International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 31 Issue 05 – September 2025.

ANALYSIS OF COMMUNITY DETECTION ALGORITHMS: GIRVAN–NEWMAN, K-CLIQUES, AND CHINESE WHISPERS

D. Dhanalakshmi[#1] , G. Rajendran*[2]

[#1]Research Scholar, Periyar University, Salem -11 and Assistant Professor of Computer Science and Applications, Vivekanandha College of Arts and Sciences for Women(Autonomous), Tiruchengode, Namakkal, Tamilnadu, India.
[#2]Associate Professor and Head, Department of Computer Science, Govt. Arts and Science College, Modakkurichi, Erode, Tamilnadu, India

***Abstract* - Community detection plays a crucial role in understanding the structural organization of complex networks such as social, biological, and information systems. This paper provides a comprehensive analysis of three prominent community detection algorithms—Girvan–Newman (GN), K-Cliques, and Chinese Whispers (CW)—focusing on their methodologies, computational complexities, and effectiveness in various network scenarios. The Girvan–Newman algorithm, based on edge betweenness and hierarchical clustering, efficiently identifies well-separated communities but is limited by its high computational cost and inability to detect overlapping structures. The K-Cliques algorithm leverages clique percolation to uncover overlapping communities, making it suitable for real-world networks with dense interconnections, albeit with significant computational expense for large graphs. In contrast, the Chinese Whispers algorithm utilizes local label propagation for rapid, unsupervised clustering, excelling in scalability and execution speed but lacking in modularity and overlap detection. Experimental comparisons using modularity (Q), Normalized Mutual Information (NMI), and execution time (T) demonstrate distinct trade-offs between accuracy, scalability, and community overlap detection. The results highlight that while GN is ideal for small hierarchical networks, K-Cliques are best for overlapping community detection, and CW is preferable for large-scale, real-time applications.**

Index Terms- Community Detection, Girvan–Newman Algorithm, K-Cliques, Chinese Whispers, Overlapping Communities, Graph Clustering, Network Analysis, Modularity, Normalized Mutual Information (NMI), Execution Time, Label Propagation, Clique Percolation, Scalability.

I. INTRODUCTION

In recent years, the study of complex networks has become one of the most significant areas of research in data science, social computing, and artificial intelligence. Networks are powerful tools for modeling relationships among entities—ranging from social interactions and biological processes to information systems and communication infrastructures. Within these networks, communities or clusters represent groups of nodes that are more densely connected to each other than to the rest of the network[1] [2]. Detecting such communities is crucial for understanding the underlying structural and functional organization of the system. Community detection algorithms aim to uncover these modular structures by analyzing topological patterns in the network. They help identify influential groups in social networks, functional modules in biological systems, or related documents in citation networks[3]. However, due to the diversity and scale of real-world networks, designing efficient and accurate community detection methods remains a challenging problem. Factors such as overlapping communities, hierarchical organization, and dynamic network evolution make the task even more complex. Over the years, several algorithms have been developed to address these challenges, each based on distinct theoretical principles and computational strategies. Among them, three representative methods stand out for their conceptual clarity and broad applicability—Girvan–Newman (GN), K-Cliques, and Chinese Whispers (CW) algorithms.

- The Girvan–Newman algorithm identifies communities through edge betweenness centrality, progressively removing edges that bridge different communities. It is highly effective for detecting hierarchical structures in small to medium-sized networks but computationally expensive for large-scale graphs.
- The K-Cliques algorithm uses clique percolation to identify overlapping communities, where nodes can belong to multiple clusters. This makes it ideal for real-world social or biological networks with complex overlapping relationships.
- The Chinese Whispers algorithm, on the other hand, employs local label propagation for fast, unsupervised community detection. It is particularly suitable for large and dynamic datasets due to its simplicity and scalability.

Each of these algorithms demonstrates unique strengths and limitations depending on network topology, size, and density[4]. While the Girvan–Newman method

emphasizes accuracy and hierarchy, K-Cliques focus on overlapping modularity, and Chinese Whispers prioritize speed and scalability. Thus, a comparative analysis of these methods provides deeper insight into their applicability across different domains and network conditions.

This paper presents a detailed comparison of the Girvan–Newman, K-Cliques, and Chinese Whispers algorithms in terms of their approaches, computational complexities, modularity performance, NMI scores, and execution time. The goal is to evaluate their efficiency and suitability for various types of network data. The study also discusses the advantages, trade-offs, and best-use scenarios of each algorithm, offering a comprehensive understanding of their behavior in real-world applications.

