IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 10, No. 3, September 2021, pp. 764~770
ISSN: 2252-8938, DOI: 10.11591/ijai.v10.i3.pp764-770

CLG clustering for dropout prediction using log-data clustering method

Agung Triayudi[1], Wahyu Oktri Widyarto[2], Lia Kamelia[3], Iksal[4], Sumiati[5]
[1]Department of Informatic and Communication Technology, Universitas Nasional, Indonesia
[2]Department of Industrial Engineering, Universitas Serang Raya, Indonesia
[3]Department of Electrical Engineering, UIN Sunan Gunung Djati, Indonesia
[4]Department of Electrical Engineering, Universitas Faletehan, Indonesia
[5]Department of Informatic, Universitas Serang Raya, Indonesia

Article Info

Article history:

Received Mar 2, 2021
Revised May 5, 2021
Accepted May 22, 2021

Keywords:

Dropout prediction
Educational data mining
k-means
Outlier detection
UNIX commands

ABSTRACT

Implementation of data mining, machine learning, and statistical data from educational department commonly known as educational data mining. Most of school systems require a teacher to teach a number of students at one time. Exam are regularly being use as a method to measure student's achievement, which is difficult to understand because examination cannot be done easily. The other hand, programming classes makes source code editing and UNIX commands able to easily detect and store automatically as log-data. Hence, rather that estimating the performance of those student based on this log-data, this study being more focused on detecting them who experienced a difficulty or unable to take programming classes. We propose CLG clustering methods that can predict a risk of being dropped out from school using cluster data for outlier detection.

Corresponding Author:

Agung Triayudi
Department of Informatic and Communication Technology
Universitas Nasional
Jl. Sawo Manila, RT.14/RW.3, Ps. Minggu, Kec. Ps. Minggu
Kota Jakarta Selatan, Jakarta, Indonesia
Email: agungtriayudi@civitas.unas.ac.id

1. INTRODUCTION

Educational data mining is a data mining implemented technique in an effort to develop data explorations of various educational information systems recommend the single linkage (SLG) dissimilarity increment distribution method, global cumulative score standard (SLG), and average linkage (ALG) dissimilarity increment distribution, global cumulative score standard (ALG) which used to analyze student learning online interaction data. The end result is a grouping model of behavior patterns and interpersonality patterns of students [1], [2]. The initial process starts from collecting data, before it is continued with data transformation, and is terminated by data analysis [3]. Educational data mining is implemented in order to achieve the goal of fulfilling the useful information needs of large amounts of electronic data recorded in the educator system [4]-[6].

Referring to the main topic of the discussion this time, the actual education system. In most schools, most teachers will teach a number of students in a class at a time [7], [8]. Of course, this will complicate the teacher in displaying the material in detail on each of its students [9]. On the other hand, teachers need to know the skill level of students to guide them and provide a high quality education. Therefore, testing periodically needs to be done to see if the students have the skills he needs [10], [11]. However, it is quite

difficult to understand the students ' ability from each subject, because the exams have depleted a lot of time as well as incriminated class members [12]. The other hand, it is common to track the activities and behaviors in programming class, as source code editing and UNIX commands save them as log-data [13], [14].

We found several studies related to log-data usages. Likes, there is a research that predicts student skills based on the log-data [15], [16]. Although the level of accuracy of the method in this prediction is not so high, and still a lack of consideration whether the evaluation is done based on the prescribed aspects. Moreover, in some research apparently found difficulties in evaluating the students' acquired skills based on log-data. Therefore, this research is aimed at obtaining data on students who cannot keeping up with the programming class, rather than to estimate the achievement of students based on data-logs. We specifically propose a method that can be applied to predict dropout by using outlier detection without any learning supervision.

2. RESEARCH METHOD

2.1. Problem setting for dropout prediction

Monitoring and supporting the students highly necessary according to the department of education. when the teacher is able to track a student with high risk being dropout from the beginning, they can take action immediately and make sure to help that student so he or she will not be expelled from the school. Hence, it is important to predict those risky students over the class, so that the teachers would give them special guidance. It is potential to control student action using log-data in the programming class. By developing a logging system that can record application traces of source code editing and UNIX command, a dataset which obtained from our programming lesson's students from 39 students. Manufacturing evaluators with learning supervision are commonly used to predict a person's dropout potential. However, the data is difficult to understand, as the size given from our narrow dataset. Also, the features in the log-data depends on the elements from every class, such as if there is a lot of training or a plenty of explanations. As a result, we use an unattended learning method with outlier detection, assuming that students as part of an outlier cluster can be compared based on students' achievements, either superior or inferior students [16], [17]. The application of this k-means clustering technique is adjusted with Euclidean distance, in order to do clustering by using the dynamic time warping and benchmarking against active time behavior. Therefore, it is possible for us to compare the flow of activities to the exception of time-series deviations [18].

