IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 6, December 2026, pp. 5289~5298
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i6.pp5289-5298

Optimizing k-means centroid initialization using genetic algorithms for student activity clustering

Abdul Rahman[1,2], Sri Mulyana[1], Agus Sihabuddin[1], Aina Musdholifah[1]
[1]Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia
[2]Department of Computer Science, Faculty of Engineering and Computer Science, Universitas Baturaja, Baturaja, Indonesia

Article Info

Article history:

Received Sep 9, 2025
Revised Aug 13, 2026
Accepted Sep 20, 2026

Keywords:

Centroid initialization
Clustering
Genetic algorithm
K-means++
Student activity data

ABSTRACT

K-means is one of the most widely used clustering algorithms due to its computational efficiency and simplicity. However, its performance depends heavily on the selection of initial centroids, which can lead to inconsistent or suboptimal outcomes. To address this limitation, k-means++ was introduced with a probabilistic initialization strategy. Although it improves stability compared to random initialization, k-means++ still encounters challenges when applied to datasets with complex or non-linear structures. This study proposes integrating a genetic algorithm (GA) with k-means to optimize centroid initialization before clustering. GA applies selection, crossover, and mutation processes to explore the solution space and identify near-optimal centroids. The performance of GA k-means was evaluated using fitness scores and silhouette scores, and compared with k-means++ and random initialization. Experiments were conducted on student online learning activity data, including module access, discussion participation, and other interactions. Results show that GA consistently produces higher fitness values, with the best score of 0.8948 at two clusters, reflecting stable centroid optimization. Meanwhile, k-means++ achieved the highest silhouette score of 0.6144 at three clusters, indicating the most coherent structure. These results demonstrate that combining GA with k-means offers a robust solution for clustering educational data and similar applications.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Sri Mulyana
Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada
Building C, 4th Floor, North Sekip, Bulaksumur, Yogyakarta 55281, Indonesia
Email: smulyana@ugm.ac.id

1. INTRODUCTION

Clustering is a data analysis technique that groups objects based on similarity. K-means is one of the most widely used algorithms due to its simplicity and efficiency, yet it suffers from high sensitivity to the selection of initial centroids, which can lead to inconsistent results. This issue becomes more critical when applied to large and complex datasets, such as student behavior data [1], [2]. To address this, this study proposes the use of a genetic algorithm (GA) to optimize centroid initialization. GA, a global search algorithm inspired by natural selection, has been widely applied to complex optimization problems [3]–[5]. The proposed hybrid approach aims to improve clustering quality by reducing dependence on random initialization [6]. The main motivation is to enhance the accuracy and consistency of clustering on student behavior data, which involves multiple variables and requires robust methods. Better

clustering quality is expected to support educational institutions in identifying behavior patterns related to academic performance [4].

The methodology includes: i) encoding centroids as chromosomes in GA, ii) defining a fitness function based on within-cluster squared distances, iii) applying selection, crossover, and mutation, and iv) using the optimized centroids as the initial centers for k-means. The dataset includes attributes such as modules accessed, discussions viewed/created, assignments submitted, and study time [7]. Unlike prior studies that relied on suboptimal initialization and lacked a thorough evaluation of consistency [4], [8]–[11], this research contributes a GA k-means hybrid method that provides more stable and consistent results. Findings show that optimizing centroid initialization with GA significantly improves k-means performance, producing more accurate and consistent clustering [12], [13]. Furthermore, an indexing mechanism is applied after clustering to accelerate the classification of new data [14]–[16], enabling faster and more efficient data-to-cluster matching [17].

The main contribution of this research lies in the development of a k-means centroid initialization method utilizing GA, where cluster distributions are randomized for each individual in the population. After obtaining the best fitness value through the iterative GA process, the resulting optimal centroids are used as the initial points for the k-means algorithm. This approach significantly improves clustering quality while also accelerating algorithm convergence.

2. METHOD

The proposed model integrates GA with k-means to optimize cluster centroid initialization. The objective is to reduce k-means' sensitivity to random initialization [18], [19], which often leads to inconsistent and suboptimal clustering, especially on large or complex datasets [20], [21]. By employing GA, the initial centroids are optimized before running k-means, thereby improving accuracy and stability [14], [22]. After clustering, an indexing step is applied to the final centroids to facilitate faster assignment of new data points. The overall flow of the proposed method is outlined in Algorithm 1.