II. RELATED WORKS

Community detection has been a central topic in network science, with extensive research focusing on identifying hidden structures within complex graphs. Over the past two decades, numerous algorithms have been developed, each adopting different strategies to reveal community boundaries and overlapping structures. These approaches can generally be categorized into hierarchical, clique-based, and label-propagation methods, which form the foundation of the algorithms compared in this paper—Girvan–Newman (GN), K-Cliques, and Chinese Whispers (CW).

The Girvan–Newman algorithm, introduced by Girvan and Newman (2002), marked a major breakthrough in community detection by proposing an edge-betweenness-based hierarchical clustering approach. The algorithm assumes that edges connecting distinct communities have higher betweenness values, as they lie on many of the shortest paths between nodes. By iteratively removing these high-betweenness edges, the network gradually splits into distinct communities [5]. Subsequent studies by Newman (2004) and Fortunato (2010) have emphasized the algorithm's ability to uncover hierarchical structures and its strong interpretability. However, despite its accuracy, its computational complexity of $O(n^3)$ restricts its scalability to large networks, making it suitable primarily for small to medium datasets such as social, citation, and biological networks [6].

In contrast, clique-based methods emerged to address the limitation of non-overlapping partitions. The K-Cliques algorithm, introduced by Palla et al. (2005), is based on clique percolation, which allows the detection of overlapping communities—a key characteristic of many real-world networks. In this method, communities are defined as chains of k-cliques (fully connected subgraphs of k nodes) that share k–1 nodes. This approach reflects the natural overlap seen in social networks, where individuals often belong to multiple groups [7]. Subsequent research, including Evans (2010) and Xie et al. (2013), demonstrated that clique-based methods effectively capture dense substructures but suffer from high computational costs, especially when identifying larger cliques in sparse networks. Despite these challenges, K-Cliques remain widely adopted in biological systems, citation analysis, and overlapping community studies [8].

The Chinese Whispers algorithm, proposed by Biemann (2006), represents a shift toward label propagation and stochastic clustering techniques. It operates by assigning unique labels to nodes and iteratively updating them based on neighboring labels and edge weights. Unlike deterministic methods, CW introduces randomness in label updates, leading to faster convergence and improved scalability [9]. Research by Cordasco and Gargano (2010) and Raghavan et al. (2007) compared CW to other label propagation algorithms and confirmed its linear-time complexity ($O(n+m)$), making it suitable for large-scale, real-time, and dynamic networks such as natural language processing (NLP) and social media analysis. However, CW produces non-overlapping communities and may yield slightly lower modularity and NMI scores compared to hierarchical or clique-based algorithms [10].Beyond these three major techniques, researchers have explored several other paradigms for community detection, including modularity optimization (Blondel et al., 2008; the Louvain method), spectral clustering (Newman, 2006), genetic algorithms (Pizzuti, 2008), and deep learning-based methods (Cavallari et al., 2017). These approaches extend the applicability of community detection to high-dimensional and dynamic environments. Nonetheless, traditional algorithms such as GN, K-Cliques, and CW remain foundational benchmarks due to their interpretability and well-understood mathematical formulations[11] [12] [13]. Overall, the existing literature demonstrates that while Girvan–Newman excels in precision and hierarchical analysis, K-Cliques effectively handle overlapping structures, and Chinese Whispers provide scalability for large datasets. The comparative evaluation of these three methods, as presented in this study, contributes to a deeper understanding of their trade-offs, computational performance, and practical applications across different network domains.

III. COMMUNITY DETECTION METHODS

3.1. Girvan-Newman Algorithm

The Girvan-Newman (GN) algorithm is a hierarchical community detection method that identifies communities by progressively removing edges with the highest edge betweenness centrality. Introduced by Girvan and Newman [14], it is based on the idea that edges connecting different communities have high betweenness scores, meaning they are frequently used in shortest paths between nodes. By iteratively removing these high-betweenness edges, the network is gradually split into smaller, densely connected communities.

Girvan-Newman Algorithm:

Step 1: Compute Edge Betweenness Centrality (EBC)

- Edge betweenness centrality measures how frequently an edge appears in the shortest paths between all pairs of nodes in the network.
- The betweenness score for an edge e is given by:

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

Where, σ_{st} is the total number of shortest paths between nodes s and t. $\sigma_{st}(e)$ is the number of shortest paths that pass through edge e**.** High betweenness edges act as bridges between communities and are the first candidates for removal.

Step 2: Identify and Remove the Edge with the Highest Betweenness

- Find the edge **e*** with the highest betweenness score:

$$e^* = \arg max_e EB(e)$$

- Remove **e*** from the network:

$$G' = G - e^*$$

- Recompute the edge betweenness scores after each removal.