2.2. Dynamic time wrapping

Dynamic time warping (DTW) is an algorithm that used to measure the similarities between the two sequences with different lengths or amounts of data. DTW matches two sequences by calculating temporal information so that both of them can be aligned. Alignment is the smallest measured cumulative distance between two synced samples. If it then assumed that there are two sequential data, Q and C, with the range of n and m severally as shown in (1) and (2) [19], [20].

$$Q = q1, q2, \ldots, qi, \ldots, qn \quad (1)$$

$$C = c1, c2, \ldots, cj, \ldots, cm \quad (2)$$

Then, to align these two sequences using dynamic time warping, a matrix is formed m × n with matrix element (i,j) in the form of distance value $d(q_i,c_j)$ between two q_i points, and declared as $d(q_i,c_j) = (q_i - c_j)^2$. Each of matrix element (i,j) relates to align between q_i and c_j points. Warping path W is a group of adjoining matrix elements that define mapping between Q and C. The k element of W is formulated as $w_k = (i,j)_k$, so we got (3).

$$W = w_1, w_2, \ldots, w_k, \ldots, w_K \quad (3)$$

with: $max(m,n) \leq K < m + n - 1$.

While the path is defined as the cumulative distance D(i,j), that's distance d (qi,cj) for the elements added with the minimum cumulative distance from adjacent elements, as shown in (4).

$$D(i,j) = d(q_i,c_j) + min\{D(i-1,j-1),D(i-1,j),D(I,j-1)\} \quad (4)$$

Once obtained the optimal warping path, the distance or warping cost is calculated based on (5).

$$\mathrm{DTW}(Q,C) = \min\left\{\sqrt{\sum_{k=1}^{K} w_k}\right\} \quad (5)$$

2.3. CLG clustering

In this study, a proposed modification method in the clustering algorithm is complete linkage dissimilarity increment distribution-global cumulative score standard (CLG), this algorithm is a combined algorithm between the complete linkage (CL) algorithm [20], the dissimilarity increment distribution (DID) algorithm [20], global cumulative score standard (GCSS) algorithm [21]. The CLG algorithm works by combining elements of free graph-based parameters and model-based approaches (which are defined by combining criteria by characterizing clusters in probabilistic terms) for grouping.

$$\text{CL}= max\{D(C_k, C_i), D(C_k, C_j)\} \quad (6)$$

$$\text{DID}= pdissinc(w;\ \lambda) = \frac{\pi\beta^2}{4\lambda^2} w\ exp\left(-\frac{\pi\beta^2}{4\lambda^2} w^2\right) \quad (7)$$

$$+\frac{\pi^2\beta^3}{8\sqrt{2}\lambda^3} \text{X}\left(\frac{4\lambda^2}{\pi\beta^2} - w^2\right) exp\left(-\frac{\pi\beta^2}{8\lambda^2} w^2\right) erfc\left(\frac{\sqrt{\pi}\beta}{2\sqrt{2}\lambda} w\right)$$

$$\text{GCSS}=gcss_{th}(C_k,\ C_{i,}, C_j, Y_{MIN}) =$$
$$gcss_{th}(css_k, N_i, \gamma_i, \mu_i, \sigma_i, N_j, \gamma_j, \mu_j, \sigma_j, Y_{MIN}) =$$
$$css_k \Upsilon(N_i, N_j)\ \Psi_G(N_i, \gamma_i, \mu_i, \sigma_i, N_j, \gamma_j, \mu_j, \sigma_j, Y_{MIN}) \quad (8)$$

The CLG algorithm provides different treatment to small cluster candidate groups. Each candidate groups whose size is lower than YMIN is not required to explain the merging criteria. In fact, the merger between Ci and Cj always occurs in the case of the two groups of candidates less than the value of the YMIN object. Regarding the cluster size threshold, it is important to note the difference between the H and YMIN parameters; because both values refer to group size, parameter H is the real value used in the calculation of the dynamic merge threshold, while YMIN is the integer threshold value used when directing the comparison with the required cluster size.

