Community Detection Based on the LO-WT Algorithm

Zainab Falah Hasan
Department of Software, University of Babylon, Iraq
wsci.zainab.falah@uobabylon.edu.iq (corresponding author)

Huda Naji Nawaf
Department of Information Networks, College of Information Technology, University of Babylon, Iraq
huda.almamory@uobabylon.edu.iq

Received: 16 February 2025 | Revised: 18 April 2025, 19 May 2025, and 15 June 2025 | Accepted: 3 July 2025

 DOI: https://doi.org/10.48084/etasr.10588

ABSTRACT

Community detection is an important and difficult task in network analysis. This study proposes the Louvain algorithm combined with the Walk-Trap algorithm (LO-WT), leveraging structural and path-based similarities to enhance the detection accuracy. The structural similarity between nodes is a sensitive issue and essential for community identification. First, computing the similarity between the nodes is applied to determine the weights assigned to the edges connecting them. On the other hand, this information is used to guide the walking process in the Walk-Trap algorithm to obtain the trajectories for each node. Then, the similarity is calculated based on the Jaccard similarity among the trajectories. This builds a similarity matrix, which is then used to create a new graph as a weighted network based on a certain threshold depending on the network's complexity. The weights represent the degree of similarity between the nodes. The Louvain algorithm can then be effectively applied to the new graph to identify the communities by maximizing the modularity rather than using agglomerative clustering in the Walk-Trap algorithm. This approach leverages the efficiency of the algorithm to uncover the meaningful community structures within the data. Four real and synthetic networks are used to validate the results, and the algorithm is evaluated against several algorithms, including baseline and state-of-the-art methods. Critical parameters, such as the structure-based similarity threshold (0.1–0.3) and trajectory length (2–4) are carefully adjusted to optimize the performance. The results show that LO-WT significantly outperforms recent related methods and traditional algorithms. Specifically, it outperforms the Louvain algorithm in terms of Normalized Mutual Information (NMI) and is competitive regarding modularity. Additionally, it surpasses related work by achieving higher performance across all four real-world networks: Karate (0.41), Dolphin (0.52), Football (0.60), and Facebook (0.83). Furthermore, LO-WT can exhibit high conductivity and density, demonstrating its robust performance. Overall, the LO-WT algorithm demonstrates its effectiveness for accurate community detection.

Keywords-Louvain algorithm; Walk-Trap algorithm; Jaccard similarity; random walks

I. INTRODUCTION

Community detection is a crucial research topic in network science, enabling the discovery of complex structures [1]. A community consists of nodes that have strong relationships within the community and weak relationships outside it. In general, the nodes within a community have the same features. Therefore, community detection plays an important role in the analysis of complex networks by helping to recognize the inner construction of the network, acquire significant information, and obtain the relationships between the nodes [1]. With the advancement of information technology, complex networks continue to grow over time. Social network analysis extracts graph structures that represent the complex relationships in these social networks.

Community detection algorithms include methods, such as hierarchical [2], spectral [2], and modularity-based algorithms. Hierarchical algorithms are easy to understand and group nodes with the strongest links while separating those with weaker links. However, they do not consistently produce good results [1].

Spectral algorithms are often used to obtain non-overlapping groups of nodes by approximating solutions to the Normalized Cut minimization problem. However, the similarity matrix used in spectral clustering affects the clustering result and speed. When the dimensionality is high, spectral clustering's performance and speed can suffer due to incomplete dimension reduction. Additionally, the final clustering outcome can vary depending on the similarity matrix used.

Most modularity-based community detection algorithms can precisely identify communities in diverse networks, particularly those with distinct community structures. However, they struggle in networks with unclear community structures. Additionally, these algorithms have limitations: they capture only part of the network's features and information and are applied directly to sparse original graphs. Consequently, they are not suitable for fully utilizing the network's structure [1].

The Louvain algorithm, a modularity optimization heuristic, is a popular technique for identifying community structures. It quickly and effectively finds high-modularity community structures in large-scale networks. Several studies have proposed improvements to the Louvain algorithm [3-5].

The Walk-Trap algorithm is an agglomerative method that builds a node similarity matrix using random walks. The similarity matrix is updated iteratively [6]. Infomap [7] is based on the idea that a random walker tends to remain within a community before moving to another. By assigning prefix codes to groups and codes to nodes within communities, Infomap reduces the length of the Huffman encoded path of the walker. Like Louvain, Infomap repeatedly transfers nodes between communities using modularity and code length criteria.

Unlike the Walk-Trap method, which uses unbiased random walks, few studies have used guided walks that bias the paths based on edge weights [8]. In this work, Walk-Trap is applied to the weighted graph where random walks are guided by weights to obtain node paths. These weights are created using the structure of the network. Additionally, the walks are used to establish relationships between the nodes in a graph. The Louvain algorithm is then applied for community detection.

Some relevant literature and previous research on community detection are reviewed below.

A. *Louvain Community Detection Methods*

Many researchers have developed and analyzed the Louvain algorithm. Authors in [3] developed an extension of the Louvain algorithm to build communities based on the connections between nodes that are linked by weights to form cluster associations. They then applied a weighting algorithm that considered the relationships between the users and the activity of user accounts when determining the weights. The weighting process obtained cluster weights, computed the trust between the users, and built a trust matrix.

To address some of the drawbacks of modularity functions, such as the resolution limit problem, authors in [4] proposed a new objective function for the modularity function. They implemented a restricted Louvain algorithm by incorporating some constraints into the original Louvain technique. This method used an objective function F_2 based on in-degree and out-degree for community evaluation. The traditional Louvain algorithm was applied with this objective function. The change in the objective function, ΔF_2, is calculated when a node is moved from its community to a neighboring community. Ultimately, every community is compressed into a new node within a newly built network. The weight of the new node's self-loop is equal to the total of the edge weights inside each community. Additionally, the weight between two new nodes is obtained from the total weights between the nodes in each pair of communities. This work is repeated until the community's goal function no longer increases. In the next phase, the weak communities are broken apart, and their nodes are assigned to other communities that satisfy the maximum increase of ΔF_2, which are named quasi-communities.

