest betweenness score is repeatedly chosen and deleted. The network is gradually divided into smaller subgraphs of closely connected nodes as the edges of the highest centrality are removed. This successive deletion process persists until one gets a desired number of communities or the entire network is disconnected.

3.6. Algorithm Steps of the Newman-Girvan

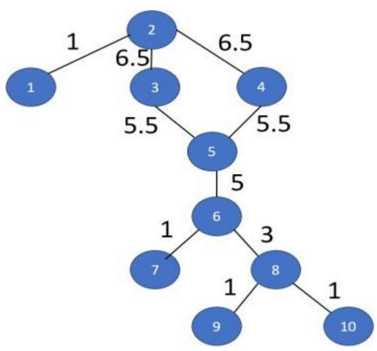
1. Calculate the betweenness centrality values for all edges in the network.
2. Identify and eliminate the edge that has the maximum betweenness value.
3. Update the betweenness scores of the remaining edges after the removal.
4. Continue the edge removal and recalculation process until well-defined communities emerge or the graph becomes fully disconnected.

This method is particularly appropriate for networks of small to medium size because it is able to distinguish certain structural links connecting two distinct communities. It is, however, limited in its applicability to large networks due to high computational cost, as calculations of shortest paths need to be repeated in the process. In general, the Girvan-Newman algorithm is conceptually simple and gives a clear, straightforward mechanism of identifying community structure by taking advantage of the importance of edge betweenness in revealing the modular structure of large networks.

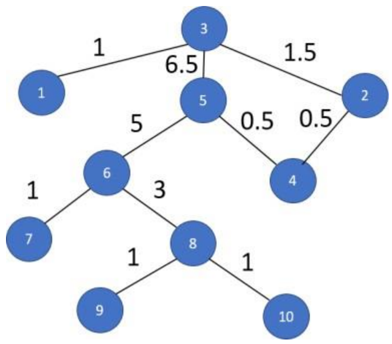
The algorithm uses each vertex in turn as a source node in finding edge betweenness centrality and forms a breadth-first search (BFS) tree with that source, which is illustrated in Graph-4(a) by Graph-4(j). With each BFS tree, each shortest path between the source node and all other nodes is found, and the contributing score is propagated to the edges that are used in the paths. This is done by repeated processing of each node as the root, and the sum of all of the BFS trees is added to get the final betweenness centrality values of each edge.



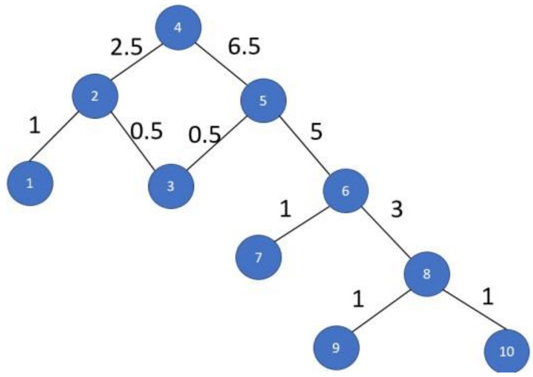
Graph-4(a)



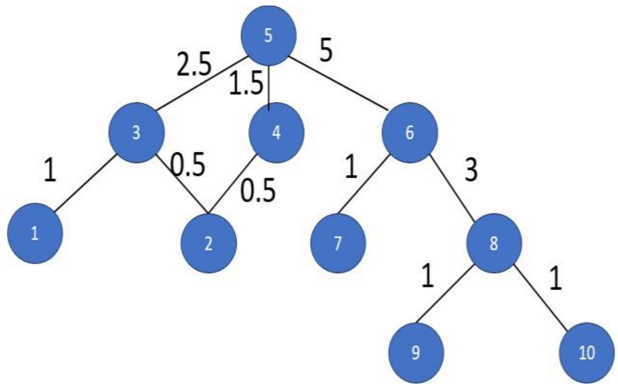
Graph-4(b)



Graph-4(c)

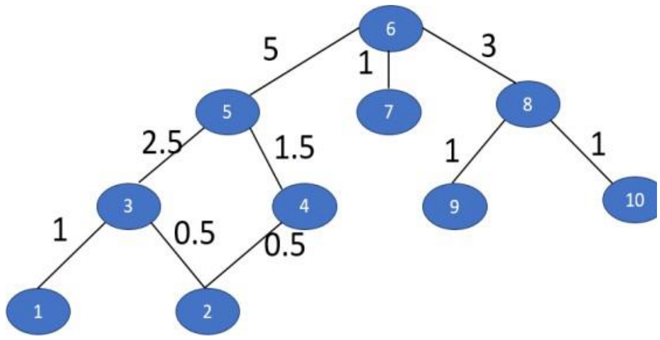


Graph-4(d)

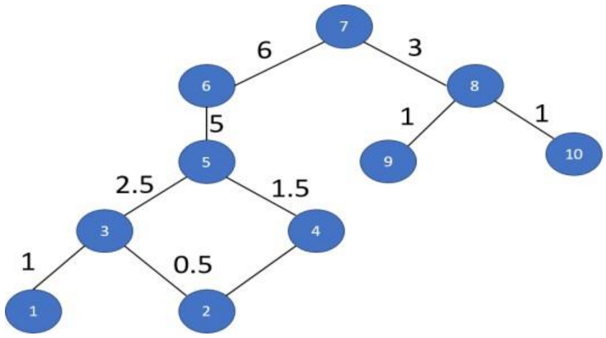


Graph-4(e)

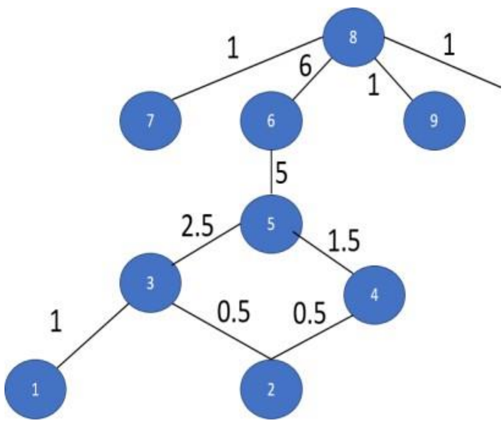
Graph-4(a) shows the process when node v1 is chosen as the starting point. The search commences again at v1, and the minimum paths to all the other vertices are then determined.



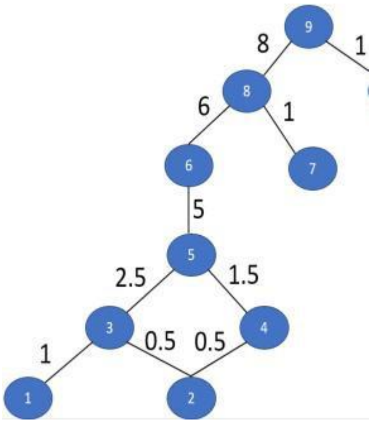
Graph-4(f)



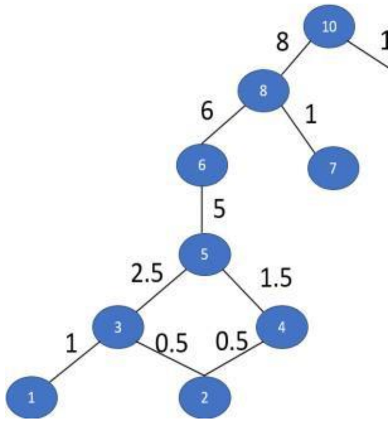
Graph-4(g)



Graph-4(h)



Graph-4(i)



Graph-4(j)

For example, the edge (1-3) obtains a betweenness centrality value of 7, since most paths from v1 to the right-hand side of the graph must pass through this edge. Similarly, the edge (1-2) carries a smaller value of 2, as it only connects v1 to the small local branch {v2, v4}. The labels on the edges represent the betweenness centrality values derived from this root.

