The role of Louvain-coloring clustering in the detection of fraud transactions

Heru Mardiansyah[2], Saib Suwilo[1,2], Erna Budhiarti Nababan[1], Syahril Efendi[2]
[1]Department of Information Technology, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia
[2]Department of Computer Science, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia

Article Info

Article history:

Received May 12, 2023
Revised Jul 13, 2023
Accepted Jul 17, 2023

Keywords:

Cluster
Clustering
Data mining
Fraud
Graph
Louvain algorithm
Transaction

ABSTRACT

Clustering is a technique in data mining capable of grouping very large amounts of data to gain new knowledge based on unsupervised learning. Clustering is capable of grouping various types of data and fields. The process that requires this technique is in the business sector, especially banking. In the transaction business process in banking, fraud is often encountered in transactions. This raises interest in clustering data fraud in transactions. An algorithm is needed in the cluster, namely Louvain's algorithm. Louvain's algorithm is capable of clustering in large numbers, which represent them in a graph. So, the Louvain algorithm is optimized with colored graphs to facilitate research continuity in labeling. In this study, 33,491 non-fraud data were grouped, and 241 fraud transaction data were carried out. However, Louvain's algorithm shows that clustering increases the amount of data fraud of 90% by accurate.

Corresponding Author:

Saib Suwilo
Department of Information Technology, Faculty of Computer Science and Information Technology,
Universitas Sumatera Utara
Medan, Indonesia
Email: saib@usu.ac.id

1. INTRODUCTION

One of data mining techniques using unsupervised learning is clusters [1], [2]. Where unsupervised learning is learning without information so that it becomes new knowledge based on large data sources [3], [4]. Clustering itself aims to group data on the same characteristics and areas based on the required characteristics of other regions [5], [6]. Various studies are often carried out by clustering, as was done [7] do managerial clustering which has a lot of data and expertise in decision makers within the company so that it can be used as a reference in changing leadership. aside from that [8] do a categorization of behavior on bitcoin. Job operation requires a different amount of time, and it is impossible to process a series of operations on a job simultaneously because an operation must be carried out sequentially. The problem is choosing the best sequence of work activities and determining which machine can perform the task to minimize machine idle and standby time [9].

Various fields can be utilized in this clustering technique as is currently developing, namely in the business sector [10]. In the field of business and various problems as well as with big data can be solved in the field of business [11]. Various types of businesses are also a concern for the need for data mining techniques to gain new knowledge [12]. As is the case in banking which has a financial transaction process where one type of crime that often occurs is fraud in the transaction process, both in the declared type of fabrication crime [13], [14].

Fraud in transactions in banking certainly causes a lot of harm to various parties [15]. As in the incident that has been quoted from [16] perform fraud detection using a machine learning approach and produce an related-party transactions (RPTs) knowledge graph to obtain performance and verify fraud in transactions. Besides that, fraud also occurs from research [17] grouping fraudulent transactions in the data mining process with the aim of increasing the profits of the companies and obtaining optimal results so that managerial can improve the quality of the company.

With these problems, of course, to get contributions in the field of computer science, the right algorithm is chosen to detect fraud [18]. Recorded transaction processes are of course large data and can be used as new knowledge [19]. Thus, of course, an optimal algorithm is needed to detect fraudulent transactions. As done [20] creating an anti-fraud system in banking to be able to monitor fraud in transactions where the system is given knowledge with a clustering approach to data mining.

However, in business processes the Louvain algorithm is often used in clustering [18]. Louvain algorithm is a community clustering algorithm based on graphs. So that data on transactions becomes a graph that can facilitate the detection of fraud as a community [21]. Louvain algorithm has also been carried out [22] in conducting clusters for local energy market travel routes in the US where the optimization model on the Louvain algorithm is located on modularity based on a combination of graphic-intrinsic parameters so that it has an accuracy of 0.05 to 0.15. So that in this study an optimization was found in the Louvain algorithm with the image function so that decision makers can quickly get labels in the fraud transaction community to be the Louvain-Coloring algorithm approach.