Authors in [9] offered a strategy based on the clique idea to reduce this algorithm's computational cost. Rather than treating every node in the network as a community as in the Louvain method, the cliques with at least three members and the other nodes that are not part of them are regarded as distinct communities. The benefit of merging neighboring communities is then computed.

Authors in [10] presented the MuLaN technique for the local alignment of multilayer networks, where a multilayer alignment graph is formed starting from some initial nodes, such as two input layers for the multilayer network: disease and drug. The graph is then analyzed by revealing regions of similarity based on the topology of the multilayer alignment graph. MuLaN applied Louvain for community detection.

An improved Louvain algorithm was presented in [11] by merging the node significance and the modularity function with the original Louvain algorithm to restrict randomness. By increasing the modularity value, the improved Louvain algorithm employed the original Louvain algorithm to identify the community structure. In the community detection phase, the node scanning order is determined by the centrality degree, which measures the node importance. Community building—also known as clustering—between nodes or users is achieved by assigning the nodes weights to form clusters, and by merging the nodes based not only on the modularity gain, but also on the node importance.

B. *Community Detection Methods based on Random Walks*

Authors in [1] proposed to detect communities in attributed networks by directly processing the original network topology along with node attribute information. Initially, seed nodes were selected, followed by seed replacement to improve quality. An interaction transition matrix was constructed, and a random walk was applied to expand the communities [1].

Most previous studies have not focused on random walks guided by weights. Authors in [8] are among the few who have investigated edge weighting strategies that bias the random walk process. Their study showed that biased random walks improve community detection tasks based on K-means clustering with different settings. The weights of a graph's edges are measured according to edge betweenness centrality and the common neighbor ratio [8].

Authors in [12] presented two methods for identifying network communities. The first method uses random walks and the Walk Likelihood Algorithm (WLA) to divide a cluster of data points or network nodes into non-overlapping communities. They analyzed the network using random walks and Bayesian feature inference. The optimal number of clusters was determined based on the number of visits to network nodes and edge weights, as well as global movements. The network was considered a community, and then it was divided. The resulting partitions were evaluated using modularity, and adjacent communities were merged accordingly.

The algorithm proposed in [13] for agglomerative community discovery is based on node influence and similarity. It consists of three key steps: (1) determining the central node based on on their influence on neighbors, (2) selecting candidate neighbors based on the node similarity, and (3) using merging small communities utilizing modularity and label propagation.

A method in [14] combines the message forwarding-based node embedding with the graph neural network architecture. Initially, node2vec creates node embeddings using Skip-Gram with Negative Sampling (SGNS) and second-order biased random walks. These embeddings are then passed to a Graph Convolutional Network (GCN). The model calculates the scaled normalised matrices and a weighted average matrix. The edge weights are derived from the embedding similarity between the nodes. Feature vectors are generated, and a 3-layer GCN is used for community prediction, utilizing the weighted average matrix and an activation function during testing.

II. LOUVAIN ALGORITHM

The Louvain algorithm is a fast and modular community detection technique used in conjunction with agglomerative hierarchical clustering. First, the initial modularity is computed with each node treated as a community. Next, to maximize the modularity value, a community is built, where each node is moved to a neighboring community. The node then joins the neighboring community with the highest modularity score. A node remains in its community and does not move when all neighboring communities yield a modularity gain of zero or negative [9]. When evaluating the community detection model, the following metrics are typically employed.

An unweighted graph $G\,(V, E)$ with nodes V and edges E between nodes is given. Assume that $u \in V$ and that the relation between u and v is $(u, v) \in E$. Given a clustering of k communities $C = \{C_1, C_2, \dots, C_k\}$, where $C_1 \neq \emptyset$ and $C_1 \quad V$, the modularity is given by [9]:

$$Q = \frac{1}{2M} \sum_{i \neq j} \left(A_{ij} - \frac{k_i k_j}{2M}\right) \delta(C_i, C_j) \tag{1}$$

$$M = \frac{1}{2} \sum_{ij} A_{ij} \tag{2}$$

where A_{ij} indicates whether the nodes i and j are connected: $A_{ij} = 1$ if connected, and $A_{ij} = 0$ otherwise; M is the number of edges, $i \in V$, and $j \in V$; C_i is the group in which node i is a member. The value of $\delta(C_i, C_j) = 1$ when nodes i and j are in the same community, otherwise $\delta(C_i, C_j) = 0$. Finally, $Q \in [-1, 1]$. The weights of the edges that are connected to node i are added together to form k_i.

III. WALK-TRAP ALGORITHM

The Walk-Trap algorithm is based on random walks that use short walks from a node, where the walker tends to remain in the same community. Subsequently, the communities are merged from the bottom up [15]. The Walk-Trap algorithm employs agglomerative hierarchical clustering and random walks to find community structures. Starting from an undirected, connected graph, the method creates partitions and evaluates modularity as follows [6]:

The clustering technique iteratively gathers the nodes into communities by measuring the similarity between them. Given a connected and undirected graph G with a given length t for random walks, P is the transition matrix, and D is the diagonal matrix of node degrees. The distance function r must be measured for every pair of nodes (i, j) [6, 16]. Then, the mean of the squared variance function is minimized using Ward's approach. The communities are merged to reduce the mean of squared distances between each node and its community. After that, the variation is computed between two pairs of communities, and the probability matrix and the $\Delta\sigma$ data are recalculated. Finally, the process continues until the entire network is in a single community.

IV. METHODOLOGY

This study proposes a non-overlapping technique for community detection that utilizes the Louvain algorithm with the Walk-Trap algorithm (LO-WT), as shown in Figure 1. Unlike methods that rely on random walks to repeatedly visit nodes for community detection, the introduced approach uses a control mechanism to quantify the node similarity. The Louvain algorithm then utilizes these similarities to construct communities.