The same procedure is applied with each of the remaining vertices (v2 through v10) as the root, generating Graph-4(b) to Graph-4(j). Summing the betweenness centrality values across all these BFS trees yields the final edge betweenness values shown in Table 4. Here, the edge (v5-v6) has the highest value (50.00), followed by (v6-v8 = 34.00) and (v3-v5 = 33.50), identifying them as critical inter-community bridges

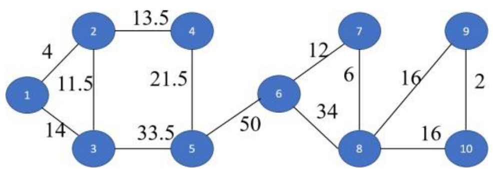
As shown in Graph-4(k), these final totals are annotated on the original network. Following the Girvan-Newman algorithm, the edge with the maximum betweenness centrality value (v5-v6) is removed first (Graph-4(l)), which disconnects the graph into two communities:

- C1: {v1, v2, v3, v4, v5}
- C2: {v6, v7, v8, v9, v10}

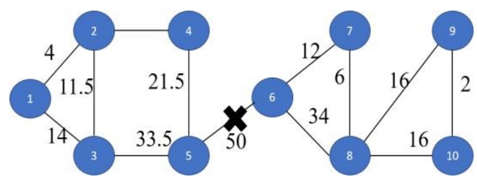
Table 4: Edge Betweenness Centrality

Edge	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	sum
v1-v2	2.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00
v1-v3	7.00	0.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	14.00
v2-v3	0.00	6.50	1.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	11.50
v2-v4	1.00	6.50	0.50	2.50	0.50	0.50	0.50	0.50	0.50	0.50	13.50
v3-v5	6.00	5.50	6.50	0.50	2.50	2.50	2.50	2.50	2.50	2.50	33.50
v4-v5	0.00	5.50	0.50	6.50	1.50	1.50	1.50	1.50	1.50	1.50	21.50
v5-v6	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	**50.00**
v6-v7	1.00	1.00	1.00	1.00	1.00	1.00	6.00	0.00	0.00	0.00	12.00
v6-v8	3.00	3.00	3.00	3.00	3.00	1.00	0.00	6.00	6.00	6.00	34.00
v7-v8	0.00	0.00	0.00	0.00	0.00	0.00	3.00	1.00	1.00	1.00	6.00
v8-v9	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	8.00	0.00	16.00
v8-v10	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	8.00	16.00
v9-v10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	2.00

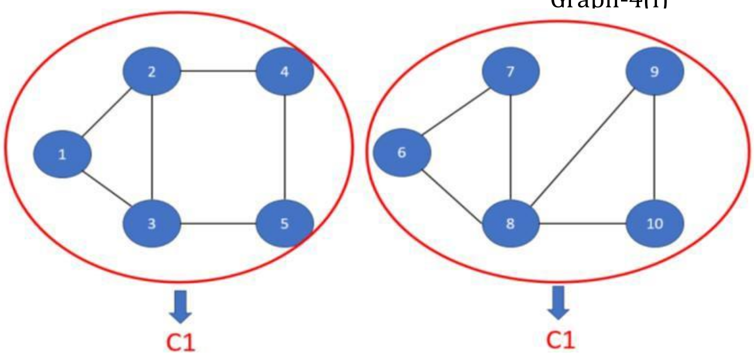
This shows that edges with high betweenness centrality act as bridges in the network, and removing them reveals the community structure, as illustrated in Graph-4(m).



Graph-4(k)



Graph-4(l)



Graph-4(m)

3.7. Clauset-Newman-Moore (CNM) Analysis

The Clauset-Newman-Moore (CNM) algorithm is a popular hierarchical method for detecting community structures in large-scale networks [29,30]. It was proposed by Aaron Clauset, Mark Newman, and Cristopher Moore as an efficient solution to uncover modular structures in graphs, particularly when dealing with networks too large for more computationally expensive methods like Girvan-Newman.

This algorithm belongs to the class of bottom-up hierarchical clustering techniques. It begins by treating each node in the network as a separate community and then iteratively merging pairs of communities that result in an increase in a quality function known as modularity.

The CNM algorithm is centered around modularity (Q), a scalar value that measures the strength of division of a network into communities. Modularity compares the density of edges inside communities with the density expected if edges were placed at random, without regard to community structure.

$$\Delta Q = \frac{e_{ab}}{m} - 2\left(\frac{k_a}{2m}, \frac{k_b}{2m}\right)$$

Where

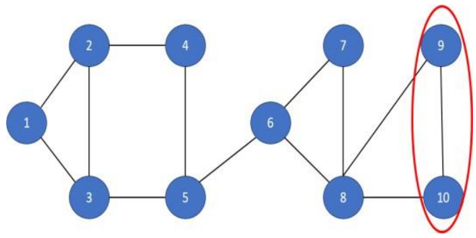
- e_{ab}: number of edges between communities a and b.
- m: total number of edges in the network.
- k_a, k_b: sum of degree of nodes in communities a and b, respectively
- First term $\frac{e_{ab}}{m}$: observed share of edges linking a and b
- Second term $2\left(\frac{k_a}{2m}, \frac{k_b}{2m}\right)$: expected share of such edges under a random baseline
- Decision rule: merge the pair with larges ΔQ;

1. Start by assigning each vertex to an individual community.
2. At each iteration, determine the pair of communities whose merger produces the maximum increase in modularity.
3. Combine the selected communities into a single community.
4. Update the internal data structures to efficiently reflect changes in modularity after the merge.
5. Repeat the merging and updating process until no additional improvement in modularity is possible or all vertices are grouped into one community.

Algorithm Steps of the CNM

Table-5(a)

C_a	C_b	E_{ab}	m	k_a	k_b	Delta Q
1	2	1	13	2	3	0.0592
1	3	1	13	2	3	0.0592
2	3	1	13	3	3	0.0503
2	4	1	13	3	2	0.0592
3	5	1	13	3	3	0.0503
4	5	1	13	2	3	0.0592
5	6	1	13	3	3	0.0503
6	7	1	13	3	2	0.0592
6	8	1	13	3	4	0.0414
7	8	1	13	2	4	0.0533
8	9	1	13	4	2	0.0533
8	10	1	13	4	2	0.0533
9	10	1	13	2	2	**0.0651**



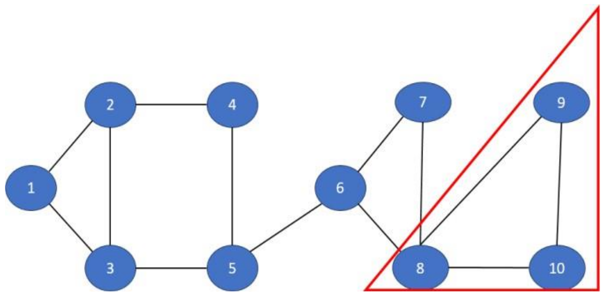
Graph-5(a)

The community configuration that attains the highest modularity score during the optimization process is chosen as the final partition.

