2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Overlapping Community Detection Algorithm Based on Enhanced Label Propagation with Graph Neural Network Optimization

Xiaoliang Zhang[1,2][0009-0000-7185-5821] , Xiaomeng Zhai[1,3], Meng Wang[1,4][0009-0007-8656-7137], and Hong Zhang[1,5,*][0000-0002-5318-9594]

[1] School of Cyber Security and Computer, Hebei University, Baoding 071002, Hebei, China
[2] zhangxiaoliang@stumail.hbu.edu.cn
[3] xiaomeng_abc@163.com
[4] wangmenghbu@hbu.edu.cn
[5] hzhang@hbu.edu.cn

Abstract. The structure of a community is essential for understanding complex networks, yet detecting communities efficiently and accurately remains a significant challenge. Although the label propagation algorithm offers linear-time complexity, it faces issues with low robustness, high randomness, and a tendency to form overly large communities. To overcome these limitations, we propose an Overlapping Community Detection Algorithm based on Enhanced Label Propagation with Graph Neural Network Optimization (ELP-GNN). Our approach consists of three phases: first, an enhanced label propagation algorithm is employed to identify initial communities by incorporating core node selection and importance-based propagation; second, a Graph Neural Network (GNN) model is trained on the initial communities to learn node embeddings and optimize the community structures; and finally, a fusion strategy is applied to combine the strengths of both methods. We evaluate ELP-GNN on both real-world and synthetic networks, comparing its performance with existing overlapping and non-overlapping community detection algorithms. The experimental results demonstrate that our algorithm outperforms state-of-the-art methods in terms of accuracy and robustness, particularly in complex network structures with high mixing parameters.

Keywords: Community Detection, Label Propagation, Density Peak Clustering, Graph Neural Networks, Graph Computation.

1 Introduction

Complex networks are ubiquitous in the real world, encompassing diverse domains such as railway transportation networks [1], biological networks[2], and social networks[3]. A defining characteristic of these networks is their community structure, which refers to groups of nodes that are densely interconnected internally but sparsely connected externally[4]. As an intrinsic property of complex networks, community structure plays a pivotal role in understanding network functionality, analyzing topological features, and predicting evolutionary trends. Consequently, it has attracted significant attention from the academic community, leading to the development of numerous community detection algorithms.

Community detection algorithms can be broadly categorized into non-overlapping and overlapping types. In real-world networks, nodes often belong to multiple communities—for example, individuals in social networks may participate in various interest groups—making overlapping community detection more practically relevant. Among existing algorithms, the label propagation algorithm (LPA) is widely adopted due to its near-linear complexity and scalability. However, traditional LPA has key shortcomings: it ignores topological context, treats all nodes equally (leading to "giant communities"), and relies on random tie-breaking during label updates, which compromises robustness and accuracy.

To overcome these limitations, we enhance LPA through a multi-stage refinement. First, we pre-select core nodes based on their structural importance using density peak clustering, combining node density and relative distance to identify central nodes. This avoids redundant initialization and improves label consistency. Peripheral nodes (degree 1) are directly assigned the label of their nearest core node, simplifying prior approaches that used compression-recovery strategies. Label propagation is then conducted from updated to unupdated nodes, followed by a correction step where inconsistent labels are reassigned to better match local neighborhoods.

Despite the proposed improvement strategies significantly enhance the robustness and accuracy of the traditional label propagation algorithm, some inherent limitations still exist. The method still relies heavily on local information and may be sensitive to core node selection. Furthermore, it struggles to capture global patterns in highly heterogeneous networks.

In recent years, Graph Neural Networks have emerged as powerful tools for learning representations of graph-structured data. Through sophisticated message-passing mechanisms, GNNs can effectively capture both local and global structural information, making them particularly suitable for community detection tasks. However, they typically require large amounts of labeled data and offer limited interpretability. Traditional methods like LPA are more transparent but lack representational depth.

