Pertanika J. Sci. & Technol. 32 (3): 1285 - 1300 (2024)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Identifying Communities with Modularity Metric Using Louvain and Leiden Algorithms

Siti Haryanti Hairol Anuar[1], Zuraida Abal Abas[2]*, Norhazwani Md Yunos[2], Mohd Fariduddin Mukhtar[3], Tedy Setiadi[4] and Abdul Samad Shibghatullah[5]

[1]*Faculty of Technology and Engineering Electronic & Computer, Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia*
[2]*Faculty of Information & Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia*
[3]*Faculty of Mechanical Technology & Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia*
[4]*Faculty of Science, Fakultas Teknologi Industri (FTI), Ahmad Dahlan University (UAD), Yogyakarta, Indonesia*
[5]*School of Information Technology, Faculty of Business & Information Science (FoBIS), UCSI University Kuala Lumpur, Malaysia*

ABSTRACT

Over the past 20 years, there has been a significant increase in publication in complex network analysis research, especially in community detection. Many methods were proposed to identify community structure. Each community identification algorithm has strengths and weaknesses due to the complexity of information. Among them, the optimisation methods are widely focused on. This paper focuses on an empirical study of two community detection algorithms based on agglomerative techniques using modularity metric: Louvain and Leiden. In this regard, the Louvain algorithm has been shown to produce a bad connection in the community and disconnected when executed iteratively. Therefore, the Leiden algorithm is designed to successively resolve the weaknesses. Performance comparisons between the two and their concept were summarised in detail, as well as the step-by-step learning process of the state-of-the-art algorithms. This study is important and beneficial to the future study of interdisciplinary data sciences of network analysis. First, it demonstrates that the Leiden method

ARTICLE INFO

Article history:
Received: 15 July 2023
Accepted: 26 October 2023
Published: 04 April 2024

DOI: https://doi.org/10.47836/pjst.32.3.16

E-mail addresses:
sitiharyanti@utem.edu.my (Siti Haryanti Hairol Anuar)
zuraidaa@utem.edu.my (Zuraida Abal Abas)
wanie.my@utem.edu.my (Norhazwani Md Yunos)
fariduddin@utem.edu.my (Mohd Fariduddin Mukhtar)
tedyasni@gmail.com (Tedy Setiadi)
abdulsamad@ucsiuniversity.edu.my (Abdul Samad Shibghatullah)
* Corresponding author

Current Affiliation:
Abdul Samad Shibghatullah
College of Computing & Informatics (CCI), Universiti Tenaga Nasional, 43000 Kajang, Selangor, Malaysia

ISSN: 0128-7680
e-ISSN: 2231-8526

outperformed the Louvain algorithm in terms of modularity metric and running time. Second, the paper displays the use of these two algorithms on synthetic and real networks. The experiment was successful as it identified better performance, and future work is required to confirm and validate these findings.

Keywords: Community detection, Leiden, Louvain, modularity, network structure

INTRODUCTION

Systems in many real-world data can be modelled as a complex network. Based on Scopus online database, the number of research studies about network analysis is projected to grow by more than 148 873 publications from 1994 until now. Scopus is a large database of peer-reviewed literature and the citation and publication index owned by Elsevier, including journals, books, and conference proceedings. The concept of community detection is crucial to both graph theory and social network analysis. A network is composed of nodes as objects, as well as edges as the interaction or relation between the objects in a particular community. Community detection is one of the methods used to represent information in a complex real-world structural network. A community is described as a set of nodes that share common properties such as affiliations, similar interests, and similar information. Community detection is used to identify low-density communities with a high density within edges.

There are several different social network analysis techniques to characterise the structural network, investigate the relationship, or determine the social network group structure, depending on the application of the communities. In that sense, several studies have provided recommendations concerning algorithm selection using criteria such as network parameters, computation time, or overlap with a simulated community structure.

Generally, community detection can be categorised into global and local scopes. Global scope is dependent on prior knowledge of the entire network. Various perspectives have been developed to detect the community structure, such as hierarchical, spectral, and fuzzy clustering, as well as optimisation methods. Among them, the optimisation methods are widely focused on (Cheng et al., 2019). These class methods define community detection as the optimisation problem of an objective function.

For the performance testing of each community detection algorithm, it is preferable to run on networks that can be modified regarding the mesoscopic characteristics, including the distribution of degrees, the extent of local clustering, and the modularity of the global structure (LaRock et al., 2020). First, the degree distribution is important in determining whether learning is feasible or beneficial. Distributions of degree in complex networks can be characterised by two simplified extremes, which are homogenous and heterogeneous distributions. Second is the extension of the local clustering coefficient. It measures the degree of connectivity between two nodes and finds the important node.

The third is modularity, also known as quality function and denoted as Q (Newman, 2006; Newman & Girvan, 2004). The modularity function can be used to determine a community's strength. In this way, modularity measures can be applied to community detection to find the density of connections within communities rather than between communities (Blondel et al., 2008). Several community detection algorithms utilise modularity as an objective function to be maximised (Yuan & Liu, 2021). Modularity can be leveraged only in undirected, static, and non-overlapping networks. It is also used to denote the dendrogram's line and mark the conclusion of the algorithm that shows the values of effective partitions.

This paper forms a relationship between degree distribution and modularity metric from these three properties. This paper mainly evaluates Louvain and Leiden, two community detection algorithms based on agglomerative methods using modularity metric as an objective function. Numerous graph analysis software uses the well-known Louvain algorithm. Despite the fact that both techniques use similar steps at the start of the two phases, the Leiden algorithm performs better than Louvain because of improvements made during the refining phase before the identification of the community.