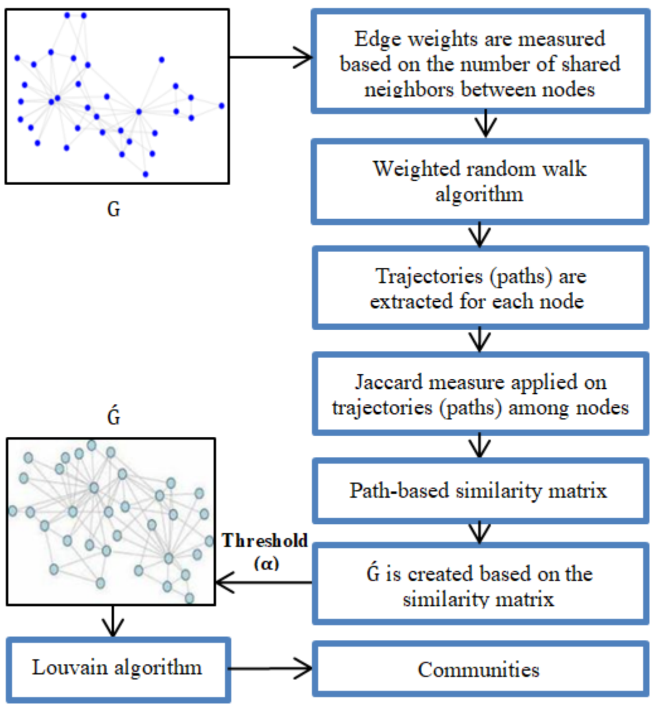


Fig. 1. Architecture of the proposed LO-WT algorithm.

A. Random Walks and Similarity Weighting

Walk-Trap begins by performing biased random walks to identify the structurally similar nodes within a network. The algorithm takes three main inputs: edge weights, a target node, and the desired walk length. The output is a sequence of nodes representing the trajectory (path) taken during each walk. This process is repeated according to a predefined number of walks. To guide the walk toward more relevant neighbors, transition probabilities are biased using edge weights. The number of walks is determined by the degree of the target node, with higher-degree nodes generating more walks, as supported in [6]. The length of each walk is selected empirically through trial and error to balance exploration and accuracy. The selection of a node's path is controlled based on the weight of each edge in the graph, which is calculated based on the similarity between the nodes in terms of their structure. These weights are used to walk toward the edges with the highest weight in the node's path. The idea behind this is to select the most similar nodes to ensure an effective community detection. The Jaccard index has been used to measure the similarity between two nodes, based on their structure, as stated in:

$$\forall(\mathrm{u},\mathrm{v}) \in \mathrm{E},\ \ \mathrm{W}(\mathrm{u},\mathrm{v}) = \mathrm{Jaccard}(\mathrm{N}[\mathrm{u}],\mathrm{N}[\mathrm{v}]) \quad (3)$$

where N is a node's set of neighbors and W is the weight between u and v. The weight W directs the random walks in the Walk-Trap algorithm, enabling the latter to favor the transitions to neighboring nodes according to the strength of their connections. This weighting mechanism increases the probability of travelling along edges that signify more significant relationships, which helps identify the densely connected communities in the graph.

For each node v in graph G, let $T(\mathrm{v})$ denote the set of trajectories associated with node v. Each trajectory in $T(\mathrm{v})$ is generated by simulating a controlled random walk starting from node v. The similarity between the nodes is then based on the overlap of their visited nodes across multiple walks. The set of shared vertices, $S(\mathrm{v})$, is defined as the intersection of all visited nodes across trajectories in $T(\mathrm{v})$:

$$\mathrm{S}(\mathrm{v}) = \cap\, \mathrm{T}(\mathrm{v}) \quad (4)$$

Using the vector of $S(\mathrm{v})$ for walks that each node has visited starting from v, the similarity matrix is constructed between all pairs of nodes. Again, Jaccard similarity is used to define the similarity between nodes based on trajectories:

$$\mathrm{Sim}_{\mathrm{Jac}}(\mathrm{S}(\mathrm{u}),\mathrm{S}(\mathrm{v})) = \mathrm{Jaccard}(\mathrm{S}(\mathrm{u}),\mathrm{S}(\mathrm{v})) \quad (5)$$

The similarity matrix resulting from (5) is needed in the next step, where it is used by the Louvain algorithm for community detection rather than the hierarchical clustering algorithm that exists in Walk-Trap.

B. Louvain Clustering for Community Detection

A new graph is built after generating paths and computing the similarity based on trajectories. For any pair of nodes i and j such that $\mathrm{i} \neq \mathrm{j}$, an edge is created between the nodes i and j in the graph $\acute{\mathrm{G}}$ if their similarity exceeds a predefined threshold. This threshold is chosen empirically to balance sparsity and connectivity:

$$Edge(i,j) = \begin{cases} 1 & \text{if } (\mathrm{Sim}_{\mathrm{Jac}}(\mathrm{S}(\mathrm{i}),\mathrm{S}(\mathrm{j}) > \alpha) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

The Louvain algorithm is then used to detect communities by maximizing modularity such that (1) is applied to $\acute{\mathrm{G}}$ as input and communities as output, where $\mathrm{A}_{\mathrm{ij}}=1$ if $\mathrm{Sim}_{\mathrm{Jac}}(\mathrm{S}(\mathrm{i}),\mathrm{S}(\mathrm{j}) > \alpha$, otherwise it is zero.

V. DESCRIPTION OF BENCHMARK DATASETS

The datasets used for evaluation are divided into two types: real-world networks and synthetic networks.

A. Real-World Networks

Several well-known real-world networks were selected to validate the proposed algorithm. These networks come from various domains and have different sizes and characteristics. Their properties are summarized in Table I. The datasets include:

- Karate club [17]: A well-known real-world dataset from a U.S. university social group, consisting of 34 members. Two groups were formed in the club based on the interpersonal relationships. It is widely used in community detection research since its community structure is known in advance.
- Dolphins' social network [18]: It includes 62 dolphins, and two communities have been based on observed interactions.
- American college football network [19]: It includes 115 nodes and 616 edges representing college football games in the US. It is divided into 12 conferences (categories), each comprising 8-12 teams.
- Facebook network (Stanford SNAP dataset) [20]: It is a large-scale real-world graph with thousands of nodes and edges representing user friendships. Although it lacks ground-truth communities, it is useful for qualitative and scalability analysis.