Algorithm 1. GA k-means with centroid indexing

Input: D, K_{min}, K_{max}, PopSize, t, p_c, stop condition
Output: k^*; C^*; I^*
Initialize $P = \{p_i\}_{i=1}^{\text{PopSize}}$ with random partitions.
Compute fitness $S(p_i)$ for all $p_i \in P$ using silhouette coefficient.
while *not stop* do
    $P^* \leftarrow \emptyset$
    while $|P^*| <$ PopSize do
        $(p_1, p_2) \sim$ Tournament (P, t)
        $(c_1, c_2) =$ Mutate $(\text{Crossover}(p_1, p_2, p_c), p_m)$
        $P^* \leftarrow P^* \cup \left\{\arg\max\{S(p_j), S(c_j)\}, j \in \{1,2\}\right\}$
    $P \leftarrow P^*$
$(k^*, C^*) = \arg\max_{p \in P} S(p)$
Run k-means with (k^*, C^*) as initial centroids $\rightarrow C_{\text{final}}$
Define index set $I^* = \{\text{Centroid}_0, \ldots, \text{Centroid}_{k^*-1}\}$
For $x \in D_{new}$: assign $x \mapsto \arg\min_{c \in C_{final}} \|x - c\|$ and record the index from I^*

In the initial stage, the GA k-means with centroid indexing algorithm begins by initializing a population P, consisting of a set of random partitions of the dataset D, with the number of individuals determined by the parameter PopSize. Each individual in this population is evaluated using fitness based on the silhouette coefficient, where the quality of clustering is measured by comparing the average intra-cluster distance and the average distance to the nearest cluster. The population evolves through iterative processes as long as the stop condition is not met. In each generation, a new population $P*$ is created, filled with individuals resulting from tournament selection (with tournament size t), followed by crossover (with probability pc) and mutation (with probability pm). For every selected pair of individuals, two offspring are produced, and only the one with the higher fitness is included in the new population. Once $P*$ is full, the main population is updated, and this process repeats.

After the evolutionary process ends, the best solution is selected-namely, the individual with the highest fitness value, which determines the optimal number of clusters ($k*$) and the initial centroid positions ($C*$). These values are then used to initialize the k-means process, resulting in the final centroids (Cfinal). All centroids obtained from the clustering are then indexed (e.g., Centroid0, .., Centroidk*-1), so that each new data point x can be immediately assigned to the nearest centroid based on the minimum distance to a member of C final. Thus, although there is no explicit fitness function notation in the pseudocode, the

evaluation of solution quality is clearly represented by the silhouette value for each partition during the evolutionary selection process.

2.1. Genetic algorithm k-means with centroid indexing

The GA k-means with centroid indexing framework integrates GA and k-means to optimize cluster initialization and stability [23], [24]. A population of candidate partitions is initialized within $K_{\min} - K_{\max}$ and evaluated by the silhouette score. Through tournament selection, crossover, mutation, and elitist replacement, the best individual k^*, C^* is evolved and used to initialize k-means, reducing sensitivity to random centroids. Finally, centroid indexing I^* assigns explicit labels to C_{final}, ensuring consistent cluster identification and efficient assignment of new data points.

2.2. Dataset description

The student activity dataset employed in this research captures a wide range of behavioral indicators that reflect students' engagement in the learning management system [25]. Each record in the dataset contains quantitative measures of several core activities, such as accessing course modules, participating in online discussions, viewing and uploading assignments, the total number of platform interactions (hits), and the total time spent in the system. These attributes were chosen as they provide a comprehensive overview of students' learning patterns and are commonly used in learning analytics studies to assess student engagement and predict learning outcomes. The full list of attributes and their descriptions is presented in Table 1.

Table 1. Student activity data

Attribute	Description	Attribute initials
module_viewed	Number of material modules viewed	A1
discussion_viewed	Number of discussion forums viewed	A2
assignment_viewed	Number of tasks viewed	A3
assignment_upload	Number of tasks uploaded	A4
sum_click	Total clicks	A5
n_time	Number of active hours accessing	A6

3. RESULTS AND DISCUSSION

The proposed GA k-means with centroid indexing algorithm effectively optimizes centroid initialization in the clustering of student activity data. The use of the silhouette score as an evaluation metric enables GA to enhance centroid selection, while the implementation of indexing accelerates data retrieval and reduces computational complexity, resulting in improved clustering quality and scalability. The dataset used as a test case for the GA k-means model is presented in Table 2. This section reports the experimental results of GA k-means in student behavior analysis, covering fitness evaluation with silhouette score, evolutionary updates, and centroid computation. The method effectively identifies the optimal number of clusters as indicated by the highest silhouette value.

