STM JOURNALS

International Journal of
Solid State Innovations & Research

Volume 3, Issue 1, 2025
January–June
DOI (Journal): 10.37591/IJSSIR

https://journals.stmjournals.com/ijssir

Research IJSSIR

Advancements in K-Means Clustering: Boosting Algorithm Performance Through Innovations

Sohan Lal Gupta[1,*], Vinod Kataria[1], Arpita Sharma[1], Vikram Khandelwal[1], Anjali Pandey[1], Vipin Gupta[2]

Abstract

K-Means clustering is a widely used unsupervised learning algorithm for partitioning a dataset into distinct clusters. Despite its popularity and simplicity, K-Means has several limitations, such as sensitivity to initial centroids, convergence to local minima, and inefficiency with large datasets. This study reviews recent advancements aimed at addressing these challenges and enhancing the performance of the K-Means algorithm. Innovations include improved initialization methods, such as K-Means++, which significantly reduce the chances of poor clustering results by selecting more optimal starting centroids. Additionally, optimization techniques, such as using advanced optimization algorithms and parallel processing, have been developed to accelerate convergence and handle larger datasets more efficiently. We also explore hybrid approaches that combine K-Means with other clustering algorithms to achieve more accurate and robust clustering outcomes. These advancements collectively contribute to the enhanced performance, scalability, and robustness of the K-Means algorithm, making it more suitable for a wider range of applications in data analysis and machine learning.

Keywords: Data mining, data clustering, centroids, SSE, k-means, distance metrices

INTRODUCTION

K-Means clustering is a popular unsupervised learning algorithm valued for its straightforward approach and efficiency in grouping data into separate clusters based on shared features. Introduced by Stuart Lloyd in 1957 and later generalized by MacQueen in 1967 [1], the algorithm has become a fundamental tool in various fields such as image processing, market segmentation, and bioinformatics. Although widely used, the traditional K-Means algorithm has its shortcomings. It is highly sensitive to the choice of initial centroids, which can lead to suboptimal clustering results and convergence to local minima. Additionally, K-Means struggles with large datasets due to its computational inefficiency and the requirement for multiple iterations to achieve convergence.

***Author for Correspondence**
Sohan Lal Gupta
E-mail: sohan.gupta@skit.ac.in

[1]Assistant Professor, Department of Computer Science and Engineering, Swami Keshvanand Institute of Technology Management, Gramothan Jaipur, Rajasthan, India
[2]Assistant Professor, School of Engineering and Technology, Suresh Gyan Vihar University, Jaipur, Rajasthan, India

Received Date: March 24, 2024
Accepted Date: April 01, 2025
Published Date: April 09, 2025

Citation: Sohan Lal Gupta, Vinod Kataria, Arpita Sharma, Vikram Khandelwal, Anjali Pandey, Vipin Gupta. Advancements in K-Means Clustering: Boosting Algorithm Performance Through Innovations. International Journal of Solid State Innovations & Research. 2025; 3(1): 30–37p.

To address these challenges, significant advancements and innovations have been introduced over the years. One major improvement is the development of better initialization methods, such as K-Means++, which selects initial centroids in a way that significantly enhances the chances of finding a globally optimal solution. This reduces the likelihood of poor clustering outcomes and enhances the overall performance of the algorithm. Optimization techniques, including advanced optimization algorithms and the utilization of parallel processing, have also been proposed to accelerate convergence and improve computational efficiency. These methods enhance K-Means' ability to efficiently process large datasets.

Moreover, hybrid approaches that integrate K-Means with other clustering algorithms have been explored to leverage the strengths of different methods and achieve more accurate and robust clustering results. These combined approaches overcome the limitations of K-Means while maintaining its straightforwardness and clarity.

This study aims to review these advancements in K-Means clustering, highlighting how these innovations have contributed to boosting the algorithmic performance, scalability, and robustness of the K-Means algorithm. By understanding and leveraging these improvements, practitioners and researchers can apply K-Means clustering more effectively in various complex and large-scale data analysis scenarios.

