An influencer is someone who has the ability to persuade a large number of people to take specific actions, regardless of space or time. The role of influencers, especially on social media platforms, has grown significantly. One common feature utilized by businesses today is follower grouping. However, this feature is limited to identifying influencers based solely on mutual followership, highlighting the need for a more advanced approach to influencer detection. This study proposes a new method that integrates the Leiden coloring algorithm with Degree centrality for influencer detection. This approach employs network analysis to identify patterns and relationships within large-scale datasets. First, the Leiden coloring algorithm partitions the network into various communities, which are considered potential influencer communities. Degree centrality then enhances this process by identifying highly connected nodes, which are indicative of influencers. The proposed method is validated using crawled data from Twitter (X) with the keyword "GarudaIndonesia". The data collection process was carried out using Tweet Harvest, resulting in a dataset of 22,623 rows. The dataset was tested across three scenarios: the first with 1,000 rows, the second with 2,000 rows, and the third with 5,000 rows. The proposed method was compared with the Louvain coloring method, showing an increase in the modularity value of the Leiden coloring algorithm by 0.0240. This increase demonstrates the Leiden method's ability to achieve more optimal network partitioning. Additionally, the Leiden coloring algorithm reduced the processing time by 14.85 seconds compared to the Louvain method, highlighting its faster performance. This is particularly important for applications requiring quick results, especially in big data analysis. Lastly, the Leiden algorithm reduced the number of communities by 1,149, producing a simpler and more organized community structure, which facilitates easier and more efficient analysis

Keywords: influencer, graph, coloring, Louvain, Leiden, optimization, centrality, community, Garuda Indonesia

UDC 658

DOI: 10.15587/1729-4061.2025.315180

AN IMPROVEMENT OF THE LEIDEN ALGORITHM FOR INFLUENCER DETECTION

Handrizal Handrizal
Corresponding author
Student Doctoral Program in Computer Science*
E-mail: handrizal@usu.ac.id

Poltak Sihombing
Professor of Computer Science*

Erna Budhiarti Nababan
Associate Professor of Information Technology
Department of Information Technology**

Mohammad Andri Budiman
Associate Professor of Computer Science*

*Department of Computer Science**
**Universitas Sumatera Utara
Dr. T. Mansur str., 9, North Sumatera, Indonesia, 20155

Received 04.11.2024
Received in revised form 23.12.2024
Accepted date 08.01.2025
Published date 27.02.2025

How to Cite: *Handrizal, H., Sihombing, P., Nababan, E. B., Budiman, M. A. (2025). An improvement of the leiden algorithm for influencer detection. Eastern-European Journal of Enterprise Technologies, 1 (2 (133)), 33–42. https://doi.org/10.15587/1729-4061.2025.315180*

1. Introduction

The role of influencers, especially on social media platforms, has grown significantly, creating a demand for more sophisticated influencer detection methods [1–3]. These methods are a crucial component of community detection [4, 5], which is essential for identifying key individuals who shape opinions and behaviors within a network [6]. In the business world, identifying the right customers and responding to their needs is vital for maintaining competitiveness. As the business landscape evolves, companies are increasingly adopting digital marketing strategies to stay ahead of the competition [7, 8]. The study of social networks has gained significant traction due to their widespread applications across various fields, including marketing, social media, and online communities. Accurately detecting influencers is crucial for effective targeted marketing, information diffusion analysis, and community management, making it a key aspect of social network analysis.

Digital marketing promotes and disseminates information and searches for markets through digital media by utilizing various means such as social media [9, 10]. Digital marketing makes it easier for business people to unite and satisfy the desires of potential consumers. When viewed from the perspective of potential consumers [11], digital marketing can provide product information simply by exploring cyberspace, making it easier and faster to search for information [12, 13]. To increase the number of consumers, business actors need to detect people or groups who have the potential to help market their business, people or groups who can increase the number of consumers are called influencers [14].

In the existing digital marketing applications that can help solve the problems faced by business people in promoting their products, especially on social media Twitter (X) [15], various features can be used by business people to help promote their products, one of which is the follower grouping feature.

The follower grouping features only groups based on accounts that follow the business person's account without displaying information in the form of topics discussed by the group and without telling business people how large the group is formed and who the most influential figures are in the group (influencers). This can be challenging for new business owners using social media for promotion. Therefore, a way is needed to detect influencers based on certain topics or keywords using the Social Network Analysis (SNA) method on social media Twitter (X), which is then represented in graph form. The process of crawling data from Twitter (X) using Tweet Harvest with the keyword "GarudaIndonesia" [16, 17].

Existing influencer detection methods often face challenges related to the scalability and accuracy of the underlying community detection algorithms. The rapid growth of social media platforms and the increasing complexity of online networks necessitate the development of robust and efficient methods for analyzing these intricate structures. Identifying influential nodes, who play a pivotal role in shaping opinions, behaviors, and trends, has become paramount in social network analysis.

The Leiden algorithm, a significant advancement over the widely used Louvain method, has demonstrated exceptional performance in identifying communities within large-scale networks. By optimizing modularity more effectively, the Leiden algorithm provides a more accurate and refined network partitioning. This has garnered considerable attention within the community detection research field.

While the Leiden algorithm excels at community detection, it may not fully capture the nuances of influencer identification within these communities. Community detection primarily focuses on identifying groups of densely connected nodes. However, pinpointing influential individuals within these communities requires a deeper understanding of their impact and influence within the network. Therefore, studies that are devoted to the accuracy and effectiveness of influencer detection within complex social networks are of scientific relevance.

2. Literature review and problem statement

The study in [18] introduced the Louvain algorithm for community detection. The Louvain algorithm is one of the most popular and efficient methods for identifying communities in complex networks. In this context, a community refers to a group of nodes within a network that are more densely connected to each other than to nodes outside the group. However, the algorithm has a notable limitation regarding modularity. It struggles to accurately detect small communities within large networks. This limitation arises because the optimization of global modularity tends to favor larger structures, often leading to the merging of smaller communities into larger ones.