Table 2. Dataset student activity

Data	A1	A2	A3	A4	A5	A6	Category
x1	54	24	23	1	..	108	Low
x2							Medium
..	..	..	..	..	..	..	..
x69	330	71	116	5	..	233	Medium
x70	311	64	137	7	..	119	Medium
..	..	..	..	..	..	..	..
x197	294	64	144	7	..	214	Low
x198	316	79	145	6	..	152	Low

3.1. Population initialization

The GA was initialized with 20 individuals in the range of 2–6 clusters, using a crossover probability of 0.8, a mutation probability of 0.3, and a tournament size of 3 over 50 generations. The best clustering quality was achieved by GA k-means, which produced a higher silhouette score and more consistent centroids compared to k-means elbow. These findings confirm that centroid initialization using a GA enhances both the robustness and stability of clustering, as demonstrated in Table 3.

Table 3. Sample of population initialization (10 data points out of 198)

Individual	Number of clusters	Cluster labels (x1-x198)
1	2	[1, 1, 0, 0, 1, 0, 1, 0, 0, 1, .., n]
2	3	[1, 2, 1, 0, 2, 1, 0, 1, 2, 1, .., n]
..	..	..
9	3	[1, 2, 0, 1, 2, 0, 1, 2, 0, 1, .., n]
10	4	[2, 1, 3, 0, 2, 1, 0, 3, 2, 1, .., n]

3.2. Initial fitness scores

The silhouette score is calculated for each clustering result represented in the chromosome, which measures how similar an object is to its assigned cluster compared to other clusters. Suppose a is the average distance d between a point and all other points in the same cluster, and b is the average distance between the point and all points in the nearest different cluster. The silhouette value for a point i is defined as (1).

$$S_{(i)} = \frac{b(i) - a(i)}{max\{a(i),b(i)\}} \quad (1)$$

3.3. Tournament selection

To evaluate the effectiveness of the candidate solutions during the clustering optimization process, a tournament selection method is employed. In this approach, a subset of individuals from the population is randomly selected, and the one with the best fitness (in this case, the highest silhouette score) is chosen as the winner for the next generation. For this experiment, a tournament size of 2 is utilized, meaning that two individuals compete in each tournament round. This process is repeated across three tournaments, resulting in three individuals being selected based on their silhouette scores. The detailed results of the tournament selection, including the individual identifiers, the corresponding number of clusters, and the silhouette scores, are summarized in Table 4.

Table 4. Tournament selection result

Individual	Number of clusters	Silhouette score
8	2	-0,003620
16	4	-0,000404
..	..	..
2	2	0.000122
15	3	-0,031871

3.4. Crossover

Before the generation of new candidate solutions, a genetic crossover operator is applied to the selected individuals. In this study, a one-point crossover mechanism is used to combine the genetic information from two parent individuals, thereby producing new offspring that inherit characteristics from both parents. Specifically, a crossover point is randomly chosen—in this case, at position 90 out of a total of 198 genes. The segment to the left of the crossover point is taken from the first parent, while the segment to the right is taken from the second parent, and vice versa for the second child. This recombination process enables the exploration of new regions in the search space by introducing novel genetic variations in the offspring. The crossover process and the resulting structure of the children, as derived from the two parents, are illustrated in Figure 1.

3.5. Mutation

After crossover, the offspring undergoes mutation with a probability of 0.3. A random number is generated in the range [0,1] for each gene. If the generated random number is less than 0.3, then mutation is applied by changing the label to the value of the maximum cluster in the offspring. The detailed process is shown in Table 5.

3.6. Population update

At the outset of the evolutionary clustering process, an initial population of individuals is generated, with each individual representing a possible solution to the clustering problem. In this context, an individual is encoded as a vector of cluster labels, where each label indicates the cluster assignment for a specific data point in the dataset. For instance, in the case of k = 2, each gene in the vector will have a value of either 0 or 1, denoting membership in one of the two clusters. This diverse initial population serves as the

foundation for subsequent genetic operations, such as crossover and mutation, which aim to explore and optimize cluster assignments. The structure and contents of the initial population, including individual indices, the number of clusters, and the corresponding cluster label vectors, are summarized in Table 6.