TABLE I. STATISTICS OF REAL-WORLD NETWORKS

Network	Number of nodes	Number of edges
Dolphins	62	159
American football	115	613
Karate club	34	78
Facebook	4,039	88,234

B. Synthetic Networks

The Lancichinetti-Fortunato-Radicchi (LFR) benchmark is a synthetic network specifically created to provide a robustness and scalability test for community detection algorithms. These networks are defined based on the power-law distributions of node degrees and community sizes, mimicking real-world network properties [12]. Four versions of the LFR dataset are generated, with parameters shown in Table II, where the number of nodes is represented by N, k is the average degree, mu stands for the degree of mixture of nodes, maxk is the maximum degree of nodes, and maxc stands for the maximum community size.

TABLE II. STATISTICS OF LFR NETWORKS

Network	N	k	maxk	mu	maxc
LFR 1	1,000	20	40	0.1	60
LFR 2	2,000	20	40	0.1	60
LFR 3	5,000	20	40	0.1	60
LFR 4	10,000	20	40	0.1	60

VI. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

The performance of the proposed algorithm is evaluated using NMI and modularity. For networks with ground truth, both NMI and modularity are used. In contrast, for networks without ground truth, only modularity is utilized to assess the performance of the algorithm. The experiments are performed using random walk lengths in the range of 2-4, which achieved the best results. The similarity threshold used to construct the new graph is set to 0.3 in all networks, except in the Facebook network, where it is 0.1 due to lower similarity between the nodes used to create the graph. Table III compares the performance of the proposed method for real networks with traditional algorithms (Walk-Trap, Louvain, Newman), learning-based methods (e.g., Node2Vec), and recent related works, as well as a baseline version of LO-WT in terms of modularity and NMI. It is worth mentioning that the baseline of LO-WT excludes the directed walks. LO-WT's modularity is comparable to that of the conventional Walk-Trap and Louvain algorithms in Karate and Football networks; it performs somewhat better in the Dolphins network.

TABLE III. PERFORMANCE OF LO-WT COMPARED WITH OTHER ALGORITHMS USING REAL-WORLD NETWORKS

Algorithm	Modularity				NMI		
	Karate	Dolphins	Facebook	Football	Karate	Dolphins	Football
Walk-Trap	0.41	0.49	0.79	0.6	0.68	0.4	0.88
Louvain	0.41	0.51	0.83	0.6	0.58	0.47	0.88
Newman	0.38	0.49	0.77	0.54	0.69	0.57	0.69
Node2vec	0.35	0.43	0.80	0.33	0.01	0.09	0.38
Baseline	0.37	0.44	0.80	0.55	0.49	0.44	0.81
LPA-CW [21]	0.39	0.50	0.78	0.53	0.47	0.52	0.88
Propinquity dynamics [22]	0.37	0.51	—	0.60	0.92	0.94	0.92
CDCE[23]	0.37	0.52	—	0.57	—	—	—
ICDR[24]	—	0.37	0.68	—	—	0.98	—
LO-WT	0.41	0.52	0.83	0.6	0.7	0.57	0.92

When comparing the NMI of LO-WT to the Walk-Trap and Louvain algorithms, there has been a clear improvement. Overall, LO-WT has higher modularity and NMI than the Newman, Node2vec, and baseline methods. While in the recent related works, it can be said that [22, 24] are superior to the proposed method in terms of NMI, this often comes at the cost of lower modularity. In contrast, LO-WT maintains a better balance between NMI and modularity. To further validate LO-WT, it is tested on the synthetic networks (LFR) of different sizes, as depicted in Table IV. LO-WT achieves modularity similar to Walk-Trap and Louvain across all synthetic networks. It generally performs better than other algorithms.

TABLE IV. MODULARITY OF LO-WT COMPARED WITH OTHER ALGORITHMS USING SYNTHETIC NETWORKS

Algorithm	LFR1000	LFR2000	LFR5000
Walk-Trap	0.78	0.81	0.82
Louvain	0.78	0.81	0.82
Newman	0.74	0.77	0.78
Node2vec	0.78	0.79	0.80
Baseline	0.78	0.80	0.81
LO-WT	0.78	0.81	0.82

Furthermore, the performance of the LO-WT algorithm is validated with real networks over different walk lengths, as portrayed in Figure 2. It is demonstrated that the best results in terms of modularity are obtained for walks lengths 2, 3, and 4, where the NMI value remained relatively high. In contrast, the modularity is not affected by different walk lengths for synthetic networks, as shown in Figure 3. It should be noted that the NMI value was not calculated for these networks, as they do not have ground truth.

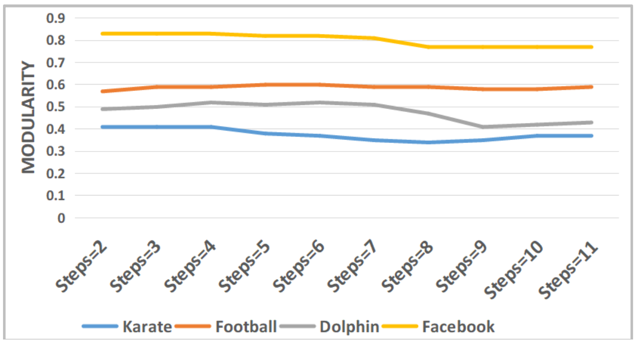


Fig. 2. Modularity of the LO-WT algorithm for different walk lengths on real-world networks.