The Leiden algorithm, introduced in [19], has been shown to produce guaranteed connected communities. Furthermore, it demonstrates that iterative application converges to a partition where all subsets of all communities are optimally assigned locally. By utilizing the fast local move approach, the Leiden algorithm operates faster than the Louvain algorithm. While the Leiden algorithm addresses several limitations of the Louvain method, such as enhancing community connectivity, it may still encounter challenges when handling networks with highly heterogeneous edge weights. The presence of significantly varying edge weights can complicate the algorithm's ability to accurately identify and delineate distinct communities.

The paper [20] states that the Louvain algorithm aims to maximize the modularity value of a network partition. Modularity is a metric that measures how well a partition (the division of a network into communities) represents the actual community structure within the network. However, similar to the standard Louvain algorithm, the approach presented in this paper may encounter limitations in detecting smaller communities. This is because modularity optimization often favors the formation of larger communities, potentially merging smaller, less prominent communities and overlooking nuanced community structures.

The paper [21] states that the Louvain algorithm works by locally optimizing modularity for each node. Each node is evaluated to determine whether moving it to a different community improves the overall modularity value. However, this local optimization can cause the algorithm to become stuck in a local minimum, where further partitioning does not significantly increase modularity, even if a better global partition exists. While the proposed algorithm in this study addresses some limitations in community detection, modularity-based methods, including Louvain, inherently struggle to accurately identify small communities within large networks. This is because modularity optimization often favors merging smaller communities into larger ones, potentially overlooking more nuanced community structures.

The paper [22] evaluates the performance of the Leiden algorithm on several benchmarks and real-world networks. However, the clustering results produced by the Leiden algorithm can be challenging to interpret, particularly in the context of large and complex networks. While the paper makes a significant contribution by defining commuter zones using the Louvain algorithm, it is important to recognize the inherent limitations of modularity-based community detection methods. One key limitation is their tendency to favor larger communities, which can lead to the merging of smaller but meaningful commuter zones. This, in turn, may result in an underestimation of the diversity and complexity of commuting patterns within a region.

In the paper [23], it is stated that a new approach to graph clustering, developed by Tel Aviv University (TAU), efficiently explores the solution space using a genetic algorithm. In this research, TAU is compared with synthetic and real datasets, demonstrating its superiority over previous methods in terms of both the modularity of the computed solutions and its similarity to ground-truth partitions, where such partitions exist. However, TAU does not address the inherent weakness of genetic algorithms, which typically have high computation times, particularly when applied to big data.

In the paper [24], LouvainNE is introduced as a novel hierarchical clustering approach for network embedding. LouvainNE leverages the Louvain algorithm, a fast and accurate community detection method, to construct a hierarchy of progressively smaller subgraphs. By recursively applying the Louvain algorithm, this approach generates node representations at different levels of the hierarchy. These hierarchical representations are then combined to learn the final node embedding. However, the paper primarily focuses on network reconstruction and node classification, and the effectiveness of this method for other tasks, such as community detection, remains unexplored.

In the paper [25], a fast community detection algorithm based on local balanced label diffusion (LBLD) is proposed. The LBLD algorithm starts by assigning an importance score to each node using a novel local similarity measure. However, a comprehensive comparison with other prominent community detection algorithms is lacking, which hinders a thorough assessment of LBLD's relative strengths and weaknesses.

In the paper [26], the clique-based Louvain algorithm (CBLA) is introduced, which can classify non-classified nodes (NCNs) obtained after finding cliques in one of the communities by applying the Louvain algorithm. The Louvain algorithm is used to classify non-overlapping communities, but with the help of cliques, it can also detect overlapping nodes. While the algorithm is claimed to exhibit good performance, the paper lacks an in-depth analysis of CBLA's scalability when applied to very large datasets. Furthermore, it is important to note that, in general, clique-based algorithms often have high time complexity and may require further optimization for efficient execution.

In the paper [27], a novel Louvain-based algorithm called "NI-Louvain" is introduced, which incorporates the influence of individual nodes within a community. This enhancement allows the algorithm to not only detect communities but also identify the most influential nodes within them. The algorithm operates in three stages: First, the input graph is processed to reduce its density by calculating cliques. Next, Louvain's multilevel algorithm is applied to the clique graph. While the algorithm considers the influence of each node, it does not clearly explain how this influence is quantified or its impact on the community formation process.

In the paper [21], the authors introduced an enhancement to the Louvain algorithm, known as the Fast Louvain algorithm. This version improves the iterative process by shifting from a cyclic to a dynamic approach, which accelerates convergence and optimizes the network's local tree structure. The network is progressively partitioned and refined using a tree-based structure, leading to better community aggregation and improved community detection results. Experimental tests on various datasets show that the Fast Louvain algorithm outperforms the traditional Louvain algorithm in both partition quality and operational efficiency. However, despite its design to improve efficiency, the algorithm still faces challenges with computational complexity, particularly in large-scale networks.

In the paper [28], a community detection algorithm was analyzed, comparing the Louvain and Leiden algorithms. The results show that the Louvain algorithm suffers from poor connectivity within communities and experiences disconnections during iterative runs. In contrast, the Leiden algorithm is regarded as the newest and fastest option compared to Louvain. However, this paper primarily emphasizes the analytical insights and advantages of both algorithms, without providing in-depth empirical analysis or case studies that demonstrate their practical performance across different types of networks.

In the paper [29], graph coloring was applied to the Louvain algorithm, yielding improved results in detecting large network communities, as evidenced by higher modularity values and reduced processing time. Through a series of rigorous tests across diverse scenarios, the Louvain Coloring algorithm demonstrated superior performance in both effectiveness and efficiency when compared to the traditional Louvain algorithm. However, the algorithm faces challenges in detecting very small or very large communities. Furthermore, modularity, while commonly used as an evaluation metric, has its limitations; high modularity does not always correlate with high-quality communities, particularly in datasets prone to noise or outliers.