The main contributions of this paper can be summarised as follows:

- Through experimental comparison, this study demonstrated that Leiden is a superior algorithm to Louvain concerning performance comparison by focusing on the modularity metric and running time.
- This paper displays the use of these two algorithms on synthetic and real networks.

Consequently, this would aid other users in selecting the best community detection algorithm based on the results that showed the Leiden approach is the latest and faster than the Louvain algorithm.

LITERATURE REVIEW

Hierarchical Clustering

In the past 20 years, several community detection techniques have been created and implemented (Gilad & Sharan, 2023). One of the approaches in community analytics methods is hierarchical clustering. Hierarchical clustering can be classified into divisive and agglomerative methods (Newman & Girvan, 2004). The divisive method uses a split-process mechanism and contains two community detection algorithms: Girvan-Newman and the leading eigenvector algorithm. Meanwhile, the agglomerative method uses a merge process consisting of four community detection algorithms: fast greedy, walktrap, Louvain, and Leiden.

Each of the past research initiatives analysed the network's content from a unique perspective and then used this analysis to discover communities by considering the topological information of a network (Ullah et al., 2022). Recent trends in the practical application of community detection in network structure especially using the Louvain and Leiden algorithm, are in healthcare (Chatterjee & Sanjeev, 2023; Evans et al., 2022; Jin et al., 2020; Kabir et al., 2019; Kramer et al., 2020; Nallusamy & Easwarakumar, 2023; Nicolini et al., 2017; Rahiminejad et al., 2019), social network (Chessa et al., 2023; Irsyad & Rakhmawati, 2019; Li et al., 2023; Park & Kwon, 2022; Torene et al., 2022), telecommunication (Ding et al., 2022; Zu et al., 2021), economic (Han et al., 2018; Wang et al., 2022), intelligent (Karyotis et al., 2018; Singhal et al., 2020) and nature (Peeples & Bischoff, 2023; Wang & Wang, 2022; Xie et al., 2022).

MATERIALS AND METHOD

Preliminary

We formulate some preliminaries and notations used in the proposed method in the form of definitions listed in Table 1.

The definition of modularity, Q, is represented in Equation 1 as follows:

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

Table 1
Notations used in this paper

Notation	Description
E	The set of edges
G	The whole network
m	The number of edges
n	The number of nodes
Q	The modularity
t	The running time in seconds
V	The set of nodes

In $G = (V,E)$, an undirected and unweighted network, the quality function of community structure can be measured by modularity metric, Q where A_{ij} represent the adjacency matrix, k_i and k_j is the degree of node i and j, respectively, C_i and C_j are the community of node i and j, respectively, and $\delta(\circ)$ represents Kronecker function as in Equation 2.

$$\delta\left(C_i, C_j\right) = \begin{cases} 1, \text{ if } C_i = C_j \\ 0, \text{ if } C_i \neq C_j \end{cases} \quad (2)$$

The purpose of the Kronecker-delta function is to ensure that if nodes i and j are in different communities, then Q is zero. Otherwise, the value of 1, provided that both node i and j belong to the same community C_i and C_j.

Framework

After introducing the key notations and description, we begin to run the experiment based on this framework in Figure 1, which summarises the complete flow of the experiment.

Network datasets from synthetics and real-world networks were collected in the first stage. An artificial network termed Lancichinetti-Fortunato-Radicchi (LFR) was adopted as a benchmark for the synthetic type with three different network sizes consisting of small, medium, and large. Similar network sizes were also applied for the six real-world networks utilised.

The three network sizes are used to evaluate the performance algorithm, and the network parameters were set up earlier based on the benchmark network. The small size contains less than one thousand nodes, the medium size network contains between one thousand and ten thousand nodes, and the large size is more than that.

The input dataset was read as an edge list for the next step and transformed into adjacency matrices using Python. Three popular graph libraries with Python bindings, namely cdlib, NetworkX and igraph, were used for performance comparison between the two techniques. Cdlib is a powerful Python package that allows the extraction, comparison, and evaluation of communities from complex networks. NetworkX was implemented using pure Python methods, whereas igraph was run using C language. In this paper, the Louvain and Leiden algorithms were applied using the three libraries and tested in the same network. The details of two representative algorithms have been explained in comparison to the algorithm.

For the final step, both algorithms were evaluated through modularity metrics, and the running time was set up in unit seconds.

