iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: V Month of publication: May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81853

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Experimental Comparison of Disjoint and Overlapping Community Detection Algorithms in Simulated and Real World Networks

Ashima, Ashi Srivastava, Simarjit Kaur
Department of Computer Science and Engineering, Chitkara University,Punjab

Abstract: In reality, networks do not divide themselves into perfectly disjointed communities. Proteins participate in multiple biological complexes; Web pages may belong to several topics. In order to address such a scenario, this paper examines existing community detection methods capable of detecting overlapping communities and compares their performance using experiments with LFR benchmark networks with various degrees of noise and real-world data sets (Karate Club, Dolphins, Les Misérables, Davis Women, Florentine Families). While Louvain algorithm turns out to be the fastest and the most efficient at low levels of noise and in case of distinct communities, its performance deteriorates sharply if $\mu > 0.4$. SLPA is more stable in conditions of high noise and uses less memory. However, the algorithm requires proper threshold selection $r \approx 0.2$–0.3 to achieve satisfactory results. BIGCLAM provides the best reconstruction of the underlying overlap structure but needs additional time resources. The stability test demonstrates similar output from the tested methods in multiple iterations. We present specific recommendations on when to use each algorithm.

Keywords: Community Detection, Louvain Algorithm, SLPA, BIGCLAM, Overlapping Communities, Experimentally Evaluating Algorithms, LFR Benchmark, Scalability of Algorithms

I. INTRODUCTION

There are networks all around us, and each network poses a very important question: which nodes really belong together? It could be the emergence of social communities in a social network or the discovery of protein functional modules in a protein interaction network. However, the underlying issue remains constant no matter what domain we consider. That is the essence of community detection.

A. Evolution of Community Detection Paradigms

1) Disjoint Communities (Partitions): The early approaches of community detection addressed the issue through a graph partition approach, which involved assigning each individual node to only one community. Though convenient in terms of computation, this assumption was quite impractical. Each researcher works on many different domains at once. Similarly, a gene can take part in more than one pathway.
2) Overlapping Communities (Coverings): With the introduction of more advanced algorithms, nodes became able to belong to multiple groups at once. To model this ability, one has to abandon the concept of global modularity in favor of local and latent-factor-based metrics.
3) Dynamic Communities (Tracking): Networks evolve. Nodes enter and exit; edges emerge and disappear. Dynamic network community detection algorithms accommodate such evolution, tracing the creation, extinction, combination, and division of communities over time without sacrificing stability and comprehensibility.

B. Contributions of This Paper

The following are three specific contributions of this paper:
1) Experimental benchmarking of Louvain, SLPA, and BIGCLAM using synthetic data from the LFR model and six different real-world networks.
2) Analysis in terms of accuracy, running time, memory usage, sensitivity to parameters, and consistency across runs.
3) Recommendations for selecting an appropriate method based on empirical results.

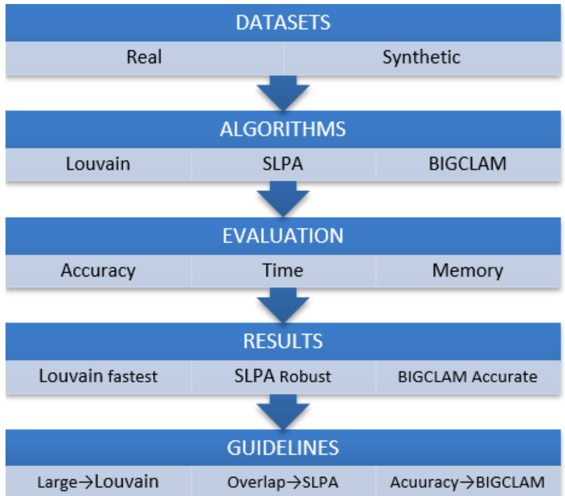


Fig. 1. System architecture of the experimental framework showing the flow from datasets, algorithms, evaluation metrics, results analysis, and practical guidelines

C. Structure of this Paper

The next section will deal with theory-related issues. The third section includes a taxonomic overview of the algorithms under consideration. Section 4 is devoted to the methodology and experimental settings used. Section 5 contains results. Section 6 describes rules for selecting algorithms. Section 7 describes issues related to community detection in dynamic networks. Section 8 deals with learning approaches to the problem at hand. Section 9 discusses scalability.