All this allows us to assert that it is expedient to conduct a study on overcoming these challenges based on a new approach that builds on existing methods.

3. The aim and objectives of the study

The aim of the study is to increase the modularity value of the Leiden algorithm in detecting communities using the graph coloring method. This will help speed up the process because assigning colors to nodes is akin to indexing a relational database.

To achieve this aim, the following objectives are accomplished:

– to deepen and understand the concept of the Leiden coloring algorithm and its implementation in influencer detection;

– to evaluate the performance of the Leiden coloring algorithm approach in obtaining better modularity values;

– to evaluate the performance of the Leiden coloring algorithm approach in obtaining better processing time values;

– to evaluate the performance of the Leiden coloring algorithm approach in obtaining a small number of communities.

4. Materials and methods

The object of the study is the community detection architecture model and how to optimize the architecture. The subject of the study is to determine the best model for the community detection architecture and how to optimize the architecture with graph coloring. The main hypothesis of this study is that graph coloring can optimize the community detection architecture, especially in influencer detection to improve the community detection model. The problem in this study is the difficulty business actors face in detecting influencers, which causes inefficiency in their business. This study uses laptop hardware with the specifications of a 16-inch Acer Aspire A314 laptop, with 4 GB of memory, an Intel(R) Celeron(R) N4120 CPU @ 1.10GHz processor, and Windows 11 operating system software, Tweet Harvest for Crawling dataset, Google Colab with Python 3.9.13 programming language. The stages of Leiden coloring research are as follows:

– business understanding. During this stage, information is gathered regarding potential influencers, encompassing indicators, phenomena, and factual data;

– Twitter data collection. Data collection was conducted by employing the tweet-harvest service to retrieve Twitter data of specified topics or keywords within the timeframe of January 1, 2020, to October 16, 2024. The dataset successfully obtained was 22623;

– network construction. This stage of dataset processing is carried out by selecting data based on needs, then continuing with cleaning the dataset so that the dataset to be used is clean from unnecessary data, and also changing the data format to graph format so that the data can be used for the next stage;

– community detection. In this stage, the dataset that has passed the previous stage will undergo the process of influencer detection using the Social Network Analysis method. The methods used are the Leiden algorithm and the Leiden coloring algorithm;

– analysis results. At this stage, analysis is carried out, concluding and evaluating the results of influencers detected in the previous stage. The information produced can be in the form of graph visualization and the number of communities formed along with who the influencers are in the community;

– evaluation. This stage involves an evaluation of the algorithm's performance, considering factors such as modularity, processing time, and the number of communities.

5. Results of an improvement of the Leiden algorithm for influencer detection

5. 1. Results of deepening and understanding the concept of the Leiden coloring algorithm and its implementation in influencer detection

5. 1. 1. Results of influencer detection with a dataset of 1,000 rows

This section presents the results of our improved Leiden algorithm for influencer detection. These results encompass a deeper understanding of the Leiden Coloring Algorithm and its implementation, an evaluation of its performance in terms of modularity, processing time, and the number of identified communities.

In the paper [30], one of the shortcomings of the Leiden algorithm is its lack of guarantee in processing large networks, which results in relatively low processing times. To modify the Leiden algorithm using graph coloring, hereinafter referred to as Leiden coloring, it is necessary to consider the coloring process when optimizing the modularity of Q. These two methods can be combined by adding coloring constraints during community updates.

The steps of the Leiden coloring algorithm are as follows:

– phase 1: community detection.

Employing the Leiden algorithm, as detailed in (1), to identify communities within the graph:

– phase 2: community coloring.

Following community detection, each identified community is assigned a distinct color.

The final outcome is a network where each node is assigned a community and color label.

Influencer detection is performed by calculating degree centrality, which is determined by the following formula:

$$C_D(v)=\frac{\deg(v)}{N-1},$$

where $C_D(v)$ – degree centrality of node v;

$\deg(v)$ – the degree of node v, representing the number of edges connected to v;

N – the total number of nodes in the graph;

N–1 – the maximum possible degree a node can have in a network.

The first test scenario was carried out with a dataset of 1,000-rows randomly. In this scenario, 5 usernames were obtained as top influencers, as seen in Table 1.

Table 1

Results of influencer detection with a dataset of 1,000 rows

No.	Leiden coloring algorithm
1	IndonesiaGaruda
2	GarudaCares
3	wandiseptian11
4	PinterPoin
5	idbcpr

In Table 1, it can be seen that for the 1,000-row dataset, the first rank as an influencer is the username "IndonesiaGaruda", meanwhile the fifth ranking is the username "idbcpr".

5. 1. 2. Results of influencer detection with a dataset of 2,000 rows

The second test scenario was carried out with a dataset of 2,000-rows randomly. In this scenario, 5 usernames were obtained as top influencers, as seen in Table 2.

Table 2

Results of influencer detection with a dataset of 2,000 rows

No.	Leiden coloring algorithm
1	IndonesiaGaruda
2	GarudaCares
3	antar_news
4	wandiseptian11
5	kompascom

In Table 2, it can be seen that for the 2,000-row dataset, the first rank as an influencer is the username "IndonesiaGaruda", meanwhile the fifth rank is the username "kompascom".

5. 1. 3. Results of influencer detection with a dataset of 5,000 rows

The third test scenario was carried out with a dataset of 5,000-rows randomly. In this scenario, 5 usernames were obtained as top influencers, as seen in Table 3.

Table 3

Results of influencer detection with a dataset of 5,000 rows

No.	Leiden coloring algorithm
1	IndonesiaGaruda
2	disemuacom
3	GarudaCares
4	astuceclover
5	TiketPesawatPro

In Table 3, it can be seen that for the 5,000-row dataset, the first rank as an influencer is the username "IndonesiaGaruda", meanwhile the fifth rank is the username "TiketPesawatPro".