Table-5(a) shows the computed ΔQ values for all potential edge-based community combinations in the network. Among these candidates, the edge connecting vertices V9 and V10 produces the largest increase in modularity, with a ΔQ value of 0.0651. As this represents the greatest modularity improvement at the current stage, vertices V9 and V10 are merged to form a single community. This initial merge is depicted in Graph-5(a), where the edge (V9-V10) is emphasized to indicate the first community formation.

Table-5(b)

C_a	C_b	E_{ab}	m	k_a	k_b	Delta Q
1	2	1	13	2	3	0.0592
1	3	1	13	2	3	0.0592
2	3	1	13	3	3	0.0503
2	4	1	13	3	2	0.0592
3	5	1	13	3	3	0.0503
4	5	1	13	2	3	0.0592
5	6	1	13	3	3	0.0503
6	7	1	13	3	2	0.0592
6	8	1	13	3	4	0.0414
7	8	1	13	2	4	0.0533
8	(9,10)	2	13	4	4	**0.1065**



Graph-5(b)

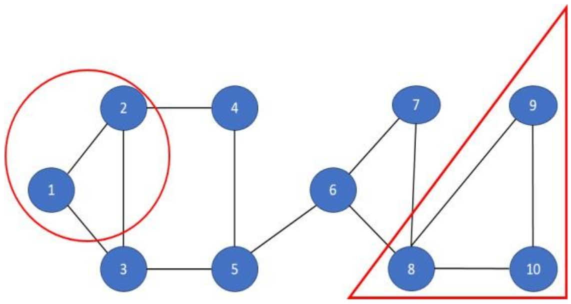
Table-5(b) shows the recalculated Delta Q values after the first merge in Table-5(a), where vertices V9 and V10 were combined into a single community. Following this merge, only the Delta Q values involving the affected node V8 were updated, since V8 was originally connected to both V9 and V10.

From the recalculations, the maximum Delta Q value of 0.1065 is observed for the merge between V8 and the newly formed community (V9, V10). Consequently, the next modularity-maximizing step merges V8, V9, and V10 into a single community, as represented in Graph-5(b).

Table-5(c) shows the updated Delta Q values after the second merge in Table-5(b), where vertices V8, V9, and V10 were combined into a single community. With this new structure, only the Delta Q values for nodes connected to the newly formed community were recalculated.

Table-5(c)

C_a	C_b	E_{ab}	m	k_a	k_b	Delta Q
1	2	1	13	2	3	**0.0592**
1	3	1	13	2	3	0.0592
2	3	1	13	3	3	0.0503
2	4	1	13	3	2	0.0592
3	5	1	13	3	3	0.0503
4	5	1	13	2	3	0.0592
5	6	1	13	3	3	0.0503
6	7	1	13	3	2	0.0592
6	(8,9,10)	1	13	3	8	0.0059
7	(8,9,10)	1	13	2	8	0.0296

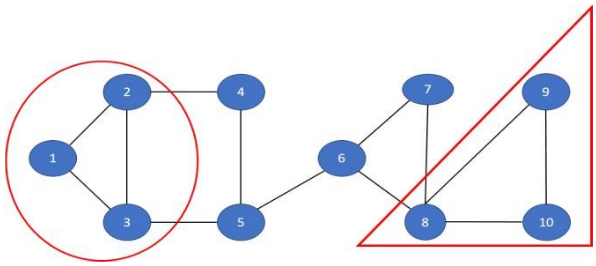


Graph-5(c)

From the results, the highest Delta Q value of 0.0592 is observed for the edge between V1 and V2. This indicates that merging vertices V1 and V2 yields the maximum modularity gain at this stage. Therefore, the next modularity-maximizing step is to merge V1 and V2 into a single community. The updated structure is illustrated in Graph-5(c), where the newly merged community {V1, V2} is highlighted.

Table-5(d)

C_a	C_b	E_{ab}	m	k_a	k_b	Delta Q
(1,2)	3	2	13	5	3	**0.1095**
(1,2)	4	1	13	5	2	0.0473
3	5	1	13	3	3	0.0503
4	5	1	13	2	3	0.0592
5	6	1	13	3	3	0.0503
6	7	1	13	3	2	0.0592
6	(8,9,10)	1	13	3	8	0.0059
7	(8,9,10)	1	13	2	8	0.0296



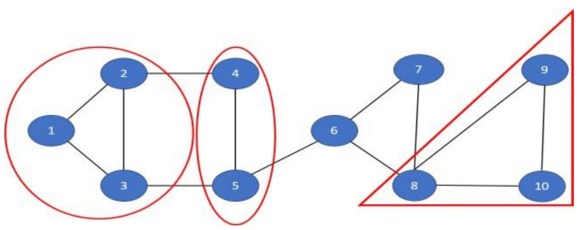
Graph-5(d)

Table-5(d) shows the recalculated Delta Q values after the merge in Table-5(c), where vertices V1 and V2 were grouped together into a single community. Once this merge is completed, the affected nodes are those directly connected to the new community (V1, V2), namely V3 and V4. Hence, Delta Q values are updated only for these connections, while the rest remain unchanged.

From the updated results, the highest Delta Q value of 0.1095 is observed for the merge between the community (V1, V2) and vertex V3. This indicates that merging these vertices provides the maximum modularity gain at this stage. Therefore, the next modularity-maximizing step is to combine V1, V2, and V3 into a single community. The revised network structure is illustrated in Graph-5(d), showing this newly formed community.

Table-5(e)

C_a	C_b	E_{ab}	m	k_a	k_b	Delta Q
(1,2,3)	4	1	13	8	2	0.0296
(1,2,3)	5	1	13	8	3	0.0059
4	5	1	13	2	3	**0.0592**
5	6	1	13	3	3	0.0503
6	7	1	13	3	2	0.0592
6	(8,9,10)	1	13	3	8	0.0059
7	(8,9,10)	1	13	2	8	0.0296



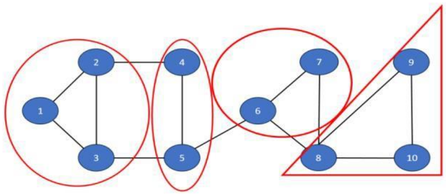
Graph-5(e)

Table-5(e) presents the recalculated ΔQ (Delta Q) values after the merge in Table-5(d), where vertices V1, V2, and V3 were grouped into a single community. Following this merge, the affected nodes are those directly connected to the new community (V1, V2, V3), namely V4 and V5, since they had edges to nodes in the merged set. Therefore, ΔQ values are updated specifically for these connections.

From the recalculations, the highest ΔQ value of 0.0592 is obtained for the edge between V4 and V5. This shows that merging V4 and V5 provides the maximum modularity gain at this stage. Consequently, the next modularity-maximizing step is to combine V4 and V5 into a single community. The resulting structure is illustrated in Graph-5(e), where both communities (V1, V2, V3) and (V4, V5) are now clearly visible alongside the previously formed group (V8, V9, V10).