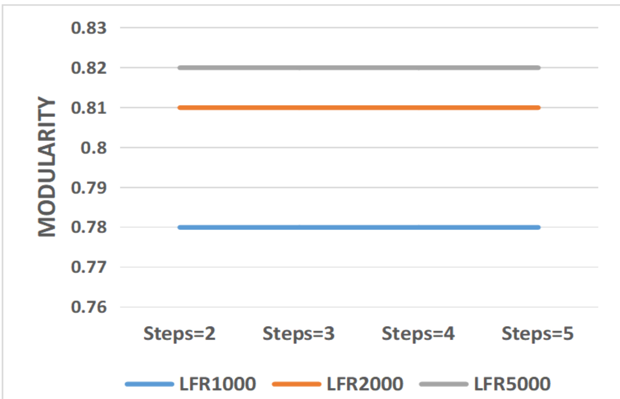
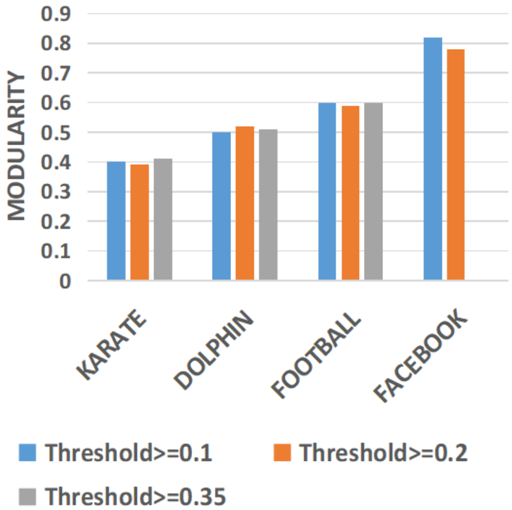
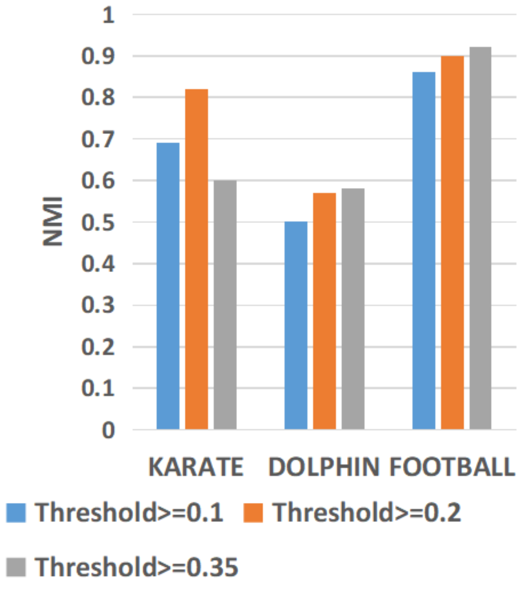


Fig. 3. Modularity of the LO-WT algorithm for different walk lengths on synthetic LFR networks.

In a further experiment, different similarity thresholds are tested to assess their effects on the performance of the LO-WT algorithm. The performance is plotted against the various threshold values in Figure 4. Across many real networks, the optimal threshold value for maximizing modularity lies between 0.1 and 0.35. Regarding the synthetic networks, as displayed in Figure 5, the LO-WT algorithm's performance remains relatively insensitive to the threshold value.



(a)



(b)

Fig. 4. Performance of the LO-WT algorithm across different similarity threshold values for real networks: (a) modularity, (b) NMI.

To evaluate the detected communities, the analysis includes both the density and the conductance measures. Density measures the ratio of the possible connections within a community that actually exists, indicating how tightly the nodes are connected within a community. Conductance measures the ratio of the external connections to the total connections of the community, and thus reflects its interconnectedness with the rest of the network. Table V presents these metrics to allow a comparative analysis of community cohesion and external interaction in real networks. The LO-WT algorithm showed the highest density in all real networks, except for Dolphins, compared to the other algorithms. In terms of conductance, LO-WT obtains superior results in almost all networks, with Louvain exhibiting similar performance in some cases.

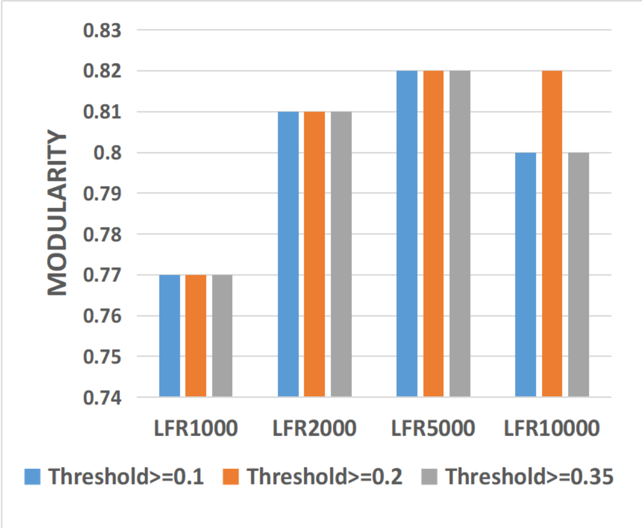


Fig. 5. Modularity and NMI of the LO-WT algorithm across different similarity threshold values for synthetic LFR networks.

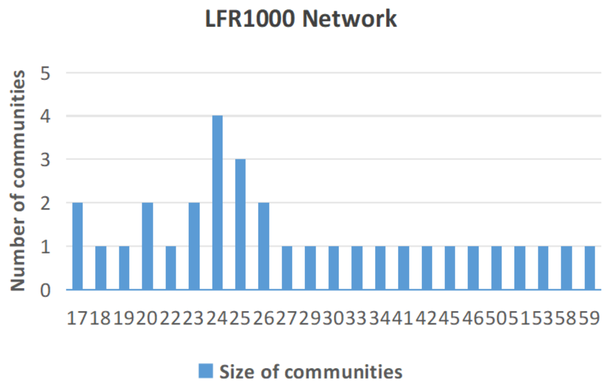
The analysis of LO-WT's performance based on the community size is illustrated in Figure 6. Facebook's network and LFR networks produced a larger number of small-sized communities; authors in [25] corroborate these findings. According to their model, the generated community sizes of LFR have two ranges: [10, 50] 'small' and [50, 100] 'big'. In the present study, the LFR model appears with roughly the same ranges and distribution, as shown in Figures 6(a-c).

As a large-scale social network, Facebook exhibits community structures whose sizes vary significantly depending on the nature and dynamics of the interactions within the network; in other words, it is challenging to pinpoint precise ranges for the sizes of the communities, and this variance in sizes is one of their defining characteristics. However, Facebook's community sizes are distributed according to power-law distributions (Figure 6(d)), which are intrinsic to many real-world social networks. The former implies a limited number of major communities along with several smaller, more connected communities.