2. MATERIAL AND METHOD

2.1. Dataset

The dataset in this research uses community banking transaction data where the data consists of 33,491 non-fraud data and 241 fraudulent transaction data. Because the Louvain algorithm can perform clustering to detect communities. However, one of the shortcomings of the Louvain algorithm on large datasets is its relatively low processing time. However, in terms of nodes and data proximity, it will certainly ensure whether there is no more data that is close to fraudulent.

2.2. Louvain-coloring algorithm in clustering

Louvain algorithm is an algorithm with unsupervised learning to be able to classify data fraud in banking transactions [23]. Where the Louvain algorithm has modularity optimization and community aggregation with maximum modularity based on (1) [24].

$$M = \frac{1}{2m}\sum_{i,j}(A_{ij} - P_{ij})\delta(c_{J,}c_{J}) = \frac{1}{2M}\sum_{I,J}\left(A_{IJ} - \frac{K_i K_j}{2M}\right)\vartheta\left(C_{I,}C_{J}\right) \tag{1}$$

A_{ij} is a weight value that can replace the input of neighboring matrix values with side weights connected between nodes i and $j, ki = \sum j\ A_{ij}$ which is the degree of node I, C_I which is a community attribute, δ-function δ (u, v) is 1 if u=v and 0, then. m=1 $\sum ij\ A_{ij}$ 2 if the weights are graphs. Louvain will randomly arrange all nodes in the network according to the modular optimization method [21]. Then, one by one, it removes and inserts each node into another community C until there is no appreciable increase in modularity (input parameter) as shown in (2):

$$\Delta M = \left|\frac{\sum_{in} + 2k_{i,in}}{2m} - \left(\frac{\sum_{tot} + K_i}{2m}\right)^2\right| - \left[\frac{\sum in}{2m} - \left(\frac{\sum tot}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \tag{2}$$

Louvain randomizes the existing nodes in the modular optimization method. For a definite increase, Louvain deletes and inserts each node in the community that is not the same as C so that a very visible increase does not occur. Like the (3) [25]:

$$\Delta M = \frac{K_{i,in}}{m} - \frac{2\sum totk_i}{(2m)^2} \rightarrow \Delta Mm = K_{i,in} - \frac{\sum totK_i}{2m} \tag{3}$$

While $K_{i,in}$ and $\sum tot$ must be calculated for each test community, node-specific K_i/(2m) is analyzed. In this way, the final expression is only recomputed when different nodes are considered during modulus optimization [26]. After completing the first stage, all nodes belonging to the same community are combined into one giant node. The links that connect the giant nodes are the sum of the previous links that have connected the nodes of the same different community.

2.3. General architecture

In this paper, of course, requires steps that are directed and focused on goals. So, to support the research process a general architecture was formed. As for Figure 1, the Louvain process of this paper. Based on Figure 1 on the Louvain process with the steps:

- Collect datasets
- Perform data preprocessing as needed
- Perform clustering with the Louvain algorithm with n scenarios.
- Perform optimization with Louvain-Coloring based on (4).

$$Q = \frac{1}{2m}\sum\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j) \tag{4}$$

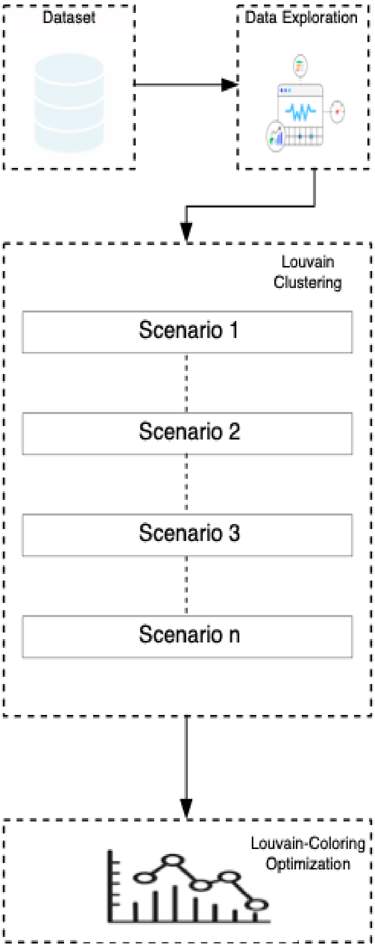

Figure 1. Louvain process

3. RESULTS AND DISCUSSION

Clustering is carried out on communities that commit fraud in transactions, especially in banking. The one-time dataset is based on public data consisting of 33,732 datasets and has been informed that 33,491 data are not fraudulent and 241 are fraudulent in transactions. But in the proof to detect how accurate the knowledge that will be achieved based on the clustering technique is of course tested with the Louvain algorithm. Where, after data exploration, it is confirmed whether transactions that are not fraudulent approach or identify fraud.

3.1. Louvain algorithm to detect community

This test uses 6 scenarios to detect communities with the Louvain algorithm, using several combinations of existing parameter values, which will be discussed in the test scenarios. The default parameter value is the new network's two-phase maxLevel (default=10), which is the level for the community hierarchy. When running, the higher the level, the larger the community, which may not be exactly what is desired. Next, maxIteration (default=10), which is the first phase, continues to repeat iteratively until the increase in modularity is negligible or the number of iterations is reached as specified. Tolerance (default=0.0001) the smaller the tolerance, the higher it is, the better, but more iterations may be required. Concurrency (default=4) which is the number of threads that can run simultaneously.

3.1.1. Scenario 1

Scenario 1 uses parameters with the default values maxLevels, maxIteration, tolerance, and concurrency. Comparison of test results from the Louvain algorithm and Louvain coloring. Louvain coloring's modularity value increased by 0.981199% compared to Louvain. Louvain coloring has a better processing time with a reduction time of 57.82%. The number of communities produced by the Louvain and Louvain coloring algorithms only has a difference of 0.07% (normal). The test results show a modularity value of 0.981199, communityID 181075, userCount 206, flaggedCount 7, flaggedRatio 0.033891. By comparing all scenarios, scenario 5 is the best scenario. So, based on the test results for scenario 1 above, the Louvain coloring algorithm is better than the Louvain algorithm.

3.1.2. Scenario 2

Scenario 2 uses parameters with a maxLevels value of 10, and the default values for maxIteration, tolerance, and concurrency. Comparison of test results for the Louvain algorithm and Louvain coloring. Where, Louvain's modularity value is 0.981199% higher compared to Louvain coloring. And the time from Louvain coloring is better than Louvain with a reduction time of 23.53%. And the number of communities produced by the Louvain algorithm and Louvain coloring only has a difference of 0.03% (normal). So, based on the test results for scenario 2, the Louvain algorithm is better than the Louvain coloring algorithm. Testing is carried out by changing the maxLevel parameter with a test value of 1 to 10, and for other parameters, namely maxIteration, tolerance, and concurrency, use the default values. Based on Table 1 scenario 2 results.

Table 1. Scenario 2 results

Max Level	Modularity	Community count	Community ID	User count	Flagged count	Ratio	Accuracy
1	0.847114	32,907	195490	2	2	1.000000	88
2	0.957023	16,527	182733	7	4	0.571429	88
3	0.974713	13,626	195231	40	6	0.150000	88
4	0.978536	12,617	187751	127	7	0.055118	88
5	0.979545	12,211	197388	145	7	0.048276	88
6	0.980187	12,001	202202	147	7	0.047619	88
7	0.980621	11,884	25771	154	7	0.045455	88
8	0.980805	11,790	60065	181	7	0.038674	88
9	0.980904	11,727	180048	155	7	0.045161	88
10	**0.980971**	**11,662**	**197265**	**159**	**7**	**0.044025**	**88**

Table 1 shows the test results for the maxLevel combination with the results parameters modularity, communityCount, communityID, userCount, FlaggedCount, Ratio having different values when the maxLevel parameter with different values is run, and the accuracy results do not change when the test is carried out with the maxLevel value different. Testing is carried out by changing the maxLevel parameter with a test value of 1 to 10, and for other parameters, namely maxIteration, tolerance, and concurrency, use the default values. When maxLevel with a value of 10 is tested, it gives the highest modularity results from the others.

3.1.3. Scenario 3

Scenario 3 uses parameters with a maxIteration value of 700, and the default values for maxLevels, tolerance, and concurrency. Comparison of test results for the Louvain algorithm and Louvain coloring. Louvain coloring's modularity value is 0.000128 higher compared to Louvain. The time from Louvain is better than Louvain coloring with a time reduction of 80.23%. And the number of communities produced by the Louvain coloring algorithm is better than Louvain. So, the test results for scenario 3 are that the Louvain coloring algorithm is better than the Louvain algorithm. The test is carried out by combining the maxIteration parameter values, and for the other parameters, namely maxLevel, tolerance and concurrency, they use the default values and in Table 2 scenario 3 results. In Table 2 the values for the maxIteration parameter used are 10, 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1,000. The test results are the maxIteratrions parameter value is 700, maximum modularity value is 0.847158.

Table 2. Scenario 3 results

Max Iterations	Modularity	Community count	Community ID	User count	Flagged count	Ratio	Accuracy
10	0.846946	329,014	195490	2	2	1.000000	88
100	0.847163	32,903	184405	3	2	0.666667	88
200	0.847097	32,907	195490	2	2	1.000000	88
300	0.847039	32,908	195490	2	2	1.000000	88
400	0.84708	32,906	195490	2	2	1.000000	88
500	0.846724	32,921	195490	2	2	1.000000	88
600	0.847041	32,905	196688	2	2	1.000000	88
700	**0.847158**	**32,904**	**195490**	**2**	**2**	**1.000000**	**88**
800	0.847061	32,907	195490	2	2	1.000000	88
900	0.847039	32,907	195490	2	2	1.000000	88
1000	0.847002	32,907	195490	2	2	1.000000	88

3.1.4. Scenario 4

Scenario 4 uses parameters with a tolerance value of 0.000000001, and maxLevels, maxIteration, and concurrency are the default. Comparison of test results from the Louvain algorithm and Louvain coloring. Louvain's modularity value is 0.0063% higher compared to Louvain coloring. And the processing time of Louvain coloring is better than Louvain with a reduction time of 17.45%. And the number of communities produced by the Louvain and Louvain coloring algorithms has a difference of 0.05% (normal). So, the test results for scenario 4 are that the Louvain coloring algorithm is better than the Louvain algorithm. Testing is carried out by combining the tolerance parameter values, and for other parameters, namely maxLevel, maxIteration, and concurrency, using the default values and shown in Table 3 scenario 4 results. In Table 3 the values for the tolerance parameters used are 0.01, 0.001, 0.0001, 0.00001, 0.000001, 0.0000001, 00000001, 0.000000001, 00000000001, with the highest test results on the parameter value 0.000000001 which has a modularity value 0.847597.

Table 3. Scenario 4 results

Tolerance	Modularity	Community count	Community ID	User count	Flagged count	Ratio	Accuracy
0.01	0.844313	33,080	199429	2	2	1.000000	88
0.001	0.846621	324,925	195490	2	2	1.000000	88
0.0001	0.847105	32,908	195490	2	2	1.000000	88
0.00001	0.847053	32,908	195490	2	2	1.000000	88
0.000001	0.847156	32,906	195490	2	2	1.000000	88
0.0000001	0.847102	32,906	195490	2	2	1.000000	88
0.00000001	0.847073	32,906	195490	2	2	1.000000	88
0.000000001	**0.847597**	**32,906**	**195490**	**2**	**2**	**1.000000**	**88**
0.0000000001	0.847205	32,906	194648	2	2	1.000000	88
0.00000000001	0.847188	32,906	195490	2	2	1.000000	88

3.1.5. Scenario 5

Scenario 5 uses parameters with a concurrency value of 1, and maxLevels, maxIteration, tolerance, and are the defaults. Comparison of test results from the Louvain and Louvain coloring algorithms. Louvain coloring's modularity value is 0.0024% higher compared to Louvain. And the time from Louvain coloring is better than Louvain with a reduction time of 55.96%. And the number of communities produced by the Louvain and Louvain coloring algorithms has a difference of 0.12%. So based on this, the test results for scenario 5 are that the Louvain coloring algorithm is better than the Louvain algorithm. Testing is carried out by combining the concurrency parameter values, and for other parameters, namely maxLevel, maxIteration, and tolerance using the default values in Table 4 scenario 5 results. In Table 4 the values for the concurrency parameters used are 1, 2, 3 and 4, with the highest test results on the value of parameter 1 which has a modularity value of 0.847234.

Table 4. Scenario 5 results

Concurrency	Modularity	Community count	Community ID	User count	Flagged count	Ratio	Accuracy
1	**0.847234**	**32,889**	**195490**	**2**	**2**	**1.000000**	**88**
2	0.846653	32,914	195490	2	2	1.000000	88
3	0.847034	32,913	195490	2	2	1.000000	88
4	0.846975	32,910	194597	2	2	1.000000	88

3.2. Louvain-coloring algorithm optimization to detect community

In this scenario, tests will be run with parameter values for maxLevel, maxIterations, tolerance and concurrency. The test results, the value of Q_m is 0.981199, communityID 181075, userCount 206, flaggedCount 7, flaggedRatio 0.033891. By comparing all of Louvain's scenarios, this scenario can be said to be the best scenario. The results are shown in Table 5.

From the several scenarios that have been carried out from Table 5 it is clear that the optimal results are clearly visible. Based on the results that have been achieved, it can be seen from the dataset analyzed that fraud has some closeness from the data and transaction community that is good and not fraudulent. So that a graph is formed in Figure 2.

Table 5. Optimization results of the Louvain-Coloring algorithm for detecting fraud transactions

Skenario	Louvain			Louvain Coloring		
	Time (ms)	Modularity	Community count	Time (ms)	Modularity	Community count
1	354	0.980996	11,678	296	0.980959	11,686
2	348	0.981066	11,672	219	0.980867	11,662
3	233	0.981239	11,635	193	0.980925	11,685
4	265	0.980977	11,682	184	0.980907	11,682
5	471	0.981008	11,667	265	0.981183	11,656

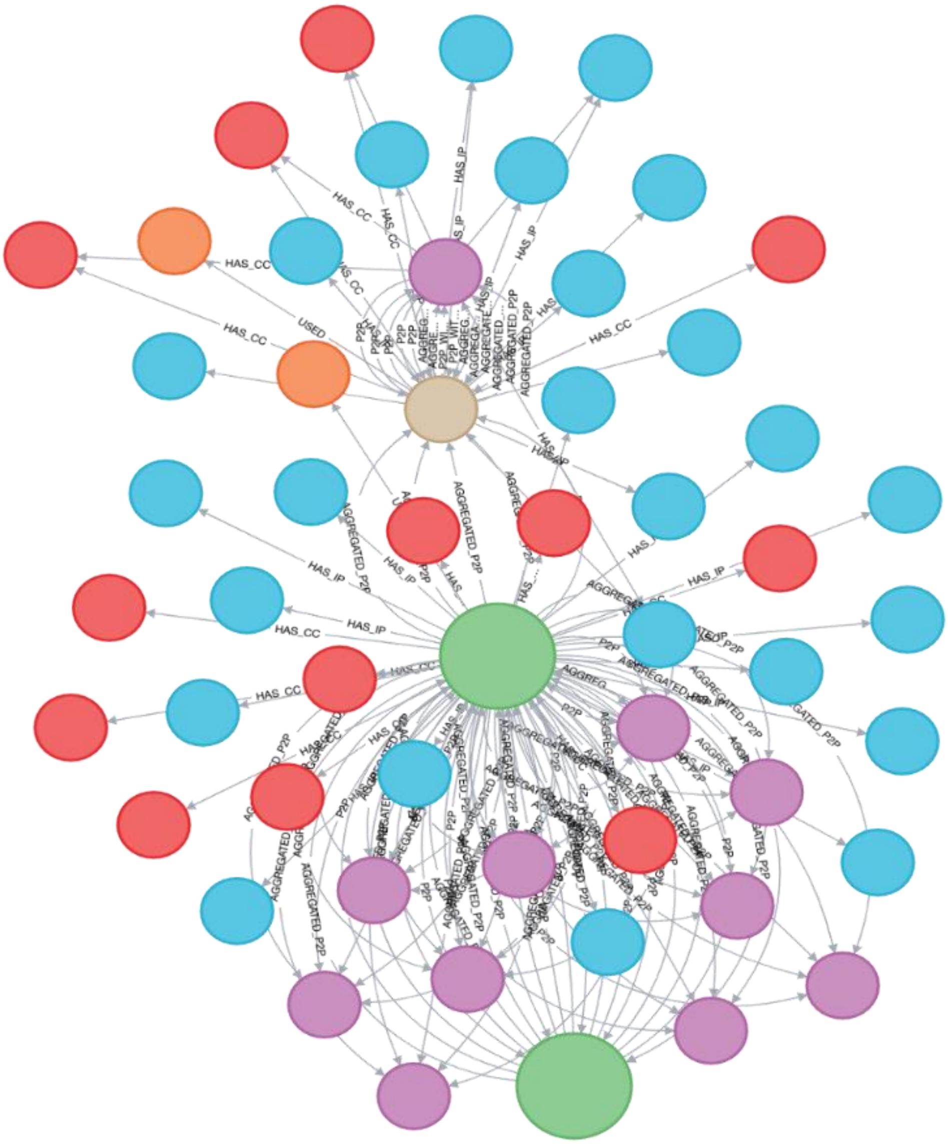

Figure 2. Graph of fraud transaction community

From Figure 2 on the graph, it can be seen that the results with the Louvain algorithm have been optimized for Louvain-coloring, there are 19 data that are close to the Fraud community. The modularity value of the Louvain coloring algorithm is better than the Louvain algorithm. From the 5 test scenarios carried out, the Louvain coloring algorithm excels in 4 scenarios, namely 1, 3, 4 and 5. Meanwhile, Louvain's algorithm only excels in scenario 2. The modularity value of the Louvain Coloring algorithm is at the lowest value of 0.980969 and the highest is 0.981207. Whereas the Louvain algorithm has the lowest value of 0.981037 and the highest of 0.981157. The highest modularity value of the Louvain coloring algorithm is obtained through scenario 1 which uses parameters with default values. While the highest modularity value of the Louvain algorithm is obtained through scenario 4 which uses a tolerance value of 0.00000001 and other parameters with default values.

The Louvain coloring algorithm has the fastest or shortest processing time produced by scenario 4, which is 123 ms, and the longest is produced by scenario 1, which is 394 ms. While the Louvain algorithm has the fastest processing time produced by scenario 2, namely 136 ms, and the longest produced by scenario 1, which is 934 ms. The longest time produced by the two algorithms in scenario 1 has a very large or quite significant time difference.

The Louvain coloring algorithm has the smallest community generated by scenario 2 with a total of 11,641 while the largest is generated by scenario 4 with a total of 11,668. Meanwhile the Louvain algorithm has the smallest community generated by scenario 2 with a total of 11,649 and the largest community is generated by scenario 4 with a total of 11,664 It is very interesting that the two algorithms together produce the smallest and largest numbers in the same scenario. And for the number of communities produced, even though it has a difference, it is in a very small range between 4 to 14 communities (<1% of the community). The researchers consider the number of communities generated can be excluded from the assessment of the algorithm. So that exposure in the Louvain-coloring algorithm is able to solve the clustering problem in transaction fraud by forming a new label so as to facilitate the decision-making process.

4. CONCLUSION

The focus of this research is cluster research on big data, namely fraudulent and non-fraud transactions in banking. Where the data that has been received is 33,491 non-fraud data and 241 fraudulent transaction data. Then to detect this, clustering is carried out on data mining techniques so that they are able to detect fraud transactions. This has been proven where the cluster process using the Louvain algorithm is able to solve clustering problems in fraudulent transactions. From the data that has been received based on the Louvain algorithm which approaches fraud data increases and has an accuracy of 90%. However, to make labeling easier, the Louvain algorithm is optimized by presenting the data in a colored graph. This makes it easier to detect. This is of course the basis for future predictions or forecasting in the cluster process in the behavior of fraud perpetrators.

REFERENCES

[1] A. Triayudi, W. O. Widyarto, L. Kamelia, I. Iksal, and S. Sumiati, "CLG clustering for dropout prediction using log-data clustering method," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 3, pp. 764–770, Sep. 2021, doi: 10.11591/ijai.v10.i3.pp764-770.

[2] W. Ramdhan, O. S. Sitompul, E. B. Nababan, and Sawaluddin, "Clustering algorithm-based pandemic multicluster framework analysis: k-means and k-medoids," in *2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM)*, Oct. 2022, pp. 1–6, doi: 10.1109/ICOSNIKOM56551.2022.10034927.

[3] A. R. Lubis, S. Prayudani, M. Lubis, and Al-Khowarizmi, "Analysis of the Markov chain approach to detect blood sugar level," *Journal of Physics: Conference Series*, vol. 1361, no. 1, Nov. 2019, doi: 10.1088/1742-6596/1361/1/012052.

[4] S. Lailiyah, E. Yulsilviana, and R. Andrea, "Clustering analysis of learning style on Anggana High School student," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 3, pp. 1409–1416, Jun. 2019, doi: 10.12928/telkomnika.v17i3.9101.

[5] A. A. Baker and Y. Ghadi, "Cancerous lung nodule detection in computed tomography images," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 5, pp. 2432–2438, Oct. 2020, doi: 10.12928/telkomnika.v18i5.15523.

[6] I. M. O. Widyantara, I. P. N. Hartawan, A. A. I. N. E. Karyawati, N. I. Er, and K. B. Artana, "Automatic identification system-based trajectory clustering framework to identify vessel movement pattern," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 12, no. 1, pp. 1–11, Mar. 2023, doi: 10.11591/ijai.v12.i1.pp1-11.

[7] S. E. Hashemi, F. Gholian-Jouybari, and M. Hajiaghaei-Keshteli, "A fuzzy C-means algorithm for optimizing data clustering," *Expert Systems with Applications*, vol. 227, Oct. 2023, doi: 10.1016/j.eswa.2023.120377.

[8] Ş. Telli and X. Zhao, "Clustering in Bitcoin balance," *Finance Research Letters*, vol. 55, Jul. 2023, doi: 10.1016/j.frl.2023.103904.

[9] E. B. Nababan, O. S. Sitompul, and J. Sianturi, "Analysis of tabu list length on job shop scheduling problem," *Journal of Physics: Conference Series*, vol. 1235, no. 1, Jun. 2019, doi: 10.1088/1742-6596/1235/1/012047.

[10] A. C. Benabdellah, A. Benghabrit, and I. Bouhaddou, "A survey of clustering algorithms for an industrial context," *Procedia*

Computer Science, vol. 148, pp. 291–302, 2019, doi: 10.1016/j.procs.2019.01.022.

[11] J. M. D. Delgado *et al.*, "Robotics and automated systems in construction: Understanding industry-specific challenges for adoption," *Journal of Building Engineering*, vol. 26, Art. no. 100868, pp. 1-11, Nov. 2019, doi: 10.1016/j.jobe.2019.100868.

[12] A. Al-Khowarizmi, R. Syah, and M. Elveny, "The model of business intelligence development by applying cooperative society based financial technology," in *Proceedings of Sixth International Congress on Information and Communication Technology*, Springer Singapore, 2022, pp. 117–125.

[13] K. W. F. Ma and T. McKinnon, "COVID-19 and cyber fraud: emerging threats during the pandemic," *Journal of Financial Crime*, vol. 29, no. 2, pp. 433–446, Mar. 2022, doi: 10.1108/JFC-01-2021-0016.

[14] A. Srivastav and S. Prajapat, "Text similarity algorithms to determine Indian penal code sections for offence report," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 1, pp. 34–40, Mar. 2022, doi: 10.11591/ijai.v11.i1.pp34-40.

[15] M. Leo, S. Sharma, and K. Maddulety, "Machine learning in banking risk management: a literature review," *Risks*, vol. 7, no. 1, Mar. 2019, doi: 10.3390/risks7010029.

[16] X. Mao, H. Sun, X. Zhu, and J. Li, "Financial fraud detection using the related-party transaction knowledge graph," *Procedia Computer Science*, vol. 199, pp. 733–740, 2022, doi: 10.1016/j.procs.2022.01.091.