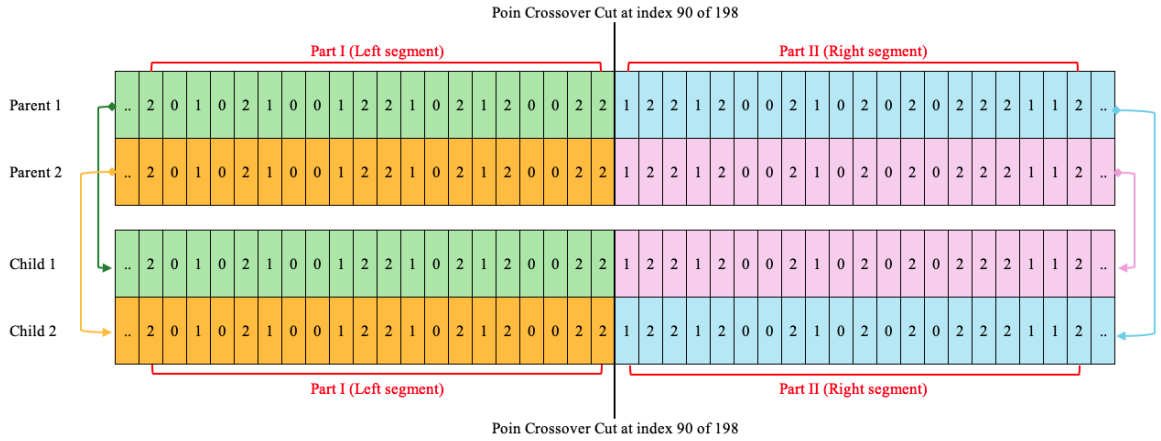

Figure 1. One-point crossover at cut = 90 (showing first 40 points)

Table 5. Sample mutation result of child 1, child 2, and child 3

Random number	Individual 1	Child 1	Individual 2	Child 2	Individual 3	Child 3
0,077225	1	0				
0,166961	1	0				
..	..	..				
0,263557	1	0				
0,109494	1	0				
0,065847			0	1		
0,244669			0	1		
..			..	..		
0,200000			1	0		
0,299086			1	0		
0,123337					0	1
0,189630					0	1
..					..	..
0,038790					0	1
0,252448					1	0

Table 6. Initial population of individuals with cluster labels (k = 2)

Individual	Number of clusters	Cluster labels
1	2	[0, 1, 0, 0, 1, 0, 0, 0, 0, 0, ...]
2	4	[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...]
..	..	..
19	5	[0, 1, 0, 0, 1, 0, 0, 0, 1, 0, ...]
20	4	[0, 0, 0, 0, 1, 1, 0, 1, 1, 0, ...]

3.7. Fitness scores after population update

The recalculated fitness scores for the updated population are presented in Table 7, providing a basis for evaluating the optimal number of clusters. The results indicate that two clusters (k = 2) achieve the highest silhouette score of 0.1287, suggesting that this configuration provides the best clustering performance among the evaluated alternatives. Although the silhouette score is relatively low, indicating limited cluster separation and potential overlap between groups, the two-cluster configuration represents the most suitable solution based on the comparative fitness evaluation presented in Table 7.

3.8. Initial best centroid positions

Determining optimal initial centroids is essential for improving clustering performance. In the GA approach, evolutionary principles are used to select starting points that minimize errors and accelerate convergence. This optimization reduces sensitivity to random initialization, yielding more cohesive clusters and clearer separation, thus providing a stronger foundation for clustering complex datasets, as presented in

Tables 8 and 9. Figure 2 compares the initial centroid positions from k-means++ and GA optimization. While k-means++ centroids tend to vary due to random initialization, the GA-based approach produces more stable and well-separated centroids, leading to improved clustering performance.

Table 7. Best silhouette score generation

Generation	Best silhouette score	k_{best}
1	0.1180	2
2	0.1180	2
3	0.1180	2
4	0.1180	2
..	..	..
49	0.1287	2
60	0.1287	2

Table 8. Best initial centroid positions from k-means++ results (original scale)

Cluster	A1	A2	A3	A4	A5	A6
0	303.12	75.13	129.41	5.72	51.80	486.53
1	154.00	39.97	63.88	1.95	268.30	475.65
2	322.57	78.71	138.85	4.71	545.85	522.4

Table 9. Best initial centroid positions from GA results (original scale)

Cluster	A1	A2	A3	A4	A5	A6
0	225.78	55.89	95.55	3.75	387.13	592.83
1	270.77	68.14	115.11	4.83	464.33	691.65