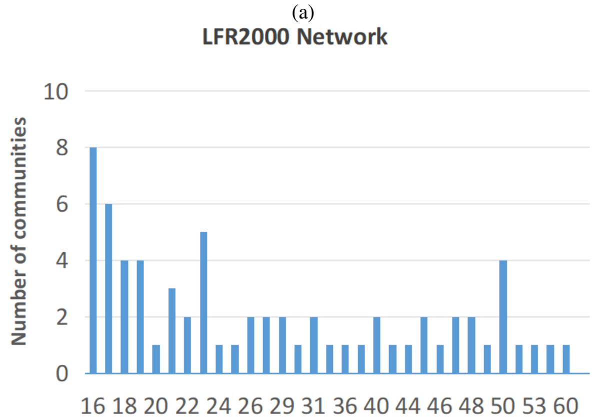
Figures 7 and 8 present the cumulative distribution of Facebook and Football, respectively. With a slope of about -1, the cumulative distribution function in the entire Facebook network exhibits a power-law-like pattern. It is assumed that a slope close to -1 represents a perfect power law distribution.

TABLE V. DENSITY AND CONDUCTANCE OF LO-WT COMPARED WITH OTHER ALGORITHMS USING REAL-WORLD NETWORKS

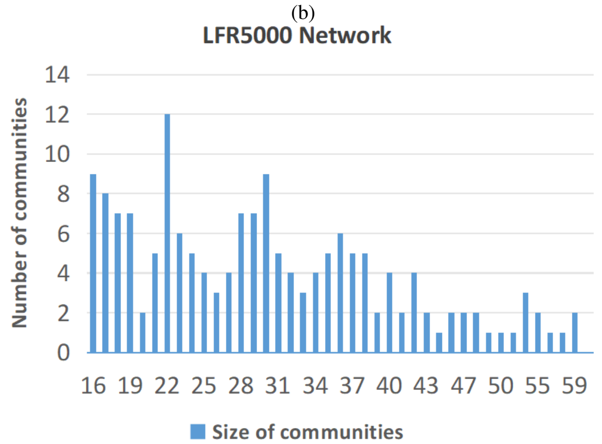
Algorithm	Metric	Karate	Dolphins	Facebook	Football
Walk-Trap	Density	0.45	0.61	0.47	0.75
	Conductance	0.287	0.356	0.196	0.294
Louvain	Density	0.49	0.35	0.29	0.75
	Conductance	0.287	0.301	0.054	0.294
Newman	Density	0.34	0.42	0.36	0.48
	Conductance	0.664	0.963	0.121	0.768
Node2vec	Density	0.14	0.07	0.01	0.07
	Conductance	0.280	0.316	0.054	0.277
Baseline	Density	0.35	0.41	0.14	0.70
	Conductance	0.213	0.335	0.023	0.351
LO-WT	Density	0.49	0.29	0.46	0.86
	Conductance	0.210	0.316	0.024	0.138



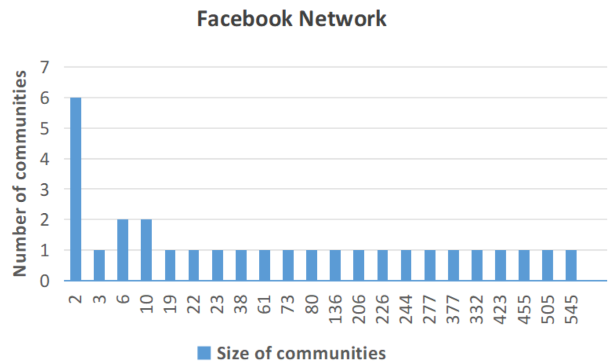
(a)



(b)



(c)



(d)

Fig. 6. Community detection results based on community sizes: (a) LFR1000 network, (b) LFR2000 network, (c) LFR5000 network, and (d) Facebook network.

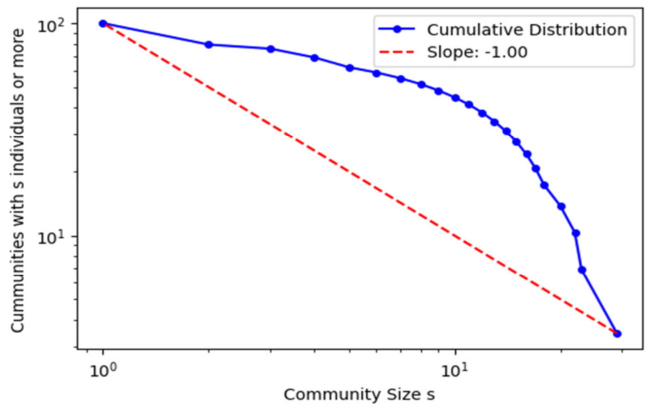


Fig. 7. Cumulative distribution function of community sizes in the Facebook network.

The cumulative distribution function in the Football network also shows a power-law-like behavior with an estimated exponent of -1.46. These findings are consistent with Newman's findings [26] on co-authorship networks for physicists; according to his study, the distribution approximately follows a power law with an exponent of -1.6.

Figures 9-11 illustrate the communities determined by the proposed algorithm in various networks.

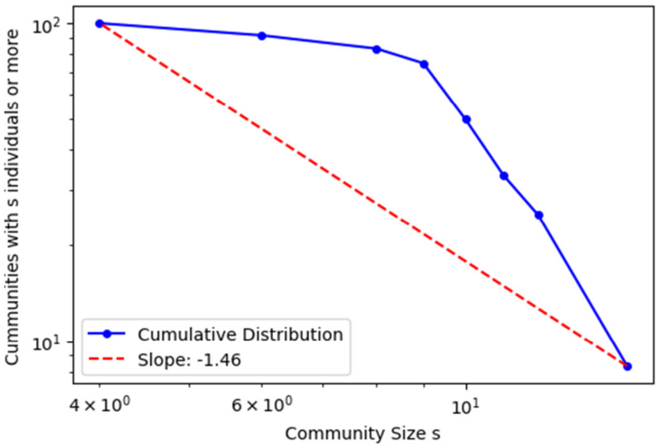


Fig. 8. Cumulative distribution function of community sizes in the Football network.