Table-5(f)

C_a	C_b	E_{ab}	m	k_a	k_b	Delta Q
(4,5)	(1,2,3)	2	13	5	8	0.0355
(4,5)	6	1	13	5	3	0.0325
6	7	1	13	3	2	**0.0592**
6	(8,9,10)	1	13	3	8	0.0059
7	(8,9,10)	1	13	2	8	0.0296



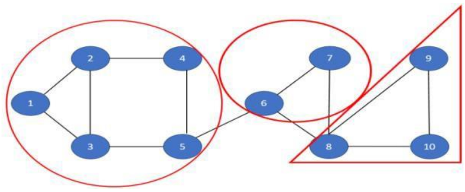
Graph-5(f)

Table-5(f) shows the recalculated ΔQ (Delta Q) values after the merge in Table-5(e), where vertices V4 and V5 were grouped into a single community. Following this merge, the affected nodes are those connected to the new community (V4, V5), namely the community (V1, V2, V3) and vertex V6, since they both had direct edges to V4 or V5. Accordingly, the ΔQ values were updated for these connections.

From the updated results, the highest ΔQ value of 0.0592 is observed for the merge between V6 and V7. This indicates that merging V6 and V7 provides the maximum modularity gain at this stage. Therefore, the next modularity-maximizing step is to combine V6 and V7 into a single community. The corresponding structure is illustrated in Graph-5(f), where the communities (V1, V2, V3), (V4, V5), (V6, V7) and (V8, V9, V10) are now clearly distinguished.

Table-5(g)

C_a	C_b	E_{ab}	m	k_a	k_b	Delta Q
(4,5)	(1,2,3)	2	13	5	8	**0.0355**
(6,7)	(4,5)	1	13	5	5	0.0030
(6,7)	(8,9,10)	2	13	5	8	0.0355



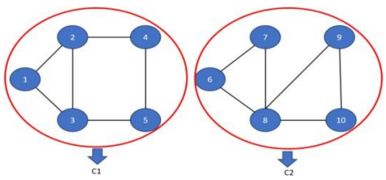
Graph-5(g)

Table-5(g) shows the recalculated ΔQ (Delta Q) values after the merge in Table-5(f), where vertices V6 and V7 were grouped into a single community. Following this merge, the affected nodes are those directly connected to the new community (V6, V7), namely the community (V4, V5) and the community (V8, V9, V10), since both had edges to V6 or V7. Accordingly, the ΔQ values were updated for these pairs.

From the recalculations, the maximum ΔQ value of 0.0355 is observed for the merge between the community (V4, V5) and the community (V1, V2, V3). This indicates that merging these two groups yields the highest modularity gain at this stage. Therefore, the next modularity-maximizing step is to combine V1, V2, V3, V4, and V5 into a single community. The resulting structure is depicted in Graph-5(g), where the larger merged community is now clearly visible along with the previously formed groups (V6, V7) and (V8, V9, V10).

Table-5(h)

C_a	C_b	E_{ab}	m	k_a	k_b	Delta Q
(1,2,3,4,5)	(6,7)	1	13	13	5	-0.1154
(6,7)	(8,9,10)	2	13	5	8	**0.0355**



Graph-5(h)

Table-5(h) shows the recalculated ΔQ (Delta Q) values after the merge in Table-5(g), where the communities (V1, V2, V3, V4, V5) and (V6, V7) were considered along with (V8, V9, V10). At this stage, the affected communities are those connected to (V6, V7), namely (V1, V2, V3, V4, V5) and (V8, V9, V10).

From the Table-5(h), the merge between (V1, V2, V3, V4, V5) and (V6, V7) gives a negative ΔQ (-0.1154), which reduces modularity and is therefore not beneficial. On the other hand, the merge between (V6, V7) and (V8, V9, V10) produces a positive ΔQ value of 0.0355, indicating a modularity gain.

Thus, the optimal step at this stage is to merge (V6, V7) with (V8, V9, V10), resulting in two well-defined final communities:

- C1 = {V1, V2, V3, V4, V5}
- C2 = {V6, V7, V8, V9, V10}

This final partition is illustrated in Graph-5(h), where the network is clearly divided into two cohesive communities.

3.8. Max-Min Analysis

The Max-Min algorithm can be used when detecting communities in a community setting as a heuristic algorithm with the goal of identifying communities that are both isolated and internally connected. As compared to most traditional methods of clustering, which rely on pre-specified parameters or are sensitive to starting configurations, the Max-Min methodology focuses on expanding the distance between communities and ensuring that groups remain compact.

The approach is driven by a greedy approach, which expands the communities around selective original nodes, usually known as seed nodes. New nodes are selected at every step, with maximum distance to current community centers, in order to facilitate a clear division between the communities. Consequently, the identified communities are highly connected internally without being similar to each other.

3.9. Algorithm Steps of the Max-Min

1. Distance Matrix Construction: Compute a pairwise distance or dissimilarity matrix for all nodes in the network. This could be based on shortest path lengths, graph embedding similarity, or structural similarity (like common neighbors or Jaccard similarity).
2. Seed Node Selection: Select the first seed node randomly or based on a centrality measure (like degree or betweenness). Select the next seed node that is at the maximum minimum distance from the already selected seeds.
3.

$$v = arg \max_{u \in V}(\min_{s \in S} d(u, s))$$

Where:

- V is the set of all nodes
- S is the set of already selected seeds

d (u , s) is the distance between node u and s

Repeat this process until K seeds are selected, where K is the desired number of communities

4. Community Assignment: Assign each remaining node to the community whose seed it is closest to (i.e., has the minimum distance).

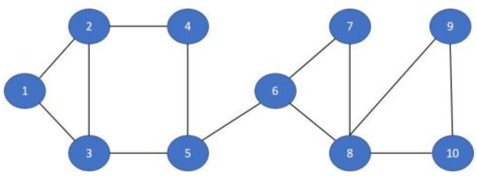
$$C_i\text{-}\{v \epsilon V|\ d(v, s_i) - \min_j d(v, s_j)\}$$

Where C_i is the i-th community and s_j are the seed nodes

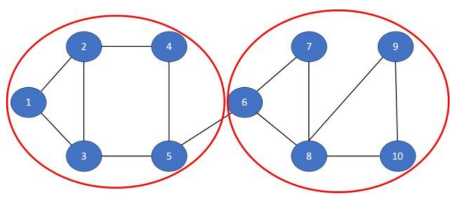
In this technique, the number of communities is predetermined by the user through the selection of seed nodes. The process begins by identifying the seed nodes, after which every remaining vertex is assigned to the community of its nearest seed based on shortest-path distance.

Seed selection strategy:

1. The node with the highest degree is chosen as the first seed (s1).
2. The second seed (s2) is selected as the vertex that lies at the maximum shortest-path distance from s1. In the case of multiple candidates, the node with the smallest label in the sequence is chosen.
3. For each additional seed node (s3,s4), the algorithm selects the vertex whose closest distance to the already chosen seeds is as large as possible. This selection strategy distributes the seeds evenly throughout the network.



Graph-5(a)



Graph-5(b)

Community assignment rule: Having been able to determine the seed nodes, the shortest-path distances between all the nodes and the seed nodes are calculated. A node is then combined in the community with the closest seed node.