Step 3: Repeat Until the Network is Partitioned into Communities

- Continue removing the highest betweenness edges until the graph breaks into disjoint components (communities)**.**
- The process stops when all edges with high betweenness are removed**,** leaving clusters of nodes that are densely connected internally.

Step 4: Evaluate the Community Structure Using Modularity

- To determine the best partitioning, the algorithm calculates modularity (Q) at each step.
- Modularity is a measure of the quality of a partition, defined as:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(C_i, C_j)$$

Where, A_{ij} is the adjacency matrix. k_i and k_j are the degrees of nodes i and j. m is the total number of edges in the network. $\delta(C_i,C_j)$ is 1 if **i** and **j** are in the same community**,** otherwise 0**.** The partition with the highest modularity value is chosen as the optimal community structure**.**

The GN algorithm is a hierarchical clustering method that detects communities in a network by progressively removing edges with high Edge Betweenness Centrality (EBC). The underlying assumption is that edges connecting different communities act as bridges, frequently appearing in shortest paths between nodes. By iteratively removing these high-betweenness edges, the network gradually splits into densely connected communities.

The algorithm begins by computing the EBC, which quantifies how often an edge is used in shortest paths between all pairs of nodes. Edges with high betweenness scores are removed first, as they are likely to be inter-community links. After each removal, the edge betweenness scores are recomputed, ensuring that the next edge to be removed remains the most critical for community separation. This iterative process continues until the network is completely partitioned into distinct communities. To determine the best community structure, the algorithm evaluates modularity (Q) at each step. Modularity measures how well-defined the detected communities are compared to a randomly connected network. The partition with the highest modularity is selected as the final community structure, ensuring optimal separation of groups [15]. Despite its effectiveness in detecting well-separated communities, the Girvan-Newman algorithm has some significant limitations. The biggest drawback is its computational complexity of $O(n^3)$, making it unsuitable for large networks. Additionally, because edges are removed, the network structure is permanently altered, meaning it cannot detect overlapping communities. However, GN remains a useful method for small- to medium-sized networks, particularly in social networks, biological networks and citation networks, where understanding the hierarchical nature of communities is important. The GN algorithm is a hierarchical clustering method that identifies communities by progressively removing high-betweenness edges. It effectively uncovers hierarchical structures within networks, making it useful for small-scale social, biological and citation networks. However, due to its high computational cost ($O(n^3)$), it is not suitable for large networks. Despite this, GN remains an important method in community detection, offering valuable insights into network topology and the role of critical connections in information flow.

3.2. K-Cliques Algorithm

The K-Cliques algorithm is a community detection method that identifies overlapping communities in a network by finding k-cliques—fully connected subgraphs of k nodes—and merging them if they share k−1 nodes. Unlike modularity-based methods, which assume non-overlapping communities, K-Cliques percolation enables nodes to belong to multiple communities, making it highly effective for real-world networks with overlapping structures (e.g., social networks, biological systems)[16].

K-Cliques Algorithm:

Step 1: Identify All K-Cliques in the Graph

- A k-clique is a complete subgraph of size k, meaning that every node in the clique is connected to every other node.
- The condition for a k-clique in an adjacency matrix A is:

$$A_{ij} = 1, \forall_{i,j} \in C, i \neq j$$

Where, C is the set of nodes in the clique. The set of all k-cliques in the graph is denoted as K(G,k).

Step 2: Construct the K-Clique Adjacency Graph

- Two k-cliques C_i and C_j are considered adjacent if they share k−1 nodes**:**

$$\left|C_i \cap C_j\right| = k - 1$$

- This adjacency relationship defines a new graph, called the k-clique adjacency graph, where:
  - Nodes represent k-cliques**.**
  - Edges exist between k-cliques if they share k−1 nodes**.**

Step 3: Identify Connected Components in the K-Clique Adjacency Graph

- A community is defined as a maximally connected component in the k-clique adjacency graph**.**
- The task reduces to finding connected components in this graph using graph traversal algorithms (e.g., BFS or DFS).

$$C = \bigcup_{C_i \in K(G,k)} C_i$$

Where, C represents a detected community. Nodes belonging to multiple k-cliques are part of multiple communities**,** leading to overlapping community detection**.**

Step 4: Output the Overlapping Community Structure

- The final output is a set of communities, where each node may belong to multiple communities**.**
- Unlike traditional modularity-based approaches, which assume strict partitions**,** K-Cliques naturally allow overlapping communities**.**