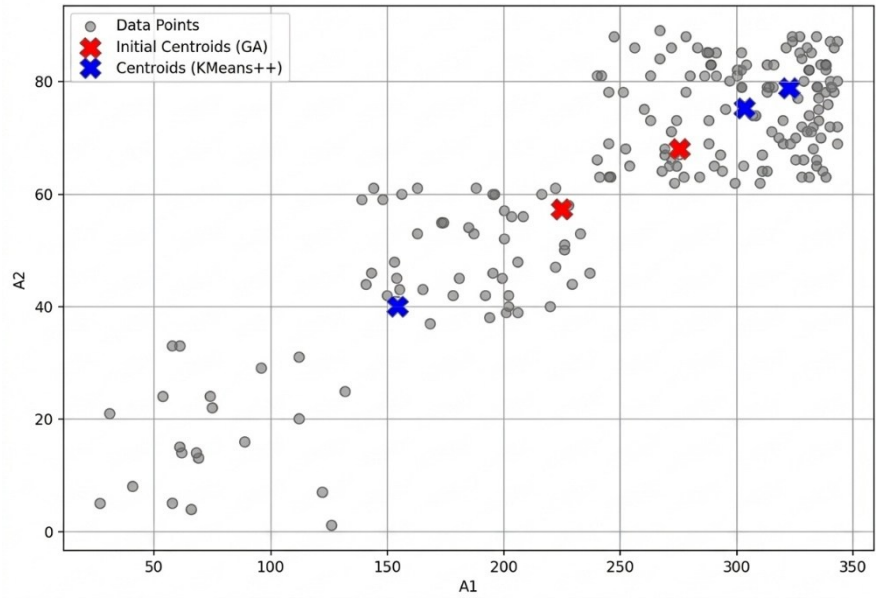

Figure 2. Optimal initial centroid positions

To assess the computational efficiency of both clustering methods, a comparison of processing time (fit time) was conducted between GA k-means and k-means++ across various sample sizes, namely 20, 70, 120, and 198 data points. The test results summarized in the following table provide insights into the scalability and performance trade-offs between the evolutionary approach and the conventional k-means++ algorithm, as illustrated in Figure 3. Figure 3 demonstrates that the fitting process of GA k-means is consistently faster than k-means++ because the initial centroid positions in GA k-means are already optimized and directly determined by the GA, as shown in Table 9, thereby reducing the number of iterations needed to reach convergence during clustering.

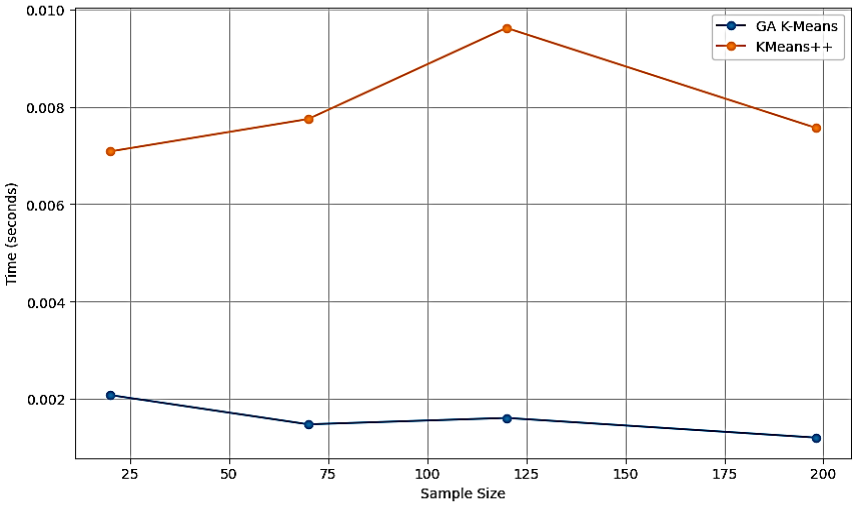

Figure 3. Clustering fit time comparison by sample size

3.9. Fitness scores with genetic algorithm k-means and k-means++

After the GA process was executed, the optimal initial centroids were obtained as the best representation of cluster centers. These optimized centroids were then used as the initialization for the conventional k-means algorithm to ensure a more stable and accurate clustering process. The silhouette score results with k-means++ indicate that the best value is obtained at $k = 3$ (0.6144) thus, the optimal number of clusters in this study is three. The final results of applying k-means with the optimal centroids are presented in Table 10.