Table 6: Distance from Seed Nodes

Node	Distance From Seed-1(node-8)	Distance From Seed-2(node-1)	community selected
1	4	0	c2
2	4	1	c2
3	3	1	c2
4	3	2	c2
5	2	2	c2
6	1	3	c1
7	1	4	c1
8	0	4	c1
9	1	5	c1
10	1	5	c1

As can be seen in Graph-2, node number 8 has the highest degree, thus it is chosen as the first seed (Seed-1). The nearest node is node 8, and the farthest node is node 1, which is taken as the second seed (Seed-2). Since the example has taken only two communities into consideration, the rest of the nodes are attributed to communities, depending on their shortest-path distance to the two seeds. Table-6 presents the calculated distances of Seed-1 (node 8) and Seed-2 (node 1) to every node as well as the respective community assignments. The community structure formed is shown as Graph-5(b). Community C2: {1, 2, 3, 4, 5} Community C1: {6, 7, 8, 9, 10}

This illustrates the effectiveness of the proposed seed-based approach in separating the network into user-specified communities, with the tasks being directed by distance to seeds that are selected strategically.

3.10. Performance Evaluation

To assess the performance of community detection algorithms, various performance measures are usually used, and they measure the accuracy, stability, and scalability. Two of them, modularity and Normalized Mutual Information (NMI), are common metrics used to evaluate the quality of the obtained community structures. All of the metrics shed light on a different aspect of the performance of the algorithm.

3.11. Modularity

One of the measures that is widely embraced for the effectiveness of community partition is called modularity. It assesses the strength of connection of nodes within the same community as compared to connections across communities. The larger the modularity is, the more apparent and well-knit a network is in terms of differing communities.

$$Q = \frac{1}{2m}\sum_{i\neq j} \left(A_{ij} - \frac{v_i v_j}{2m}\right) \delta(C_i, C_j)$$

Where

- A_{ij} is the element of the adjacency matrix A. representing the presence of an edge between nodes i and j
- m is the total number of edges in the network
- v_i *and* v_j represents the degree
- $\delta(C_i,\ C_j)$ is an indicator function from which equals 1 if nodes i and j belong to the same community and 0 otherwise

The modularity score Q lies within the interval [-1,1]. Larger values of Q, commonly above 0.3, suggest a pronounced community organization, while a value of Q=1corresponds to a perfect division in which all connections occur within communities and no links exist between different groups.

3.12. Normalized Mutual Information (NMI)

Normalized Mutual Information (NMI) is a frequently used metric for comparing two different partitions of a network (usually reference (ground truth) communities and those obtained by a detection algorithm). Based on information theory, NMI is used to determine how much information is shared between the two partitions and normalizes it so that it can be fairly used in comparisons between networks of differing sizes and community structure.

NMI (A, B) = $\frac{2.I(A;B)}{H(A)+H(B)}$

Where:

- $I(A; B)$ is the mutual information between two partitions
- H(A)and H(B)are the entropies of A and B

The NMI score ranges from 0 to 1:

1 A value of 1 signifies complete correspondence between the two partitions, meaning that the community structures are the same,

2 A value of 0 indicates completely independent or unrelated partitions.

NMI is especially applicable in cases where the ground truth communities are known (i.e., in benchmark networks), and one wants to quantitatively evaluate the performance of a community detection algorithm. It is invariant to differences in the size and number of communities.

4. RESULT AND ANALYSIS

The evaluation of community detection algorithms is typically done in steps by the researchers. First, they select a popular dataset, e.g. Dolphin social network utilized in the present research. Then they establish the real communities in the dataset to refer to them when making comparisons. The third step involves them examining whether the dataset has excellent community structure by considering such measures as modularity. Lastly, they draw comparisons among various algorithms, both old and new, to determine the performance of each of them according to the chosen evaluation metrics.

4.1. Experimental Results

On the Dolphin datasets, researchers get 10 distinct partitions for the threshold η, which they take at regular intervals ranging from 0.1 to 1.

The outcome is assessed for every segment using the following metrics: NMI, R-measure, modularity Q, extensive modularity eQ, and NOC (number of communities). Table 7 shows 10 experimental results with different threshold values (η) from the Dolphin dataset. From the results, the maximum level of modularity (Q) is achieved when η=0.3 and represents a Q value of 0.488.

Table 7: Experimental findings on the Dolphin dataset under various thresholds.

η	NOC	Q	eQ	*F*-measure	NMI
0.1	2	0.382	0.758	1.000	1.000
0.2	2	0.392	0.781	0.972	0.825
0.3	4	0.488	0.648	0.899	0.685
0.4	6	0.438	0.530	0.803	0.495
0.5	9	0.426	0.489	0.740	0.463
0.6	16	0.328	0.351	0.559	0.375
0.7	18	0.284	0.305	0.583	0.362
0.8	20	0.269	0.285	0.448	0.346
0.9	20	0.267	0.279	0.450	0.340
1.0	21	0.238	0.251	0.435	0.344

Additionally, both the F-measure (0.899) and NMI (0.685) indicate a poor match with ground truth communities. On the other hand, the maximum value of extensive modularity (eQ = 0.781) is observed when η = 0.2. In this scenario, both F-measure (0.972) and NMI (0.825) are significantly high, indicating that the detected communities are highly consistent with the real community structure (as seen in Figure 3).

Modularity optimization can give high QQQ values, but inaccurate community assignments can sometimes be generated by modularity optimization. Conversely, maximum modularity not only achieves a high eQ score, but it also has more similarity with real-life community patterns, and thus is more practical to use. Maximizing extensive modularity, therefore, provides a more middle ground between structural fidelity and accuracy of detection than does the use of modularity maximization alone.

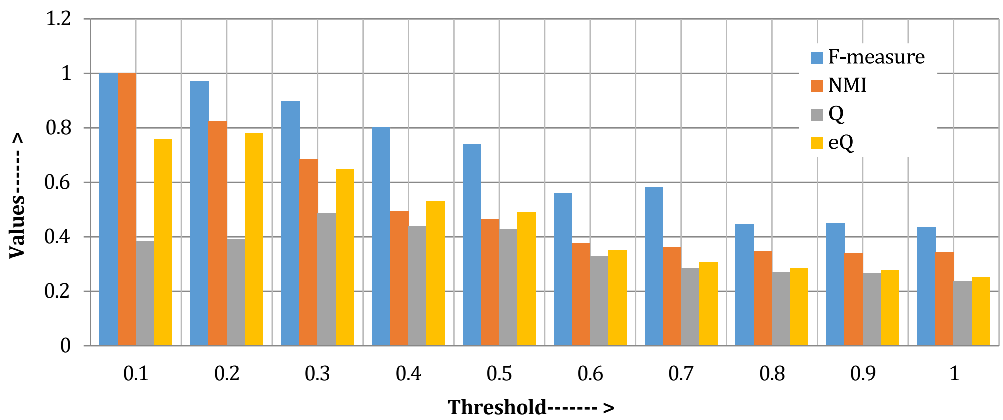


Figure 3: Graph of different thresholds η on the dataset Dolphin

The proposed models identify the communities on the Dolphin network, as shown in Figure 4, where the community structure can be seen.

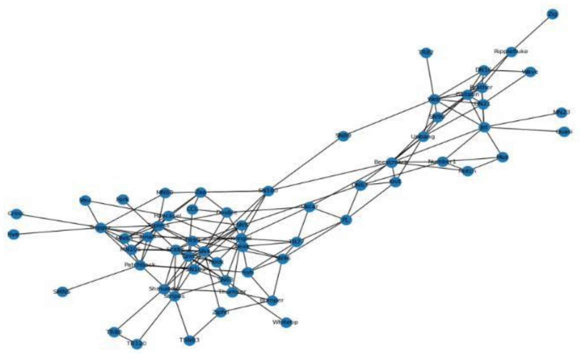


Figure 4: Communities detected in Dolphin

4.2. Performance Evaluation of Discussed Algorithms

As indicated above, extensive modularity is a superior evaluation criterion as compared to standard modularity, and as such, the configuration that gives rise to the greatest value of extensive modularity is selected as the best and final solution.