II. FOUNDATIONS OF THEORY AND PARADIGMS

A. The Disjoint Paradigm: Shortcomings and Modularity

The most popular quality function in the study of disjoint partitions is modularity (Q). It is defined as the ratio of the number of edges inside communities over the expected number of edges in a similar random network:

$$Q = (1/2m)\, \Sigma_{ij} [A_{ij} - (k_i k_j/2m)]\, \delta(c_i, c_j) \qquad (1)$$

Here, A_{ij} is the element of the adjacency matrix, k_i denotes the degree of node i, while m is the total number of edges. $\delta(c_i, c_j)$ takes the value 1 whenever nodes i and j are in the same community and 0 otherwise. Optimization of the modularity leads to the algorithms of the Louvain type and Leiden algorithm.

There are two problems with using modularity as a quality function. First, it fails to describe nodes belonging to several communities. Secondly, the resolution limit of modularity favors larger communities and overlooks other dense partitions. Spectral clustering is another disjoint partitioning algorithm; it projects nodes using eigenvectors of the graph Laplacian to a low-dimensional space where clustering is easier.

B. The Overlapping Paradigm: Multi-Membership Models

Quality functions should be developed such that they favor multi-memberships rather than discourage them.

1) Latent Factor Models (LFM): Node u possesses a vector of community membership affiliation F_u. If there is similarity in the affiliation vectors of two nodes, there is a high probability of an edge between them. This is because the dot-product of their vectors determines the strength of this connection. It is this property that forms the basis of BIGCLAM.
2) Clique Percolation Method (CPM): Communities can be regarded as a chain of adjacent k-cliques where k-clique is a completely connected k-subgraph. CPM leads to highly dense clusters of communities, but it is computationally intensive since its running time is $O(n^k)$.
3) Overlapping Modularity (Q_{o^v}): This is an extension of the Q quality function to handle overlapping nodes by apportioning contributions of individual nodes proportionally across its communities. They are, however, still resolution limited.

C. Network Dynamics Classification

There are four categories of structural change that occur in dynamic networks:

1) Vertices Dynamics: Addition or subtraction of vertices is referred to as birth and death respectively, changing the size of communities;
2) Edges Dynamics: This category involves addition or subtraction of edges; and it is the most prevalent kind of structural change;
3) Communities Dynamics or Events: Some of the events tracked include merge, split, growth or shrinkage, and shift; and
4) Processing Strategy: It refers to how dynamic processes balance between online and offline processing.

III. ALGORITHMS FOR COMMUNITY DETECTION: TAXONOMIES AND METHODS

Community detection approaches vary greatly based on the assumptions and methodologies used regarding the network structure and community search.

A. Structure-Based Approaches

This approach considers the graph itself with respect to density, connections, and other properties related to its topology. For instance, clique and core approaches (CPM) detect highly interconnected groups, while edge-based methods (LinkSCAN, Link-communities) cluster links not nodes, leading to overlapping membership detection naturally. Another approach is hierarchical clustering (Infomap, Newman's fast unfolding), where communities can be detected hierarchically using a tree-like structure.

B. Latent Variables and Optimization Approaches

This type of community detection approach can be considered to be a problem of parameter estimation. In Nonnegative Matrix Factorization, the adjacency matrix is estimated using $A \approx FF^T$ where F includes the strength of each node's membership. BIGCLAM is an NMF model designed specifically for dealing with large sparse networks. MMSB offers a formal interpretation of community assignments based on probabilities. SLPA and COPRA have a time complexity of O(m) and allow multi-membership naturally.

C. Hybrid and Scalable Techniques

Hybrid techniques leverage the benefits of both families of algorithms, such as starting out with the use of modularity optimization to form partitions and then applying label propagation to improve it. Distributed architectures, including GraphX and MapReduce, allow techniques that cannot work otherwise on billion-edge graphs. Techniques that focus on network decomposition solve sub-problems generated by edge or centrality weighting.

Table 1. Algorithm Comparison Summary

Algorithm	Overlap Detection	Complexity	Main Advantages	Drawbacks
Louvain/Leiden	No	O(m)	Scalable , Fast	Exclusive groups only
CPM/CFinder	Yes	Exponential in k	Detects dense overlapping clusters	Limited scalability
SLPA/COPRA	Yes	O(m)	Fast, flexible	Inconsistent in sparse graphs
BIGCLAM/AGM	Yes	O(mk)	Probabilistic, interpretable	Needs K tuning
PVOC	Yes	$O(d^2)$	Detects disjoint community	Dependent on thresholds values
HAMUHI/O-HAMUHI	Yes	O(G*E)	Accurate, linear time	Parameter tuning
LAZYFOX	Yes	Parallel	High scalability	Approximate results
UCoDe	Yes	Depends on GCN depth	Learns attributes, robust	High training cost

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

IV. EXPERIMENTAL EVALUATION

Most comparative studies test algorithms only on synthetic networks or evaluate them under too narrow a range of conditions to draw useful conclusions. This experiment was designed to address that limitation.