Table 10. Fitness score for different number of clusters

Number of clusters *(k)*	Fitness score GA–k-means	Fitness score k-means++
2	0.8948	0.5705
3	0.5489	0.6144
4	0.4527	0.6045
5	0.4490	0.3876
6	0.4459	0.3777
7	0.4703	0.2824
8	0.4789	0.2878

3.10. Plot comparison of fitness scores from genetic algorithm and k-means++

The results show that the GA yields higher fitness scores compared to the conventional k-means initialization across all tested cluster numbers. GA achieves its highest fitness score of 0.8948 with two clusters, and although the score gradually decreases as the number of clusters increases, it remains relatively stable. On the other hand, the silhouette score analysis with k-means++ indicates that the optimal clustering solution is obtained at $k = 3$ with a score of 0.6144, which reflects the best balance between cohesion and separation. These findings suggest that GA provides a robust method for optimizing centroid initialization, while the elbow method with k-means++ effectively determines the most appropriate number of clusters, with both approaches complementing each other to improve clustering performance, as presented in Figure 4.

Figure 4 shows that GA k-means achieves higher silhouette scores than k-means++ because the initial centroids in GA k-means are optimized using a GA, as presented in Table 9, resulting in clusters that are more distinct and cohesive. This study demonstrates the clear advantage of the evolutionary-based GA k-means over conventional k-means++. GA k-means achieved a higher silhouette score (0.8948 with two clusters) compared to k-means++ (0.6144 with three clusters), indicating superior cluster quality and clearer group separation. Furthermore, GA–k-means identified a different optimal number of clusters, reflecting its ability to explore the solution space more broadly and avoid local optima [6], [8]. The centroids produced by GA–k-means were also more stable across experiments, supporting findings by [3], [18] on the importance of global optimization. As illustrated in Figure 4, GA yields more stable fitness scores across cluster numbers, while k-means++ shows greater fluctuation. However, this improved quality comes with higher computational cost, as GA k-means requires more processing time, whereas k-means++ is faster but may yield suboptimal results on complex data [18], [22]. Overall, GA-based centroid initialization is recommended as an adaptive and robust solution for complex clustering tasks.

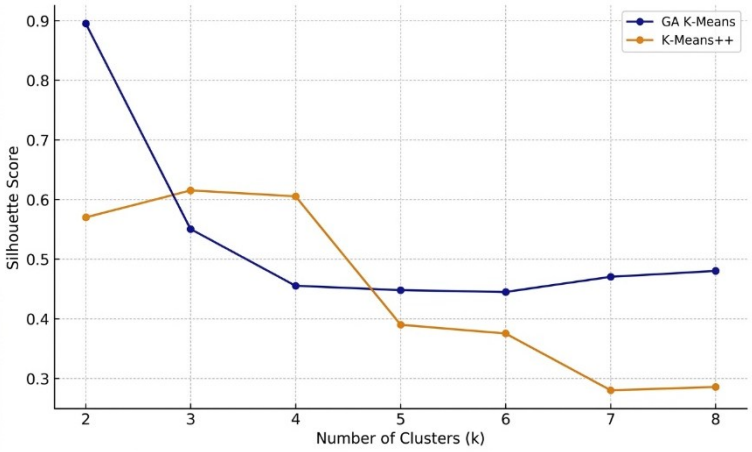

Figure 4. Plot comparison of fitness scores from GA k-means and k-means++

4. CONCLUSION

This study demonstrates that integrating the GA with k-means clustering is effective in optimizing cluster initialization and improving clustering performance. The GA achieved its best fitness score of 0.8948 at two clusters ($k = 2$), showing superior stability compared to random or conventional initialization. Meanwhile, the k-means++ evaluation based on the silhouette score indicated that the optimal cluster structure was obtained at $k = 3$ with a score of 0.6144, reflecting a well-defined balance between cohesion and separation. These findings suggest that GA is valuable for enhancing centroid initialization, while k-means++ is effective for determining the most appropriate number of clusters. Future research should explore hybrid approaches, alternative fitness metrics, and broader application domains to further improve clustering quality and robustness.

ACKNOWLEDGEMENTS

This work was partially supported by the Department of Computer Science and Electronics, Universitas Gadjah Mada, under the Publication Funding Year 2026. Additionally, the authors express gratitude for the valuable comments and suggestions provided by the reviewers, contributing to the improvement of the presentation of this research.

FUNDING INFORMATION

This research was funded by Department of Computer Science and Electronics, Universitas Gadjah Mada under the publication funding year 2026.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Abdul Rahman	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	
Sri Mulyana	✓	✓		✓		✓		✓		✓	✓	✓	✓	
Agus Sihabuddin	✓			✓		✓				✓	✓	✓	✓	
Aina Musdholifah	✓			✓		✓				✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

DATA AVAILABILITY

The data used in this study were derived from the publicly available Open University Learning Analytics Dataset (OULAD) [25]. This study used only a subset of the original dataset, consisting of selected attributes relevant to the research objectives. The original dataset is available at https://doi.org/10.1038/sdata.2017.171. The data selection and preprocessing procedures applied in this study are described in the relevant sections of this article.