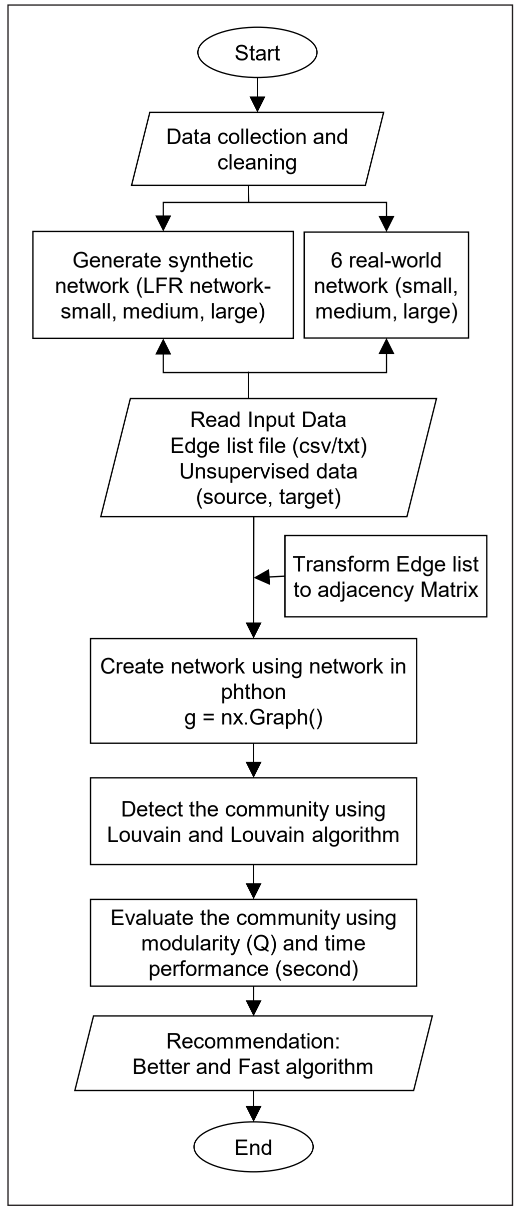


Figure 1. Workflow of the experiment

Comparison of Algorithms

A hierarchical clustering strategy can rapidly generate highly modular communities in a large network. There are two phases in the Louvain algorithm (Blondel et al., 2008). The first stage is the local moving of nodes for modularity optimisation, while the second stage is the community merging or network aggregation process. A static network is required for Louvain to produce an efficient output. In a large network, this approach, which belongs to the hierarchical clustering category, may quickly create communities with a high degree of modularity.

Meanwhile, the Leiden algorithm enhances the Louvain algorithm (Traag et al., 2019). Despite the fact that it is more complicated than its counterpart, this algorithm is able to derive a faster and more precise computation time. As opposed to Louvain's, the Leiden algorithm comprises three phases, with the modularity optimisation process being the first, followed by partition refinement, and the community aggregation process in the last step. In addition, this algorithm works well on large-scale, medium, and small networks. Figure 2 compares hierarchical clustering, and Figure 3 shows the infographic of the Louvain and Leiden algorithms.

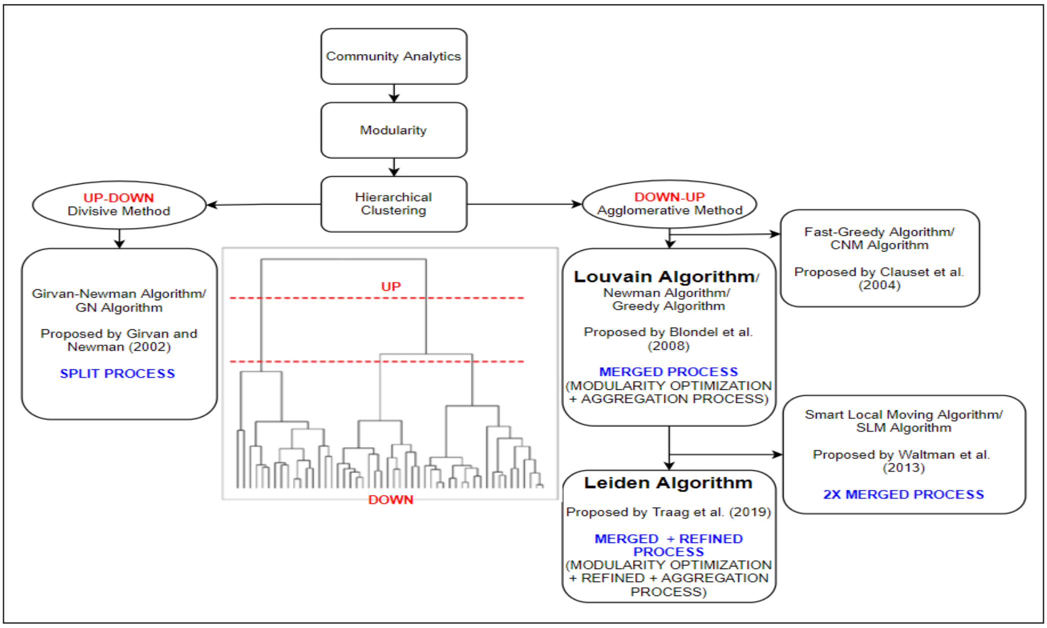
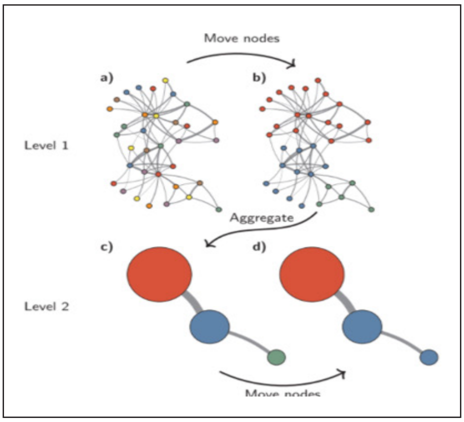
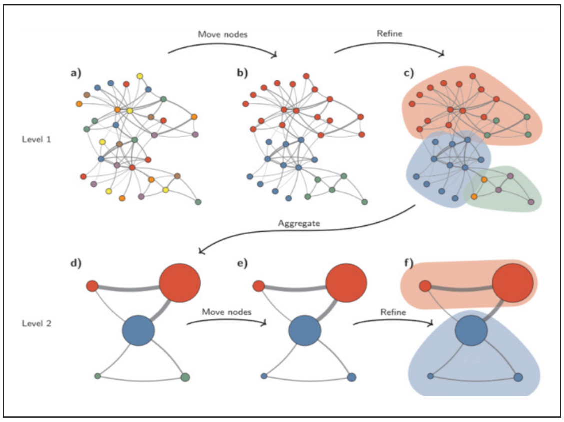


Figure 2. Hierarchical clustering (Anuar et al., 2021)





(a) (b)

Figure 3. Infographic of: (a) Louvain; and (b) Leiden algorithms method (Traag et al., 2019)

EXPERIMENTS

Experimental Environment

All the algorithms were executed on a device with installed Python running on an Intel Core i7-7700 CPU @ 2.8GHz and 24GB of RAM.