To bridge this gap, we propose a novel hybrid approach that integrates our enhanced label propagation algorithm with GNNs. The interpretable communities from LPA serve as guidance for the GNN, which refines them using its ability to capture complex node interactions. This combination preserves the strengths of both approaches—interpretability from LPA and representational power from GNNs—yielding a more accurate and robust community detection framework. The key contributions of this paper are summarized as follows:

1. We propose an enhanced label propagation algorithm for overlapping community detection, incorporating node importance and strategic label assignment to optimize

the process, reduce randomness, and improve the accuracy of detecting both overlapping and non-overlapping communities.

2. We design a GNN-based community optimization framework that learns topological features and membership information, capturing complex overlapping patterns to enhance community boundary detection and improve the quality of identified communities.
3. We introduce an adaptive fusion strategy that combines enhanced label propagation and GNN optimization, leveraging their strengths to improve accuracy and robustness, particularly in complex networks with overlapping communities.

2 Related Work

Community detection, as a core task in complex network analysis, has attracted extensive research attention. Based on the characteristics of community structures, existing algorithms are mainly divided into non-overlapping and overlapping types. In this section, we systematically review related work from these two perspectives.

Non-overlapping community detection aims to partition nodes into disjoint groups. Traditional methods typically rely on modularity optimization to increase intra-community edge density. For example, Qiao et al.[5] combined the Mountain model and Landslide algorithm for efficient optimization, while Shi et al. [6] proposed quasi-Laplacian centrality with adaptive merging. However, such methods struggle with scalability and cannot capture overlapping structures.

For large-scale networks, label propagation-based methods have gained attention due to their efficiency. Yang et al.[7] introduced GLPA based on label similarity, Li et al.[8] incorporated neighbor label frequency and node influence, and Tang et al.[9] proposed a parallel algorithm using weights and random walks. Although effective, these methods are prone to randomness and giant community formation. Meanwhile, strategies based on node importance and community centers also emerged. For example, Zhao et al.[10] proposed compressing low-degree vertices and selecting seed nodes via density, Li et al.[11] introduced a stable method combining density peak clustering and label propagation, and Roghani et al.[12] developed a parallel Spark-based framework that computes node importance via multiple criteria.

In real-world networks, nodes often belong to multiple communities, making overlapping community detection more applicable. Roy et al.[13] proposed a fuzzy method using neighborhood similarity and improved local random walks. Gao et al.[14] used constrained personalized PageRank to suppress redundant label diffusion, while Tang et al.[15] employed local maximal cliques and label propagation. Despite their strengths, these methods often lack robustness in networks with heterogeneous community sizes and densities.

Recent advances in Graph Neural Networks (GNNs) have enabled powerful representation learning on graph-structured data. Jin et al.[16] proposed a graph attention model that adaptively weighs neighbor importance, enhancing detection in heterogeneous networks. However, GNN-based methods often require large amounts of labeled data and suffer from poor interpretability, limiting their practical deployment[17].

Although existing algorithms have made significant progress, challenges remain in terms of accuracy, stability, and efficiency in processing large-scale networks with complex overlapping community structures[18]. Most GNN-based methods require extensive supervision, while traditional algorithms struggle with network heterogeneity[19]. To address these issues, we propose a hybrid approach combining the interpretability and efficiency of enhanced label propagation with the expressive power of GNNs, offering a more robust and accurate solution for community detection in diverse network environments.

3 Method

In this section, we present our novel community detection approach that combines enhanced label propagation with Graph Neural Network optimization. Our method consists of three main components: (1) Enhanced Label Propagation, (2) GNN-based Community Optimization, and (3) Adaptive Fusion Strategy. Figure 1 illustrates the overall framework of our approach.

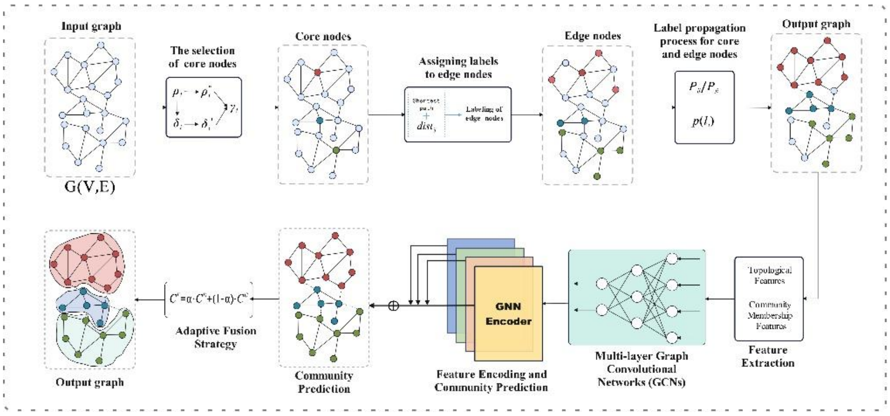

Fig. 1. ELP-GNN Framework.