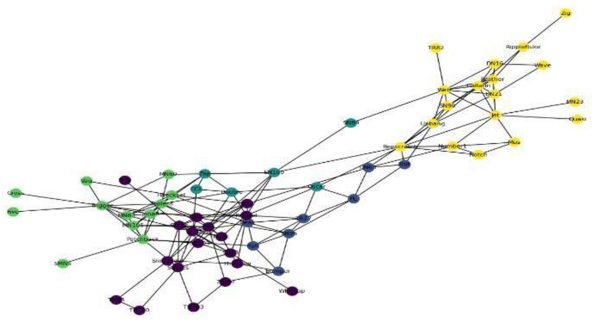


Figure 5: Louvain Community Detection

The following figure (Figure 5) shows the community organization of the Dolphin social network, which was obtained by means of the Louvain community detection algorithm. In this model, the dolphins are represented as the vertices, and a social interaction is represented as a link. The modular structure of the network is quite obvious since nodes of the same color are grouped based on the communities detected by the algorithm. The Louvain method is useful for identifying the strongest internal connectivity groups of dolphins and minimal connectivity between them by maximizing modularity. The visualization achieved indicates the ability of the Louvain approach to identify significant social divisions, which split the network into five communities, which are agreeable to familiar behavioral patterns within the dolphin community.

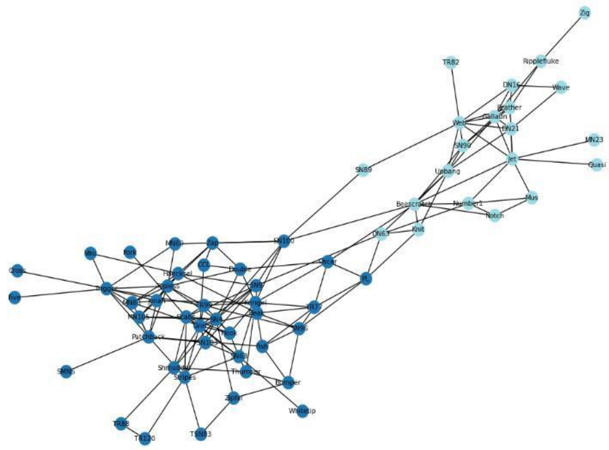


Figure 6: Newman-Girvan Community detection

Figure 6 shows the Dolphin social network partitioned with the Newman-Girvan community detection algorithm, which reveals communities by continuing to remove edges whose betweenness centrality is maximum. In this representation, the nodes are associated with single dolphins, and the edges are connected with social interactions. The network has two separate communities, which are indicated by the colors of nodes: light blue and dark blue, and thus, there is a structural division between the two communities. Here, the emphasis is placed on the bridge edges, which join various parts of the network and reflect coarse-grained divisions. The Newman-Girvan algorithm focuses on larger separations, unlike other algorithms (e.g., Louvain), which result in finer community partitions and potentially simplify the interpretation of underlying social processes.

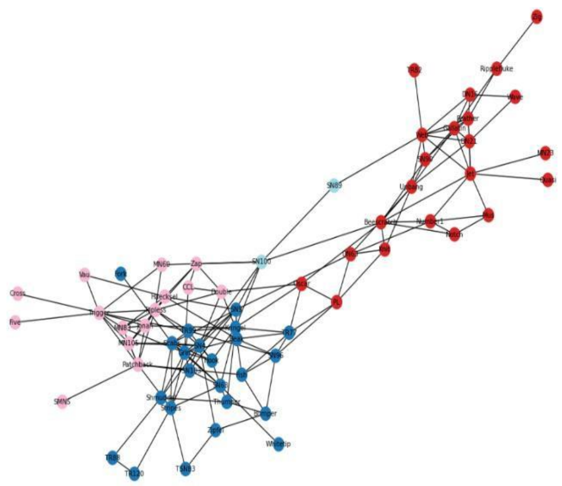


Figure 7: Clauset-Newman-Moore Community Detection

Figure 7 shows the Dolphin social network, which has been computed using the CNM community detection algorithm, which uses the greedy strategy to provide maximum modularity. The following representation displays the social interactions among dolphins and the social network of individuals in terms of vertices and links, respectively. The algorithm divides the network into three distinctly defined communities, with the colors of red, blue, and pink nodes, respectively. CNM forms communities hierarchically by sequentially fusing community pairs that lead to the largest growth in modularity. In doing so, the technique actually uncovers the internal social structure to produce a balanced partition that balances the strongly bonded core groups and the smaller marginal groups, hence giving an educative insight into dolphin social interactions.

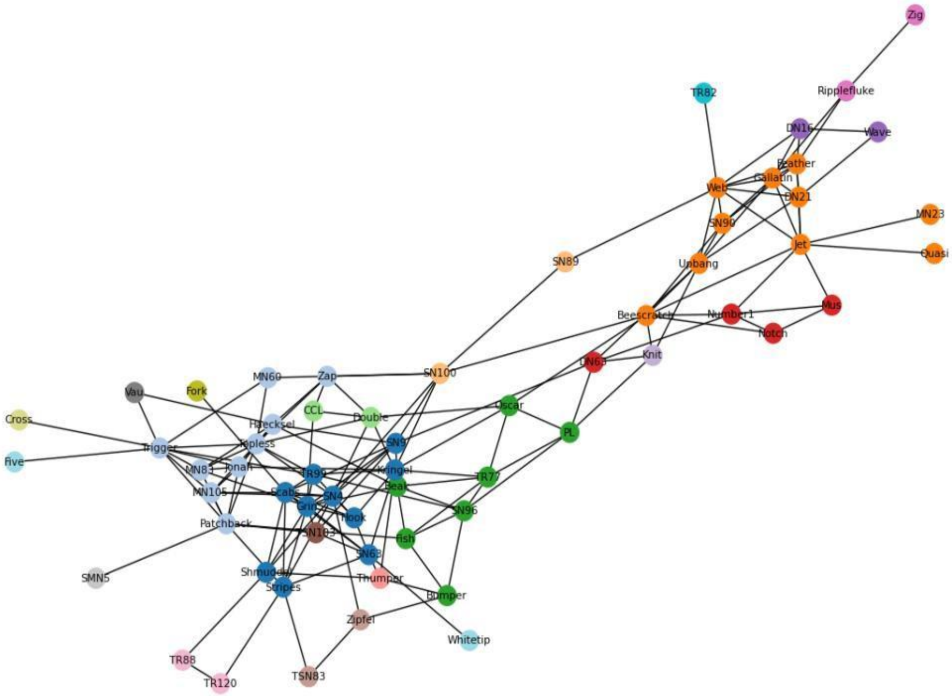


Figure 8: Max-Min Community Detection

Figure 8 shows the Dolphin social network using the Max-Min community detection method, which tends to optimize the minimum similarity within a community. In this visualization, the nodes represent dolphins, and the edges represent interactions between the dolphins, and the various colors are used to represent different smaller communities. The method records fine-grained partitions and identifies delicate but important social groupings, which might not be observed using methods that are less fine-grained. Its resulting structure gives a close-up of dolphin interactions with central and peripheral subgroups of nodes in the network.