Experimental Data

Many researchers have proposed their community detection and experiment on various types of networks as a benchmark. (Chunaev, 2020) The small dataset consists of less than 10^3 nodes, the medium dataset consists of nodes between 10^3 and 10^5, while the large dataset consists of nodes of more than 10^5. (Chunaev, 2020) For convenience, the datasets were distinguished by size, namely, small, medium, and large. Louvain and Leiden's algorithms were tested on a range of synthetic networks, as well as six real-world datasets.

Synthetic Networks

To evaluate the performance of two community detection algorithms, Louvain and Leiden, we generated a synthetics network with a known ground truth called extended LFR. The details are shown in Table 2. Three artificial network data sets were constructed. The difference between the six artificial network data sets lies in the blend factor number of nodes, representing the small, medium, and large networks.

LFR network as a benchmark of synthetic network can be built very quickly, and the complexity of the construction algorithms is linear in the number of links of the graph. So, one can perform tests on very large systems, provided the study method is fast enough to analyse them.

Table 2
Synthetic network used in experiment

LFR Network	Parameters					
	n	$\tau 1$	$\tau 2$	c_{min}	k	μ
Small	500	3	1.5	20	5	0.1-1.0
Medium	7000	3	1.5	20	5	0.1-1.0
Large	10000	3	1.5	20	5	0.1-1.0

Real-world Networks

We also performed experiments on six real-world networks. The dataset retrieved from the UCI machine learning library is detailed in Table 3.

Real-world datasets can include complex networks from sociology, communication, biology, and transportation domains. Fundamentally, real-world or empirical networks are unknown ground truths. The description of each network is provided as follows:

1. Zachary network: A social network of connections formed by 34 karate club members of a US university's karate club in the 1970s.
2. Democratic National Committee (DNC): The official administration body of the United States Democratic Party. It undirected a network of individuals who received the same email in 2016.
3. Cora: A seven-class network of scientific publications in the citation network. The classes include genetic algorithms, case-based reasoning, neural networks, probabilistic techniques, rule learning, reinforcement learning, and theory.
4. Wikipedia: A Wikipedia voting for the promotion of administrator ship. A to B directed edge indicates that user A voted on B to become a Wikipedia administrator.
5. Enron email: An undirected network of communication emails sent around 500,000 emails from the Federal Energy Regulatory Commission.
6. Amazon: A network of products derived from the process of crawling the Amazon website. It is based on the feature 'Customers Who Bought This Item Also Bought' on the Amazon website.

Table 3
The real-world network features

Size datasets	Network	Domain	Feature	
			Nodes (*n*)	**Edges (*m*)**
Small	Zachary	Social network	34	78
	DNC	Communication network	906	12100
Medium	Cora	Publication network	2,708	5,429
	Wikipedia	Wiki-vote network	7,115	103689
Large	Email Enron	Communication network	36,692	183,831
	Amazon	Product network	334,863	925,872

RESULTS AND DISCUSSION

The result elaboration is divided into performance evaluation index and statistical analysis.

Performance Evaluation Index

The detailed performance result of Louvain and Leiden algorithms is explained for the 36 datasets of networks with different nodes and mixing parameters in the form of modularity metrics and running time. First, tests are performed on well-known synthetic networks.

Synthetic Networks

A set of networks was created by the LFR benchmark (Lancichinetti & Fortunato, 2009). LFR generation consists of network size N, the mixing parameters, the average degree k, the maximum degree, the minimum, and the maximum community size. Generally, the degrees of the nodes are governed by power laws with an exponent of τ1=3 and τ2=1.5, respectively. The parameters of LFR networks are set as shown in Table 2. Figure 4 shows the detection effect in the LFR networks with n = 500, 7000, and 10000.

The modularity values range between 0 and 1 (Needham & Hodler, 2021). Larger values indicate better communities, while a modularity value of less than 1 signifies that each node is a community. However, the optimal partition indicates 0.41 and above, which is the best value of partition using modularity metrics.

By comparison, both the Louvain and Leiden algorithms have good performance at the value of μ from 0.1 to 1.0 in terms of modularity. However, in terms of running time,