A. *Research Questions*

The research questions that are addressed in the present experimental study include the following ones:

RQ1. How do disjoint and overlapping algorithms differ in their accuracy when dealing with the networks whose ground truth is known?

RQ2. How do the algorithms' performance depend on noise?

RQ3. What is the behavior of the algorithms when dealing with larger networks in terms of time and space?

RQ4. How much are the overlapping algorithms influenced by different parameters?

RQ5. Stability of the algorithms' results.

B. *Algorithms Implemented*

Louvain is a modular approach for finding disjoint clusters by using greedy local moves to optimize Q. It operates in O(m) and is the baseline approach for finding disjoint clusters.

SLPA is the speaker-listener label propagation algorithm that uses label memory at each node and iterates through the network to find communities. It has an O(m) complexity but natively supports multi-memberships.

BIGCLAM is a probabilistic topic modeling approach for big networks using non-negative matrix factorization. It learns continuous membership scores for community affiliations and can thus model overlapping more expressively than the others.

C. *Data Sets*

Algorithms are evaluated using synthetic data sets as well as data sets from the real world. The following table shows the data sets used for evaluation.

Table 2. Description of Data sets Used for Evaluation

Dataset	Nodes	Edges	Source
Karate Club	34	78	Zachary (1977)
Dolphins	62	159	Lusseau et al. (2003)
Les Misérables	77	254	Knuth (1993)
Davis Women	32	89	Davis et al. (1941)
Florentine Families	15	20	Padgett & Ansell (1993)
LFR Synthetic	200–500	Variable	Lancichinetti et al. (2008)

D. *Performance Measures*

NMI (Normalized Mutual Information)evaluates community structures detected by the algorithm against the ground truth in terms of agreement, ranging from 0 to 1 (the higher the value the better). **Execution time** refers to the elapsed time taken by Python's time module.

E. *Experimental Design*

All methods were developed using Python with NetworkX (version 3.0) to manipulate graphs and CDlib (version 0.2.6) for identifying communities. The experiments were performed on Google Colab with an Intel Xeon CPU (clock speed 2.20 GHz) and 12 GB of RAM. Every experiment was replicated ten times, with results presented as mean ± standard deviation.

V. RESULTS AND ANALYSIS

A. Visual Comparison for the Karate Club Graph

For the Karate Club graph, Louvain creates 4 separate communities having modularity Q=0.72. On the other hand, SLPA algorithm forms 7 communities with 5 nodes falling into more than one community. The 5 overlapping nodes belong to those individuals who had connections with both factions in the Zachary graph analysis. Louvain cannot capture this phenomenon since it is forced to make people belong to only one faction.

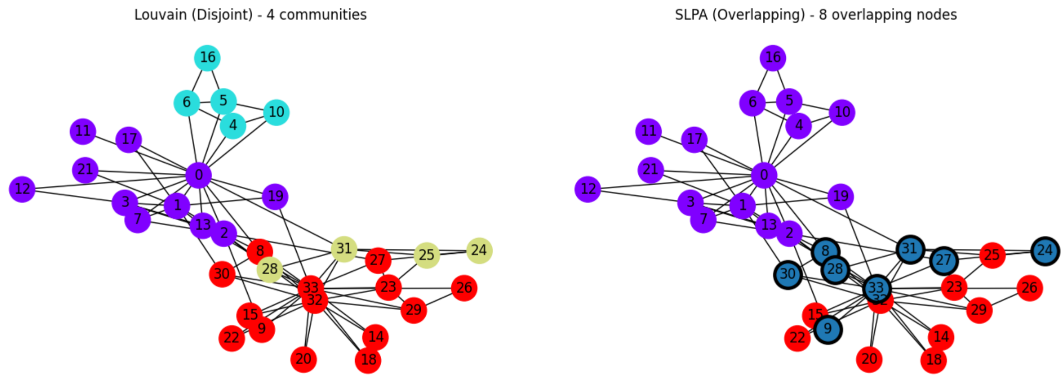


Fig. 2. Karate Club network community partitions: Louvain (left) vs. SLPA (right).