The input is a graph $G(V, E)$, where $V = \{v_1, \dots, v_n\}$ represents a set of n nodes and E represents the set of edges connecting pairs of nodes in V. Our approach begins with an enhanced label propagation algorithm that identifies initial communities through strategic core node selection and importance-based label propagation. Then, a GNN model is trained on these initial communities to learn node embeddings and refine community boundaries. Finally, an adaptive fusion strategy combines the results from both methods to produce the final community structure.

3.1 Enhanced Label Propagation

Traditional label propagation algorithms suffer from several limitations, including high randomness, poor robustness, and the tendency to form "giant communities." To address these issues, we propose an enhanced label propagation algorithm that incorporates node importance and strategic label assignment.

Core Node Selection. The first step in our enhanced label propagation algorithm is to identify core nodes as initial community centers. Unlike traditional methods that initialize each node with its own label, we select core nodes based on two key metrics: local density and minimum distance. The local density of a node is calculated as:

$$\rho_i = \frac{2*(EN(i)+d(i))}{|N(i)|*(|N(i)|+1)} * d(i)^2, \tag{1}$$

where $EN(i)$ is the number of edges between neighbors of node i, $d(i)$ is the degree of node i, and $N(i)$ is the set of neighbors of node i.

To measure the relative distance between nodes, we propose a "trust degree" metric based on common neighbors and edge relationships:

$$dist_{ij} = \frac{1}{\log(1+\alpha_{i,j})\cdot\log(1+\beta_{i,j})}, \tag{2}$$

where $\alpha_{i,j}$ and $\beta_{i,j}$ capture the similarity and edge density between nodes i and j, respectively.

$$\alpha_{i,j} = \frac{|CN(i,j)|+1}{max(|N(i)|,|N(j)|)} \tag{3}$$

$$\beta_{i,j} = \begin{cases} \frac{2*(E(CN(i,j)))}{|CN(i,j)|*(|CN(i,j)|-1)}, & |CN(i,j)| > 2 \\ 1, & |CN(i,j)| \le 2 \end{cases} \tag{4}$$

The distance metric $dist_{ij}$ is based on the "trust degree." By integrating this into the density peak distance formula, the relative distance between nodes is defined as:

$$\delta_i = \begin{cases} \min\limits_{j:\rho_i<\rho_j} (dist_{ij}), & if\, \exists j s.t. \rho_i < \rho_j \\ \max\limits_{j}(dist_{ij}), & otherwise \end{cases} \tag{5}$$

To identify the core nodes, we multiply the normalized density ρ_i^* by the normalized distance δ_i^* to obtain the node importance measure γ_i:

$$\gamma_i = \rho_i^* * \delta_i^* \tag{6}$$

$$\rho_i^* = \frac{\rho_i - min(\rho)}{max(\rho)-min(\rho)} \tag{7}$$

$$\delta_i^* = \frac{\delta_i - min(\delta)}{max(\delta)-min(\delta)} \tag{8}$$

We employ Chebyshev's inequality to select the core nodes, where nodes satisfying the following condition are identified as core nodes:

$$P(|X - E(X)| \ge \varepsilon * \sigma(X)) \le \frac{1}{\varepsilon^2}, \tag{9}$$

where $E(\gamma)$ represents the expected value of node importance, $\sigma(\gamma)$ denotes the standard deviation, and ε is a positive real number.