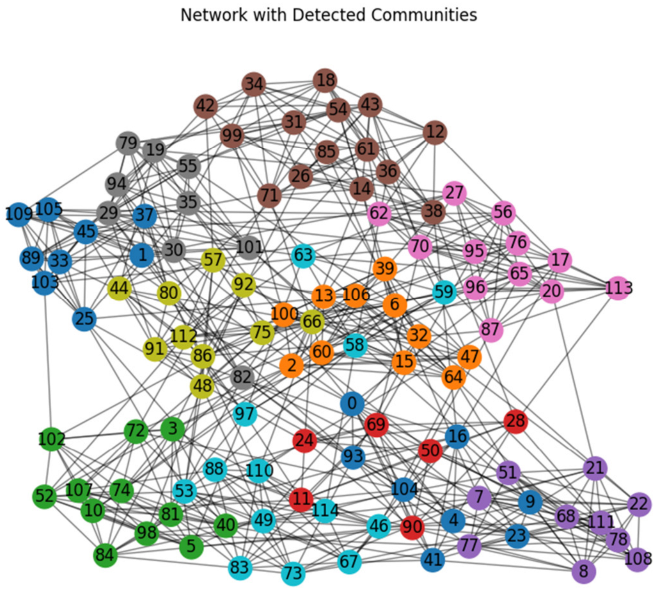


Fig. 9. Community detection results for the Football network.

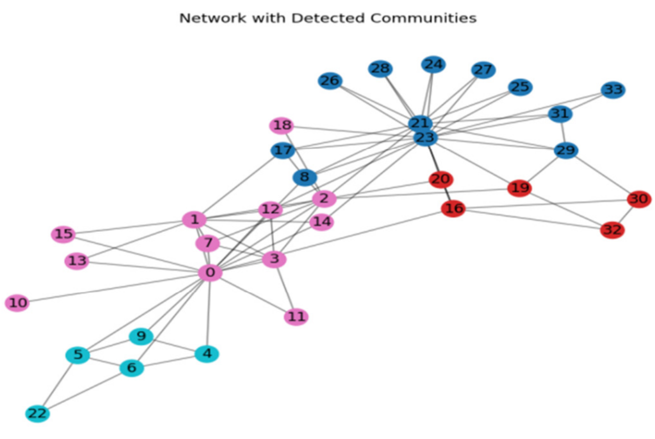


Fig. 10. Community detection results for the Karate club network.

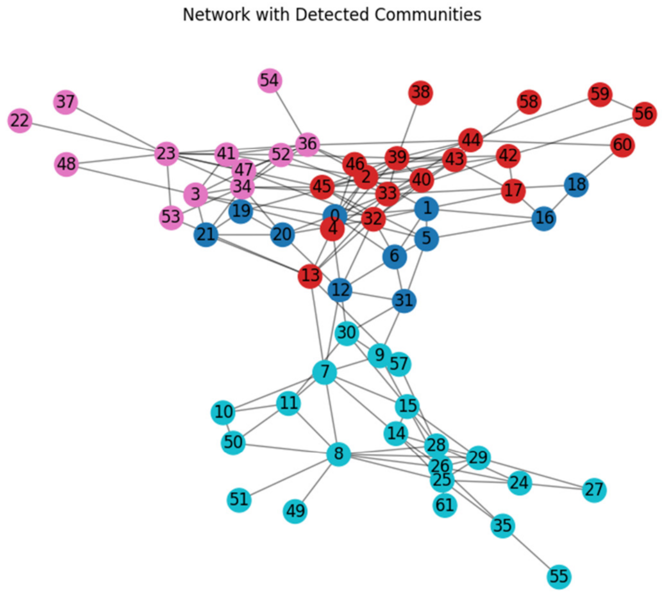


Fig. 11. Community detection results for the Dolphins network.

VII. CONCLUSION

This study presents the Louvain algorithm with the Walk-Trap algorithm (LO-WT), a hybrid community detection algorithm that demonstrates the crucial role of structural similarity through controlled random walks with Louvain clustering. The experimental results show that LO-WT outperforms its baseline, where edge weighting is not used. In addition, it is worth noting that LO-WT performs as well on real networks as on synthetic networks when compared to competing and standard algorithms.

A key finding from the experiments is the significance of structure-based edge weighting, which enhances the ability of the algorithm to detect cohesive communities. This is especially true in real-world networks, where LO-WT consistently performs with higher modularity and Normalized Mutual Information (NMI) compared to Louvain, Walk-Trap, Node2Vec, and other baseline methods.

Experimentally, the number of walking steps plays a critical role: while maintaining a high NMI value across all networks, it is shown that the modularity of the LO-WT algorithm stabilizes or slightly deteriorates when the number of walking steps exceeds four. This indicates that the walker moves outside the boundary of the community after more than four steps, reducing the algorithm's effectiveness.

The threshold experiment suggests that the optimal similarity threshold is network-specific and depends on unique structural characteristics, such as density, degree distribution, community size, and the strength of connections within and between communities. These factors influence the optimal threshold for community detection. In contrast, for synthetic networks, such as Lancichinetti-Fortunato-Radicchi (LFR), using different walk lengths or thresholds do not affect the performance of the proposed algorithm, likely due to the predefined nature of these networks.

Furthermore, the results demonstrate the proposed community detection algorithm's strong capacity to recognize communities. The potential relevance of the algorithm for social network analysis is evidenced by the high density and low conductance values, which emphasize its ability to identify communities.

Finally, the proposed technique exhibits the power-law distribution of the community sizes, a common feature of complex networks. This outcome aligns with the natural characteristics of social networks, where several smaller communities coexist alongside a few larger ones.

In conclusion, LO-WT is a powerful and efficient model for non-overlapping community detection, particularly adept at handling complex real-world networks. Its capacity to combine structural similarity with modularity optimization makes it a promising approach for future research in graph mining and social network analysis.

REFERENCES

[1] Y. Chang, H. Ma, L. Chang, and Z. Li, "Community detection with attributed random walk via seed replacement," *Frontiers of Computer Science*, vol. 16, no. 5, Jan. 2022, Art. no. 165324, https://doi.org/10.1007/s11704-021-0482-x.