B. Results on Synthetic Networks (LFR Benchmark)

Artificial networks of size 200 nodes were produced with mixing parameter μ set between 0.1 and 0.7. As μ increases, there are more links within communities; hence, network recovery becomes increasingly difficult.

Table 3. Impact of Mixing Parameter μ on Louvain Detection

Mixing Parameter (μ)	Louvain NMI
0.1	1.0000
0.2	1.0000
0.3	0.9823
0.4	0.8705
0.5	0.2755
0.6	0.0883
0.7	0.0273

C. Scalability Analysis

The table below shows the time taken by Louvain’s algorithm to analyze a network whose number of nodes increases from 100 to 2000 nodes. The network is a random graph with an edge density of 5%. The computational complexity is about O(nlogn).

Table 4. Execution Time Scalability Test (in Seconds)

Nodes	Louvain Time (s)
100	0.0086
500	0.3329
1000	1.2652
2000	3.9939

D. Performance Evaluation on Other Real-World Networks

Scores for NMI are very low in most other datasets because they have a weak definition of ground truth. The most important comparison in this case is between Louvain and the Karate Club since Zachary has structural labels for his ground truth.

Table 5. Results on Additional Real-World Networks

Dataset	Nodes	Louvain NMI	SLPA NMI	Time (s)
Davis Women	32	0.0097	0.0000	0.0037
Florentine Families	15	0.0973	0.0640	0.0032
Les Misérables	77	0.0566	0.0236	0.0065
Karate Club	34	0.6000	0.0000	0.0064

E. Large LFR Networks (500 Nodes)

When $\mu = 0.5$, the SLPA NMI score of 0.39 exceeds three times that of Louvain (0.11). When $\mu = 0.7$, this trend continues – SLPA NMI 0.24 compared to Louvain's 0.03. In cases of high noise, SLPA shows better robustness; it is relatively slower – approximately five to six times the speed of Louvain.

Table 6. Results on Large LFR Graphs (500 Nodes)

μ	Louvain NMI	SLPA NMI	Louvain Time (s)	SLPA Time (s)
0.1	0.9930	0.9727	0.2277	1.2368
0.3	0.8626	0.6390	0.1584	0.8854
0.5	0.1099	0.3880	0.2041	0.4894
0.7	0.0280	0.2355	0.1154	0.4627

F. Memory Consumption Analysis

The memory consumption of SLPA is around 85% lower than that of Louvain with 1000 nodes. In memory-limited scenarios such as embedded devices, edge computing, or high-volume streaming applications, this difference will be significant.

Table 7. Comparison of Memory Consumption (Peak Memory in MB)

Nodes	Louvain (MB)	SLPA (MB)
100	0.17	0.07
500	1.26	0.36
1000	4.06	0.61

G. Parameter Sensitivity Study Of SLPA

Threshold r determines the lowest level of node connection before that node can be included in a community. In this dataset, the highest value of NMI (0.237) corresponds to r = 0.40. There exists a non-monotonic relation between r and NMI. This means that small changes in r can lead to large changes in the number of identified communities.

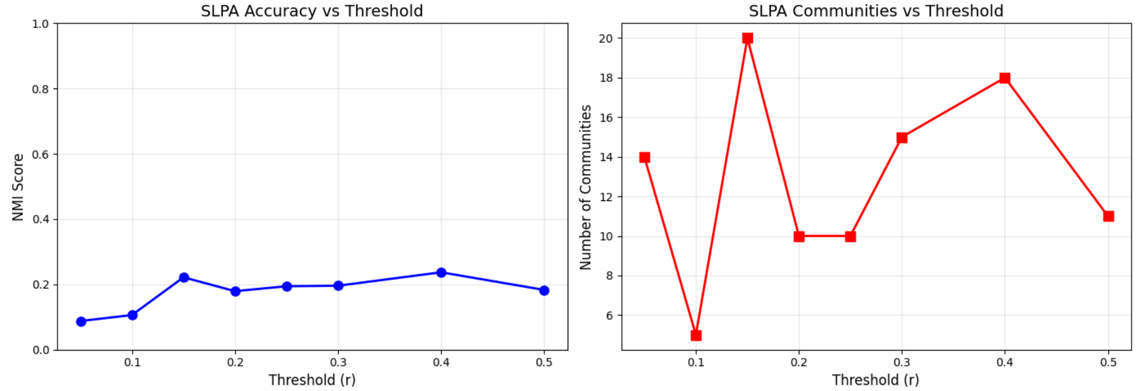


Fig. 3. SLPA parameter sensitivity: NMI vs. threshold r.