Label Propagation Process. We propose an enhanced shortest path algorithm to assign edge nodes the label of the nearest core node based on computed distances. Core nodes propagate their labels and distances to neighbors, which select the label with the shortest distance, increment the distance by 1, and forward the updated information to un-updated neighbors until all nodes are labeled. For edge nodes equidistant from multiple core nodes, a trust-based distance metric (Equation 2) determines the nearest core node. The algorithm processes large-scale networks efficiently using parallel batches of equidistant nodes.

Nodes are initialized post-labeling: core nodes with their ID and "isUpdated" set to true, edge nodes with assigned labels and true, and others with 0 and false. The label propagation probability, considering node importance and similarity, is calculated as:

$$p_{ij} = \gamma_i \frac{Sim(i,j)}{\sum_{k\in N(i)} Sim(i,j)} + Sim(i,j) \tag{10}$$

Here, p_{ij} represents the probability of node i propagating its label to node j. Leveraging the characteristic that the greater the distance between two nodes, the smaller their similarity, we derive the similarity formula as follows:

$$Sim(i,j) = \frac{1}{dist_{ij}} \tag{11}$$

When a node receives multiple labels, we employ the following formula to calculate the weight of each label:

$$P(l_i) = \sum_{j\in N(i)} p_{ji} * TF(l_i, l_j) \tag{12}$$

Here, $TF(l_i, l_j)$ indicates whether the labels of the nodes are identical (1 if identical, 0 otherwise). Based on the weights, we sort the labels in descending order. Subsequently, we define a custom threshold φ, which specifies the maximum number of labels a node can possess. If the number of labels is less than or equal to the threshold, all labels are selected as the node's labels; if the number of labels exceeds the threshold, only the top φ labels are chosen.

3.2 GNN-based Community Optimization

After deriving initial communities via enhanced label propagation, we refine them using a Graph Neural Network (GNN). This process includes feature extraction, model design, training, and final prediction.

Feature Extraction. We extract both topological and community features:

1. **Topological Features**: Node-level metrics such as degree centrality, clustering coefficient, eigenvector centrality, and PageRank.
2. **Community Features**: One-hot or multi-hot vectors representing initial community assignments.

These are concatenated into the input matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$.

GNN Model Architecture. Our GNN model consists of multiple graph convolutional layers and a final prediction layer:

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \tag{13}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, and $\mathbf{H}^{(0)} = \mathbf{X}$.

The final output is a probability matrix:

$$\mathbf{P} = \text{sigmoid}(\mathbf{H}^{(L)}\mathbf{W}^{(L)}), \tag{14}$$

where $\mathbf{P}_{ij}$ represents the probability of node i belonging to community j.

Model Training.

We train the GNN using a joint loss function:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda\mathcal{L}_{struct}, \tag{15}$$

where $\mathcal{L}_{struct}$ encourages dense intra-community and sparse inter-community connections:

$$\mathcal{L}_{struct} = \sum_{(i,j)\in E} \parallel \mathbf{P}_i - \mathbf{P}_j \parallel_2^2 - \beta \sum_{(i,j)\notin E} \parallel \mathbf{P}_i - \mathbf{P}_j \parallel_2^2. \tag{16}$$

Community Prediction. Final community assignments are derived by thresholding:

$$C_{GNN}(i,j) = \begin{cases} 1, & \text{if } P_{ij} \geq \theta \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

3.3 Adaptive Fusion Strategy

To combine outputs from the label propagation (C_{LPA}) and GNN (C_{GNN}), we propose a weighted fusion:

$$C_{fused}(i,j) = \begin{cases} 1, & \alpha \cdot C_{LPA}(i,j) + (1-\alpha) \cdot C_{GNN}(i,j) \geq \gamma \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where α controls method preference, and γ is a threshold.

We further refine assignments by enforcing local consistency:

$$C_{final}(i,j) = \begin{cases} 1, & \frac{\sum_{k\in N(i)} C_{fused}(k,j)}{|N(i)|} \geq \delta \text{ or } C_{fused}(i,j) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

This fusion balances the efficiency of label propagation with the expressiveness of GNNs, yielding robust, accurate community detection across diverse networks.