REFERENCES

[1] A. Nabil, M. Seyam, and A. A.-Elfetouh, "Prediction of students' academic performance based on courses' grades using deep neural networks," *IEEE Access*, vol. 9, pp. 140731–140746, 2021, doi: 10.1109/ACCESS.2021.3119596.

[2] H. Turabieh *et al.*, "Enhanced harris hawks optimization as a feature selection for the prediction of student performance," *Computing*, vol. 103, no. 7, pp. 1417–1438, 2021, doi: 10.1007/s00607-020-00894-7.

[3] M. A. Damos *et al.*, "Enhancing the k-means algorithm through a genetic algorithm based on survey and social media tourism objectives for tourism path recommendations," *ISPRS International Journal of Geo-Information*, vol. 13, no. 2, 2024, doi: 10.3390/ijgi13020040.

[4] A. Farissi, H. M. Dahlan, and Samsuryadi, "Genetic algorithm based feature selection with ensemble methods for student academic performance prediction," *Journal of Physics: Conference Series*, vol. 1500, no. 1, 2020, doi: 10.1088/1742-6596/1500/1/012110.

[5] M. Anusha, "An enhanced k-means genetic algorithms for optimal clustering," in *International Conference on Computational Intelligence and Computing Research*, 2016, vol. 14, pp. 1–13, doi: 10.1109/ICCIC.2014.7238422.

[6] K. P. Sinaga and M. S. Yang, "Unsupervised k-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020, doi: 10.1109/ACCESS.2020.2988796.

[7] A. Al Malki, M. M. Rizk, M. A. E.-Shorbagy, and A. A. Mousa, "Hybrid genetic algorithm with k-means for clustering problems," *Open Journal of Optimization*, vol. 5, no. 2, pp. 71–83, 2016, doi: 10.4236/ojop.2016.52009.

[8] K. J. Kim and H. Ahn, "A recommender system using GA–K-means clustering in an online shopping market," *Expert Systems with Applications*, vol. 34, no. 2, pp. 1200–1209, 2008, doi: 10.1016/j.eswa.2006.12.025.

[9] S. Kapil, M. Chawla, and M. D. Ansari, "On k-means data clustering algorithm with genetic algorithm," in *2016 4th International Conference on Parallel, Distributed and Grid Computing,* 2016, pp. 202–206, doi: 10.1109/PDGC.2016.7913145.

[10] S. Bhatia, "New improved technique for initial cluster centers of k-means clustering using genetic algorithm," in *2014 International Conference for Convergence of Technology,* 2014, doi: 10.1109/I2CT.2014.7092112.

[11] B. Lu and F. Ju, "An optimized genetic k-means clustering algorithm," in *2012 International Conference on Computer Science and Information Processing,* 2012, pp. 1296–1299, doi: 10.1109/CSIP.2012.6309099.

[12] Z. Yuan, X. Wang, F. Chen, and X. Ma, "Improved k-means algorithm for nearby target localization," *IEEE Access*, vol. 13, pp. 14872–14880, 2025, doi: 10.1109/ACCESS.2024.3479091.

[13] H. Mahmood, T. Mehmood, and L. A. A.-Essa, "Optimizing clustering algorithms for anti-microbial evaluation data: a majority score-based evaluation of k-means, gaussian mixture model, and multivariate t-distribution mixtures," *IEEE Access*, vol. 11, pp. 79793–79800, 2023, doi: 10.1109/ACCESS.2023.3288344.

[14] A. Bhadana and M. Singh, "Fusion of k-means algorithm with Dunn's index for improved clustering," in *2nd International Conference on Computational Systems and Information Technology for Sustainable Solutions, CSITSS 2017*, 2018, doi: 10.1109/CSITSS.2017.8447798.

[15] Z. Lin *et al.*, "Research on power market user credit evaluation based on k-means clustering and contour coefficient," in *2020 3rd International Conference on Robotics, Control and Automation Engineering, RCAE 2020*, 2020, pp. 64–68, doi: 10.1109/RCAE51546.2020.9294725.

[16] H. Santoso and H. Magdalena, "Improved k-means algorithm on home industry data clustering in the Province of Bangka Belitung," in *ICoSTA 2020: 2020 International Conference on Smart Technology and Applications: Empowering Industrial IoT by Implementing Green Technology for Sustainable Development*, 2020, doi: 10.1109/ICoSTA48221.2020.1570598913.