5. 1. 4. Results comparison of influencer detection

This section will show the results of a comparison of the proposed Leiden coloring algorithm with the Louvain coloring algorithm. The comparisons made are influencer detection, modularity value, processing time, and number of communities. In this study, three scenarios of the dataset testing process were carried out. The first test scenario was carried out with a dataset of 1,000 rows randomly. The second test scenario was carried out with a dataset of 2,000 rows randomly. The third test scenario was carried out with a dataset of 5,000 rows randomly.

5. 1. 5. Results comparison of influencer detection with a dataset of 1,000 rows

The first test scenario was carried out with a dataset of 1,000-rows randomly. In this scenario, 5 usernames were obtained as top influencers, as seen in Table 4.

Table 4

Comparison of influencer detection with a dataset of 1,000 rows

No.	Louvain coloring algorithm	Leiden coloring algorithm
1	IndonesiaGaruda	IndonesiaGaruda
2	GarudaCares	GarudaCares
3	astuceclover	wandiseptian11
4	TiketPesawatPro	PinterPoin
5	disemuacom	idbcpr

In Table 4, it can be seen that the Louvain coloring algorithm and the Leiden coloring algorithm on the 1,000-row dataset produce the same influencer detection for the first rank, which is the username "IndonesiaGaruda", and the second rank is the username "GarudaCares". Meanwhile, the ranks up to the third rank produce different influencer detections.

5. 1. 6. Results comparison of influencer detection with a dataset of 2,000 rows

The second test scenario was carried out with a dataset of 2,000 rows randomly. In this scenario, 5 usernames were obtained as top influencers, as seen in Table 5.

Table 5

Comparison of influencer detection with a dataset of 2,000-rows

No.	Louvain coloring algorithm	Leiden coloring algorithm
1	IndonesiaGaruda	IndonesiaGaruda
2	GarudaCares	GarudaCares
3	astuceclover	antar_news
4	TiketPesawatPro	wandiseptian11
5	disemuacom	kompascom

In Table 5, it can be seen that the Louvain coloring algorithm and the Leiden coloring algorithm on the 2,000-row dataset produce the same influencer detection for the first rank, which is the username "IndonesiaGaruda", and the second rank is the username "GarudaCares". Meanwhile the third to tenth ranks produce different influencer detections.

5. 1. 7. Results comparison of influencer detection with a dataset of 5,000 rows

The third test scenario was carried out with a dataset of 5,000-rows randomly. In this scenario, 5 usernames were obtained as top influencers, as seen in Table 6.

Table 6

Comparison of influencer detection with a dataset of 5,000 rows

No.	Louvain coloring algorithm	Leiden coloring algorithm
1	IndonesiaGaruda	IndonesiaGaruda
2	GarudaCares	disemuacom
3	TiketPesawatPro	GarudaCares
4	disemuacom	astuceclover
5	astuceclover	TiketPesawatPro

In Table 6, it can be seen that the Louvain coloring algorithm and the Leiden coloring algorithm on the 5,000-row dataset produce the same influencer detection for the first rank, namely the username "IndonesiaGaruda". Meanwhile, the second to fifth ranks produce different influencer detections.

Based on Tables 4–6, it can be seen that the Louvain coloring algorithm and the Leiden coloring algorithm produce the same influencer detection for the first rank, namely the username "IndonesiaGaruda". Meanwhile, the second to fifth ranks produce different influencer detections.

5. 2. Results of evaluating the performance of the Leiden coloring algorithm approach in obtaining better modularity values

By incorporating coloring constraints into the modularity function Q, we can improve results. Let $g(i, j)=1$ if vertices i and j have different colors, and 0 otherwise. The modularity function Q with graph coloring becomes:

$$Q=\frac{1}{2m}\Sigma A_{i,j}-\frac{k_i k_j}{2m}\delta(c_i,c_j)*g(j,i), \quad (1)$$

where $A_{i,j}$ represents the edge weight between nodes i and j;

k_i and k_j are the sum of the weights of the edges attached to nodes i and j;

m is the sum of all edge weights in the graph;

c_i and c_j are the node communities;

δ is the Kronecker delta function ((c_i, c_j)=1 if $c_i=c_j$, 0 otherwise);

$g=(i, j)$, which is 1 if nodes i and j have different colors and 0 if they have the same color.

In Table 7, it can be seen that for the 1,000-row dataset, the modularity value is 0.9396, for the 2,000-row dataset, the modularity value is 0.9367 and for the 5,000-row dataset, it is 0.9381.

Table 7

Results of the modularity matrix

No.	Dataset	Modularity of Leiden coloring
1	1,000 rows	0.9396
2	2,000 rows	0.9367
3	5,000 rows	0.9381
Average		0.9381

The matrix above can also be seen in graphical form, as shown in Fig. 1.

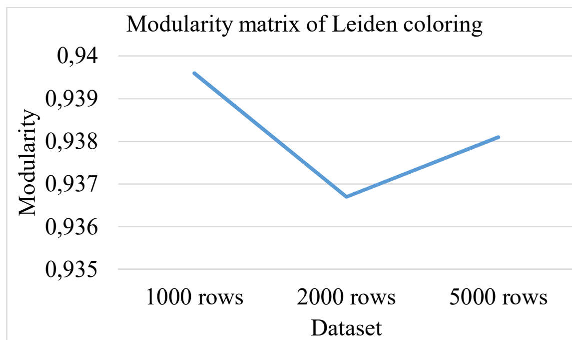

Fig. 1. Results of the modularity matrix

In the graph in Fig. 1, it can be seen that the modularity value for the 1,000-row dataset is 0.9396, then decreases for the 2,000-row dataset to 0.9381 and increases again for the 5,000-row dataset to 0.9381.

The modularity value of the Leiden coloring algorithm is better than that of the Louvain coloring algorithm. Of the 3 test scenarios carried out, the Leiden coloring algorithm excels in all scenarios, as seen in Table 8.

Table 8

Comparison of the modularity matrix

No.	Dataset	Modularity	
		Louvain coloring	Leiden coloring
1	1,000 rows	0.9114	0.9396
2	2,000 rows	0.9259	0.9367
3	5,000 rows	0.9050	0.9381
Average		0.9141	0.9381

The modularity value of the Leiden coloring algorithm is at the lowest value of 0.9367 and the highest of 0.9396, with an average of 0.9381. Meanwhile, the Louvain coloring algorithm has the lowest value of 0.9050 and the highest of 0.9252 with an average of 0.9141.