4 Experimental Evaluation

In this section, we evaluate our enhanced label propagation with GNN optimization approach on both real-world and synthetic networks. Our experiments were conducted on a cluster of six machines, each with an Intel(R) Core i9-10900K 3.70 GHz 20-core CPU, 64 GB of memory, and 1000 MB/s bandwidth. We evaluate both non-overlapping and overlapping community detection capabilities.

4.1 Experimental Datasets

We used both real-world networks (Table 1) and synthetic networks generated using the LFR[20] benchmark. For synthetic networks, we generated both non-overlapping (LFR1) and overlapping (LFR2) community networks with parameters shown in Table 2.

Table 1. Description of real-world networks

Name	Vertices	Edges	CN	Overlap Density
Karate[21]	34	78	2	Low
Football[22]	115	613	12	Medium
Amazon[23]	334863	925873	75149	High
Youtube[23]	1134890	2987624	8385	High
DBLP[23]	317080	1049866	13477	Medium

Table 2. Description of synthetic networks

Type	Name	N	K	Maxk	Minc	Maxc	$t1/on$	$t1/om$	μ
Non-overlapping	LFR1	10000	20	100	50	100	2	1	0.1-0.8
Overlapping	LFR2	10000	20	100	50	100	0.1	3	0.1-0.8

For evaluation, we used several metrics including Normalized Mutual Information (NMI), Overlap modularity, F1-Score, and Accuracy (ACC). These metrics measure the similarity between detected communities and ground truth communities, with values closer to 1 indicating better performance.

4.2 Baseline Methods

We compare our method against several baselines, including traditional algorithms PCOPRA[24] and PSCAN[25], as well as three variants of our own approach: ELP (enhanced label propagation alone), GNN (GNN-based detection without propagation), and ELP-GNN (our full model integrating enhanced label propagation, GNN optimization, and adaptive fusion).

4.3 Performance Comparison

Non-overlapping Community Detection. Table 3 shows the comparison of NMI and F1-Score values for non-overlapping community detection on real-world networks. Our full approach (ELP-GNN) consistently outperforms both traditional methods and pure GNN-based methods across most datasets.

Table 3. NMI and F1-Score values for non-overlapping communities on real-world networks

NMI values						
Algorithm	karate	dolphins	football	polbooks	polblogs	email-Eu-core
ELP-GNN	**0.952**	**1**	**0.923**	**0.702**	**0.614**	**0.712**
ELP	0.837	**1**	0.900	0.673	0.580	0.687
GNN	0.891	0.923	0.885	0.654	0.592	0.673
PCOPRA	0.223	0.563	0.823	0.531	0.540	0.455
PSCAN	0.503	0.202	0.914	0.522	0.190	0.588
F1-Score values						
ELP-GNN	**0.985**	**1**	**0.788**	**0.803**	**0.782**	**0.412**
ELP	0.970	**1**	0.743	0.770	0.753	0.376
GNN	0.942	0.957	0.762	0.751	0.764	0.381
PCOPRA	0.614	0.596	0.410	0.562	0.488	0.138
PSCAN	0.389	0.144	0.773	0.400	0.101	0.150

Overlapping Communities. To comprehensively evaluate the performance of our proposed ELP-GNN method in detecting overlapping communities, we conducted extensive experiments on both synthetic networks generated using the LFR benchmark and real-world networks. We compared our approach with several state-of-the-art methods including ELP (our enhanced label propagation algorithm without GNN), standalone GNN, PCOPRA, PSCAN.