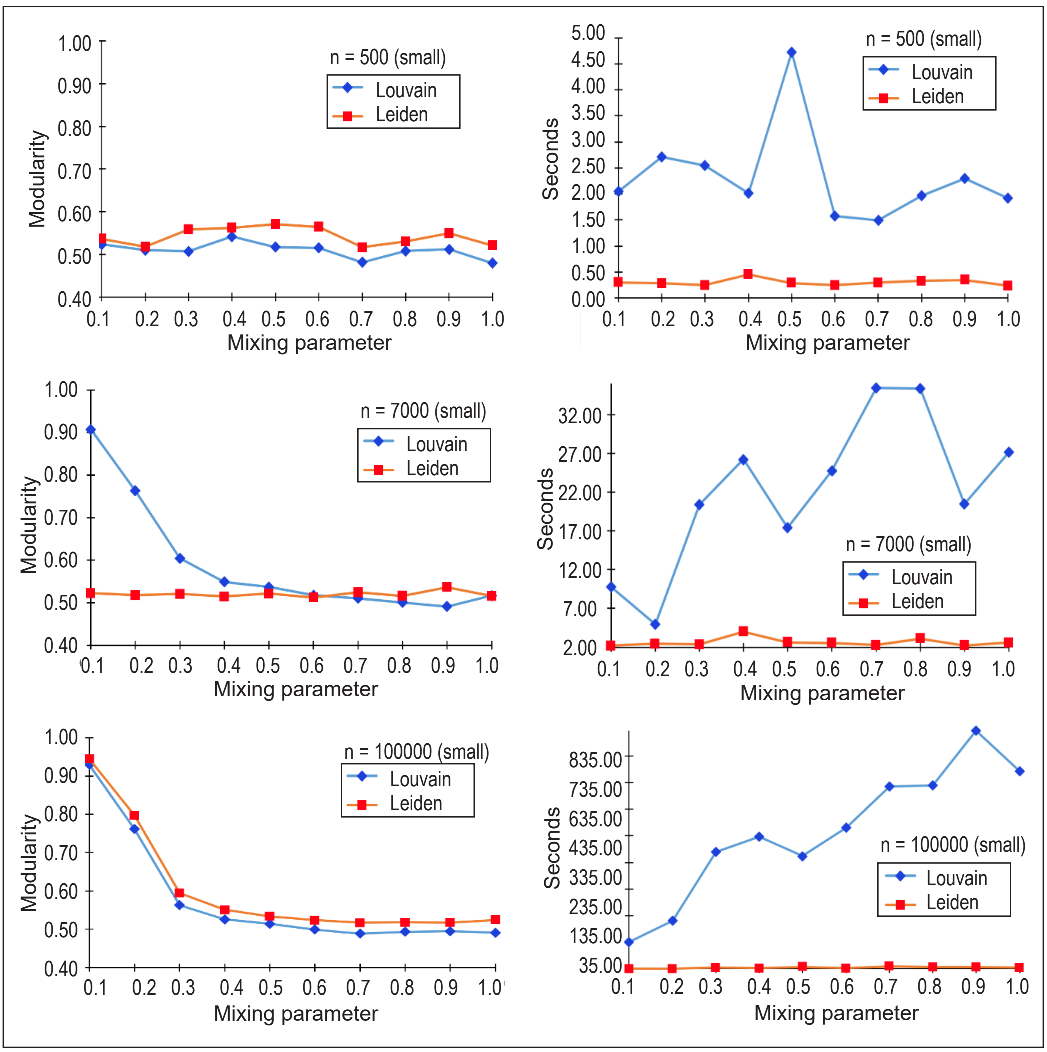


Figure 4. Results on LFR benchmark networks. Small, medium, and big notation indicates that the community sizes are in the range mixing parameter μ [0.1, 1.0]

Leiden performed well and fast. With increasing of μ, the Leiden algorithm has stable performance in the networks with k = 5 and c_{min} = 20. The algorithm has no significant difference in the networks with varying numbers of nodes. It indicates that the Leiden algorithm is stable in dense networks and is unaffected by the number of nodes and the scale of mixing parameters. However, when $\mu < 0.4$ with a big network scale, the value of modularity increases in the network with $k = 5$ and $c_{min} = 20$. When $\mu > 0.4$, the modularity value is suddenly stable at the optimal range [0.5, 0.6] for both the Louvain and Leiden algorithms.

When the mixing parameter is increased, each node is more closely connected to a local central node. This situation allows too many nodes to be merged into the same community, rapidly deteriorating the detection effect.

Real-world Networks

The selected networks include Zachary Karate Club, DNC, Cora, Wikipedia, email Enron and Amazon, which are in Table 3. Figures 5 to 7 show the detection effect in real networks with small, medium, and large sizes.

By comparison, both the Louvain and Leiden algorithms perform well in real-world networks in terms of modularity. However, in terms of running time, Leiden still performed

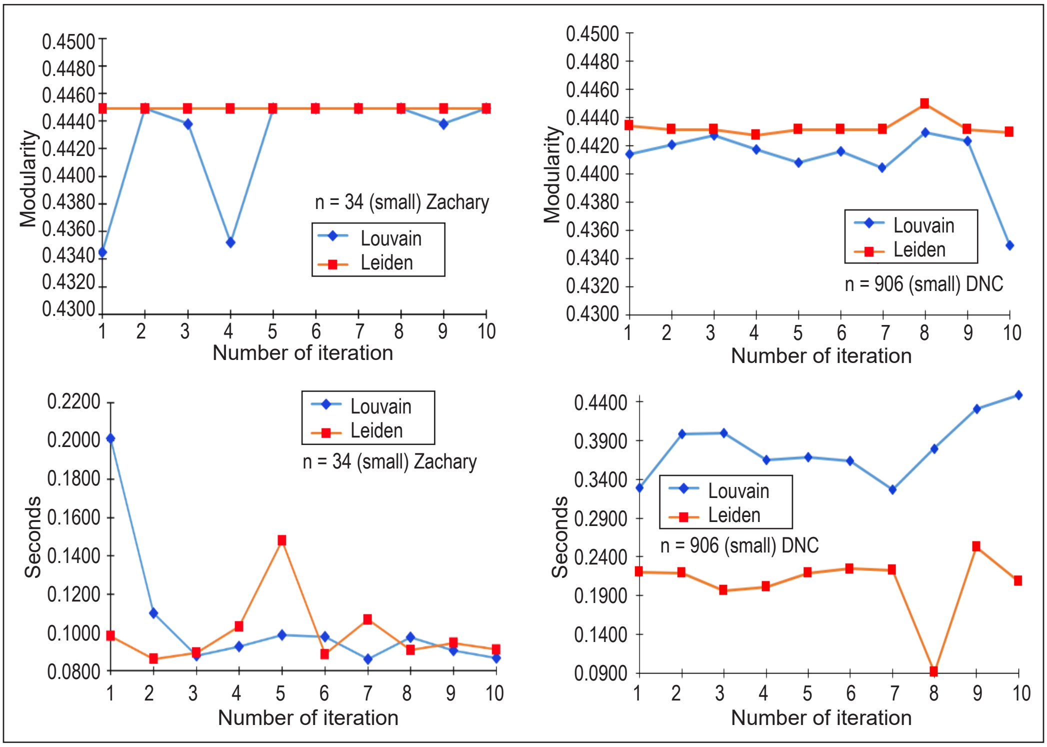


Figure 5. Results on real-world networks (small scale < 10^3 nodes)