The comparison graph of the modularity matrix is shown in Fig. 2. The graph illustrates an increase of 0.0240 in the modularity value when using the Leiden coloring algorithm.

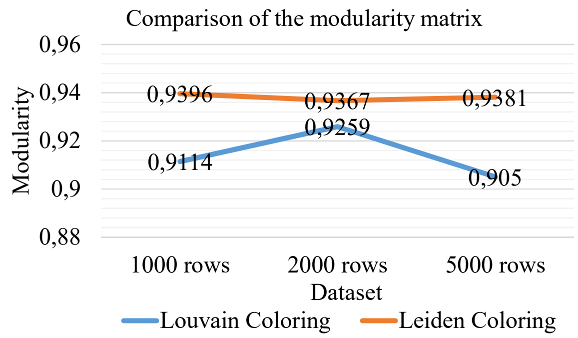

Fig. 2. Comparison of the modularity matrix

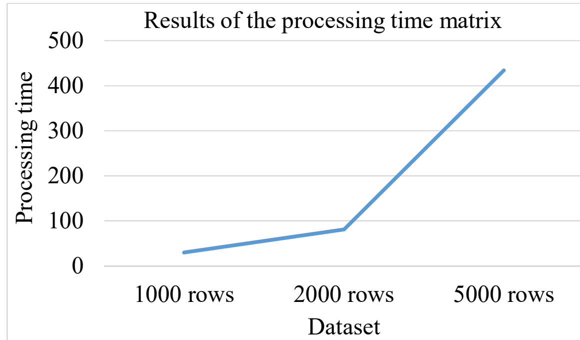

Fig. 3. Results of the processing time matrix

5. 3. Results of evaluating the performance of the Leiden coloring algorithm approach in obtaining better processing time values

The processing time of the Leiden coloring algorithm can be calculated by considering the time complexity of its core operations, which include the modularity optimization phase and the graph coloring constraints. Here's a general formula expressed in English:

Processing Time (T)=*T_partition*+*T_coloring*+*T_update*,

where:

1. *T_partition*: The time taken to partition the graph into communities.

This depends on the size of the graph (number of nodes n and edges m) and is generally $O(m)$ for modularity optimization.

2. T_*coloring*: The time required to apply graph coloring constraints during the community updates.

This depends on the graph structure and is influenced by the coloring process, which can be approximated as $O(n+e)$, where e is the number of conflicting edges that require recoloring.

3. *T_update*: The time taken to update modularity values after applying coloring constraints.

Typically $O(n)$ per iteration.

In Table 9, it can be seen that for a dataset of 1,000 rows, the processing time value is 29.5491 seconds, for a dataset of 2,000 rows, the processing time value is 80.7600 seconds, and for a dataset of 5,000 rows, it is 434.1838 seconds.

Table 9

Result of the processing time matrix

No.	Dataset	Processing time of Leiden coloring (second)
1	1,000 rows	29.5493
2	2,000 rows	80.7600
3	5,000 rows	434.1838
Average		181.4977

The matrix above can also be seen in graphical form, as shown in Fig. 3.

In the graph in Fig. 3, it can be seen that the processing time value for the 1,000-row dataset is 29.5491 seconds, then increases for the 2,000-row dataset, namely 80.7600 seconds and continues to increase for the 5,000-row dataset, namely 434.1838 seconds.

The processing time value of the Leiden coloring algorithm is better than that of the Louvain coloring algorithm. Of the 3 test scenarios carried out, the Leiden coloring algorithm excels in all scenarios, as seen in Table 10.

Table 10

Comparison of the processing time matrix

No.	Dataset	Processing time (second)	
		Louvain coloring	Leiden coloring
1	1,000 rows	41.85	29.5493
2	2,000 rows	96.32	80.7600
3	5,000 rows	450.86	434.1838
Average		196.3433	181.4977

The processing time value of the Leiden coloring algorithm is at the lowest value of 29.5493 seconds and the highest of 434.1838 seconds with an average of 181.4977 seconds. Meanwhile, the Louvain coloring algorithm has the lowest value of 41.85 seconds and the highest of 450.86 seconds with an average of 196.3433 seconds.

The comparison graph of the processing time matrix is shown in Fig. 4. The graph indicates a reduction of 14.85 seconds in the processing time achieved by the Leiden coloring algorithm.

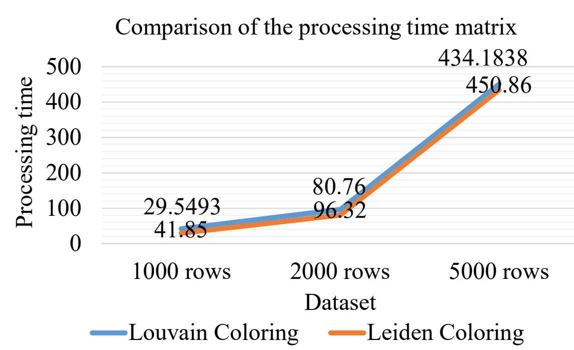

Fig. 4. Comparison of the processing time matrix

5. 4. Results of evaluating the performance of the Leiden coloring algorithm approach in obtaining a small number of communities

In the Leiden Coloring algorithm, the formula to calculate the number of communities can be described as:

Number of Communities (C)=Sum of all unique community labels identified after graph partitioning.

Here's how it works in detail:

1. Partitioning the graph: the Leiden coloring algorithm partitions the graph into communities based on modularity optimization and graph coloring constraints.

2. Community identification: each node in the graph is assigned a community label.

3. Counting unique labels: the total number of unique community labels after partitioning represents the number of communities.