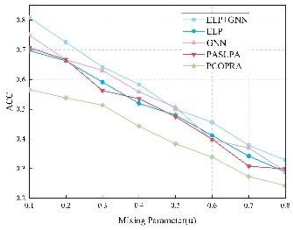

Fig. 2. Accuracy Comparison on Synthetic Networks

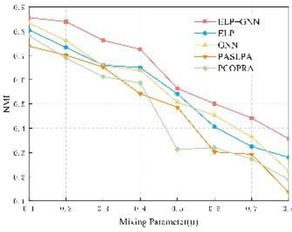

Fig. 3. NMI Comparison on Synthetic Networks

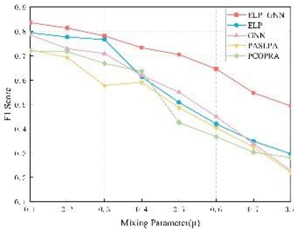

Fig. 4. F1 Score Evaluation on Synthetic Networks

Fig.2 shows the accuracy of five models on synthetic networks as the mixing parameter μ increases. ELP-GNN achieves the highest accuracy (0.75 at μ=0.1) and remains superior (0.35 at μ=0.5, 0.15 at μ=0.8), indicating strong robustness. Fig.3 presents NMI trends, where ELP-GNN consistently outperforms others, especially at low μ (0.65 at μ=0.1, 0.35 at μ =0.4), while PASLPA and PCOPRA show limited adaptability. As shown in Fig.4, ELP-GNN also leads in F1 score (0.85 at μ=0.1), outperforming all baselines, with a notable gap over PASLPA and PCOPRA, particularly at high μ values.

Across all three metrics, a pronounced downward trend is observed for all methods as μ increases. This trend is expected and aligns with the inherent difficulty of community detection in synthetic networks. As μ rises, the community structure becomes increasingly indistinct, thereby escalating the detection challenge.

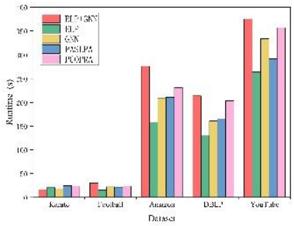

Fig. 5. Computational Efficiency Analysis

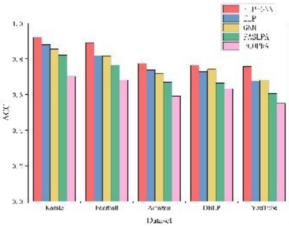

Fig. 6. ACC Evaluation on Real-world Networks

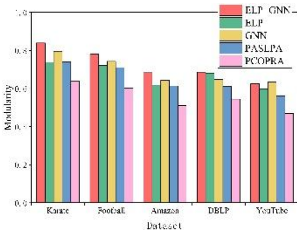

Fig. 7. Modularity on Real-world Networks

Fig.5 compares the runtime of five methods on real-world datasets. While all methods perform similarly on small datasets (Karate, Football), ELP-GNN shows significantly higher runtime on large datasets (Amazon, DBLP, YouTube), indicating scalability limitations. Fig.6 shows that ELP-GNN and ELP achieve high accuracy on small datasets, with ELP-GNN maintaining a clear advantage on larger ones (e.g., 31.9% higher than PCOPRA on YouTube, 34.6% on Amazon). As shown in Fig.7, all methods attain high modularity on simple datasets, but ELP-GNN stands out on Amazon and DBLP, with improvements of 23.2% over PCOPRA and 6% over GNN, highlighting its ability to detect tightly connected communities.

In summary, ELP-GNN demonstrates significant advantages in terms of accuracy and modularity, particularly on complex and large-scale datasets, although it is less efficient computationally. ELP performs similarly to ELP-GNN but is slightly inferior. GNN shows stable performance but is generally less effective than the former two. PASLPA and PCOPRA are computationally efficient but fall short in terms of accuracy and modularity compared to other methods.

5 Conclusion

In this paper, we propose a novel community detection approach that integrates enhanced label propagation with GNN optimization. By integrating node importance metrics and refined label assignment, our label propagation alleviates issues like randomness and giant communities. A GNN framework further captures complex relationships and overlapping structures. An adaptive fusion strategy leverages both components to enhance robustness and accuracy. Experiments on real and synthetic networks show that our method consistently outperforms state-of-the-art approaches, especially under high mixing conditions, while maintaining efficiency for large-scale distributed settings.

Acknowledgments. This work was supported by the Key Laboratory on High Trusted Information System in Hebei Province, Hebei University, Baoding 071000, Hebei, China.