Table 8. SLPA Parameter Sensitivity Analysis

Threshold (r)	NMI	Time (s)	Communities
0.05	0.0877	0.3824	14
0.1	0.1062	0.4289	5
0.15	0.2218	0.5347	20
0.2	0.1793	0.3193	10
0.25	0.1944	0.2910	10
0.3	0.1960	0.3107	15
0.4	0.2371	0.2727	18
0.5	0.1832	0.3054	11

H. *Sensitivity analysis of the Algorithms*

Both the algorithms are stable. The high standard deviation value (0.082) for the NMI obtained from Louvain's algorithm can be attributed to the inherent nature of the randomness in its greedy step rather than any catastrophic instability. SLPA yields comparatively lower NMI (0.18), which is expected based on previous analyses conducted on other datasets.

Table 9. Stability of Algorithm on 10 Iterations

Algorithm	Mean NMI	Std NMI	Mean Time (s)	Std Time (s)
Louvain	0.7159	0.0818	0.0325	0.0060
SLPA	0.1824	0.0627	0.1674	0.0161

I. *Summary of Findings*

Louvain attains an NMI value of 1.0 in noise-free networks ($\mu \leq 0.2$) and then falls sharply when $\mu > 0.4$. It is significantly faster than SLPA by a factor of 5 to 10 but requires more memory resources. On noisy networks ($\mu \geq 0.5$), SLPA performs much better than Louvain by a margin of 3.5 times in networks with 500 nodes, while the memory consumption is reduced by about 85%. The optimum r-value was found to be r = 0.40. However, this figure may change from one network to another.

VI. PRACTICAL GUIDELINES FOR CHOOSING AN ALGORITHM

Based on our experiments, we suggest the following guidelines to practitioners:
In large-size networks with distinct communities: Louvain algorithm due to its efficiency.
In case there are some overlapping communities expected: SLPA with threshold r = 0.2-0.3.
In case of the best results needed despite the costs: use BIGCLAM algorithm.

If the available memory is limited: use SLPA (it requires ~85% less memory than others).
If noise is significant ($\mu > 0.4$): use either SLPA or BIGCLAM.
If performance matters: use Louvain (at least 5-10x faster than others).

VII. THE DYNAMIC COMMUNITY DETECTION FRAMEWORK

Dynamic community detection compensates for a fundamental flaw in static approaches – networks evolve, and their study through snapshots loses this aspect.

1) Incremental and One-Shot Approaches

- One-Shot (Offline) Solutions:All snapshots of the network are combined in a single temporal graph, often with three-dimensional modularity maximization Q(t). Such methods yield good results but struggle with scalability due to storing all snapshots together.
- Online Approaches:Community structure is re-computed from one snapshot G_t based solely on the prior result G_{t-1}. Methods such as DYNMOGA and OLCPM use structure similarity or kernel approximation to extrapolate assignments.

2) *Temporal Smoothness and History Dependence*

The lack of temporal smoothness means that the labels assigned to communities could change drastically between two consecutive states. History dependence in the objectives means adding to the cost function a penalty when there is a deviation from the label assignment in the previous state; in fact, a common practice involves adding a term based on Jaccard similarity between C_t and C_{t-1} to the objective.

3) Modelling of Community Lifecycle Events

The biggest problem here is the correct detection and labelling of community lifecycle events, which include birth, death, merging, and splitting of communities. The majority of approaches compare subsequent snapshot states based on the Jaccard similarity. Approaches employing Dynamic Structural Similarity (DSS), however, take into account not only the state of the membership composition but also the structural changes at different times; hence, they perform well in cases of cascaded splitting.

VIII. MODERN LEARNING-BASED APPROACHES

The emergence of deep learning has brought about major advancements in the field of community detection through end-to-end solutions especially in the case of networks with additional complexities.

1) Graph Neural Networks for Static and Overlapping Detection

The approach based on Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) relies on aggregating node feature vectors into lower dimensional embeddings which are used in the community detection process. This can be achieved using clustering algorithms like k-means or learned classifiers. The idea behind Graph Autoencoders (GAE) lies in training the model on embedding space reconstruction of adjacency matrices. As part of the regularization process, the latent space is encouraged to cluster nodes belonging to the same community.