Mathematically:

$$C = \left|\{c_i : c_i :\in \text{Community Labels}\}\right|,$$

where c_i represents the community label of node i;

$\{c_i\}$ is the set of all community labels;

$|\cdot|$ denotes the cardinality (size) of the set, which gives the total number of unique communities.

This formula ensures that each distinct group or cluster formed by the algorithm is counted as one community.

In Table 11, it can be seen that for the 1000-rows dataset, the number of communities is 505, for the 2,000-row dataset, the number of communities is 934 and for the 5,000-row dataset, it is 1,969.

Table 11

Results of the number of communities matrix

No.	Dataset	Number of communities of Leiden coloring
1	1,000 rows	505
2	2,000 rows	934
3	5,000 rows	1,969
Average		1,136

The matrix above can also be seen in graphical form, as shown in Fig. 5.

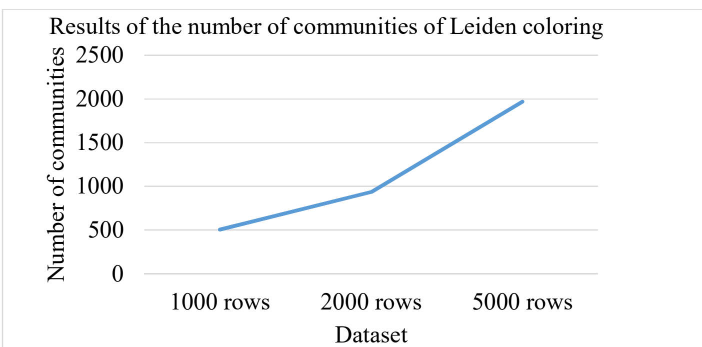

Fig. 5. Results of the number of communities matrix

In the graph in Fig. 5, it can be seen that the number of communities for the 1,000-row dataset is 505, then increases for the 2,000-row dataset to 934 and continues to increase for the 5,000-row dataset to 1,969.

The number of communities of the Leiden coloring algorithm is less than that of the Louvain coloring algorithm. Of the 3 test scenarios carried out, the Leiden coloring algorithm excels in all scenarios, as seen in Table 12.

Table 12

Comparison of the number of communities matrix

No.	Dataset	Number of communities	
		Louvain coloring	Leiden coloring
1	1,000 rows	936	505
2	2,000 rows	1,800	934
3	5,000 rows	4,119	1,969
Average		2,285	1,136

The sum value of the Leiden coloring algorithm is at the lowest value of 505 and the highest in 1,969 with an average of 1,136. Meanwhile, the Louvain coloring algorithm has the lowest value of 936 and the highest of 4,119 with an average of 2,285.

The comparison graph of the number of communities matrix is shown in Fig. 6. The graph indicates a reduction in the number of communities of the Leiden coloring algorithm by 1,149.

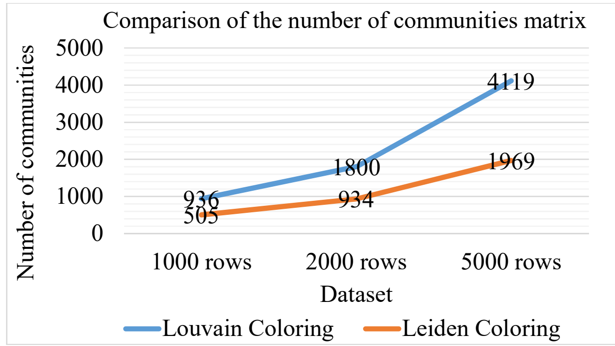

Fig. 6. Comparison of the number of communities matrix

6. Discussion of the results of an improvement of the Leiden algorithm for influencer detection

The research results on the Leiden coloring algorithm to detect influencers provide valuable insights into the approach's effectiveness. This study shows that the Leiden coloring algorithm approach consists of two phases, namely the initial phase of community detection utilizing the Leiden algorithm, followed by the assignment of unique colors to each identified community.

The test results using the Leiden coloring algorithm to detect influencers with a dataset of 1,000-rows recommend five influencers: IndonesiaGaruda, GarudaCares, astuceclover, TiketPesawatPro, and disemuacom. Meanwhile, the test results using the Louvain coloring algorithm recommend five influencers: IndonesiaGaruda, GarudaCares, Wandiseptian11, PinterPoin, and idbcpr, as shown in Table 4.

The test results using the Leiden coloring algorithm to detect influencers with a dataset of 2,000-rows recommend five influencers: IndonesiaGaruda, GarudaCares, astuceclover, TiketPesawatPro, and disemuacom. Meanwhile, the test results using the Louvain coloring algorithm recommend five influencers: IndonesiaGaruda, GarudaCares, antar_news, wandiseptian11, and kompascom, as shown in Table 5.

The test results using the Leiden coloring algorithm to detect influencers with a dataset of 5,000-rows recommend five influencers: IndonesiaGaruda, GarudaCares, TiketPesawatPro, disemuacom, and astuceclover. Meanwhile, the test results using the Louvain coloring algorithm recommend five influencers: IndonesiaGaruda, disemuacom, GarudaCares, astuceclover, and TiketPesawatPro, as shown in Table 6.

Based on Fig. 1, the modularity value of the Leiden coloring algorithm for the 1000-row dataset is 0.9396. This value decreases slightly to 0.9381 for the 2,000-row dataset but increases again to 0.9396 for the 5,000-row dataset.

According to Fig. 2, which compares the modularity results of the Leiden coloring algorithm, the modularity value ranges from a minimum of 0.9367 to a maximum of 0.9396, with an average of 0.9381. In contrast, the Louvain coloring algorithm achieves a modularity value ranging from 0.9050 to 0.9252, with an average of 0.9141. Thus, the Leiden coloring algorithm shows an improvement in modularity value by 0.0240, as shown in Table 8.

Fig. 3 shows that the processing time for the Leiden coloring algorithm is 29.5491 seconds for the 1,000-row dataset. This increases to 80.7600 seconds for the 2,000-row dataset and further rises to 434.1838 seconds for the 5,000-row dataset.