[2] F. Belloum, L. Houichi, and M. Kherouf, "The Performance of Spectral Clustering Algorithms on Water Distribution Networks: Further Evidence," *Engineering, Technology & Applied Science Research*, vol. 12, no. 4, pp. 9056–9062, Aug. 2022, https://doi.org/10.48084/etasr.5116.

[3] Murniyati, A. B. Mutiara, S. Wirawan, T. Yusnitasari, and D. Anggraini, "Expanding Louvain Algorithm for Clustering Relationship Formation," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, no. 1, pp. 701–708, 31 2023, https://doi.org/10.14569/IJACSA.2023.0140177.

[4] B. Yao, J. Zhu, P. Ma, K. Gao, and X. Ren, "A Constrained Louvain Algorithm with a Novel Modularity," *Applied Sciences*, vol. 13, no. 6, Mar. 2023, Art. no. 40445, https://doi.org/10.3390/app13064045.

[5] J. Zhang, J. Fei, X. Song, and J. Feng, "An Improved Louvain Algorithm for Community Detection," *Mathematical Problems in Engineering*, vol. 2021, no. 1, Nov. 2021, Art. no. 1485592, https://doi.org/10.1155/2021/1485592.

[6] P. Pons and M. Latapy, "Computing Communities in Large Networks Using Random Walks," in *Computer and Information Sciences - ISCIS 2005: 20th International Symposium*, Istanbul, Turkey, 2005, pp. 284–293, https://doi.org/10.1007/11569596_31.

[7] M. Rosvall, D. Axelsson, and C. T. Bergstrom, "The map equation," *The European Physical Journal Special Topics*, vol. 178, no. 1, pp. 13–23, Nov. 2009, https://doi.org/10.1140/epjst/e2010-01179-1.

[8] A. Kosmatopoulos, K. Loumponias, D. Chatzakou, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "Random-Walk Graph Embeddings and the Influence of Edge Weighting Strategies in Community Detection Tasks," in *Proceedings of the 2021 Workshop on Open Challenges in Online Social Networks*, Virtual Event, Ireland, 2021, pp. 9–13, https://doi.org/10.1145/3472720.3483621.

[9] E. A. Abbas and H. N. Nawaf, "Improving Louvain Algorithm by Leveraging Cliques for Community Detection," in *2020 International Conference on Computer Science and Software Engineering*, Duhok, Iraq, 2020, pp. 244–248, https://doi.org/10.1109/CSASE48920.2020.9142102.

[10] M. Milano, P. Cinaglia, P. H. Guzzi, and M. Cannataro, "A novel local alignment algorithm for Multilayer networks," *Informatics in Medicine Unlocked*, vol. 44, Jan. 2024, Art. no. 101425, https://doi.org/10.1016/j.imu.2023.101425.

[11] A. Aldabobi, A. Sharieh, and R. Jabri, "An Improved Louvain Algorithm based on Node Importance for Community Detection," *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 23, pp. 7055–7063, Dec. 2022.

[12] A. Ballal, W. B. Kion-Crosby, and A. V. Morozov, "Network community detection and clustering with random walks," *Physical Review Research*, vol. 4, no. 4, Nov. 2022, Art. no. 043117, https://doi.org/10.1103/PhysRevResearch.4.043117.

[13] Y. Xu, T. Ren, and S. Sun, "Community Detection Based on Node Influence and Similarity of Nodes," *Mathematics*, vol. 10, no. 6, Mar. 2022, Art. no. 970, https://doi.org/10.3390/math10060970.

[14] R. Bhattacharya, N. K. Nagwani, and S. Tripathi, "A community detection model using node embedding approach and graph convolutional network with clustering technique," *Decision Analytics Journal*, vol. 9, Dec. 2023, Art. no. 100362, https://doi.org/10.1016/j.dajour.2023.100362.

[15] R. George, K. Shujaee, M. Kerwat, Z. Felfli, D. Gelenbe, and K. Ukuwu, "A Comparative Evaluation of Community Detection Algorithms in Social Networks," *Procedia Computer Science*, vol. 171, pp. 1157–1165, Jan. 2020, https://doi.org/10.1016/j.procs.2020.04.124.

[16] D. D. Hieu and P. T. Ha Duong, "Detecting communities in large networks using the extended Walktrap algorithm," in *2022 RIVF International Conference on Computing and Communication Technologies*, Ho Chi Minh City, Vietnam, 2022, pp. 100–105, https://doi.org/10.1109/RIVF55975.2022.10013880.

[17] W. W. Zachary, "An Information Flow Model for Conflict and Fission in Small Groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, Dec. 1977, https://doi.org/10.1086/jar.33.4.3629752.

[18] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, Sept. 2003, https://doi.org/10.1007/s00265-003-0651-y.

[19] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, Feb. 2004, Art. no. 026113, https://doi.org/10.1103/PhysRevE.69.026113.

[20] J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, Red Hook, NY, USA, 2012, pp. 539–547.

[21] M. Ning, J. Gong, and Z. Zhou, "A Novel Label Propagation Method for Community Detection Based on Game Theory," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, no. 5, pp. 924–938, May 2023, https://doi.org/10.14569/IJACSA.2023.0140597.

[22] A. Kanavos, Y. Voutos, F. Grivokostopoulou, and P. Mylonas, "Evaluating Methods for Efficient Community Detection in Social Networks," *Information*, vol. 13, no. 5, May 2022, Art. no. 209, https://doi.org/10.3390/info13050209.

[23] K. R. Žalik and M. Žalik, "Density-Based Entropy Centrality for Community Detection in Complex Networks," *Entropy*, vol. 25, no. 8, Aug. 2023, Art. no. 1196, https://doi.org/10.3390/e25081196.

[24] J. Yao and B. Liu, "Community-Detection Method of Complex Network Based on Node Influence Analysis," *Symmetry*, vol. 16, no. 6, June 2024, Art. no. 754, https://doi.org/10.3390/sym16060754.

[25] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, "Finding Statistically Significant Communities in Networks," *Plos One*, vol. 6, no. 4, Apr. 2011, Art. no. e18961, https://doi.org/10.1371/journal.pone.0018961.

[26] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, June 2004, Art. no. 066133, https://doi.org/10.1103/PhysRevE.69.066133.