The K-Cliques algorithm is an overlapping community detection method that identifies groups of densely connected nodes in a network. Unlike traditional approaches that force non-overlapping partitions, K-Cliques recognizes that real-world networks often exhibit overlapping structures—for example, a person in a social network may belong to multiple friend groups. The algorithm works by first identifying all k-cliques in the graph, where a k-clique is a fully connected subgraph of k nodes. Next, it constructs a k-clique adjacency graph, where each node represents a k-clique and edges exist between k-cliques that share k−1 nodes. This step ensures that closely related k-cliques are linked together, capturing local community structures. Once the k-clique adjacency graph is built, the algorithm extracts connected components, treating each component as a community. This means that communities are formed by chains of k-cliques that percolate through the network, rather than being strictly isolated clusters. Importantly, nodes can belong to multiple k-cliques, allowing the natural detection of overlapping communities [17]. One of the major advantages of K-Cliques is its ability to capture overlapping structures, making it ideal for social networks, biological networks and citation graphs, where nodes frequently belong to multiple groups. However, the algorithm has some computational limitations, as the process of finding all k-cliques can be expensive, with a worst-case complexity of $O(n^k)$. As a result, it is best suited for moderately sized networks or cases where overlapping communities are essential for analysis.

The K-Cliques algorithm is a highly effective method for detecting overlapping communities by finding chains of k-cliques that percolate through the network. Unlike modularity-based approaches, which force strict partitions, K-Cliques naturally captures nodes that belong to multiple communities, making it particularly useful for social networks, biological systems and citation networks. While K-Cliques provides an intuitive way to detect densely connected groups, its computational complexity $O(n^k)$ can be expensive, especially for large networks. However, for moderate-sized graphs where overlapping communities are important, K-Cliques remain one of the most effective and widely used community detection methods.

3.3. Chinese Whispers Algorithm

The Chinese Whispers (CW) algorithm is an unsupervised graph clustering method that detects communities by iteratively propagating labels among neighboring nodes. It is similar to the LPA but differs in how it updates labels. Unlike LPA, which selects the most frequent neighboring label, Chinese Whispers uses randomized label updates, making it fast and scalable for large networks. It is commonly applied in Natural Language Processing (NLP), SNA and biological networks [18].

Chinese Whispers Algorithm:

Step 1: Initialize Each Node with a Unique Label

- Every node v in the network is assigned a unique initial label**:**

$$L(v) = v, \forall_v \in V$$

Where, V is the set of all nodes in the graph.

Step 2: Iterate Over Nodes in Random Order

- The algorithm randomly selects a node v at each iteration.
- The node updates its label based on neighboring labels, but instead of choosing the most frequent label (as in LPA), it selects a label probabilistically based on the weighted sum of its neighbors' labels:

$$L(v) = arg\ max_i \sum_{u \in N(v)} I(L(v) = l)$$

Where, N (v) is the set of neighbors of node v. wuv is the edge weight between nodes u and v. I(L(u)=l) is an indicator function that returns 1 if node **u** has label **l**, otherwise 0**.** Unlike LPA, where label updates are deterministic, Chinese Whispers introduces randomness in label updates to avoid local optima.

Step 3: Repeat until Convergence

- The algorithm continues updating labels until no further label changes occur or a maximum number of iterations is reached. The stopping condition is:

$$L_t(v) = L_{t+1}(v), \forall_v \in V$$

Where, $L_t(v)$ is the label at iteration t and $L_{t+1}(v)$ is the label at iteration t+1.

Step 4: Extract Communities Based on Final Labels

- After convergence, nodes sharing the same label form a community**.**
- The output consists of disjoint communities where each node is assigned to a single cluster.

The CW algorithm is a graph-based clustering method that partitions a network by iteratively propagating labels among neighboring nodes. The name "Chinese Whispers" comes from the idea that information spreads and evolves locally, leading to emergent structures. It is widely used for NLP, social network analysis and biological data clustering due to its fast and unsupervised nature. The algorithm begins by assigning each node a unique label, treating every node as its own community. Then, in each iteration, nodes randomly update their labels based on the labels of their neighbors [19]. Chinese Whispers updates labels in a probabilistic manner, using edge weights to affect label choices, in contrast to the LPA, which chooses the most frequent label. This randomness helps prevent local optima, ensuring a more natural clustering process. Chinese Whispers is highly efficient, with a time complexity of O(n+m), making it well-suited for large-scale networks. However, due to its stochastic nature, results may vary across different runs, requiring multiple executions to obtain stable clusters. Additionally, it only detects disjoint communities and does not support overlapping clusters, unlike the K-Cliques algorithm. Despite these limitations, Chinese Whispers remains a powerful clustering tool, especially for real-time applications where speed and simplicity are essential. Its ability to detect patterns in large networks with minimal computation makes it a valuable algorithm in graph-based machine learning, NLP entity recognition and social network segmentation.