Fig. 4 presents the processing time measurements for the Leiden coloring algorithm. The processing time ranges from a minimum of 29.5493 seconds to a maximum of 434.1838 seconds, with an average of 181.4977 seconds. Meanwhile, the Louvain coloring algorithm exhibits processing times ranging from 41.85 seconds to 450.86 seconds, with an average of 196.3433 seconds. This indicates a reduction in processing time for the Leiden coloring algorithm by 14.85 seconds, as detailed in Table 10.

Fig. 5 indicates that the number of communities generated by the Leiden coloring algorithm is 505 for the 1,000-row dataset. This number increases to 934 for the 2,000-row dataset and further rises to 1,969 for the 5,000-row dataset.

As shown in Fig. 6, the number of communities generated by the Leiden coloring algorithm ranges from a minimum of 505 to a maximum of 1,969, with an average of 1,136. In contrast, the Louvain coloring algorithm produces a minimum of 936 communities and a maximum of 4,119, with an average of 2,285. This represents a reduction in the number of communities generated by the Leiden coloring algorithm by 1,149, as shown in Table 12.

Overall, the results obtained from the Leiden coloring algorithm for detecting influencers demonstrate its effectiveness in recommending the top 5 influencers with higher modularity values. High modularity indicates the algorithm's ability to achieve a more optimal network partitioning. The results also show a lower processing time, indicating faster performance, which is particularly important for applications requiring quick results, especially in big data analysis. Additionally, the Leiden coloring algorithm produces a smaller number of communities, leading to a simpler and more organized community structure, which facilitates easier and more efficient analysis. The study's results suggest that influencer recommendations can be made more accurately. These recommendations are highly valuable for business owners in making informed decisions about influencer selection. However, additional details and experimental evidence are needed to assess the performance of this approach on larger datasets. Further studies and comparisons with existing research could provide more comprehensive insights into the potential advantages and limitations of the Leiden coloring algorithm for detecting influencers across various types of data.

7. Conclusions

1. Influencer detection using the Leiden coloring algorithm using a dataset from Twitter (X) with the keyword "GarudaIndonesia" provides recommendations for 5 accounts selected as influencers.

2. The modularity value of the Leiden coloring algorithm in influencer detection is better than that of the Louvain coloring algorithm. Of the 3 test scenarios carried out, the Leiden coloring algorithm excels in all scenarios. The modularity value of the Leiden coloring algorithm is at the lowest value of 0.9367 and the highest of 0.9396, with an average of 0.9381. Meanwhile, the Louvain coloring algorithm has the lowest value of 0.9050 and the highest of 0.9252 with an average of 0.9141. Thus, there is an increase in the modularity value of the Leiden coloring algorithm by 0.0240.

3. The processing time value of the Leiden coloring algorithm in influencer detection is better than that of the Louvain coloring algorithm. Of the 3 test scenarios carried out, the Leiden coloring algorithm excels in all scenarios. The processing time value of the Leiden coloring algorithm is at the lowest value of 29.5493 seconds and the highest is 434.1838 seconds with an average of 181.4977 seconds. Meanwhile, the Louvain coloring algorithm has the lowest value of 41.85 seconds and the highest is 450.86 seconds with an average of 196.3433 seconds. Thus, there is a reduction in the processing time value of the Leiden coloring algorithm by 14.85 seconds.

4. The number of communities of the Leiden coloring algorithm is less than that of the Louvain coloring algorithm. Of the 3 test scenarios carried out, the Leiden coloring algorithm excels in all scenarios. The sum value of the Leiden coloring algorithm is at the lowest value of 505 and the highest in 1,969 with an average of 1,136. Meanwhile, the Louvain coloring algorithm has the lowest value of 936 and the highest of 4,119 with an average of 2,285. Thus, there is a reduction in the sum value of the Leiden coloring algorithm community by 1,149.

Conflict of interest

The authors declare that they have no conflict of interest about this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

The study was conducted without financial support.

Data availability

Data will be made available on reasonable request.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Chen, C.-W., Nguyen, D. T. T., Chih, M., Chen, P.-Y. (2024). Fostering YouTube followers' stickiness through social contagion: The role of digital influencer' characteristics and followers' compensation psychology. Computers in Human Behavior, 158, 108304. https://doi.org/10.1016/j.chb.2024.108304