References

1. Feng, L., Kang, L., Duan, Y., Cheng, S., Fei, D.: Modeling the heterogeneous traffic correlations in urban road systems using traffic-enhanced community detection approach. Physica A: Statistical Mechanics and its Applications. 501, (2018).
2. M'Barek, M.B., Borgi, A., Bedhiafi, W., Hmida, S.B.: Genetic Algorithm for Community Detection in Biological Networks. Procedia Computer Science. 126, 195–204 (2018).
3. Pattanayak, H.S., Sangal, A.L., Verma, H.K.: Community detection in social networks based on fire propagation. Swarm and Evolutionary Computation. (2018).
4. Paoletti, G., Gioacchini, L., Mellia, M., Vassio, L., Almeida, J.: CoDÆN: Benchmarks and Comparison of Evolutionary Community Detection Algorithms for Dynamic Networks. ACM Transactions on the Web. (2025).
5. Chen, N., Liu, Y., Cheng, J., Liu, Q.: A novel parallel community detection scheme based on label propagation. World Wide Web. 21, 1377–1398 (2018).
6. Shi, T., Ding, S., Xu, X., Ding, L.: A community detection algorithm based on Quasi-Laplacian centrality peaks clustering. Applied Intelligence. 51, 7917–7932 (2021).
7. Yang, G., Zheng, W., Che, C., Wang, W.: Graph-based label propagation algorithm for community detection. International Journal of Machine Learning and Cybernetics. 11, 1319–1329 (2020).
8. Li, C., Tang, Y., Tang, Z., Cao, J., Zhang, Y.: Motif-based embedding label propagation algorithm for community detection. International Journal of Intelligent Systems. 37, 1880–1902 (2022).

9. Tang, M., Pan, Q., Qian, Y., Tian, Y., Al-Nabhan, N., Wang, X.: Parallel label propagation algorithm based on weight and random walk. Mathematical Biosciences and Engineering. 18, 1609–1628 (2021).
10. Zhao, X., Liang, J., Wang, J.: A community detection algorithm based on graph compression for large-scale social networks. Information Sciences. 551, 358–372 (2021).
11. Li, C., Chen, H., Li, T., Yang, X.: A stable community detection approach for complex network based on density peak clustering and label propagation. Applied Intelligence. 52, 1188–1208 (2022).
12. Roghani, H., Bouyer, A., Nourani, E.: PLDLS: a novel parallel label diffusion and label selection-based community detection algorithm based on spark in social networks. Expert Systems with Applications. 183, 115377 (2021).
13. Roy, U.K., Muhuri, P.K., Biswas, S.K.: NeSiFC: Neighbors' Similarity-Based Fuzzy Community Detection Using Modified Local Random Walk. IEEE Transactions on Cybernetics. (2021).
14. Gao, Y., Yu, X., Zhang, H.: Overlapping community detection by constrained personalized PageRank. Expert Systems with Applications. 173, 114682 (2021).
15. Tang, Z., Li, C., Tang, Y.: An Efficient Method Based on Label Propagation for Overlapping Community Detection. In: 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD). pp. 168–173. IEEE (2021).
16. Jin, W.: Graph mining with graph neural networks. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. pp. 1119–1120 (2021).
17. Shchur, O., Günnemann, S.: Overlapping Community Detection with Graph Neural Networks. Deep Learning on Graphs Workshop, KDD. (2019).
18. Liu, H., Wei, J., Xu, T.: Community detection based on community perspective and graph convolutional network. Expert Systems with Applications. 231, 120748 (2023).
19. Yuan, S., Wang, C., Jiang, Q., Ma, J.: Community detection with graph neural network using Markov stability. In: 2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). pp. 437–442. IEEE (2022).
20. Andrea, Lancichinetti, Santo, Fortunato, Filippo, Radicchi: Benchmark graphs for testing community detection algorithms. Physical Review E. (2008).
21. Rozemberczki, B., Kiss, O., Sarkar, R.: Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20). pp. 3125–3132. ACM (2020).
22. Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proceedings of the national academy of sciences. 99, 7821–7826 (2002).
23. Yang, J., Leskovec, J.: Defining and Evaluating Network Communities based on Ground-truth, (2012).
24. Gregory, S.: Finding overlapping communities in networks by label propagation. New Journal of Physics. 12, 103018 (2010).
25. Zhao, W., Martha, V., Xu, X.: PSCAN: A Parallel Structural Clustering Algorithm for Big Networks in MapReduce. In: 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA). pp. 862–869 (2013).