The CW algorithm is a fast and efficient graph clustering method based on local label propagation. It iteratively updates node labels based on neighboring labels and edge weights, allowing clusters to emerge organically. Unlike the LPA, CW introduces randomness in label updates, preventing premature convergence to local optima. This makes it particularly useful for large-scale, real-time applications such as social network segmentation, NLP entity clustering and biological data analysis. Despite its strengths, CW has limitations, including non-deterministic results and the inability to detect overlapping communities. However, its speed and simplicity make it an attractive choice for unsupervised clustering in large networks.

IV.RESULTS AND DISCUSSION

The experimental setup for evaluating community detection methods involves testing their performance on diverse real-world datasets, comparing them with established algorithms, and assessing key evaluation metrics. The evaluation is conducted on four real-world network datasets, each representing different types of interactions. The Reddit Hyperlink Network (RH-NW) captures hyperlink relationships between subreddits and includes temporal interactions, making it suitable for dynamic community detection. The Amazon Co-purchasing Network (ACP-NW) represents product co-purchasing relationships, helping analyze hierarchical structures in e-commerce networks. The DBLP Collaboration Network (DBLP-NW) models academic collaborations based on co-authored research papers, making it useful for detecting research communities. The Twitch Gamers Network (TG-NW) showcases interactions among Twitch users, including demographic attributes, and is ideal for testing overlapping community detection methods. These datasets provide a robust and varied platform for evaluating the effectiveness and scalability of community detection algorithms.

Table 1 provides a comparative analysis of various community detection algorithms, outlining their approaches, strengths, weaknesses, scalability, ability to handle overlapping communities and ideal use cases.

Table 1 Comparison of Existing Community Detection Algorithms

Algorithm	Approach	Strengths	Weaknesses
Girvan-Newman (GN)	Edge Betweenness and Hierarchical Clustering	Identifies hierarchical community structures; Good for small networks	Not scalable; Cannot detect overlapping community
K-Cliques	Clique Percolation	Works well in dense networks, capturing highly modular substructures Flexibility in community size detection.	Computationally expensive for large and sparse networks. .
Chinese Whispers	Local Label Propagation	Unsupervised and parameter-free, reducing the need for tuning.	Lower modularity and NMI scores, making it less effective for high-quality community detection.

Community detection algorithms are evaluated based on various metrics to determine their effectiveness in uncovering meaningful structures in networks. The following key evaluation metrics are widely used:

Modularity (Q): Modularity is a graph-based metric that measures the strength of the community structure by comparing the actual edge density within communities to the expected edge density in a random network. A higher modularity value (Q) indicates a better-defined community structure.

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(C_i, C_j)$$

Where, A_{ij} is the adjacency matrix (1 if an edge exists between nodes i and j, 0 otherwise). k_i and kj are the degrees of nodes i and j. m is the total number of edges in the network. C_i and C_j represent the communities of nodes i and j, respectively. $\delta(C_i,C_j)$ is the Kronecker delta function, which is 1 if nodes i and j are in the same community and 0 otherwise.

Table 2 and Figure 1 presents the modularity (Q) values obtained for three community detection algorithms—Girvan–Newman (GN), K-Cliques, and Chinese Whispers (CW)—across four network datasets (RH-NW, ACP-NW, DBLP-NW, and TG-NW) of varying sizes (n = 1,000; 5,000; 25,000). Modularity serves as a key performance metric that quantifies the quality of the detected community structure.

Table 2 Modularity Analysis for Community Detection Algorithms

Network Size (n)	Data set	GN	K-Cliques	Chinese Whispers
n=1000	RH-NW	0.721	0.653	0.589
	ACP-NW	0.684	0.619	0.524
	DBLP-NW	0.812	0.744	0.657
	TG-NW	0.869	0.779	0.693
n=5000	RH-NW	0.658	0.589	0.472
	ACP-NW	0.687	0.612	0.487
	DBLP-NW	0.788	0.721	0.579
	TG-NW	0.588	0.522	0.391
n=25000	RH-NW	0.744	0.693	0.589
	ACP-NW	0.722	0.645	0.478
	DBLP-NW	0.689	0.638	0.539
	TG-NW	0.752	0.693	0.521

Higher modularity values indicate that the algorithm successfully identifies dense intra-community connections and sparse inter-community links, reflecting a more meaningful partitioning of the network.