[17] J. Srinivasan, R. Murthy, S. Sundara, N. Agarwal, and S. DeFazio, "Extensible indexing: a framework for integrating domain-specific indexing schemes into Oracle8i," in *16th International Conference on Data Engineering*, 2000, pp. 91–100, doi: 10.1109/icde.2000.839396.

[18] M. Laszlo and S. Mukherjee, "A genetic algorithm using hyper-quadtrees for low-dimensional k-means clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 533–543, 2006, doi: 10.1109/TPAMI.2006.66.

[19] M. S. Yang and I. Hussain, "Unsupervised multi-view k-means clustering algorithm," *IEEE Access*, vol. 11, pp. 13574–13593, 2023, doi: 10.1109/ACCESS.2023.3243133.

[20] M. Somantri, R. I. Pramudya, A. Sugiharto, and T. A. Kurniawan, "Hybrid genetic algorithm and particle swarm optimization as centroid initialization in k-means clustering for national nutrition dataset," *Journal of ICT Research and Applications*, vol. 20, no. 2, pp. 183-204, 2026, doi: 10.5614/itbj.ict.res.appl.2026.20.2.5.

[21] H. Wang and Y. Song, "Portrait of college students' online learning behavior based on artificial intelligence technology," *IEEE Access*, vol. 12, pp. 6318–6328, 2024, doi: 10.1109/ACCESS.2024.3349448.

[22] S. R. Zhu, "Application of improved genetic algorithm in optimization computation," in *5th International Conference on Natural Computation, ICNC 2009*, 2009, vol. 4, pp. 144–148, doi: 10.1109/ICNC.2009.393.

[23] K. Krishna and M. N. Murty, "Genetic k-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 29, no. 3, pp. 433–439, 1999, doi: 10.1109/3477.764879.

[24] G. R. Corominas, M. J. Blesa, and C. Blum, "Accelerating the k-means++ algorithm by using geometric information," *IEEE Access*, vol. 13, pp. 67693–67717, 2025, doi: 10.1109/ACCESS.2025.3561293.

[25] J. Kuzilek, M. Hlosta, and Z. Zdrahal, "Data descriptor: Open University learning analytics dataset," *Scientific Data*, vol. 4, 2017, doi: 10.1038/sdata.2017.171.

BIOGRAPHIES OF AUTHORS



Abdul Rahman is a lecturer in the Informatics Study Program at Universitas Baturaja with a strong academic background in Informatics Engineering and Computer Science. He earned his bachelor's degree in Informatics Engineering from Universitas Amikom, Yogyakarta, Indonesia, in 2016, followed by a master's degree in Computer Science from Universitas Budi Luhur, Jakarta, Indonesia, in 2019. His primary research interests include intelligent systems, machine learning, and case-based reasoning. His master's thesis, titled "Case-based reasoning adaptive e-learning system based on visual-auditory-kinesthetic learning styles," highlights his contributions to developing adaptive learning systems tailored to visual, auditory, and kinesthetic learning styles. Currently, he is pursuing a doctoral degree in Computer Science at Universitas Gadjah Mada. He can be contacted at email: abdulrahman@unbara.ac.id.



Sri Mulyana is a lecturer and researcher in Computer Science at Universitas Gadjah Mada, Indonesia. He holds a doctorate, master's, and bachelor's degree in Computer Science from Universitas Gadjah Mada. His areas of expertise include expert systems, computer reasoning, and decision support systems. He currently serves as the head of the Intelligent Systems Laboratory, Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada. He can be contacted at email: smulyana@ugm.ac.id.



Agus Sihabuddin is a lecturer and researcher in Computer Science at Universitas Gadjah Mada, Indonesia. He holds a doctorate and master's in Computer Science from the Department of Computer Science and Electronics Universitas Gadjah Mada, Yogyakarta, Indonesia. He is an assistant professor at the Department of Computer Science and Electronics Universitas Gadjah Mada, Yogyakarta, Indonesia. His research interests include computation, machine learning, and time series forecasting. He can be contacted at email: a_sihabudin@ugm.ac.id.



Aina Musdholifah is a lecturer and researcher at the Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Indonesia. She earned her bachelor's and master's degrees in Computer Science from Universitas Gadjah Mada, and a Ph.D. in Computer Science from Universiti Teknologi Malaysia. Her research interests include machine learning, spatio-temporal data mining, bioinformatics, and intelligent systems. She has published several scientific papers in reputable journals and is actively involved in academic leadership as head of the master in Computer Science Program at Universitas Gadjah Mada. He can be contacted at email: aina_m@ugm.ac.id.