2.4. k-means++

Algorithm in k-means often applied in clustering techniques that aim to minimizes the squared distance that has been leveled between points in the same cluster. But the algorithm of k-means algorithm has a disadvantage that cannot provide precise accuracy even using simple and fast calculations [22], [23]. If k-means added with randomized seeding technique will improve the accuracy from the algorithm of k-means. The accuracy from the algorithm of k-means heavily depends on a value of centroid (C) at the beginning of the calculation, then if using a different C value will give different result even if requires a lot of iterations to determine the member of a cluster if the value C inappropriate. By adding formulas randomized seeding technique, then it will determine the value of C at the beginning of the calculation. Each member has the opportunity become a centroid so the value of opportunities of each member is counted to found which one is the most appropriate. Here is a randomized seeding technique formula.

$$c_i = \frac{D(x)^2}{\sum_{x \in X} D(x)^2}$$

2.5. k-methods

The k-medoids algorithm is a classic partitioning technique of clustering that performs clustering dataset of n objects into k clusters, known as a priori. This algorithm operates on principle to minimize the amount of similarity between each object appropriate reference point. The k-medoids algorithm can be done as being as [24], [25]:

- In the first step, initialise the center of the cluster by k (the amount of clusters).
- In the second step, count each entity to a nearby cluster using Euclidian Distance size equations.
- The third step, after calculating the Euclidian Distance, initialize the center of the new cluster each object as a non-medoids candidate.
- The fourth step, measure the gap between each entity located on each cluster with non-applicant medoids.
- The fifth step, measure the total deviation (S) by processing the new total distance – the old total distance. If S < 0, then exchange entity with non medoids cluster data to form a new set of k objects as medoids.

– The sixth step, repeat steps 3-5 until no more changes to the medoid, then we are already got cluster members and their respective cluster members.

2.6. Experimentation dataset

UNIX command input history are used during exam due to programming class consist of 39 students. Assumed that the log-data can rate many aspects, such as motivation, individual skills, and others. Then we need to break the teaching signals from this log-data to create a new one binary linear classifier that separates a large of student based on their level.

Then we consider a way to easily classify about the group of students with unsupervised learning without firstly prepared a quantitative evaluation machine. The outlier class considered from one subset of this group. Accumulated time-series data of five-minutes UNIX commands input with k-medoid methods will integrate k-means++ for initial value definition. Hereafter, we inspect the trend of the clusters belonged then set the outlier cluster to every lesson from the evaluation.

3. RESULTS AND DISCUSSION

3.1. Feature vector verification

Figure 1 shows the command input ratio of each student's classes, where the picture given presents the executable files as "ls", "cd", and "gcc". Looking at the graph, can be seen that the ratio of command input used depending on the subjects. Therefore, it is not appropriate to be used as an input guide for the performance of each class.

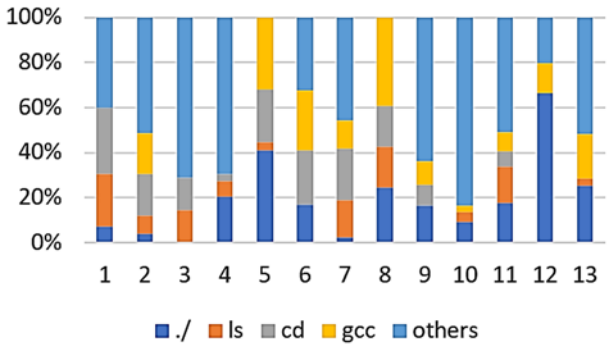

Figure 1. Evaluation from student grade E

3.2. Clustering methods for outlier detection

For instance, the result by grouping each other class can be seen in Figures 2 and 3. In this case, the number of clusters are arranged on a scale of 0 to 4, because at the end of the lesson, the number of inputs from the class medoid cluster is low. It is commonly known that in this case, clusters tend to simply classify the students based on the quantity of input. Except, for those users who suddenly experience an increasing the number of inputs will create their own cluster. Figure 2 suit this phenomenon. When the data located outside from the cluster less than 10% of the amount clusters, then the tendency of this dislocated cluster can be described as: i) less command input than other clusters as shown in Figure 3; and ii) the input increases rapidly in a short period of time as shown in Figure 2.