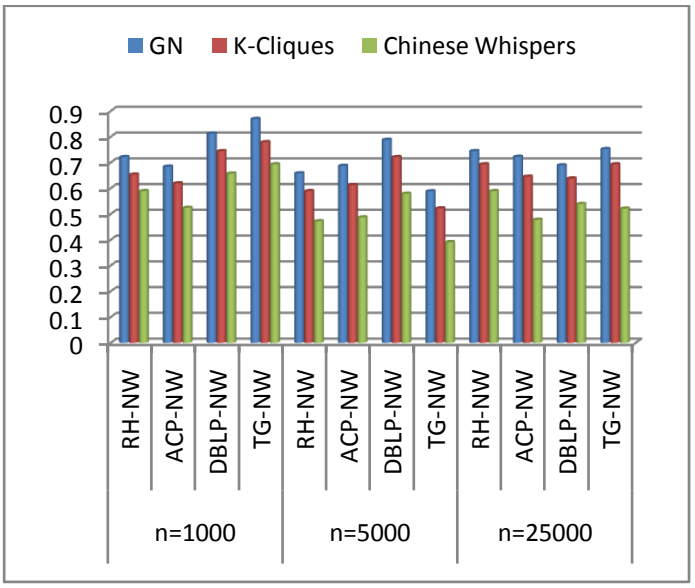

Figure 1: Modularity for all the methods , Dataset a) RH-NW b)ACP-NW c) DBLP-NW d) TG-NW

Across all datasets and network sizes, the Girvan–Newman algorithm consistently achieves the highest modularity values, demonstrating its strong ability to detect well-defined community boundaries. For example, in smaller networks (n = 1,000), GN achieves modularity scores of 0.869 (TG-NW) and 0.812 (DBLP-NW),

highlighting its accuracy in identifying hierarchical and tightly-knit communities. Even as the network size increases to n = 25,000, GN maintains relatively high modularity values (ranging between 0.689–0.752), indicating its robustness in preserving community structure. However, the slight decline in modularity with increasing network size suggests that the algorithm's hierarchical edge-removal approach becomes less efficient as the graph grows denser and more complex. Overall, GN performs exceptionally well in terms of community cohesion and separation quality, though at the cost of higher computational complexity ($O(n^3)$), making it more suitable for small to medium-scale networks.

The K-Cliques algorithm shows moderately high modularity values across all datasets, slightly below those of GN but higher than Chinese Whispers. For instance, at n = 1,000, its modularity ranges between 0.653 (RH-NW) and 0.779 (TG-NW). These results confirm that K-Cliques effectively capture overlapping and dense modular structures, which traditional non-overlapping algorithms may overlook. However, as the network size increases, a gradual decrease in modularity is observed (e.g., TG-NW drops from 0.779 to 0.693). This reduction can be attributed to the computational burden of detecting all k-cliques, especially in large or sparse graphs where fully connected subgraphs become rare. Despite this, K-Cliques retain high modularity consistency, confirming their suitability for networks with overlapping community structures, such as social and biological systems.

The Chinese Whispers algorithm yields the lowest modularity values across all datasets, with results ranging from 0.391 to 0.693, depending on network size. This outcome aligns with CW's design philosophy—it focuses on speed and scalability rather than maximizing modularity. The algorithm's randomized label propagation mechanism efficiently detects disjoint communities but often merges smaller clusters prematurely, leading to less-defined community boundaries and hence lower modularity. Despite the lower modularity, CW remains advantageous in scenarios demanding real-time analysis or large-scale data processing (e.g., n = 25,000 networks), where other algorithms like GN and K-Cliques become computationally expensive. Among all, GN demonstrates the most stable modularity performance, followed by K-Cliques, while CW exhibits the largest modularity degradation with scale.

Normalized Mutual Information (NMI): NMI is used to quantify the similarity between the detected community structure and the ground truth partition. It is based on information theory and measures how much information is shared between two partitions.

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

Where, X and Y are the detected communities and ground truth communities, respectively.

- I(X;Y) is the Mutual Information (MI):

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) log \frac{P(x,y)}{P(x)P(y)}$$

Where, P(x), P(y) and P(x,y) represent the probability distributions of clusters.

Table 3: NMI Results on Real Datasets for N = 7500

Dataset	GN	K-Cliques	Chinese Whispers
RH-NW	0.68	0.55	0.41
ACP-NW	0.76	0.67	0.48
DBLP-NW	0.74	0.61	0.43
TG-NW	0.81	0.74	0.55

Table 3 and Figure 2 presents the Normalized Mutual Information (NMI) results, which measure the similarity between detected communities and the true (ground-truth) community structure. Higher NMI values indicate more accurate and meaningful community detection. The Girvan–Newman (GN) algorithm achieves the highest NMI scores across all datasets (ranging from 0.68 to 0.81), showing its strong ability to accurately reproduce the actual community structure. This reflects GN's precise hierarchical edge-removal strategy that effectively isolates true clusters. The K-Cliques algorithm performs moderately well (0.55–0.74), demonstrating its strength in capturing overlapping communities, especially in dense datasets like TG-NW and ACP-NW. Its slightly lower NMI compared to GN is due to its focus on overlapping detection rather than strict partitioning.