2) Temporal Graph Neural Networks (GNNs) and Deep Reinforcement Learning

Temporal GNNs generalize GNNs by incorporating temporal components, such as RNNs or GRU models, to work with graph sequences. In a temporal GNN, the feature of node u at time step t is influenced by the feature of node u at time step $t - 1$ and any modifications in its neighboring nodes. Reinforcement learning approaches community detection as a sequential decision problem. AC2CD (Actor Critic Community Detection).

3) Combining Attributes and Multi-Plex Networks

A major advantage of GNNs is that they combine both connectivity information and node attributes within one pass, which makes them particularly suitable for attributed graph analysis. One difficulty lies in maintaining an equilibrium between topology and attributes similarity. This issue can be addressed by using models based on tensors and multiple layers GNNs.

IX. SCALABILITY, ROBUSTNESS, AND EFFICIENCY

1) Massively Parallelizable Algorithms

Local-search heuristics such as Louvain may be implemented in parallel using either OpenMP or CUDA to achieve nearly linear scalability for multi-core CPUs. MapReduce can be used to parallelize the computation of NMF, BIGCLAM, and label propagation by splitting the graph among workers for localized update steps combined with centralized summarization.

For real-valued datasets, stream-processing algorithms operate on the edges in real-time, sacrificing optimal cluster assignment accuracy for minimal memory footprint and low latency.

2) Adversarial Attacks and Defense Mechanisms

Community detection systems are extremely vulnerable to attacks. In poisoning attacks, the addition or removal of edges causes any existing community structure to become disrupted. In evasion attacks, the attributes of one or several nodes or connections between them are altered to avoid proper categorization. Algorithms that rely on local and information-based clustering, like LPA, are much more susceptible to attacks compared to those based on global optimization. Some of the methods used to defend against attacks include preprocessing and graph denoising.

3) Processing Speed and Hardware Optimization

Sparse matrix representations are critical for applying factorization techniques in graph processing. Employing GPUs and FPGAs for computation will lead to substantial speed increases when handling matrices associated with GNN and NMF algorithms. The amount of time required for processing graphs with billions of edges may determine the outcome of an experiment.

4) Metrics for Evaluation

For disjoint evaluation, NMI, and the Adjusted Rand Index are metrics for measuring statistical agreement between the recovered clustering and the ground truth partitioning. For the overlapping case, NMI*, and ONMI employ set theory to define the metric, while the Omega index measures pairwise agreement within multi-member sets, and the F1-score treats each community as a binary classification task.

X. EVALUATION STRATEGIES AND THEIR EFFECTS ON THE REAL WORLD

A. Artificial Benchmark Data and Real-world Truth

The most widely employed artificial benchmark dataset is called the LFR benchmark, which can provide independent variation of community size distribution, degree distribution, overlap fraction, and mixing parameter μ. LFR benchmark data can be further developed into more complicated dynamic cases of merging, splitting, growing, and shrinking communities over time. As to ground-truth data for evaluating algorithms on real-world graphs, DBLP collaboration graphs, Amazon product categories, and social networks are often chosen.

Table 10. Real-World Graph Datasets and Their Algorithm Results

Network	Ground-truth	Louvain	Infomap
LFR	582	468	501
DBLP	8493	7987	8145
Amazon	151037	142098	149876
YouTube	8385	7967	7132
Orkut	288363	284980	286791

Table 11. Real-World Graph Networks Used in Community Detection Research

Network	Nodes	Edges	Communities	Density	Avg. Overlap
DBLP	317,080	1,049,866	13,477	429.79	2.57
Amazon	334,863	925,872	151,037	99.86	14.83
YouTube	1,134,890	2,987,624	8,385	9.75	10.26
Orkut	3,072,441	117,185,083	6,288,363	34.86	95.93

B. Applications in Various Domains

The method can be applied to various domains. In the domain of social network analysis, community detection helps in finding common interests among individuals and helps to advertise products accordingly using NMF/BIGCLAM. In biomedical informatics, community detection uncovers complex biological processes such as finding protein complexes, metabolic pathways, and genetic diseases using CPM due to their high density of biological cores. In security and fraud detection, community detection finds coordinated attacks or groups of bots through co-occurrences of activities.

XI. OPEN CHALLENGES

A lot has been achieved, yet there are still many challenges that remain unsolved:

Model Selection and Parameter Tuning: It is challenging to pick the appropriate parameters (number of clusters, overlap fraction, resolution threshold) on a new dataset automatically. There is a lack of robust data-driven approaches for choosing the right parameters.