3.3. Outlier cluster interpretation

This is the characteristic of the students belonged to the outlying clusters using the clustering features such as the student's five-grade prediction (A to E). Figure 1 belong to a student whose evaluation is E and Figure 4 belong to a student who got prediction D as his index of achievement. These typical items probably suspect as supporter for solving the issues.

Figure 2 applied to a plenty of students, especially for students with evaluation grade A and C. While it is possible that the program may not work well. We detect a high number of command input entered during debugging work, or their task was finished earlier when they still proceeded the task during personal learnings.

Figure 3 belongs to a cluster of each class on one test. The evaluation is about the similarity of application techniques in students that are balanced even with low motivation. Because, in a test there are

two main problems, namely regarding on programming and writing, where most of the time drained by writing.

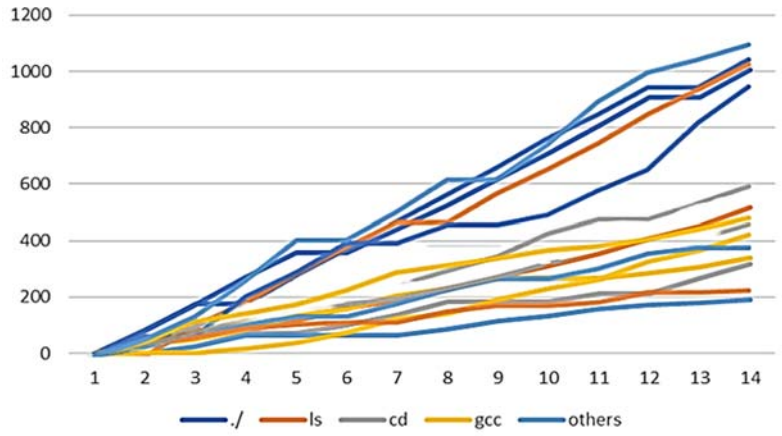

Figure 2. Output 1 from time-series clustering

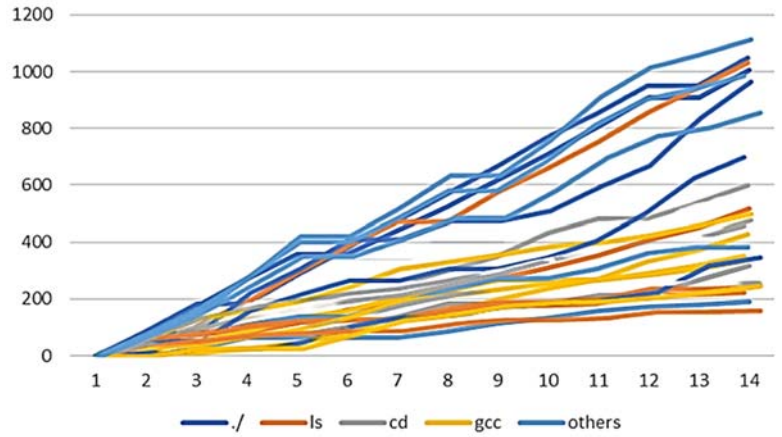

Figure 3. Output 2 from time-series clustering

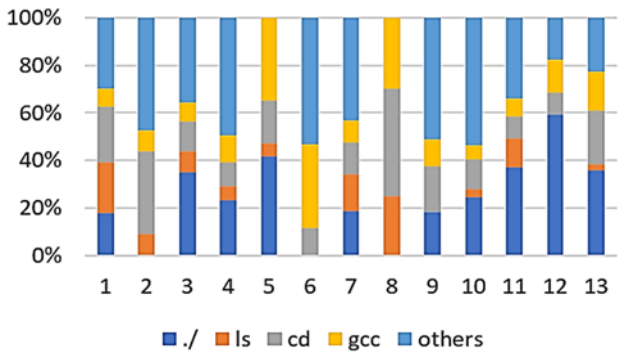

Figure 4. Evaluation from student grade D

3.4. Resolution concerning the amount of clusters

As far as has been predicted, the number of final inputs is more dominant in these methods than the way number of command input increase during the analysis. This is expected to be the basis in monitoring

and grouping these five groups in all analyses. It is just that, for the students in the Figure 3, there is a tendency that more students are in the group with less input. Clustering is unattended learning but looking at the resulting data there will be no right answer of the clusters is needed.

3.5. Transition regarding the number of clusters

Since discussed it earlier, the results of the grouping on Figure 3 are classified based on the number of command input during the classes, except when they experience with increases unexpectedly in a short period. In this study, we can said that Cluster 0 consists of the student who only have a few inputs, while Cluster 4 consists of the student who have a lot of inputs. Students can be predicted as a line of numbers. In this case, students on the left of the center point tend to have a few commands input, while those on the right will have many command inputs. The study focused on the detection of outliers applied to specify trends acknowledge the frequentative behaviors of many classes, then inspecting the trends of each lesson.

3.6. Attitude investigation by questionnaires

All of this time, we try to manage a questionnaire about to ensure how much the outlying cluster means. There are several considerations were made, two of them are likes: “How much did you understand the today’s content?”, and “How much did you paid attention for today’s lesson?” The format of the solution we created is self-evaluation and being organized into four point systems, as the students will get it once during the classes and exam. Then the result there will be no significant difference compared to the variation and outlier clusters since the scores from the answer either that low. For this reason, we cannot argued that our hypothesis incredibly valid, because another method of evaluation is still required. Overall, our suggest plays a major part as a visualizer of student motivation when unsupervised learning started.

3.7. UNIX command log manual verification

Manually, we investigated different type of student’s behavior during lessons using log-data from clusters outliers. The results of this investigation have been concluded: i) Students who increasingly pressed the keyboard in a period of time resulting “gcc” command then run the program. Some students may experience problems within compilation errors or program bugs based on our investigation. On the other hand, coding goes well without any problems. By looking at this phenomenon, we cannot classify the predicate of those students based on how much the number of input and duration while pressing the keyboard. ii) Students whose only pressed the keyboard a few times may not be able to complete the task and had a high risk being dropped out as we predicted earlier. iii) Students who suddenly experienced the increasing number of input while pressing keyboard and UNIX commands when pasted source code into the command line made this information powerless if only being investigated by the number of UNIX command issues. It is necessary to add information such as command values or implementation results.

4. CONCLUSION

This study proposes ways or methods to evaluate those students who are being risk of dropped out from school, by grouping them with unsupervised study using outlier detection. We use the data depend on the lesson’s purpose, makes it difficult while created by evaluation engine. For this reason, we investigated the group of outliers by divide them into three trends with a predictable cause, so that students who have learning problems can be detected as soon as possible. However, as our proposed prediction methods are still need further development, this research need another proper method such as visualization of student behavior based in log-data.

REFERENCES

[1] A. Triayudi and I. Fitri, “ALG Clustering to Analyze the Behavioral Patterns of Online Learning Student,” *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 16, pp. 5327-5337, 2018.

[2] A. Triayudi and I. Fitri, “A new agglomerative hierarchical clustering to model student activity in online learning,” *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 3, pp. 1226-1235, doi: 10.12928/telkomnika.v17i3.9425.

[3] A. Dutt, M. A. Ismail, and T. Herawan, “A Systematic Review on Educational Data Mining,” *IEEE Access*, vol. 5, pp. 15991-16005, 2017, doi: 10.1109/ACCESS.2017.2654247.

[4] A. Triayudi and I. Fitri, “Comparison of parameter-free agglomerative hierarchical clustering methods,” *ICIC Express Letters*, vol. 12, no. 10, pp. 973-980, 2018, doi: 10.24507/icicel.12.10.973.

[5] S. T. Ahmed, R. Al-Hamdani, and M. S. Croock, “Developed third iterative dichotomizer based on feature decisive values for educational data mining,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 1, pp. 209-217, 2020, doi: 10.11591/ijeecs.v18.i1.pp209-217.

[6] R. Asif, A. Merceron, S. A. Ali, and N. G. Haider, "Analyzing undergraduate students' performance using educational data mining," *Computers and Education*, vol. 113, pp. 177-194, 2017, doi: 10.1016/j.compedu.2017.05.007.
[7] A. Dowah, H. Al-Samarraie, and W. M. Fauzy, "Educational data mining and learning analytics for 21st century higher education: A review and synthesis," *Telematics and Informatics*, vol. 37, pp. 13-49, 2019, doi: 10.1016/j.tele.2019.01.007.
[8] M. W. Rodrigues, S. Isotani, and L. E. Zarate. "Educational Data Mining: A review of evaluation process in the e-learning," *Telematics and Informatics*, vol. 35, no. 6, pp. 1701-1717, 2018, doi: 10.1016/j.tele.2018.04.015.
[9] C. Romero and S. Ventura, "Educational data mining and learning analytics: An updated survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 3, Art. No. e1355, 2019, doi: 10.1002/widm.1355.
[10] B. Bakhshinategh, O. R. Zaiane, S. Elatia, and D. Ipperciel, "Educational data mining applications and tasks: A survey of the last 10 years," *Education and Information Technologies*, vol. 23, no. 1, pp. 537-553, 2018, doi: 10.1007/s10639-017-9616-z.
[11] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Systematic ensemble model selection approach for educational data mining," *Knowledge-Based Systems*, vol. 200, Art. No. 105992, 2020, doi: 10.1016/j.knosys.2020.105992.
[12] T. Devasia, Vinushree T P, and V. Hegde, "Prediction of students performance using Educational Data Mining," in *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*, 2016, pp. 91-95, doi: 10.1109/SAPIENCE.2016.7684167.
[13] E. Fernandes, M. Holanda, M. Victorino, V. Borges, R. Carvalho, and G. V. Erven, "Educational data mining: Predictive analysis of academic performance of public school students in the capital of Brazil," *Journal of Business Research*, vol. 94, pp. 335-343, 2019, doi: 10.1016/j.jbusres.2018.02.012.
[14] R. Ahuja, A. Jha, R. Maurya, and R. Srivastava, "Analysis of educational data mining," in *Harmony Search and Nature Inspired Optimization Algorithms*, pp. 897-907, 2019, doi: 10.1007/978-981-13-0761-4_85.
[15] E. B. Costa, B. Fonseca, M. A. Santana, F. F. De Araujo, and J. Rego, "Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses," *Computers in Human Behavior*, vol. 73, pp. 247-256, 2017, doi: 10.1016/j.chb.2017.01.047.
[16] C. S. Silva and J. M. Fonseca, "Educational Data Mining: a literature review," in *Europe and MENA Cooperation Advances in Information and Communication Technologies*, 2017, pp. 87-94, doi: 10.1007/978-3-319-46568-5_9.
[17] Ryan Baker, "Challenges for the future of educational data mining: The Baker learning analytics prizes," *JEDM*| *Journal of Educational Data Mining*, vol. 11, no. 1, pp. 1-17, 2019.
[18] R. M. Awangga, S. F. Pane, K. Tunnisa, and I. Supriana, "K means clustering and meanshift analysis for grouping the data of coal term in Puslitbang Tekmira," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 16, no. 3, pp. 1351-1357, 2018, doi: 10.12928/telkomnika.v16i3.8910.
[19] S. Slater, S. Joksimović, V. Kovanovic, R. S. Baker, and D. Gasevic, "Tools for educational data mining: A review," *Journal of Educational and Behavioral Statistics*, vol. 42, no. 1, pp. 85-106. 2017, doi: 10.3102/1076998616666808.
[20] A. Triayudi, O.W. Widyarto, and V. Rosalina, "CLG Clustering for Mapping Pattern Analysis of Student Academic Achievement." *ICIC Express Letters,* vol. 14, no. 12, pp. 1225-1234, 2020.
[21] Cobo Rodríguez, "Parameter-free agglomerative hierarchical clustering to model learners' activity in online discussion forums," Doctoral programme, Universitat Oberta de Catalunya, 2014.
[22] K. Xing, C. Hu, J. Yu, X. Cheng, and F. Zhang, "Mutual Privacy Preserving k -Means Clustering in Social Participatory Sensing," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2066-2076, Aug. 2017, doi: 10.1109/TII.2017.2695487.
[23] I. Wahyudin, T. Djatna, and W. A. Kusuma, "Cluster analysis for SME risk analysis documents based on Pillar K-Means," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 14, no. 2, pp. 674-683, 2016, doi: 10.12928/telkomnika.v14i1.2385.
[24] G. Gan and M. Kwok Po Ng, "K-means clustering with outlier removal," *Pattern Recognition Letters*, vol. 90, pp. 8-14, 2017, doi: 10.1016/j.patrec.2017.03.008.
[25] D. P. Sari, D. Rosadi, A. R. Effendie, and Danardono, "K-means and Bayesian networks to determine building damage levels," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 2, pp. 719-727. 2019, doi: 10.12928/telkomnika.v17i2.11756.