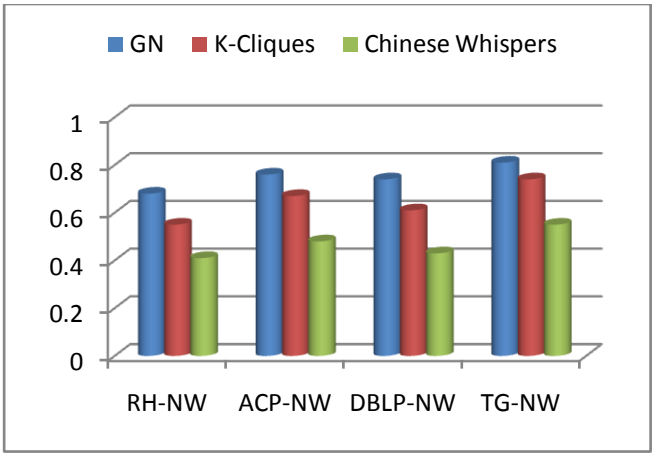

Figure 2: NMI Results on Real Datasets for N = 7500

The Chinese Whispers (CW) algorithm yields the lowest NMI values (0.41–0.55), as its randomized label propagation often merges communities and results in less accurate boundaries. However, it remains useful for large-scale or real-time clustering where execution speed is

more critical than accuracy. Overall, GN provides the most accurate results, K-Cliques offer balanced accuracy with overlap handling, and CW prioritizes computational efficiency over precision.

Execution Time (T): Execution time is a critical metric for evaluating the **computational efficiency** of community detection algorithms. It measures **how long** an algorithm takes to **process a given network** and identify communities.

$$T=t_{end}-t_{start}$$

Where, t_{start} is the time before the algorithm starts. t_{end} is the time after the algorithm completes execution.

Table 3 presents a comparison of evaluation metrics for community detection algorithms, including Modularity (Q), NMI with ground truth and Execution Time (T). These evaluation metrics provide a comprehensive assessment of community detection algorithms. Modularity is crucial for evaluating the internal consistency of communities, while NMI helps compare results with ground truth labels. Execution time determines the scalability of the algorithm for real-world applications.

Table 4 Execution Times of All Algorithms for N = 7500 (in milliseconds)

Dataset	GN	K-Cliques	Chinese Whispers
RH-NW	2872	2398	1156
ACP-NW	2628	2251	1294
DBLP-NW	2397	2143	1203
TG-NW	2135	1984	997

Table 4 and Figure 3compares the execution times of the Girvan–Newman (GN), K-Cliques, and Chinese Whispers (CW) algorithms for datasets of size N = 7500. The results clearly show significant differences in computational efficiency among the three methods. The Chinese Whispers (CW) algorithm demonstrates the fastest execution times across all datasets, ranging from 997 ms (TG-NW) to 1294 ms (ACP-NW). Its lightweight label propagation mechanism allows rapid convergence, making it highly suitable for large-scale or real-time network analysis. The K-Cliques algorithm shows moderate execution times (1984–2398 ms), performing faster than GN but slower than CW. This is because detecting all k-cliques and evaluating overlaps requires more computation, though still manageable for medium-sized networks.

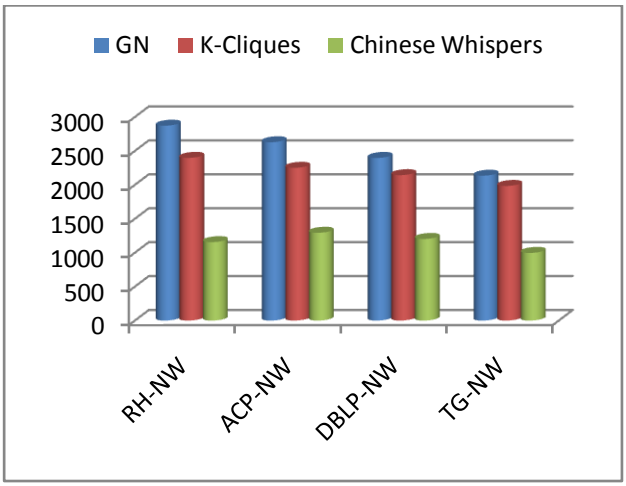

Figure 3: Execution times for community algorithms

The Girvan–Newman (GN) algorithm records the highest execution times, between 2135 ms and 2872 ms, due to its iterative calculation of edge betweenness centrality, which is computationally intensive ($O(n^3)$). While it provides the most accurate community structures, it is the least efficient in terms of runtime. Overall, CW is the most time-efficient, K-Cliques offer a balance between speed and accuracy, and GN trades higher accuracy for longer computation time.