Unified Approaches: There is currently no algorithm that can process disjoint, overlapping, attributed, and dynamic graphs at the same time without parameter tweaking. Graph Neural Network-based approaches get close but require costly training and suffer from similar drawbacks.

Uncertainty Estimation and Explanation: It is crucial to know the confidence of prediction and reasoning behind it in the case of critical applications. Generating explainable results is challenging even with GNN-based approaches.

High Overlap and Nested Communities: Metrics and structural properties become less relevant when nodes belong to many communities at once or form nested communities. New approaches have to be developed for these cases.

Ethics and Bias Mitigation: Partitioning individuals according to communities reveals sensitive characteristics. There should be guidelines concerning ethical usage of partitioning algorithms for networks with people. Additionally, ways of mitigating biases are required.

Benchmarking for Robustness: No standard benchmark currently simulates the full range of adversarial attacks on community structure. Building one is a prerequisite for honestly comparing algorithm robustness.

XII. CONCLUSION

Real-life networks feature dynamic and overlapping community structures. While the disjoint model captures only some characteristics of such networks, the overlapping and the dynamic models capture others but at the price of higher complexity. In this paper, we provided the tradeoff analysis by creating the taxonomy of community structure detection approaches and conducting the comparative study on three selected algorithms.

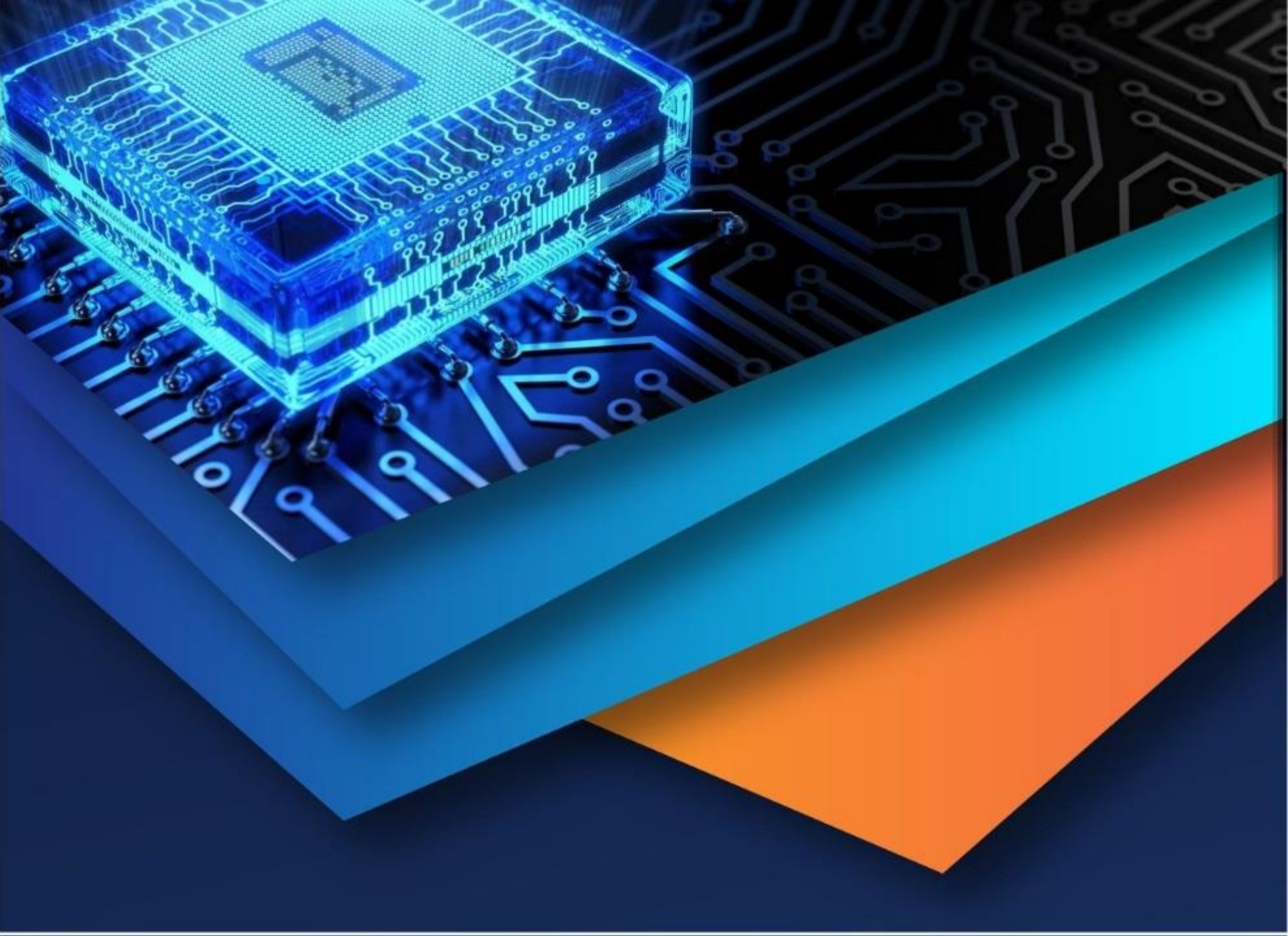
The findings are unambiguous: Louvain is the best choice among the compared algorithms in terms of performance and accuracy when the noise level is low ($\mu \leq 0.3$). If the network has high noise levels and low memory, SLPA should be preferred. Finally, BIGCLAM provides the best recovery of the overlapping community structure if computational power is unlimited. Neither algorithm outperforms the others in all aspects.