Table 8: Algorithm experimental findings on the Dolphin dataset

Algorithm	Modularity Scores	Number of Communities Detected	NMI
Louvain Algorithm	0.5188	5	0.911
Newman-Girvan Algorithm	0.3787	2	0.693
Clauset-Newman-Moore Algorithm	0.4955	4	0.814
Max-Min Algorithm	0.4014	21	0.580

A comparative summary of the performance of four community detection algorithms, namely Louvain, Newman-Girvan, CNM, and Max-Min, on the Dolphin social network is displayed in Table 8. The Louvain algorithm has the highest modularity value of 0.5188, and its NMI score is 0.911, which implies that it provides a clear partition of five communities. Generally, values of NMI, which exceed 0.5, are said to indicate a good degree of correspondence with ground-truth community structures. By comparison, the Newman-Girvan algorithm finds only two communities, but this has a lower modularity value of 0.3787 and a moderate NMI value of 0.693; however, gives a decent correspondence to the known community divisions.

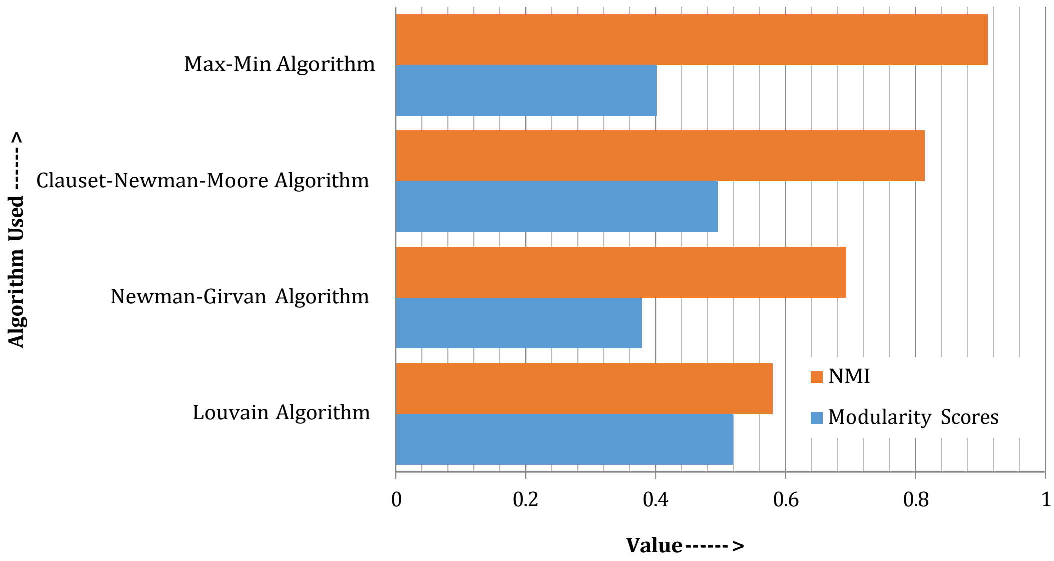


Figure 9: Experimental comparison of community detection algorithms on the Dolphin dataset

The CNM algorithm performs well with the modularity level of 0.4955 and the relatively high NMI =0.814, yielding four distinct communities. On the other hand, the Max-Min algorithm scores lower modularity (0.4014) and finds more communities (21), yet the NMI score is 0.580 meaning that the algorithm is more consistent with the real community structure than the other methods as illustrated in Figure 9. This finding proves that the Max-Min algorithm could create finer-grained social relations in the network.

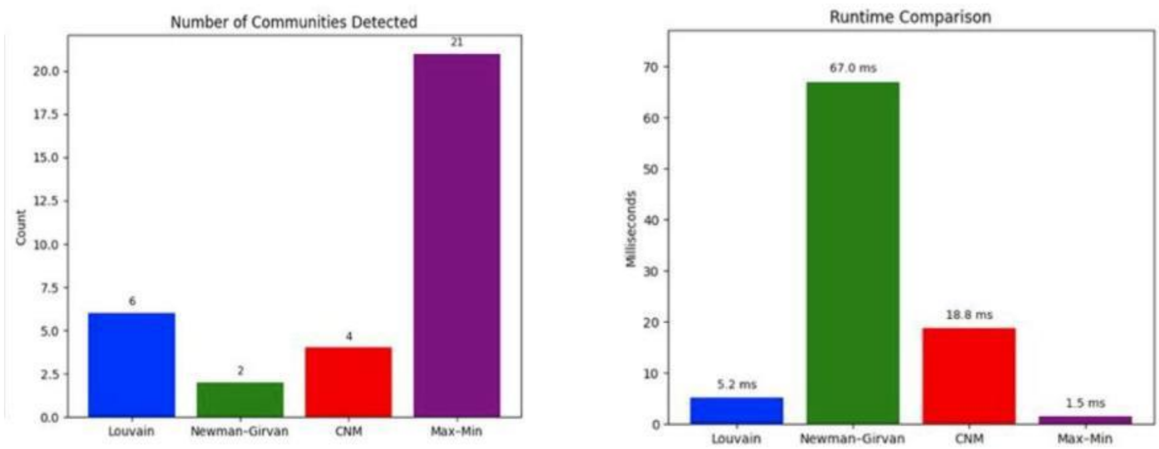


Figure 10: Communities Detected and Time Complexity

Figure 10 compares the results of the Louvain, Girvan-Newman, CNM, and Max-Min algorithms based on the number of communities that each algorithm identifies and the time taken by the algorithm. It is compared that all methods differ greatly in their partitioning behaviour and computational efficiency.

Regarding the communities found, the Max-Min algorithm has the highest number of communities found, 21. This behavior is caused by the fact that its design is that the community formation is based on seed selection and assignment based on distance. This malleability would be useful in determining the finer structure of communities, although it can also oversize against known ground-truth communities. Instead, the Louvain algorithm identifies six communities that are more consistent with the estimated anticipated modular organization of the network. The Girvan-Newman algorithm and CNM algorithm have weaker results because they detect two and four communities, respectively, and this implies that they are more inclined to generalized partitions.

The algorithms also have a differentiation in the analysis of the execution time. Girvan-Newman is the most expensive to compute, requiring a runtime of 67.0ms, but this is largely because of the repetitive recalculation of edge betweenness centrality during every iteration. The CNM algorithm is more efficient and takes only 18.8ms to complete the execution of the algorithm. Louvain is again better at 5.2ms, and this is credited to its greedy and hierarchical optimization of modularity. The Max-Min algorithm has the lowest runtime, 1.5ms, because the assignment algorithm used is distance-based; hence, it does not repeat the calculation of modularity or centrality.

In general, the results indicate that the Louvain algorithm also has a favorable trade-off between the quality of the detection and efficiency. It creates well-consistent community structures, which are ground truth, as well as low execution time. Although the Girvan-Newman algorithm is conceptually important, it is computationally expensive, so it cannot be applied to large networks. CNM is not a modular solution, but it is not as fast as Louvain is, however. The most appropriate approach in the scenario when great importance is given to the speed of computation and control of the community by the users is the Max-Min approach that offers the best execution and the best-grained partitions.