VI. CONCLUSION

Community detection continues to be one of the most essential problems in complex network analysis, offering insights into how entities organize, interact, and evolve within large systems. This study presented a comprehensive comparison of three widely used algorithms—Girvan–Newman (GN), K-Cliques, and Chinese Whispers (CW)—each representing a distinct methodological perspective in community detection: hierarchical clustering, clique percolation, and label propagation, respectively. The Girvan–Newman algorithm, based on edge betweenness centrality, provides a clear and interpretable hierarchical view of network structures. It effectively identifies well-separated communities and has proven useful in analyzing small-scale social, citation, and biological networks. However, its high computational cost ($O(n^3)$) and inability to detect overlapping communities make it unsuitable for large or complex datasets. The K-Cliques algorithm, using clique percolation, excels in uncovering overlapping communities, which often occur in real-world networks where entities belong to multiple groups simultaneously. Its strength lies in capturing local density and modularity, making it ideal for dense and interconnected networks such as social or biological systems. Nonetheless, its exponential complexity ($O(n^k)$) limits its scalability to very large graphs, requiring computational trade-offs. The Chinese Whispers algorithm stands out for its speed, simplicity, and scalability, utilizing randomized label propagation to detect communities efficiently in massive datasets. It is particularly suited for real-time clustering applications like social media analytics, NLP, and

biological data analysis. However, CW's results can vary between runs due to its stochastic nature, and it lacks support for overlapping community detection, which limits its interpretive richness in certain contexts. Comparative analysis based on modularity (Q), Normalized Mutual Information (NMI), and execution time (T) revealed distinct trade-offs among the algorithms. While GN offers the highest modularity for small networks, K-Cliques provides strong results in overlapping community scenarios, and CW achieves the best scalability and computational efficiency. Therefore, the optimal choice of algorithm depends on the specific application requirements, network size, and desired community structure—hierarchical, overlapping, or disjoint.

In conclusion, no single algorithm universally outperforms the others across all network conditions. The Girvan–Newman method remains valuable for structural exploration and academic studies, K-Cliques for overlapping modular detection, and Chinese Whispers for large-scale, real-time clustering tasks. Together, these methods provide complementary perspectives for understanding the multifaceted nature of complex networks.

VII.REFERENCES

[1]. Biemann, C. (2006). *Chinese whispers: An efficient graph clustering algorithm and its application to natural language processing problems*. In HLT-NAACL Workshop TextGraphs (pp. 73–80). Association for Computational Linguistics.

[2]. Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008.

[3]. Clauset, A., Newman, M. E. J., & Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, 70(6), 066111.

[4]. Despalatović, L., Vojković, T., & Vukičević, D. (2015). Community structure in networks: Improving the Girvan–Newman algorithm.

[5]. Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3-5), 75–174.

[6]. Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12), 7821–7826.

[7]. Kumpula, J. M., Kivelä, M., Kaski, K., & Saramäki, J. (2008). A sequential algorithm for fast clique percolation. *Physical Review E*, 78(2), 026109.

[8]. Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2), 026113.

[9]. Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23), 8577–8582.

[10]. Newman, M. E. J., & Girvan, M. (2002). Mixing patterns and community structure in networks. *Physical Review E*, 66(2), 026126.

[11]. Palla, G., Derényi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043), 814–818.

[12]. Palla, G., Derényi, I., & Vicsek, T. (2007). The critical point of k-clique percolation in the Erdős–Rényi graph. *Journal of Statistical Physics*, 129(3), 445–467.

[13]. Porter, M. A., Onnela, J.-P., & Mucha, P. J. (2009). Communities in networks. *Notices of the American Mathematical Society*, 56(9), 1082–1097.

[14]. Ribeiro, E., et al. (2020). Semantic frame induction through the detection of communities in verb-argument graphs. *Applied Network Science*, 5(1), 63.

[15]. Rustamaji, H. C., et al. (2022). A network analysis to identify lung cancer comorbid diseases. *Applied Network Science*, 7(1), 44.

[16]. Salatino, A. (2019). Clique percolation method in Python: Implementation and applications. *[Technical report]*.

[17]. Yang, L., Cao, X., He, D., Wang, C., Wang, X., & Zhan, W. (2016). Modularity-based community detection with deep learning. *Proceedings of IJCAI* (pp. 1888–1894).

[18]. D. Wang, L. Li, X. Zhang, and X. Yu, "Dynamic Community Detection Based on Label Propagation Algorithm in Weighted Networks," IEEE Access, vol. 9, pp. 61583-61594, 2021.

[19]. Xie, J., Kelley, S., & Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys*, 45(4), 1–35.