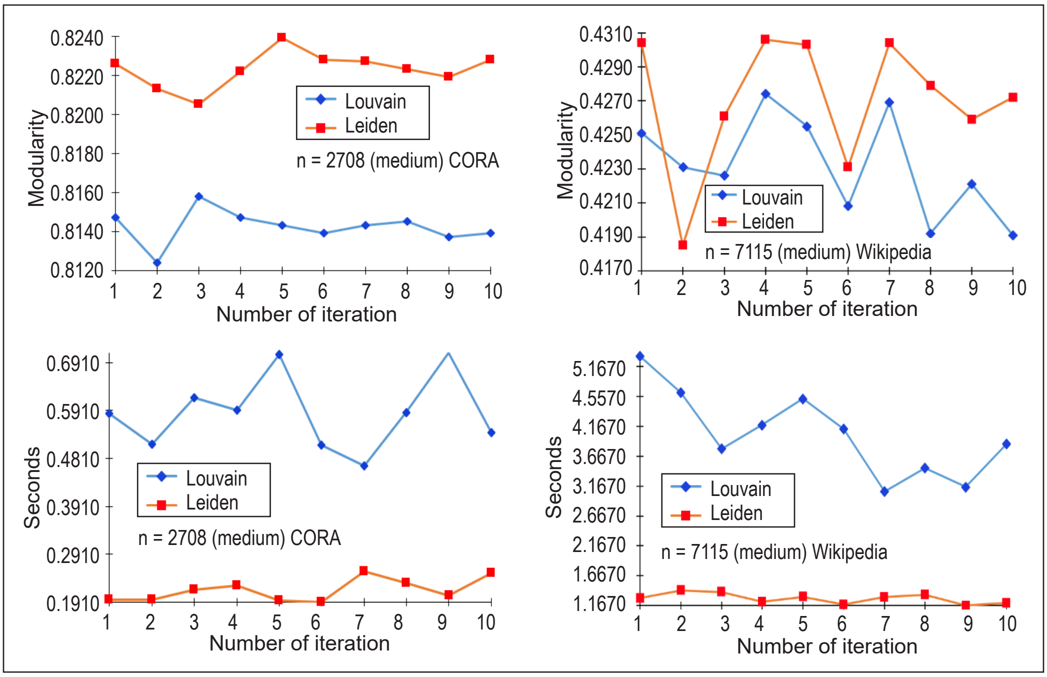


Figure 6. Results on real-world networks (medium scale 10^3 <nodes< 10^5)

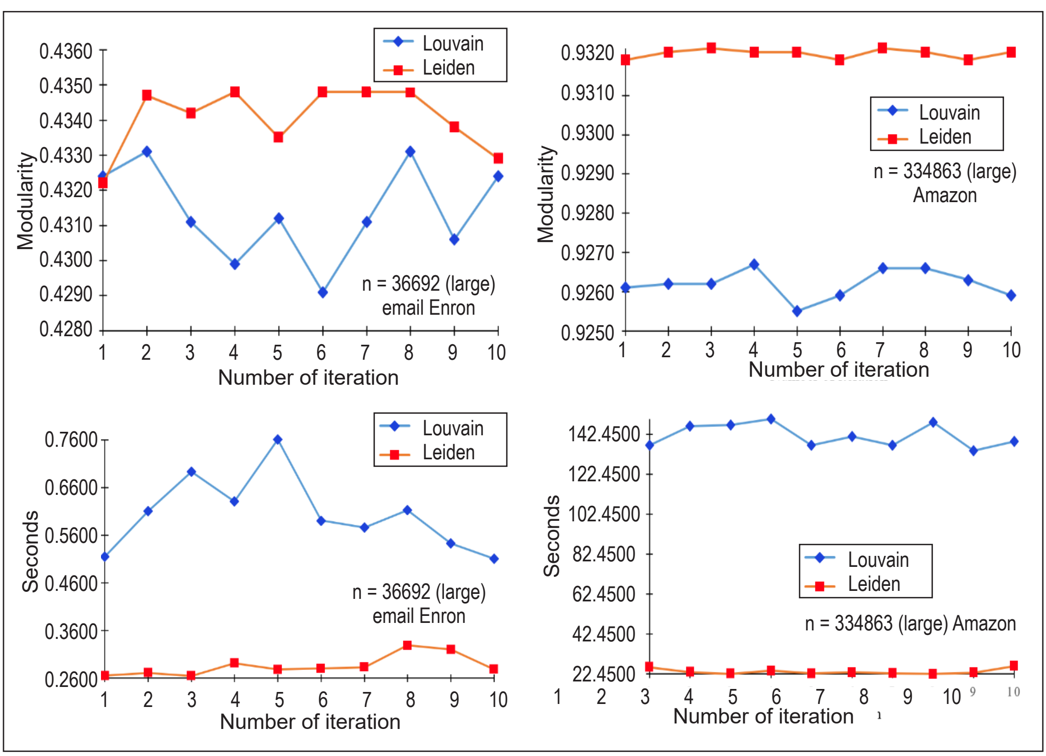


Figure 7. Results on real-world networks (large-scale nodes > 10^5)

well and fast compared to Louvain. The algorithm has no significant difference in the networks, with varying types of networks of different sizes. It indicates that the Leiden algorithm is more stable in dense networks and is unaffected by the number of nodes in the ground truth dataset.

Statistical Analysis

Statistical analysis was performed to demonstrate that the Leiden algorithm has a higher quality function (modularity metric) than the Louvain algorithm. Leiden algorithm takes less running time, as measured in seconds. Thus, a hypothesis test of unknown standard deviation, σ, and a small sample size ($n < 30$) was carried out based on the critical value technique. The standard test statistic, t, was applied following a t-distribution with a degree of freedom ($d.f.$) equal to $n - 1$.