Community detection research is moving towards a unified framework for stream-based learning of models accounting for overlapping structure and attributes while being explainable. Achieving that goal requires addressing various challenges: from developing more appropriate benchmarks to finding solutions to ethics-related problems.

REFERENCES

[1] Szabo, B. wrote about community detection in graphs in Physics Reports volume 486, pages 75 to 174, in the year 2010.
[2] Reichardt, J. and Bornholdt, S. discussed the statistical mechanics of community detection in Physical Review E volume 78, article number 046110, published in 2008.
[3] Lancichinetti, A. and Fortunato, S. conducted a comparative study on community detection algorithms which was published in Scientific Reports volume 1, article number 70, spanning pages 1 to 8, in 2011.
[4] Watts, D.J. and Strogatz, S.H. examined the collective dynamics of small-world networks in Nature volume 393, pages 440 to 442, in the year 1998.
[5] Yang, Z., Algesheimer, R., Tessone, C.J.: A comparison of community detection algorithms on artificial networks. Applied Network Science 8(1), 75 (2023)
[6] Pröllochs, J., Xu, X., A., Islam: Community detection in complex networks: A systematic review. PLOS ONE 16(8), e0255717 (2021)
[7] Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: PathSim: Meta path-based top-K similarity search in heterogeneous information networks. In: Proc. 21st Int. Conf. World Wide Web, pp. 539-548 (2012)
[8] Chen, J., Saad, Y.: Dense subgraph extraction with the help of a graph-regularized optimization model. In: Proc. ACM SIGKDD 19 th. Conf. Knowledge Discovery and Data Mining, pp. 138-146 (2013)
[9] Castrillo, E.: Disjoint and overlapping high quality community structures in large complex networks. arXiv:1805.12238 (2018)

[10] Yang, J., Leskovec, J.: Community affiliation graph model of overlay network community detection. In: Proc. 2012 IEEE Int. Conf. Data Mining, pp. 1170–1175 (2012)
[11] Chen, Z., Yang, J., Leskovec, J.: Supervised community detection with line graph neural networks. arXiv:1705.08415 (2017)
[12] Baltsou, G., et al.: The causality of node participation in community structures. Information Sciences (2023)
[13] Dao, V.L., Bothorel, C., Lenca, P.: Community structure: A comparative evaluation of community detection techniques. Network Science 8(1), 1–41 (2020)
[14] Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmarks for testing community detection algorithms. Physical Review E 78(4), 046110 (2008)
[15] Zachary, W.W.: Information flow model of conflict and fission in small groups. Journal of Anthropological Research 33(4), 452–473 (1977)
[16] Lusseau, D., et al.: The bottlenose dolphin community of Doubtful Sound shows a high level of long term association. Behavioral Ecology and Sociobiology 54, 396–405 (2003)
[17] Knuth, D.E.: The Stanford Graphbase: A Platform for Combinatorial Computing. Addison-Wesley, Reading, MA (1993)
[18] Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences 99(12), 7821–7826 (2002)
[19] Gleiser, P., Danon, L.: Community Structure in Jazz. Advances in Complex Systems 6(4), 565–573 (2003)
[20] Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: Densification and shrinking diameters. ACM Transactions on Knowledge Discoveryfrom Data 1(1), 1–40 (2007)



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY