www.czasopisma.pan.pl PAN www.journals.pan.pl

IINTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2023, VOL. 69, NO. 3, PP. 593-598
Manuscript received July 28, 2023; revised July, 2023. DOI: 10.24425/ijet.2023.146512

Community Clustering on Fraud Transactions Applied the Louvain-Coloring Algorithm

Heru Mardiansyah, Saib Suwilo, Erna Budiarti Nababan, and Syahril Efendi

***Abstract*—The contribution main from this research is modularity and better processing time in detecting community by using K-1 coloring. Testing performed on transaction datasets remittance on P2P platforms where the Louvain Coloring algorithm is better in comparison to Louvain Algorithm Data used is data transfer transactions made by customers on the P2P Online platform. The data is the User data that has information transfer transactions, Card data that has information card, IP data that has IP information, and Device data that has information device. Every user owns unique 128-bit identification, and other nodes representing card, device, and IP are assigned a random UUID. The Device node has the guide, and device properties. IP nodes only have property guide and node User has property fraud Money Transfer, guide, money Transfer Error Cancel Amount, first Charge back Date. Each node has a unique 128-bit guide, with the amount whole of as many as 789,856 nodes. Application technique K-1 staining on Louvain algorithm shows enhancement value modularity and better processing time for detecting community on the network large scale. Through a series of exercises and tests carried out in various scenarios, it shows that the experiments carried out in this paper, namely the Louvain Coloring algorithm, are more effective and efficient than the Louvain algorithm in scenario 1,3, and 5 meanwhile For Scenarios 2 and 4 Louvain Algorithm is better.**

***Keywords*—fraud Money Transfer; Louvain Coloring algorithm; Louvain Algorithm**

I. INTRODUCTION

DETECTION community in large and complex networks is a topic to get deep attention in various fields including computers, physics, and biology. In this study, we focused on Louvain algorithm. Louvain algorithm can produce partition maximizing network modularity, a metric that measures the strength distribution network to become a community. The algorithm was first proposed by (Blondel et al., 2008) and has been Lots used in various applications for detecting community in networks. This method uses a greedy approach to maximizing modularity, for showing the extent to of the network can be shared to become a clear community. However, in a number of the case, Louvain algorithm is lacking capable detect structure optimal communities, especially on networks that have level density different communities.

The Louvain algorithm (Blondel et al., 2008) is several theme algorithm detection community. This algorithm is created as a method of disclosure fast community for big networks. This algorithm works with two stages which are local node transfers and aggregation network. There is also a Surprise Community Detection algorithm (Traag et al., 2015) that was created to overcome limitations of modularity, with size based on probability known as classic as shock for evaluating the quality partition network inside community. This algorithm works with moving nodes from one community to another community so shock then improved in a manner massive. Graph coloring techniques such as K-1 Coloring (Catalyurek et al., 2012) are capable speed up the process because with gift the colors at the nodes are likened to giving an index on a relational database.

II. MATERIALS AND METHODS

Data used is data transfer transactions made by customers on the P2P Online platform. The data is the User data that has information transfer transactions, Card data that has information card, IP data that has IP information, and Device data that has information device. Every user owns unique 128-bit identification, and other nodes representing card, device, and IP are assigned a random UUID.

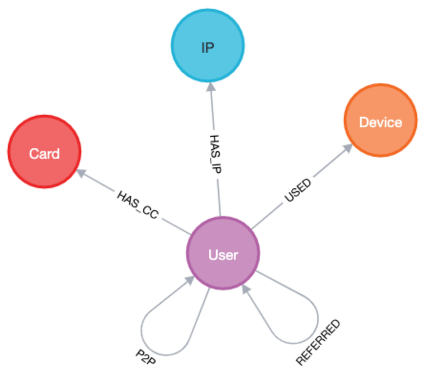


Fig. 1. Schematic Chart

Heru Mardiansyah is with the Department Computer Science, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia (e-mail: heru.mardiansyah@students.usu.ac.id).

Saib Suwilo is with the Department Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Sumatera Utara, Medan, Indonesia (e-mail: saib@usu.ac.id).

Erna Budhiarti Nababan is with the Department Data Science and Artificial Intelligence, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia (e-mail: ernabrn@usu.ac.id).

Syahril Efendi is with the Department Computer Science, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia (e-mail: syahril1@usu.ac.id).

P2P transactions are represented by cascading relationships from the customer's sender to the customer's receiver. In addition, there are also identifier nodes for Users is *Cards, Devices,* and *IPs*. Each User node has a variable indicator for money transfer scam (given Name *MoneyTransferFraud*) is worth 1 for fraud and 0, for not a fraud. This indicator is determined by events reject pay on transactions that occur and manual review.

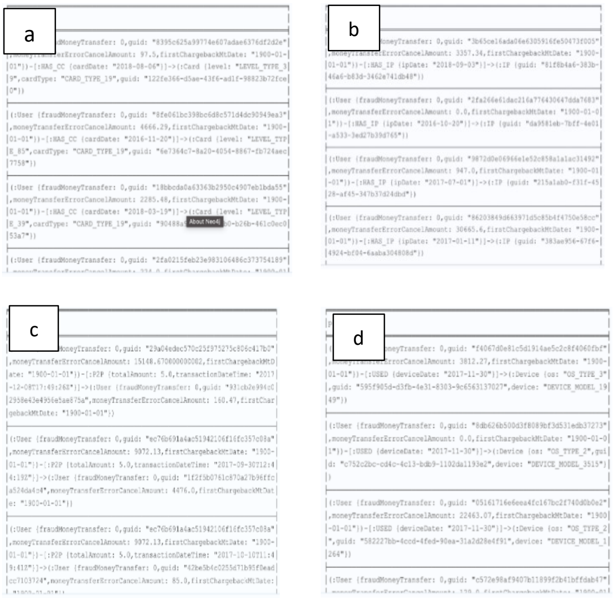


Fig. 2. Relationships a. Has CC, b. Has IP, c. P2P and d. Used

Finally, the Properties component. If on a relational database We know the column, then can be equated with properties on the graph database. The properties used are quite a lot as shown in Table I.

TABLE I
PROPERTIES

No	Properties	No	Properties
1	cardDate	11	Levels
2	card type	12	moneyTransferErorrCancelAmount
3	data	13	name
4	device	14	nodes
5	delicate	15	number of transaction
6	firstChagerBackMtDate	16	os
7	fraud Money Transfer	17	relationships
8	guid	18	styles
9	id	19	totalAmount
10	ipDate	20	transactionDateTime

Graphic data stored in the neo4j graphics database can be visualized in the knowledge graph form. Knowledge graphs are capable give a better understanding of the connection between data.

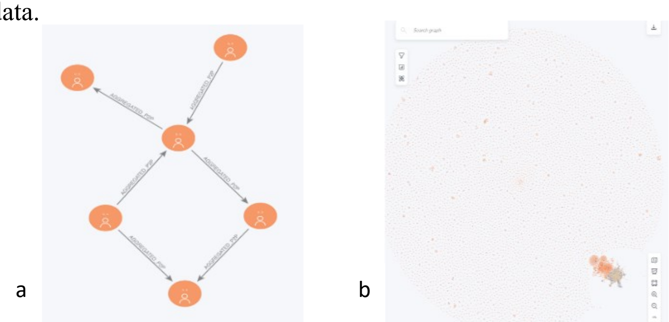


Fig. 3. Knowledge Graph, a) KG of all data, b) KG P2P

After various processes are carried out, the shape final graph database changed become Figure 4. There is the addition of 3 nodes viz FlaggedUser, PredictedFraudRisk, and FraudRiskUser.

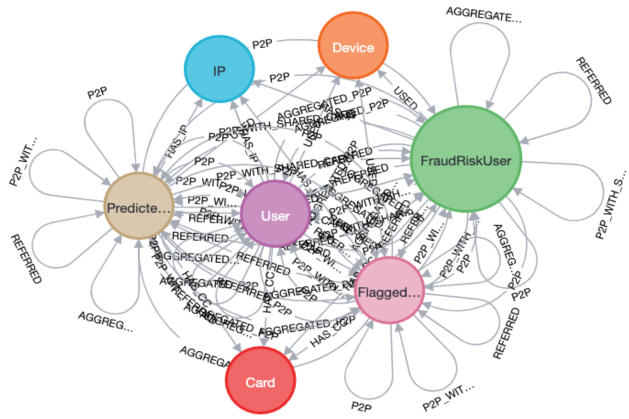


Fig. 4. Schematic Chart End

This is graph database, there are components mainly consisting of nodes, relationships and properties, and labels. In a relational database, rows are contained in tables that can be equated with nodes on the graph database. Therefore, the Card node or customer data has 3 level properties, card Type, and guide. The Device node has the os, guide, and device properties. IP nodes only have property guide and node User has property fraud Money Transfer, guide, money Transfer Error Cancel Amount, first Charge back Date. Each node has a unique 128-bit guide, with the amount whole of as many as 789,856 nodes.

In addition, component relationships have over 1.7 million CC, Has IP, used and P2P relationships. If there is a join in a relational database, it is considered equivalent to a chart database relationship. Has CC is the relationship between user nodes and card nodes. A Has IP is a relationship between a user and an IP node. Used is a User and Device relationship. P2P is a relationship between a user and a user who transfers money.

III. RESULTS AND DISCUSSION

The contribution main from this research is modularity and better processing time in detecting community by using K-1 coloring. Testing performed on transaction datasets remittance on P2P platforms where the Louvain Coloring algorithm is better in comparison to Louvain Algorithm.

A. Louvain Algorithm

Louvain algorithm is a method very heuristic and popular for the detection-optimized community with modularity. steps Louvain algorithm as follows:

1. Initiation, every knot in the network assigned to the community alone. That is, if there are N vertices, then there are N communities.
2. Phase 1 – Optimization Local, for every node, consider community knot neighbors and try to move this node to the community neighbors who will produce the biggest increase in modularity. If No There is possible improvement made, node the still is at in community original.

3. Phase 2 – Aggregation. After all, knots are checked and modularity optimized, we make a new network where every knot Now is the community from the network before. If there is a number of edges between the same community, these edges are combined in the new network, and their weight becomes the amount of weight the edge original. This phase produces more network small with amount reduced node.
4. Iteration: phases 1 and 2 are repeated so that No There is enhancement and more carry-on in modularity. In other words, the algorithm ends at modularity and reaches peak local. The result is a partitioned community in a network.

B. *Louvain Coloring Algorithm*

To modify Louvain algorithm using the next K-1 coloring called Louvain Coloring, it is necessary to consider the coloring process when optimizing Q. Combining modularity second this method with add constraints coloring in the updated community. Next, with add constraints coloring in modularity Q. Suppose We use function $g = (i, j)$which is 1 if vertices i and j are of different-colors and 0 if they are of the same color. So the Q modularity formula with K-1 coloring becomes

$$Q\_m = \frac{1}{2m}\sum\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j) * g(i, j)$$

Whereas steps Louvain coloring algorithm, are as follows:
Phase 1: Detection Community
We use Louvain algorithm to detect community in the graph. steps from algorithms and formulas math used.
Phase 2: Staining Community
After the community was identified, then applied principle graph K-1 coloring to the resulting communities. Each community is given a color unique, with an amount of color No more than the amount of community (K-1).

The final result of this process is every knot in the network has labeled according to community and color the community. In this way, have combined the Louvain algorithm and K-1 Coloring.

The input stage uses money transfer transaction data on the P2P platform. Followed by the stages of the divided process into 2 parts using graph algorithm is Louvain Coloring algorithm, using a rule with Entity Relationship, weakly Connected Component algorithm, algorithm PageRank and Degree Centrality algorithms.

The modularity value of the Louvain Coloring algorithm is better than Louvain Algorithm. From 5 scenarios the tests carried out, the Louvain coloring algorithm excels in 4 scenarios namely 1, 3, 4, and 5. Meanwhile, the Louvain algorithm is only superior in scenario 2. The modularity value of the Louvain Coloring algorithm is in value Lowest at 0.980969 and the highest at 0.981207. Whereas for Louvain algorithm has mark Lowest of 0.981037 and the highest of 0.981157. The highest modularity value from the Louvain Coloring algorithm is obtained through scenario 1 which uses the parameter with default value. Whereas the highest modularity value from Louvain algorithm is obtained through Scenario 4 uses a tolerance value of 0.00000001 and other parameters default values.

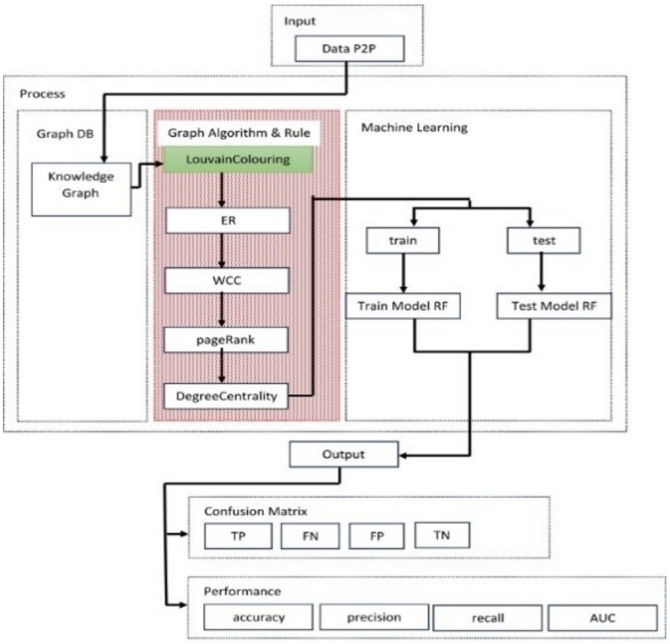


Fig. 5. Louvain Coloring

Louvain Coloring algorithm has the fastest or shortest processing time generated by scenario 4 which is 123-ms, and the longest generated by scenario 1 is 394-ms. Whereas the Louvain algorithm has the fastest processing time produced by scenario 2, namely 136-ms, and the longest generated by scenario 1, namely 934-ms. The longest-time generated second algorithm in scenario 1 has a very large or significant time difference.

Louvain Coloring Algorithm has the smallest community generated by scenario 2 with the number 11641 while the largest is generated by scenario 4 with the number 11668. Meanwhile, Louvain algorithm has the smallest community generated by scenario 2 with the number 11649 and the largest community generated by scenario 4 with the number 11664. Very interesting, that second algorithm the produce together amounts smallest and largest in the same scenario. And for the amount the resulting community, though own difference However in very range small in between 4 to 14 communities (< 1 % of the community). So the researcher considers the amount the resulting community can issue from evaluation to algorithm.

C. *K-1 Coloring Algorithm*

Introduce A graph method with gift color. Giving the next color is called K-1 coloring. With gift color on each node, which has the objective For find coloring with the amount min color.

K-1 Coloring gives color on the vertices in graphs, with the rule that No There are two adjoining nodes that own the same color. For example, there is a graph $G(V, E)$, where V are the vertices in the graph and E are the edges in the graph. We want to color each vertex $v \in V$with color $w(v) \epsilon \{1, 2, \ldots, k\}$, where k is the number of colors used, so that :

$$\forall (u, v) \epsilon, w(u) \neq w(v)$$

The P2P dataset consists of 4 nodes and 5 relations that have Lots property.

TABLE II
P2P DATASETS

Node Labels (789,856)		Relationship Types (1,998,658)	
Users	33,732	Used	55,026
Device	51,451	Has_IP	1,488,989
Cards	118,818	Has_CC	128,066
IP	585,855	Referred	1,870
		P2P	102,832

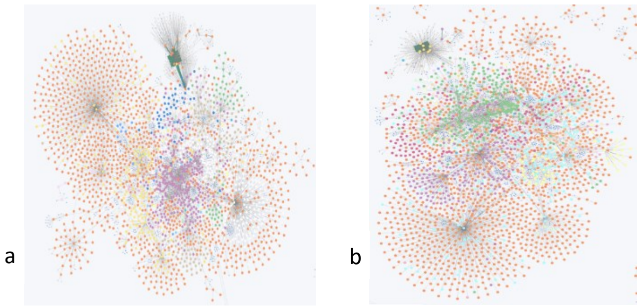


Fig. 6. Visualization Community , a) Louvain b) Louvain Coloring

From the visualization, the communities formed by the Louvain Algorithm and Louvain Coloring are different. This is caused by the coloring process that is run after the Louvain process.

There are some parameters that can increase the performance Louvain algorithm, this study uses general parameters, namely :

I. maxLevels (default: 10), every two phases is a new network, which is the level for the hierarchy community. When run, the increasingly tall the level, the bigger the community, which is possibly No exactly as desired. This parameter defines the maximum possible Level. The algorithm can return more beginning before achieving that.
II. maxIterations (default 10), in round two phases, phase First Keep going repeated iteratively until enhancement modularity can ignored or numbered iteration achieved as specified.
III. tolerance (default 0.0001), the smaller the tolerance, the higher, the better, but more Lots possible iterations are required.
IV. concurrency (default 4), the number of threads running together at the same time.

Because the value of high modularity is Not yet Of course generated by the highest parameter value nor default. So, need done testing to get the parameter configuration of the lowest value until the highest.

TABLE III
LOUVAIN PARAMETERS TEST

maxLevels	1	2	3	4	5	6	7	8	9	10
times	595	291	134	177	139	203	185	201	152	119
modularity	0.847107	0.957722	0.974399	0.978376	0.979675	0.980223	0.980431	0.980763	0.980986	0.980926
maxIterations	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
times	310	141	123	142	170	263	179	144	235	185
modularity	0.980634	0.98092	0.980947	0.981119	0.981019	0.981165	0.981113	0.981033	0.981029	0.981177
tolerance	**0.01**	**0.001**	**0.0001**	**0.00001**	**0.000001**	**0.000000001**	**0.0000000001**	**0.000000001**	**0.0000000001**	
times	176	125	173	147	179	229	115	156	178	
modularity	0.973495	0.979473	0.980772	0.98112	0.980922	0.98111	0.98108	0.981136	0.981019	
concurrency	**1**	**2**	**3**	**4**						
times	438	317	266	210						
modularity	0.981055	0.980993	0.981101	0.980863						

Parameter maxLevels with values 1 – 10 result in modularity with a mark Lowest of 0.847107 and the highest of 0.980986 (more height 15.80 %), parameter maxLevels with a mark of 9 is the best. The maxIterations parameter has values 1 – 10 resulting in modularity with a mark Lowest of 0.980634 and the highest of 0.981177 (more height 0.001166 %), in the maxIteration parameter with a mark of 10 is the best. Parameter tolerance that has a value of 0.01 to 0.000000001 yields the lowest modularity value of 0.9739495 and the largest of 0.981136 (more high 0.78490 %), in the tolerance parameter with a mark of 0.000000001 is the best. Concurrency parameters that have value 1 – 4, produce modularity with the mark Lowest of 0.980993 and the highest of 0.981101 (more high 0.01100 %), on the concurrency parameter with the mark of 3 is the best. So, we get a parameter configuration that produces the highest modularity that is maxLevels parameter value of 9, the value in the maxIteration parameter of 10, the tolerance parameter is 0.000000001 and the concurrency parameter is 3.

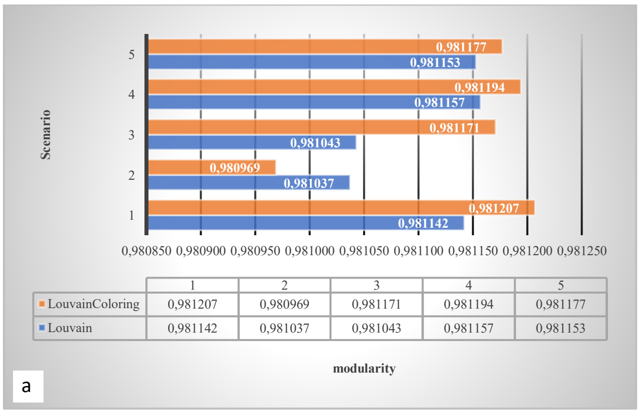
After getting the best parameter configuration, the next study continued with using 5 test scenarios that combine some parameters. Scenario these, namely:

TABLE IV
SCENARIO TESTING

Scenario	Parameter			
	max levels	maxIterations	tolerance	concurrency
1 (default)	10	10	0.0001	4
2	9	10	0.0001	4
3	10	10	0.000000001	4
4	10	10	0.0001	3
5	9	10	0.000000001	3

A good scenario is obtained when carrying out an assessment process based on:

i. The modularity value in Louvain's algorithm is greater than that of Louvain coloring, so if Louvain gets a good score then the opposite is true
ii. Louvain algorithm testing time is more effective than Louvain Coloring, so if Louvain gets a good score then the opposite is true
iii. If in one test scenario Louvain algorithm generates value modularity and better processing time than Louvain Coloring then Louvain algorithm is better, and vice versa.



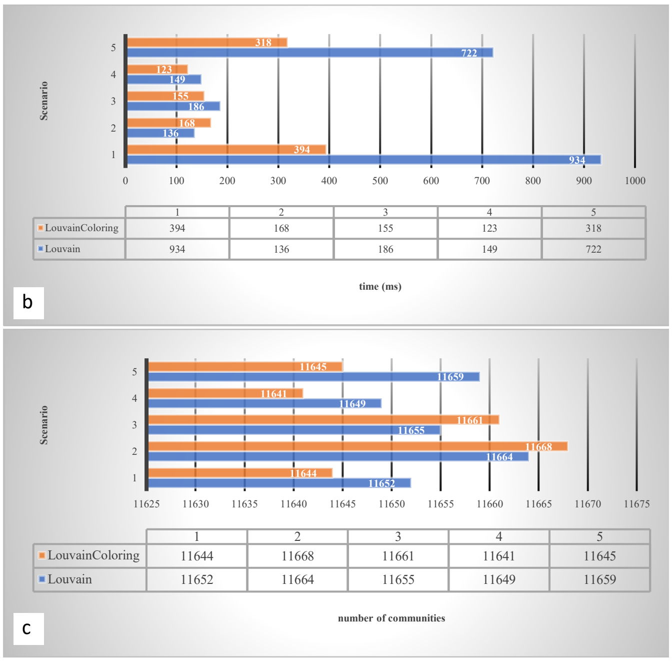


Fig.7. Comparison of Louvain Algorithm and Louvain Coloring, a) modularity, b) time, c) the amount of community

Scenario 1

Scenario 1 uses the parameter with mark maxLevels, maxIteration, tolerance, and concurrency as default. comparison results testing from Louvain and Louvain Coloring algorithms.The value of Louvain Coloring's modularity increases more tall of 0.0064% compared with Louvain. And Louvain Coloring has a better processing time with a reduction time of 57.82%. As well as the amount of community generated by the Louvain algorithm and Louvain Coloring only own difference by 0.07% (normal). So, based on the results testing for scenario 1 above is Louvain Coloring algorithm is better than Louvain Algorithm.

Scenario 2

Scenario 2 uses the parameter with mark max levels by 10, and for mark maxIteration, tolerance, and concurrency are default. comparison from results testing on The Louvain and Louvain Coloring algorithms are shown in Figure 7. Where the Louvain modularity value is higher tall of 0.0068% compared with Louvain Coloring. And the time of Louvain Coloring is better than Louvain with a reduction time of 23.53%. As well as with the amount of community generated by the Louvain algorithm and Louvain Coloring only own a difference of 0.03% (normal). So, based on the results of testing for scenario 2 Louvain algorithm is better than Louvain Coloring Algorithm.

Scenario 3

1) Scenario 3 uses the parameter with mark maxIteration by 700, and for mark max levels, tolerance, and concurrency are default. comparison from results testing on The Louvain and Louvain Coloring algorithms are shown in Figure 7. The Louvain Coloring modularity value is higher tall of 0.000128 compared with Louvain. And the time from Louvain is better than Louvain Coloring with a reduction time of 80.23%. As well as the amount of community generated by the Louvain Coloring algorithm is better than Louvain. So, the results testing For scenario 3 is Louvain Coloring algorithm is better than Louvain Algorithm.

Scenario 4

Scenario 4 uses the parameter with a tolerance value is 0.000000001, and maxLevels, maxIteration, and concurrency are default. comparison results testing from The Louvain and Louvain Coloring algorithms are shown in Figure 7. The Louvain modularity value is taller at 0.0063% compared with Louvain Coloring. And the processing time of Louvain Coloring is better than Louvain with a reduction time of 17.45%. As well as the amount the community generated by the Louvain algorithm and Louvain Coloring has selfish by 0.05% (normal). So the results testing For scenario 4 is Louvain Coloring algorithm is better than Louvain Algorithm.

Scenario 5

Scenario 5 uses the parameter with mark concurrency as 1, and maxLevels, maxIteration, tolerance, and are default. comparison results testing from Louvain and Louvain Coloring algorithms as shown in Figure 7. The value of Louvain Coloring modularity is taller by 0.0024% compared with Louvain. And the time of Louvain Coloring is better than Louvain with a reduction time of 55.96%. As well as the amount the community generated by the Louvain algorithm and Louvain Coloring has a difference of 0.12%. So, based on matters such, as the results testing For scenario 5 is Louvain Coloring algorithm is better than Louvain Algorithm.

Application technique K-1 staining on Louvain algorithm shows enhancement value modularity and better processing time for detecting community on the network large scale. Through a series of exercises and tests carried out in various scenarios, it shows that the experiments carried out in this paper, namely the Louvain Coloring algorithm, are more effective and efficient than the Louvain algorithm in scenario 1,3, and 5 meanwhile For Scenarios 2 and 4 Louvain algorithm is better.

CONCLUSION

Problem detection community in the great network already Lots investigated during last few years. Although Louvain algorithm is a method of detecting effective community, efficiency and time become decrease with an increased amount of data. Maintain high modularity and spend less time, improved algorithm based on application K-1 stain.

Therefore, the Louvain Coloring algorithm can obtain better modularity results and spend more little time in comparison with Louvain Algorithm. This study uses a transaction dataset P2P remittance has nodes and links between very large nodes (more from hundreds of thousand). Experiment results show that results best obtained by scenario 1 with maxLevels, maxIteration, tolerance, and concurrency parameters which have default values. Modularity Louvain Coloring algorithm has increased by 0.0064% and reduced processing time up to 57.82 %, as well as the amount of community increased by 0.07%.

Based on matter the above, the proposed Louvain Coloring algorithm own a better effect and performance. Therefore, in the time future, the necessary study used a larger variety of datasets and combine them with various algorithms to get a high modularity value.

REFERENCES

[1] Abdallah, A., Maarof, M. A., and Zainal, A. Fraud detection system: A survey. Journal of Network and Computer Applications, 2016. 68, 90–113.

[2] Albrecht, J., Belger, A., Blum, R., & Zimmermann, R. (n.d.). Business Analytics on Knowledge Graphs for Market Trend Analysis.

[3] Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. Data mining for credit card fraud: A comparative study. 2011. Decision Support Systems, 50(3), 602–613.

[4] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10), P10008.

[5] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data-SIGMOD '08, 1247.

[6] Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (n.d.). LOF: Identifying Density-Based Local Outliers.