The significance level (α) was first identified to produce the total area under the rejection region's distribution curve before determining the $d.f.$ = – 1. IBM SPSS software was used to calculate the value of t and derive the conclusion from the results.

The hypothesis tests for null and alternative are denoted by:

 # Null hypothesis (H0): There is no difference (equal) for the value in modularity and running time between Louvain and Leiden algorithms.

 # Alternative hypothesis (H1): There is a difference in the value for the modularity and running time between the Louvain and Leiden algorithms.

An independent t-test on a small-size Zachary network of 34 nodes showed no difference in modularity value and time performance for the statistically significant Louvain algorithm with t (9) = -0.17 and p = 0.123. The value of p = 0.123 was chosen because Levene's test produced a statistically significant p-value, where $p < 0.001$ and was less than 0.05. Otherwise, p = 0.106 would be chosen. p = 0.123 is greater than 0.05, thus failing to reject the null hypothesis that there is no difference in modularity value between the Louvain and Leiden algorithms for a small-size network (Zachary and LFR 500).

However, the results of medium- and large-size networks contradict the small-size network. This test proved a difference in modularity value and execution time performance between Louvain and Leiden algorithms. For the LFR network with 7000 nodes, the test generated a statistically significant result with $t(18)$ = -4.064 and p = 0.001. Thus, the null hypothesis was rejected due to the statistical difference in value in modularity and running time between Louvain and Leiden algorithms.

These results might provide insight into the impact of the number of nodes against the modularity metric's value and how long each community detection technique takes to run. The Louvain and Leiden algorithms may be suitable for usage in any network size.

CONCLUSION

This research focuses on two community detection algorithms, the Louvain and Leiden methods, which are based on agglomerative techniques using modularity. A detailed summary of the concept and benefit is provided through an experimental comparison. This study demonstrates the state-of-the-art algorithm's step-by-step learning. This study presents two-fold contributions. First, it demonstrates that the Leiden method performs better in modularity and running time than the Louvain algorithm. Second, it shows the application of both synthetic and real networks using these two approaches.

This study exhibits the experimental findings of several different-sized networks using Louvain and Leiden algorithms. Both Louvain and Leiden have an optimal value of the result, but there is an improvement from Leiden. The Leiden method was found to perform better in terms of execution time and the modularity metric. The researchers may consider this information for their project. Future research on network analysis using interdisciplinary data sciences would benefit from these findings, especially in the healthcare field.

ACKNOWLEDGEMENT

This work was supported by the Faculty Publication Fund (Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer, FTKEK) and Centre for Research and Innovation Management (CRIM), Universiti Teknikal Malaysia Malacca (UTeM).

REFERENCES

Anuar, S. H. H., Abas, Z. A., Yunos, N. M., Mohd Zaki, N. H., Hashim, N. A., Mokhtar, M. F., Asmai, S. A., Abidin, Z. Z., & Nizam, A. F. (2021). Comparison between Louvain and Leiden algorithm for network structure: A review. *Journal of Physics: Conference Series*, *2129*(1), Article 012028. https://doi.org/10.1088/1742-6596/2129/1/012028

Blondel, V. D., Guillaume, J. L. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*(10), 1–12. https://doi.org/10.1088/1742-5468/2008/10/P10008

Chatterjee, S., & Sanjeev, B. S. (2023). Community detection in Epstein-barr virus associated carcinomas and role of tyrosine kinase in etiological mechanisms for oncogenesis. *Microbial Pathogenesis*, *180*, Article 106115. https://doi.org/10.1016/j.micpath.2023.106115

Cheng, J., Su, X., Yang, H., Li, L., Zhang, J., Zhao, S., & Chen, X. (2019). Neighbor similarity based agglomerative method for community detection in networks. *Complexity*, *2019*, Article 8292485. https://doi.org/10.1155/2019/8292485

Chessa, A., D'Urso, P., De Giovanni, L., Vitale, V., & Gebbia, A. (2023). Complex networks for community detection of basketball players. *Annals of Operations Research*, *325*(1), 363–389. https://doi.org/10.1007/s10479-022-04647-x

Chunaev, P. (2020). Community detection in node-attributed social networks: A survey. *Computer Science Review*, *37*, Article 100286. https://doi.org/10.1016/j.cosrev.2020.100286

Ding, R., Fu, J., Du, Y., Du, L., Zhou, T., Zhang, Y., Shen, S., Zhu, Y., & Chen, S. (2022). Structural evolution and community detection of china rail transit route network. *Sustainability*, *14*(19), Article 12342. https://doi.org/10.3390/su141912342

Evans, J. C., Lindholm, A. K., & König, B. (2022). Family dynamics reveal that female house mice preferentially breed in their maternal community. *Behavioral Ecology*, *33*(1), 222–232. https://doi.org/10.1093/beheco/arab128

Gilad, G., & Sharan, R. (2023). From Leiden to Tel-Aviv University (TAU): Exploring clustering solutions via a genetic algorithm. *PNAS Nexus*, *2*(6), Article pgad180. https://doi.org/10.1093/pnasnexus/pgad180

Han, Z., Mo, R., Yang, H., & Hao, L. (2018). Module partition for mechanical CAD assembly model based on multi-source correlation information and community detection. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, *12*(1), Article JAMDSM0023. https://doi.org/10.1299/jamdsm.2018jamdsm0023

Irsyad, A., & Rakhmawati, N. A. (2019). Community detection in twitter based on tweets similarities in Indonesian using cosine similarity and Louvain Algorithms. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, *6*(1), Article 22. https://doi.org/10.26594/register.v6i1.1595