5. CONCLUSION

This study introduced and compared four popular community methods, which are Louvain, Girvan-Newman, Clauset-Newman-Moore (CNM), and Max-Min, on the Dolphin social network as a reference data set. As it has been experimentally demonstrated, the Louvain algorithm provides the best overall performance, with a modularity of 0.5188 and an NMI of 0.911, that is, a well-defined internal community structure and a high level of conformity to the known network partition. The CNM approach also exhibits competitive performance, providing a fair trade-off between modularity maximization and accuracy of detection. Despite the fact that the Max-Min method is efficient in revealing finer and detailed patterns in the community, its relatively lower modularity and NMI values reflect that this method is more appropriate to unearth smaller and localized group structures than global ones. Girvan-Newman algorithm generates meaningful partitions based on edge betweenness, but with higher computational costs, which makes it less scalable to larger networks. In general, the results highlight the importance of scalable and context-sensitive community detection algorithms in the context of studying complex, dynamic, and possibly overlapping structures such as those found in Online Social Networks (OSNs).

Author 1: **Srinivas Amedapu** is a full-time Ph.D. Research Scholar at the National Institute of Technology (NIT), Tiruchirappalli, Tamil Nadu, India. He completed his B.E. in Computer Technology from Nagpur University and his M.Tech from Jawaharlal Nehru Technological University (JNTU), Hyderabad. He has over 20 years of teaching experience in the field of education. His research interests include Social Network Analysis and Big Data Analytics.

Author 2: **Professor Dr. (Mrs.) Leela Velusamy R** is a Professor in the Department of Computer Science and Engineering at the National Institute of Technology (NIT), Tiruchirappalli, Tamil Nadu, India. She has also served as the Head of the Department of Computer Science and Engineering at NIT Trichy. She has published numerous research papers in reputed international journals such as IEEE, Elsevier, Wiley, and Springer. Her research interests include Mobile Ad Hoc Networks, Social Networks, and Quality of Service (QoS) in Routing.

5.1. Declarations

1. Ethics approval and consent to participate: Not applicable.
2. Consent for publication: All authors have read and approved the final manuscript and consent to its publication.
3. Clinical trial number: Not applicable.
4. Clinical trial registration: Not applicable.
5. Availability of data and materials: The Dolphin Social Network dataset is publicly available in the Network Repository at https://networkrepository.com/soc-dolphins.php
6. Competing interests: Not applicable.
7. Funding: No funding was received for this study.
8. Authors' contributions: Srinivas Amedapu conceptualized the study, implemented the community detection algorithms, and performed the comparative analysis. He analyzed the results, prepared the figures and tables, and drafted the manuscript. The work was carried out under the supervision of Dr. Leela Velusamy R, who provided overall guidance, critical review, and valuable suggestions for improving the research and manuscript quality.
9. Acknowledgements: Not applicable.

REFERENCES

[1] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75-174, 2010.

[2] M. E. J. Newman, *Networks: An Introduction*. Oxford, U.K.: Oxford Univ. Press, 2010.

[3] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 105, no. 4, pp. 1118-1123, 2008.

[4] D. Kempe, J. Kleinberg, et É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, 2003, pp. 137-146.

[5] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM SIGKDD Explorations*, vol. 1, no. 2, pp. 47-53, 2007.

[6] A. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761-764, 2010.

[7] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 99, no. 12, pp. 7821-7826, 2002.

[8] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, vol. 70, no. 6, Art. no. 066111, 2004.

[9] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech.: Theory Exp.*, vol. 2008, no. 10, Art. no. P10008, 2008.

[10] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 104, no. 1, pp. 36-41, 2007.

[11] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Phys. Rev. E*, vol. 80, no. 5, Art. no. 056117, 2009.

[12] S. Sushma, S. K. Nayak, and M. V. Krishna, "Enhanced toxic comment detection model through deep learning models using word embeddings and transformer architectures," *Future Technology*, vol. 4, no. 3, pp. 76-84, 2025.

[13] Y. Zhang, "Media framing and public risk communication using deep learning," *Future Technology*, vol. 4, no. 3, pp. 227-238, 2025.

[14] C. Chen, N. H. B. Zakaria, W. Deng, X. Xu, and Y. Xu, "AI-driven marketing innovation in educational technology," *Future Technology*, vol. 4, no. 4, pp. 146-158, 2025.

[15] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground truth," *Knowl. Inf. Syst.*, vol. 42, no. 1, pp. 181-213, 2015.

[16] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75-174, 2010.

[17] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 103, no. 23, pp. 8577-8582, 2006.

[18] A. De Santo, G. De Pietro, and A. Rinaldi, "A deep learning approach for semi-supervised community detection in online social networks," *Knowledge-Based Systems*, vol. 213, Art. no. 106683, 2021.

[19] M. Ali, A. Hussain, and M. Khan, "Social media content classification and community detection using deep learning," *Technological Forecasting and Social Change*, vol. 188, Art. no. 122292, 2023.

[20] A. Ferraro, D. Di Lorenzo, and A. Petrosino, "Deep learning-based community detection on multimedia social networks," *Applied Sciences*, vol. 11, no. 6, Art. no. 2584, 2021.

[21] O. Rashnodi, M. Jalili, and H. Alizadeh, "Community detection in attributed social networks using deep learning," *J. Supercomput.*, vol. 80, pp. 1-24, 2024.

[22] D. Jin, H. Wang, P. Li, and D. He, "Using deep learning for community discovery in social networks," in *Proc. IEEE Int. Conf. Tools with Artificial Intelligence (ICTAI)*, 2017, pp. 605-612.

[23] M. Asadi and F. Ghaderi, "Incremental community detection in social networks," in *Proc. IEEE Conf. FRUCT*, 2018, pp. 25-32.

[24] P. Holme and J. Saramäki, "Temporal networks," *Physics Reports*, vol. 519, no. 3, pp. 97-125, 2012.

[25] A. Sarswat, A. Sharma, and R. Kumar, "A novel two-step approach for overlapping community detection," *Social Netw. Anal. Mining*, vol. 7, no. 1, Art. no. 47, 2017.

[26] F. Huang, S. Zhang, and W. Li, "Overlapping community detection for multimedia social networks," *IEEE Trans. Multimedia*, vol. 19, no. 12, pp. 2856-2869, 2017.

[27] N. Kasoro, R. U. Khan, and M. Shahzad, "PercoMCV: A hybrid approach of community detection," *Procedia Computer Science*, vol. 151, pp. 306-313, 2019.

[28] K. Georgiou, M. Papoutsidakis, and E. Stiakakis, "A distributed hybrid community detection methodology," *Algorithms*, vol. 12, no. 8, Art. no. 168, 2019.

[29] A. Azaouzi, S. Rhouma, and L. B. Said, "A comprehensive literature review on community detection," *Procedia Computer Science*, vol. 159, pp. 1069-1078, 2019.
[30] C. Chen, N. Zakaria, and W. Deng, "AI-driven marketing innovation and user clustering," *Future Technology*, 2025.
[31] S. Sushma, S. K. Nayak, and M. V. Krishna, "Deep learning models for social media behavior analysis," *Future Technology*, 2025.
[32] Y. Wang and H. Hamid, "AI-informed competency modeling and clustering," *Future Technology*, 2025.
[33] E. Dritsas, A. Trigka, and M. V. Giannakis, "Explainable analytics in social network data," *IEEE Access*, vol. 8, pp. 164212-164225, 2020.
[34] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech.: Theory Exp.*, vol. 2008, no. 10, Art. no. P10008, 2008.