[17] M. Jans, J. M. van der Werf, N. Lybaert, and K. Vanhoof, "A business process mining application for internal transaction fraud mitigation," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13351–13359, Sep. 2011, doi: 10.1016/j.eswa.2011.04.159.

[18] J. M. Arockiam and A. C. S. Pushpanathan, "MapReduce-iterative support vector machine classifier: novel fraud detection systems in healthcare insurance industry," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 1, pp. 756–769, Feb. 2023, doi: 10.11591/ijece.v13i1.pp756-769.

[19] I. A. Ajah and H. F. Nweke, "Big data and business analytics: trends, platforms, success factors and applications," *Big Data and Cognitive Computing*, vol. 3, no. 2, Jun. 2019, doi: 10.3390/bdcc3020032.

[20] I. Vorobyev and A. Krivitskaya, "Reducing false positives in bank anti-fraud systems based on rule induction in distributed tree-based models," *Computers and Security*, vol. 120, Sep. 2022, doi: 10.1016/j.cose.2022.102786.

[21] S. Shirazi, H. Baziyad, N. Ahmadi, and A. Albadvi, "A new application of Louvain algorithm for identifying disease fields using big data techniques," *Journal of Biostatistics and Epidemiology (Quarterly)*, vol. 5, no. 3, 2019, doi: 10.18502/jbe.v5i3.3613.

[22] W. Zhang, "Improving commuting zones using the Louvain community detection algorithm," *Economics Letters*, vol. 219, Oct. 2022, doi: 10.1016/j.econlet.2022.110827.

[23] J. Zhang, J. Fei, X. Song, and J. Feng, "An improved Louvain algorithm for community detection," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–14, Nov. 2021, doi: 10.1155/2021/1485592.

[24] N. S. Sattar and S. Arifuzzaman, "Scalable distributed Louvain algorithm for community detection in large graphs," *The Journal of Supercomputing*, vol. 78, no. 7, pp. 10275–10309, May 2022, doi: 10.1007/s11227-021-04224-2.

[25] C. Makris, D. Pettas, and G. Pispirigos, "Distributed community prediction for social graphs based on Louvain algorithm," in *IFIP Advances in Information and Communication Technology*, Springer International Publishing, 2019, pp. 500–511.

[26] J. Qin, M. Li, and Y. Liang, "Minimum cost consensus model for CRP-driven preference optimization analysis in large-scale group decision making using Louvain algorithm," *Information Fusion*, vol. 80, pp. 121–136, Apr. 2022, doi: 10.1016/j.inffus.2021.11.001.

BIOGRAPHIES OF AUTHORS

Heru Mardiansyah born in Medan, Indonesia, in 1980. Student of the Faculty of Computer Science and Information Technology, University of North Sumatra and Masters from the University of North Sumatra in 2019 and currently pursuing a Ph.D. program at the University of North Sumatra. Research areas of data science, big data, machine learning, neural networks, artificial intelligence, and business intelligence. He can be contacted at email: heru@students.usu.ac.id.

Saib Suwilo Professor from Universitas Sumatera Utara, Medan Indonesia. Saib Suwilo was born in Simalungun, North Sumatera. Drs. in mathematics at USU Medan, 1992, M.Sc. in mathematics at University of Wyoming, 1994, and Ph.D. in mathematics at University of Wyoming, 2001. He can be contacted at email: saib@usu.ac.id.

Erna Budhiarti Nababan associate professor from Universitas Sumatera Utara, Medan Indonesia. Erna Budhiarti Nababan was born in Bandung, West Java. S.Pd. at Institute Keguruan dan Ilmu Pendidikan (IKIP Bandung), 1981, MIT at University Kebangsaan Malaysia (UKM Malaysia), 2005; Dr. at Universiti Kebangsaan Malaysia (UKM Malaysia), 2010. He can be contacted at email: ernabrn@usu.ac.id.

Syahril Efendi professor from Universitas Sumatera Utara, Medan Indonesia. His main research interest is data science, big data, machine learning. He can be contacted at email: syaril1@usu.ac.id and syahnyata1@gmail.com.