Jin, D., Li, R., & Xu, J. (2020). Multiscale community detection in functional brain networks constructed using dynamic time warping. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *28*(1), 52–61. https://doi.org/10.1109/TNSRE.2019.2948055

Kabir, K., Hassan, L., Rajabi, Z., Akhter, N., & Shehu, A. (2019). Graph-based community detection for decoy selection in template-free protein structure prediction. *Molecules*, *24*(5), Article 854. https://doi.org/10.3390/molecules24050854

Karyotis, V., Tsitseklis, K., Sotiropoulos, K., & Papavassiliou, S. (2018). Big data clustering via community detection and hyperbolic network embedding in IoT applications. *Sensors*, *18*(4), Article 1205. https://doi.org/10.3390/s18041205

Kramer, J., Boone, L., Clifford, T., Bruce, J., & Matta, J. (2020). Analysis of medical data using community detection on inferred networks. *IEEE Journal of Biomedical and Health Informatics*, *24*(11), 3136–3143. https://doi.org/10.1109/JBHI.2020.3003827

Lancichinetti, A., & Fortunato, S. (2009). Community detection algorithms: A comparative analysis. *Physical Review E, 80*(5), Article 056117. https://doi.org/10.1103/PhysRevE.80.056117

LaRock, T., Sakharov, T., Bhadra, S., & Eliassi-Rad, T. (2020). Understanding the limitations of network online learning. *Applied Network Science*, *5*(1), Article 60. https://doi.org/10.1007/s41109-020-00296-w

Li, S., Zhao, C., Li, Q., Huang, J., Zhao, D., & Zhu, P. (2023). BotFinder: A novel framework for social bots detection in online social networks based on graph embedding and community detection. *World Wide Web*, *26*(4), 1793–1809. https://doi.org/10.1007/s11280-022-01114-2

Needham, M., & Hodler, A. E. (2021). *Graph algorithms Practical examples in Apache Spark and Neo4j*. *O'reilly*. O'Reilly Media.

Nallusamy, K., & Easwarakumar, K. S. (2023). Classifying schizophrenic and controls from fMRI data using graph theoretic framework and community detection. *Network Modeling Analysis in Health Informatics and Bioinformatics*, *12*(1), Article 19. https://doi.org/10.1007/s13721-023-00415-4

Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, *69*(2), Article 026113. https://doi.org/10.1103/PhysRevE.69.026113

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, *103*(23), 8577–8582. https://doi.org/10.1073/pnas.0601602103

Nicolini, C., Bordier, C., & Bifone, A. (2017). Community detection in weighted brain connectivity networks beyond the resolution limit. *NeuroImage*, *146*, 28–39. https://doi.org/10.1016/j.neuroimage.2016.11.026

Park, J. H., & Kwon, H. Y. (2022). Cyberattack detection model using community detection and text analysis on social media. *ICT Express*, *8*(4), 499–506. https://doi.org/10.1016/j.icte.2021.12.003

Peeples, M. A., & J. Bischoff, R. (2023). Archaeological networks, community detection, and critical scales of interaction in the U.S. Southwest/Mexican Northwest. *Journal of Anthropological Archaeology*, *70*, Article 101511. https://doi.org/10.1016/j.jaa.2023.101511

Rahiminejad, S., Maurya, M. R., & Subramaniam, S. (2019). Topological and functional comparison of community detection algorithms in biological networks. *BMC Bioinformatics*, *20*(1), 1-25. https://doi.org/10.1186/s12859-019-2746-0

Singhal, A., Cao, S., Churas, C., Pratt, D., Fortunato, S., Zheng, F., & Ideker, T. (2020). Multiscale community detection in Cytoscape. *PLOS Computational Biology*, *16*(10), Article e1008239. https://doi.org/10.1371/journal.pcbi.1008239

Torene, S., Follmann, A., Teague, T., Chang, P., & Howald, B. (2022). Automated hashtag hierarchy generation using community detection and the Shannon Diversity Index, with applications to Twitter and Parler. *International Journal of Semantic Computing*, *16*(04), 473–496. https://doi.org/10.1142/S1793351X22500052

Traag, V. A., Waltman, L., & van Eck, N. J. (2019). From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports*, *9*(1), Article 5233. https://doi.org/10.1038/s41598-019-41695-z

Ullah, A., Wang, B., Sheng, J. F., Long, J., Khan, N., & Ejaz, M. (2022). A novel relevance-based information interaction model for community detection in complex networks. *Expert Systems with Applications*, *196*, Article 116607. https://doi.org/10.1016/j.eswa.2022.116607

Wang, C., & Wang, F. (2022). GIS-automated delineation of hospital service areas in Florida: From Dartmouth method to network community detection methods. *Annals of GIS*, *28*(2), 93–109. https://doi.org/10.1080/19475683.2022.2026470

Wang, J., Zhou, C., Rong, J., Liu, S., & Wang, Y. (2022). Community-detection-based spatial range identification for assessing bilateral jobs-housing balance: The case of Beijing. *Sustainable Cities and Society*, *87*, Article 104179. https://doi.org/10.1016/j.scs.2022.104179

Xie, L., Cui, J., Qin, Y., & Qiu, L. (2022). Analysis of deformation characteristics of reverse slope under the influence of reservoir water based on community detection. *Environmental Earth Sciences*, *81*(4), Article 110. https://doi.org/10.1007/s12665-022-10252-9

Yuan, Q., & Liu, B. (2021). Community detection via an efficient nonconvex optimization approach based on modularity. *Computational Statistics and Data Analysis*, *157*, Article 107163. https://doi.org/10.1016/j.csda.2020.107163

Zu, J., Hu, G., Yan, J., & Tang, S. (2021). A community detection based approach for service function chain online placement in data center network. *Computer Communications*, *169*, 168–178. https://doi.org/10.1016/j.comcom.2021.01.014