2. Laor, T. (2024). Do micro-celebrities preserve social roles? Differences between secular and religious female Instagram lifestyle influencers. Technology in Society, 78, 102642. https://doi.org/10.1016/j.techsoc.2024.102642
3. Kurniasari, F., Prihanto, J. N., Andre, N. (2023). Identifying determinant factors influencing user's behavioral intention to use Traveloka Paylater. Eastern-European Journal of Enterprise Technologies, 2 (13 (122)), 52–61. https://doi.org/10.15587/1729-4061.2023.275735
4. Deng, F., Tuo, M., Chen, S., Zhang, Z. (2024). Born for marketing? The effects of virtual versus human influencers on brand endorsement effectiveness: The role of advertising recognition. Journal of Retailing and Consumer Services, 80, 103904. https://doi.org/10.1016/j.jretconser.2024.103904
5. Wang, Z.-Y., Zhang, C.-P., Othman Yahya, R. (2024). High-quality community detection in complex networks based on node influence analysis. Chaos, Solitons & Fractals, 182, 114849. https://doi.org/10.1016/j.chaos.2024.114849
6. Morisada, M., Miwa, Y., Dahana, W. D. (2019). Identifying valuable customer segments in online fashion markets: An implication for customer tier programs. Electronic Commerce Research and Applications, 33, 100822. https://doi.org/10.1016/j.elerap.2018.100822
7. Abdelkader, O. A. (2023). ChatGPT's influence on customer experience in digital marketing: Investigating the moderating roles. Heliyon, 9 (8), e18770. https://doi.org/10.1016/j.heliyon.2023.e18770
8. Tataryntseva, Y., Pushkar, O., Druhova, O., Osypova, S., Makarenko, A., Mordovtsev, O. (2022). Economic evaluation of digital marketing management at the enterprise. Eastern-European Journal of Enterprise Technologies, 2 (13 (116)), 24–30. https://doi.org/10.15587/1729-4061.2022.254485
9. Armutcu, B., Tan, A., Amponsah, M., Parida, S., Ramkissoon, H. (2023). Tourist behaviour: The role of digital marketing and social media. Acta Psychologica, 240, 104025. https://doi.org/10.1016/j.actpsy.2023.104025
10. Vasylyshyna, L., Yahelska, K., Aldankova, H., Liashuk, K. (2024). Development of marketing research technologies as the basis of a socially responsible marketing strategy. Eastern-European Journal of Enterprise Technologies, 5 (13 (131)), 76–85. https://doi.org/10.15587/1729-4061.2024.312227
11. Novytska, I., Chychkalo-Kondratska, I., Chyzhevska, M., Sydorenko-Melnyk, H., Tytarenko, L. (2021). Digital Marketing in the System of Promotion of Organic Products. WSEAS TRANSACTIONS ON BUSINESS AND ECONOMICS, 18, 524–530. https://doi.org/10.37394/23207.2021.18.53
12. Vrontis, D., Makrides, A., Christofi, M., Thrassou, A. (2021). Social media influencer marketing: A systematic review, integrative framework and future research agenda. International Journal of Consumer Studies, 45 (4), 617–644. https://doi.org/10.1111/ijcs.12647
13. Peter, M. K., Dalla Vecchia, M. (2020). The Digital Marketing Toolkit: A Literature Review for the Identification of Digital Marketing Channels and Platforms. New Trends in Business Information Systems and Technology, 251–265. https://doi.org/10.1007/978-3-030-48332-6_17
14. Veleva, S. S., Tsvetanova, A. I. (2020). Characteristics of the digital marketing advantages and disadvantages. IOP Conference Series: Materials Science and Engineering, 940 (1), 012065. https://doi.org/10.1088/1757-899x/940/1/012065
15. Khanom, M. T. (2023). Using social media marketing in the digital era: A necessity or a choice. International Journal of Research in Business and Social Science (2147- 4478), 12 (3), 88–98. https://doi.org/10.20525/ijrbs.v12i3.2507
16. Cai, Y., Wang, H., Ye, H., Jin, Y., Gao, W. (2023). Depression detection on online social network with multivariate time series feature of user depressive symptoms. Expert Systems with Applications, 217, 119538. https://doi.org/10.1016/j.eswa.2023.119538
17. Abdelhamid, S., Aly, M., Katz, A. (2020). Harvesting tweets for a better understanding of Engineering Students' First-Year Experiences. 2020 First-Year Engineering Experience Proceedings. https://doi.org/10.18260/1-2--35771
18. Blondel, V. D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008 (10), P10008. https://doi.org/10.1088/1742-5468/2008/10/p10008
19. Traag, V. A., Waltman, L., van Eck, N. J. (2019). From Louvain to Leiden: guaranteeing well-connected communities. Scientific Reports, 9 (1). https://doi.org/10.1038/s41598-019-41695-z
20. De Meo, P., Ferrara, E., Fiumara, G., Provetti, A. (2011). Generalized Louvain method for community detection in large networks. 2011 11th International Conference on Intelligent Systems Design and Applications, 88–93. https://doi.org/10.1109/isda.2011.6121636
21. Zhang, J., Fei, J., Song, X., Feng, J. (2021). An Improved Louvain Algorithm for Community Detection. Mathematical Problems in Engineering, 2021, 1–14. https://doi.org/10.1155/2021/1485592
22. Zhang, W. (2022). Improving commuting zones using the Louvain community detection algorithm. Economics Letters, 219, 110827. https://doi.org/10.1016/j.econlet.2022.110827
23. Gilad, G., Sharan, R. (2023). From Leiden to Tel-Aviv University (TAU): exploring clustering solutions via a genetic algorithm. PNAS Nexus, 2 (6). https://doi.org/10.1093/pnasnexus/pgad180
24. Bhowmick, A. K., Meneni, K., Danisch, M., Guillaume, J.-L., Mitra, B. (2020). LouvainNE. Proceedings of the 13th International Conference on Web Search and Data Mining. https://doi.org/10.1145/3336191.3371800
25. Roghani, H., Bouyer, A. (2023). A Fast Local Balanced Label Diffusion Algorithm for Community Detection in Social Networks. IEEE Transactions on Knowledge and Data Engineering, 35 (6), 5472–5484. https://doi.org/10.1109/tkde.2022.3162161

26. Gupta, S. K., Singh, Dr. D. P. (2023). CBLA: A Clique Based Louvain Algorithm for Detecting Overlapping Community. Procedia Computer Science, 218, 2201–2209. https://doi.org/10.1016/j.procs.2023.01.196
27. Singh, D., Garg, R. (2022). NI-Louvain: A novel algorithm to detect overlapping communities with influence analysis. Journal of King Saud University - Computer and Information Sciences, 34 (9), 7765–7774. https://doi.org/10.1016/j.jksuci.2021.07.006
28. Hairol Anuar, S. H., Abas, Z. A., Yunos, N. M., Mohd Zaki, N. H., Hashim, N. A., Mokhtar, M. F. et al. (2021). Comparison between Louvain and Leiden Algorithm for Network Structure: A Review. Journal of Physics: Conference Series, 2129 (1), 012028. https://doi.org/10.1088/1742-6596/2129/1/012028
29. Mardiansyah, H., Suwilo, S., Nababan, E. B., Efendi, S. (2023). Community Clustering on Fraud Transactions Applied the Louvain-Coloring Algorithm. International Journal of Electronics and Telecommunications, 593–598. https://doi.org/10.24425/ijet.2023.146512
30. Sahu, S., Kothapalli, K., Banerjee, D. S. (2024). Fast Leiden Algorithm for Community Detection in Shared Memory Setting. Proceedings of the 53rd International Conference on Parallel Processing, 11–20. https://doi.org/10.1145/3673038.3673146