

Integrating k-means clustering into automatic programming assessment tool for student performance analysis

Rina Harimurti¹, Ekohariadi², Munoto³, I. G. P Asto Buditjahjanto⁴

¹Department of Electrical and Engineering, Universitas Negeri Surabaya, Indonesia

^{2,3,4}Post Graduate of Vocational Education, Universitas Negeri Surabaya, Indonesia

Article Info

Article history:

Received Mar 8, 2020

Revised Feb 25, 2021

Accepted Mar 1, 2021

Keywords:

Automatic programming

Assessment

K-means clustering

Programming

ABSTRACT

Computer programming is a subject involving a large number of logic programming activities. A programmer is compulsory to master skills of algorithms, logic, and programming language to conduct programming. An automatic programming assessment tool is an automated tool used to assist instructors in assessing programming tasks. The technology used in this application is open-source based with an evaluation module that will evaluate the sent program code, assessment, and classification. The evaluation results were then processed in the assessment module, where a comparison process with the test case was performed along with the point calculation. The classification module was used to divide students into five groups based on the point of each practicum. This study used k-means clustering classification method. The entities included were lecturers, assistants, students, and compilers. This application had 2 levels of users namely admin and students. Scoring results were then used in the process of determining the classification of student's performance based on the k-means clustering method. In connection with the classification test results with three iterations, three practicum scores resulted that the classification process was successfully carried out with student's performance divided into five groups covering very good, good, sufficient, less, and very less. The data used in the clustering process consisted of 41 students with 10 attributes which were then grouped into 3 groups (clusters).

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

I.G.P Asto Buditjahjanto

Department of Electrical and Engineering

Universitas Negeri Surabaya

Lidah Wetan St., Lidah Wetan, Lakarsantri, Kota Surabaya, Jawa Timur 60213, Indonesia

Email: asto@unesa.ac.id

1. INTRODUCTION

Programming is the main competency that must be mastered because it is a basic concept in computer science. However, many people find it difficult to learn computer programming languages. The concept of programming is important in learning programming even starting from the introductory level which is commonly started at the age of 14 years [1], [2]. Computer programming is a subject that involves a large number of logic programming activities. Programming refers to writing code from the initial process as well as modifying and debugging code. In particular, the last two activities also include understanding the code as the main activity. Although other essential life cycle activities such as analysis, design, testing, and deployment are important, programming is the dominant activity of software development [3]. Programming is more than just coding, because it exposes students to computational thinking which involves problem-solving activities using computer science concepts such as abstraction and decomposition [4]. Students and

instructors are served with several problems during the learning process, either from the concept of programming or language syntax, in which they can be obstacles for students to learn programming and become inhibiting factors of their motivation [5].

The purpose of programming is to load a program that can do a calculation or work in accordance with the programmer's wishes. Programming requires skills of algorithms, logic, and programming languages, including but not limited to mathematics. These skills are mandatory to be mastered by those who want to be successful in programming. To improve one's understanding of the programming concept, the use of visual programming language (VPL) is applied [6]. Most students learn programming by reading books or listening to teacher explanations so that the results are not optimal because programming is a set of skills that requires a lot of practice. Actually, programming can be obtained from formal and non-formal education [7], [8]. It is best learned by practice and if students want to learn effectively, some practices must be self-directed. Instructor's key role is to direct students to do certain processes or activities and to motivate them [9]. The programming method starts by turning the problem into an algorithm and translating the algorithm into the program code. As a result, students are required to have abilities in all processes: analyzing problems, designing algorithms, translating algorithms into program code, and writing program code with the correct syntax [10].

A study conducted by [11] indicated that the times spent by students on coding was the factor that potentially had significant impact on the exam results. A student's performance is associated with their programming experience and self-confidence in programming [12]. In the field of education, the existence of learning devices is very important to help students and teachers [13]. Likewise, through practical activities students understand the basic concepts of programming and problem solving and programming skills. Learning of basic practical programming is applied to enhance students' programming skills with lots of hands-on program writing practice. Actually, in-classroom practicum activities are not enough for students to practice more complex programming [14]. As a result, many practical assignments ended up becoming homework for them and the results had to be disclosed to the teacher. The teacher then provides some questions and feedback on the student's work. This process is an integral part of classroom learning so it takes a long time [12]. It took a long time when it came to assessing the results of student practicum.

A study conducted by [15] portrayed that teachers often had difficulty in validating students' programming tasks. So that, a framework is created to ease the task of lecturer called an automated assistance system for correcting programming exercises (SAC) using a web platform. This application allows teachers to determine and describe assignments/training programs while students just need to upload the results of their work. In a relatively short time, students can see feedbacks from the answers. Another advantage that can be obtained from the use of automatic assessment tools is that human factors no longer significantly involved [16]. When a program is assessed manually, it is usually possible that the influence of subjectivity will be influenced by giving excessive assessments or awards to certain students. However, when the use of automatic assessment is applied this will not happen anymore.

An automatic programming assessment tool is an automated tool used to assist instructors in assessing programming tasks [17]. It is important to note that Automatic Programming Assessment Tools are good examples for intelligent systems that would rather promote human-machine coexistence in the era of artificial intelligence driven automation [18]. Grouping students according to their level of competence will make teachers easier to monitor the each student's ability. This grouping assignment will be more directed in accordance with the target [19], [20]. Data grouping is the process of extracting patterns that were previously unknown, valid, useful, and hidden from large data sets. The main goal is to divide students into homogeneous groups in which this technique is most widely used in mapping [21]-[23]. Educational data mining (EDM), based on tasks, is divided by many categories, namely: classification [24], [25], clustering [26], association analysis [27] This application helps teachers and students to improve programming skills. This study uses cluster analysis to divide students into groups according to their characteristics [28].

2. RESEARCH METHOD

The technology used in this application is based on open source, namely Ubuntu, Linux, MySQL, Laravel and Nginx. Compile and evaluate programs with GNU compiler collection (GCC) and G++ installed on the operating system. The main modules in the application were evaluation, assessment, and classification. The evaluation module conducted an in-depth evaluation of the program code sent and became a determining factor for further processing. The results of the process in the evaluation module were then processed in the assessment module, where a comparison process with the testcase was performed as well as the point calculation. The classification module was used to divide students into five groups based on the point of each practicum. The classification was conducted by using k-means clustering method.

The application was developed with model, view, controller (MVC) so that there was a separation of functions and program structures that referred to the laravel framework structure. The model was responsible for organizing, preparing, manipulating and organizing data (from the database) according to the instructions given by the controller. View was to present information (which was easy to understand) to the user according to the instructions given by the controller. The controller was responsible for managing what the model had to do, and which views should be displayed based on requests from the user.

Testing the evaluation and assessment process was performed by sending the program code in C ++ language with several different cases and scenarios. From each test, analysis based on the output produced by the application was carried out. The trial results showed that, regarding the process of evaluating program code, applications could properly conduct in-depth evaluations of program code. This was evidenced by the appearance of messages according to the evaluation results including size limit exceeded, compilation error, memory limit exceeded, wrong answer, internal error, and runtime error. If there was no error during the compilation process, an accepted (AC) message would be displayed. In regard to the algorithm, Figure 1 shows the evaluation process of the sent program code.

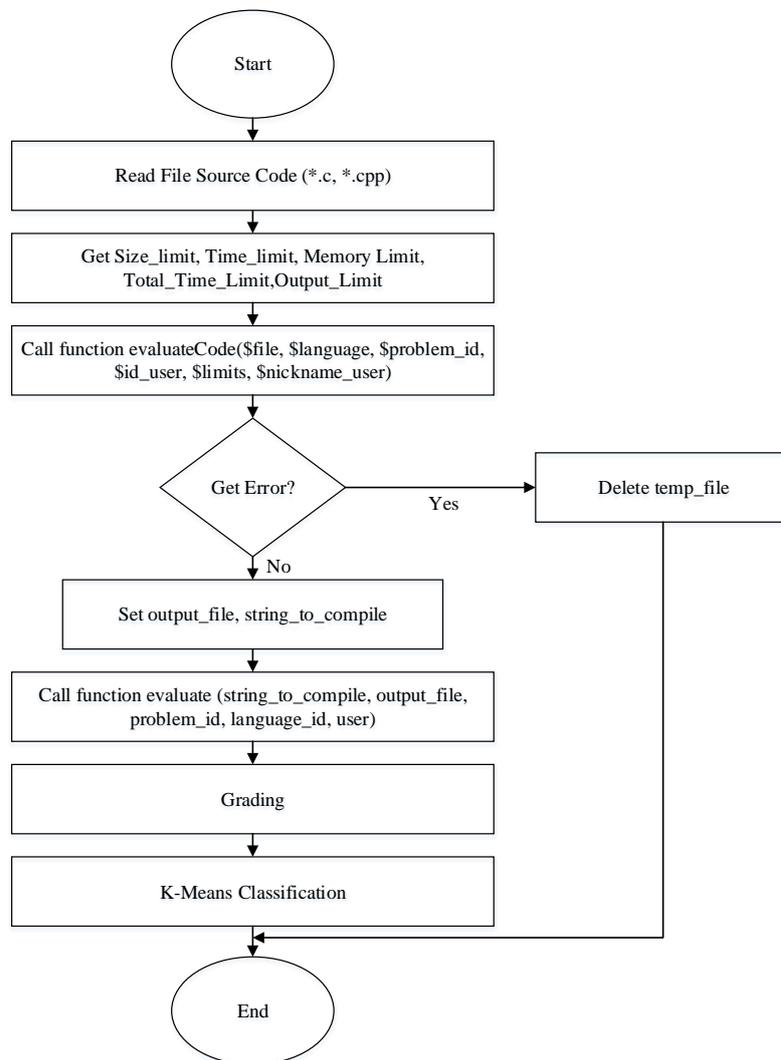


Figure 1. Flowchart automatic assessment

3. RESULTS AND ANALYSIS

3.1. Business proses

Implementation of business processes was described in data flow diagrams (DFD). Entities involved covered lecturers, assistants, students, and compilers. Figure 2 conveys DFD level 1. The main processes in DFD level 1 included input data, problem setting, and uploading program code.

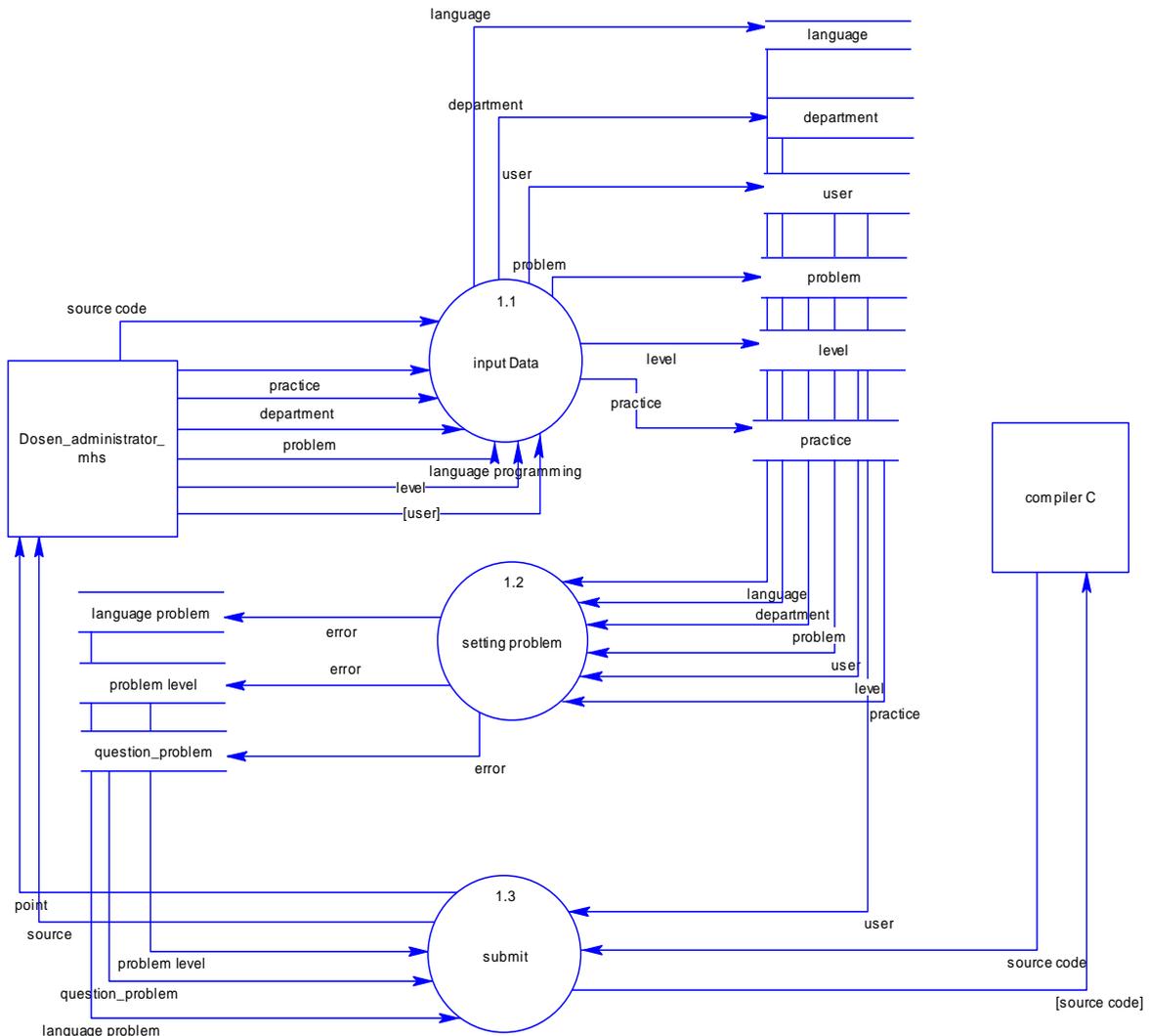


Figure 2. DFD level 1

3.2. Implementation

This application had 2 levels of users namely admin and students. Students must first register then logged in. Afterwards, they continued working on the questions. The task list could be selected in Figure 3 and they could work directly on the application in Figure 4. When working on assignments, their choice can be appropriate based on the problem group in Figure 5 the difficulty level in Figure 6.

3.3. K-means clustering

Scoring results were then used in the process of determining the classification of student’s performance based on the k-means clustering method. In connection with the classification test results with three iterations, three practicum scores resulted that the classification process was successfully carried out with dividing student’s performance into five groups namely very good, good, sufficient, less, and very less. Based on the results of trials that had been conducted, it was concluded that the application function had been functioning properly and could be used in the programming practicum process. All existing features had been tested and run well according to existing scenarios and data. The main features namely evaluation, assessment, and classification worked well. Bugs was still found in the application that displayed the execution time. The results of classification testing using k-means could be seen in Figure 7.

Figure 7 displays the results of the classification process using the k-means clustering algorithm with 3 iterations based on the data of practicum scores. The result showed that students were classified as very good, good, sufficient, less, and very less. Based on the results of the classification, it could be seen that the group of students was based on the practicum score that had been carried out.

Data grouping aimed to determine student’s performance in using Automatic Programming Assessment Tool application, data grouping (clustering) using the k-means algorithm. In this study, the numbers of clusters were 3. The next step was to determine the chosen centroid randomly with the range of data between the lowest to the highest scores. The next process was determining the distance of each data with a centroid using Euclidean Distance algorithm. The final step was to group data based on the level of similarity. This process was repeated until no data transferred between groups.

ID	TITLE	SCORE	ACCURACY	SOLVED	SUBMISSIONS
1	A +B Problem	50	44.39 %	83	187
2	Problema 2	50	0.00 %	0	50
8	A Very Big Sum	30	50.97 %	79	155
9	Diagonal Difference	30	37.40 %	49	131
10	Plus Minus	30	1.06 %	1	94

Figure 3. List of assignment

A +B Problem

Problem Limits

Time Limit Per Case	Total Time Limit	Memory Limit	Source Limit
200 Ms	200 Ms	200 Mb	2000 Kb

Problem Description

Description
For this problem you must calculate $A + B$, numbers given in the input.

Input Specification
The only line of input contain two space separated integers A, B ($0 \leq A, B \leq 10$).

Output Specification
The only line of output should contain one integer: the sum of A and B .

Sample Input
1 2

Sample Output
3

Hints
no se weno si se pero no te va decir

Figure 4. Problems description

Problems

Title or ID

All tags

ID	TITLE	SCORE	ACCURACY	SOLVED	SUBMISSIONS
1	A +B Problem	50	44.39 %	83	187
2	Problema 2	50	0.00 %	0	50
8	A Very Big Sum	30	50.97 %	79	155
9	Diagonal Difference	30	37.40 %	49	131
10	Plus Minus	30	1.06 %	1	94

- All tags
- Ad-hoc (AH)
- Arithmetic-Algebra (AA)
- Brute Force (BF)
- Combination (CO)
- Data Structures (DS)
- Dynamic Programming (DP)
- Game Theory (GA)
- General Geometry (GE)
- Graph Theory (GT)

Figure 5. Group of problem

Problems

Title or ID

All tags

All levels

ID	TITLE	SCORE	ACCURACY	SOLVED	SUBMISSIONS
1	A +B Problem	50	44.39 %	83	187
2	Problema 2	50	0.00 %	0	50
8	A Very Big Sum	30	50.97 %	79	155
9	Diagonal Difference	30	37.40 %	49	131
10	Plus Minus	30	1.06 %	1	94

- All levels
- Very Easy
- Easy
- Medium
- Hard
- Very Hard

Figure 6. Problem stage

K-means										
Proses Data Rata-Rata						Proses Data akhir				
Centroid 1	Sangat Baik					96				
Centroid 2	Baik					76				
Centroid 3	Cukup					66				
Centroid 4	Kurang					53				
Centroid 5	Kurang Sekali					14				

ID	Nama	Nilai	P1	P2	P3	Rata-Rata	Distance				Predikat	
32	M. FAHRUR ROZI	160	30	50	80	53	43	23	13	0	39	Kurang
33	MAULIDA NURSANTI	160	20	80	60	53	43	23	13	0	39	Kurang
34	INTAN PRASTIKA	160	30	70	60	53	43	23	13	0	39	Kurang
35	KURNITA SARI	160	40	40	80	53	43	23	13	0	39	Kurang
36	LILING APRISTIANA	160	20	60	80	53	43	23	13	0	39	Kurang
37	IIS TIANAH	163	40	50	73	54	42	22	12	1	40	Kurang
38	ARIFATUS SHOLEHA	163	50	40	73	54	42	22	12	1	40	Kurang

Figure 7. K-means clustering classification test results

4. CONCLUSION

The data used in the clustering process consisted of 41 students with 10 attributes which were then grouped into 3 groups or clusters. Based on the existing data, the process of determining centroid was chosen randomly with the range of data between the lowest to the highest scores based on the number of clusters specified, in this case 3 clusters. Then the distance of the data to the centroid. Euclidian distance algorithm was used to calculate the distance of data by centroid. After the data distance to the centroid was determined, the next step was to compare the distance to the centroid from each cluster. The new centroid point was then determined by calculating the average score of data located in the same centroid. The new cluster center was used to do the next iteration once the results obtained were not yet convergent. The iteration process would stop if it met the maximum iteration entered by the user or the results achieved were convergent (the new cluster center was the same as the previous cluster center).

REFERENCES

- [1] A. Jimoyiannis, "Teaching of programming and algorithmic problem solving in senior high school. In Educational Material from "Informatics Teachers' Education" Project. CTI," *Encyclopedia of the Sciences of Learning*, pp. 2845–2845, 2008.
- [2] S. Markantonatos, C. Panagiotakopoulos, and V. Verykios, "Design, Implementation, and Evaluation of an Educational Software for the Teaching of the Programming Variable Concept," *Res. e-Learning ICT Educ.*, pp. 315–322, 2018, doi: 10.1007/978-3-319-95059-4_19.
- [3] G. R. Bergersen, D. I. K. Sjøberg, and T. Dyba, "Construction and validation of an instrument for measuring programming skill," *IEEE Transactions on Software Engineering*, vol. 40, no. 12, pp. 1163–1184, Dec. 2014, doi: 10.1109/TSE.2014.2348997.
- [4] S. Y. Lye and J. H. L. Koh, "Review on teaching and learning of computational thinking through programming: What is next for K-12?," *Comput. Human Behav.*, vol. 41, pp. 51–61, 2014, doi: 10.1016/j.chb.2014.09.012.
- [5] D. Topalli and N. E. Cagiltay, "Improving programming skills in engineering education through problem-based game projects with Scratch," *Comput. Educ.*, vol. 120, pp. 64–74, 2018, doi: 10.1016/j.compedu.2018.01.011.
- [6] C. Y. Tsai, "Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy," *Computers in Human Behavior*, vol. 95, pp. 224–232, Jun. 2019, doi: 10.1016/j.chb.2018.11.038.
- [7] P. Tuomi, J. Multisilta, P. Saarikoski, and J. Suominen, "Coding skills as a success factor for a society," *Educ. Inf. Technol.*, vol. 23, no. 1, pp. 419–434, 2018, doi: 10.1007/s10639-017-9611-4.
- [8] S. Sentance and A. Csizmadia, "Computing in the curriculum: Challenges and strategies from a teacher's perspective," *Educ. Inf. Technol.*, vol. 22, no. 2, pp. 469–495, 2017, doi: 10.1007/s10639-016-9482-0.
- [9] T. Jenkins, "The motivation of students of programming," *Proc. Conf. Integr. Technol. into Comput. Sci. Educ. ITiCSE*, pp. 53–56, 2001, doi: 10.1145/507758.377472.
- [10] A. McDougall and M. Boyle, "Student Strategies for Learning Computer Programming: Implications for Pedagogy in Informatics," *Education and Information Technologies*, vol. 9, no. 2, pp. 109–116, Jun. 2004, doi: 10.1023/B:EAIT.0000027924.69726.b5.
- [11] M. Satratzemi, S. Xinogalos, D. Tsompanoudi, and L. Karamitopoulos, "Examining Student Performance and Attitudes on Distributed Pair Programming," *Sci. Program.*, vol. 2018, 2018, doi: 10.1155/2018/6523538.
- [12] M. Thuné and A. Eckerdal, "Analysis of Students' learning of computer programming in a computer laboratory context," *Eur. J. Eng. Educ.*, vol. 44, no. 5, pp. 769–786, 2019, doi: 10.1080/03043797.2018.1544609.
- [13] I. A. Kautsar and R. Sarno, "A supportive tool for project based learning and laboratory based education," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 2, pp. 630–639, 2019, doi:10.18517/ijaseit.9.2.7067.

- [14] J. C. Rodríguez-del-Pino, E. Rubio-Royo, and Z. Hernández-Figueroa, "A Virtual Programming Lab for Moodle with automatic assessment and anti-plagiarism features," *Conf. e-Learning, e-Business, Entrep. Inf. Syst. e-Government*, 2012.
- [15] B. Auffarth, M. Lopez-Sanchez, Jordi Campos i Miralles, and Anna Puig, "System for Automated Assistance in Correction of Programming Exercises," *V Int. Congr. Univ. Teach. Innov.*, 2008.
- [16] S. Gupta, "Automatic assessment of programming assignment," *Comput. Sci. Eng. An Int. J.*, vol. 2, no. 1, pp. 67–74, 2012, doi: 10.5121/csit.2012.2129.
- [17] R. Saikkonen, L. Malmi, and A. Korhonen, "Fully automatic assessment of programming exercises," in *ACM SIGCSE Bulletin*, vol. 33, no. 3, pp. 133–136, 2004, doi: 10.1145/507758.377666.
- [18] O. H. Hamid, N. L. Smith, and A. Barzanji, "Automation, per se, is not job elimination: How artificial intelligence forwards cooperative human-machine coexistence," *Proc.-2017 IEEE 15th Int. Conf. Ind. Informatics, INDIN 2017*, pp. 899-904, 2017, doi:10.1109/INDIN.2017.8104891 .
- [19] S. Gulwani, I. Radiček, and F. Zuleger, "Automated clustering and program repair for introductory programming assignments," *Proc. ACM SIGPLAN Conf. Program. Lang. Des. Implement.*, pp. 465-480, 2018, doi: 10.1145/3296979.3192387
- [20] K. Danutama and I. Liem, "Scalable Autograder and LMS Integration," *Procedia Technology*, vol. 11, pp. 388–395, 2013, doi: 10.1016/j.protcy.2013.12.207.
- [21] L. Huang, "Teaching management data clustering analysis and implementation on ideological and political education of college students," *Proc.-2016 Int. Conf. Smart Grid Electr. Autom. ICSGEA 2016*, pp. 308–311, 2016, doi: 10.1109/ICSGEA.2016.61.
- [22] Y. Pang, F. Xiao, H. Wang, and X. Xue, "A clustering-based grouping model for enhancing collaborative learning," *Proc.-2014 13th Int. Conf. Mach. Learn. Appl. ICMLA 2014*, pp. 562–567, 2014, doi: 10.1109/ICMLA.2014.94.
- [23] Harwati, A. P. Alfiani, and F. A. Wulandari, "Mapping Student's Performance Based on Data Mining Approach (A Case Study)," *Agric. Agric. Sci. Procedia*, vol. 3, pp. 173–177, 2015, doi:10.1016/j.aaspro.2015.01.034.
- [24] V. M. and M. Nagar, "Classification based data mining algorithms to predict slow, average and fast learners in educational system using WEKA," *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, 2017, pp. 475-479, doi: 10.1109/ICCMC.2017.8282735 .
- [25] V. Shanmugarajeshwari and R. Lawrance, "Analysis of students' performance evaluation using classification techniques," *2016 International Conference on Computing Technologies and Intelligent Data Engineering, ICCTIDE 2016*. 2016, doi: 10.1109/ICCTIDE.2016.7725375 .
- [26] Yeonjeong Park, Ji Hyun Yu, and Il-Hyun Jo, "Clustering blended learning courses by online behavior data: a case study in a Korean higher education institute," *Internet and Higher Education*, vol. 29, pp. 1–11, 2016, doi: 10.1016/j.iheduc.2015.11.001.
- [27] E. García, C. Romero, S. Ventura, and C. De Castro, "A collaborative educational association rule mining tool," *Internet and Higher Education*, vol. 14, no. 2, pp. 77–88, 2011, doi: 10.1016/j.iheduc.2010.07.006.
- [28] Shashikant Pradip Borgavakar and Mr. Amit Shrivastava, "Evaluating Student's Performance using k-means Clustering," *International Journal of Engineering Research and*, vol. 6, no. 05, 2017.