[7] Carcillo, F., Le Borgne, Y.-A., Caelen, O., & Bontempi, G. Streaming active learning strategies for real-life credit card fraud detection: Assessment and visualization. 2018. International Journal of Data Science and Analytics, 5(4), 285–300.

[8] Carneiro, N., Figueira, G., & Costa, M. A data mining based system for credit-card fraud detection in e-tail. 2017. Decision Support Systems, 95, 91–101. https://doi.org/10.1016/j.dss.2017.01.002

[9] Catalyurek, U., Feo, J., Gebremedhin, A., Halappanavar, M., & Pothen, A. Graph Coloring Algorithms for Muti-core and Massively Multithreaded Architectures. 2012. (arXiv:1205.3809).

[10] Chicco, D., & Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. 2020. BMC Genomics, 21(1), 6. https://doi.org/10.1186/s12864-019-6413-7

[11] Crotti Junior, A., Orlandi, F., Graux, D., Hossari, M., O'Sullivan, D., Hartz, C., & Dirschl, C. Knowledge Graph-Based Legal Search over German Court Cases. 2020.

[12] Ernst, P., Siu, A., & Weikum, G. KnowLife: A versatile approach for constructing a large knowledge graph for biomedical sciences. 2015. BMC Bioinformatics, 16(1), 157.

[13] Jemima Jebaseeli, T., Venkatesan, R., & Ramalakshmi, K. Fraud Detection for Credit Card Transactions Using Random Forest Algorithm. 2021. (Vol. 1167, pp. 189–197).

[14] Kalaycı, E. G., Grangel González, I., Lösch, F., Xiao, G., ul-Mehdi, A., Kharlamov, E., & Calvanese, D. 2020. (Eds.), The Semantic Web – ISWC 2020 (Vol. 12507, pp. 464–481).

[15] Lucas, Y., Portier, P.-E., Laporte, L., Calabretto, S., Caelen, O., He-Guelton, L., & Granitzer, M. Multiple perspectives HMM-based feature engineering for credit card fraud detection. 2019.

[16] Madhubabu, B. N. V., Vyshnavi, T., & Ashok, K. Credit Card Fraud Detection Algorithm using Decision Trees- based Random Forest Classifier. 2021.

[17] Maes, S., Tuyls, K., Vanschoenwinkel, B., & Manderick, B. Credit Card Fraud Detection. Applying Bayesian and Neural networks. 2023.

[18] Murorunkwere, B. F., Tuyishimire, O., Haughton, D., & Nzabanita, J. Fraud Detection Using Neural Networks: A Case Study of Income Tax. Future Internet. 2022. 14(6), 168.

[19] Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. 2011. Decision Support Systems, 50(3), 559–569.

[20] Rb, A., & Kr, S. K. Credit card fraud detection using artificial neural network. 2021. Global Transitions Proceedings, 2(1), 35–41.

[21] Sopiyan, M., Fauziah, F., & Wijaya, Y. F. Fraud Detection Using Random Forest Classifier, Logistic Regression, and Gradient Boosting Classifier Algorithms on Credit Cards. 2022. JUITA: Jurnal Informatika, 10(1), 77.

[22] Traag, V. A., Aldecoa, R., & Delvenne, J.-C. Detecting communities using asymptotical Surprise. 2015. Physical Review E, 92(2), 022816.

[23] Traag, V. A., Waltman, L., & van Eck, N. J. From Louvain to Leiden: Guaranteeing well-connected communities. 2019. Scientific Reports, 9(1), 5233.

[24] Yang, Y., Chen, R., Bai, X., & Chen, D. Finance Fraud Detection with Neural Network. 2020. E3S Web of Conferences, 214, 03005.

[25] Zhang, D., Bhandari, B., & Black, D. Credit Card Fraud Detection Using Weighted Support Vector Machine. 2020. Applied Mathematics, 11(12), 1275–1291.

[26] Zhao, X., Zhang, J., & Qin, X. LOMA: A local outlier mining algorithm based on attribute relevance analysis. 2017. Expert Systems with Applications, 84, 272–280.