The strategy behind the approach is to make k clusters by data defined say n. Distance between each cluster and centroid is obtained. K number of clusters must always be less than the number of data sets also called as data objects i.e. n. The research work is so designed to have literature survey part after introduction; followed by proposed work methodology; after that result in the form of graphical representation of samples; which again is followed by future work and scope of proposed methodology. The methodology approach gives us a solution that how system can select the number of clusters and how much iterations are to be formed. The results also produce the 2-D samples on X and Y axis; and 3-D samples' results are shown in X, Y, and Z axis.

LITERATURE SURVEY

Singh and Bansal introduced a concept related to clustering techniques and how noise affects them [2]. Their approach offers a straightforward method for classifying a dataset into a predetermined number of clusters (denoted as k). The selection of k depends on the specific problem and domain, and users typically experiment with different values. The algorithm is designed to minimize an objective function, specifically a squared error function, where the chosen distance measured between a data point and the cluster center indicates how far the n data points are from their respective cluster centers.

This paper by Lamirel *et al.* proposed little efficient clustering algorithm with improved cluster quality [3]. For improvement in the cluster quality, this paper is using clustering aggregation and spectra analysis by understanding the properties of data before actual clustering. Then this modified algorithm is applied to online retails data set. And the results show that proposed algorithm is little efficient and but producing quality clusters. The performance of proposed and standard k-means is compared by four performance metrics such Clustering Accuracy, Sum of Square Error, Compactness and Running Time.

Joshi *et al.* present a survey in their paper on enhancements made to the traditional K-means algorithm to overcome its limitations [4]. Additionally, they compare the K-means clustering algorithm with other clustering techniques. Qi *et al.* gave a technique that has three principles, known as novel optimized hierarchical clustering method [5]. This approach effectively increased the chance of getting the best local optima, as well top-n nearest clusters.

The papers by Ikotun *et al*. and Tang *et al*. explored the incorporation of nature-inspired optimization algorithms into K-means clustering. They discussed how integrating bio-inspired optimization techniques enhances clustering performance [6, 7]. Extended versions of these algorithms demonstrate improved outcomes, and experiments are conducted to validate the effectiveness of the proposed approach.

Arthur and Vassilvitskii introduced K-Means++, an initialization technique that selects initial centroids in a probabilistic manner, ensuring that they are spread out. This method significantly reduces the chances of poor clustering outcomes and often leads to faster convergence and better clustering results [8].

Likas *et al.* introduced the Global K-Means algorithm, which progressively incorporates one cluster center at a time through a systematic global search approach [9]. This method avoids the local minima problem by considering multiple potential solutions during each iteration.

Ackermann *et al*. developed Streaming K-Means; an algorithm designed to cluster data in a single pass [10]. This method works especially well for real-time applications where data is constantly coming in.

PROPOSED WORK

The aim of this proposed work is to further enhance the performance of the K-Means clustering algorithm by addressing its current limitations through innovative approaches [11]. Building upon the advancements highlighted in the literature survey, this work will focus on developing and integrating new techniques that improve initialization, optimization, and scalability. Additionally, hybrid models and application-specific enhancements will be explored to ensure the algorithm's robustness and versatility in diverse real-world scenarios [12].

They are given a collection of data instances that need to be grouped based on a defined similarity criterion. According to the study, the number of clusters should not exceed the database length. This research work introduces two mathematical formulas to determine the total number of similar slots for a large set of data elements that do not fall under the category of local optima [13–19]. k-means clustering defines to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as prototype of the cluster [20].

Applied K-Means

The clustering method used in the study is easy, and we instigate with an explanation of the initial version of algorithm. Algorithm steps initially pick C initial centroids, where C is an initially defined limit by user, namely, the number of clusters required. Each element is allocated its place to the nearby centroid, and group of same elements allocated to a centroid is a cluster. After that, center point of cluster is modernized based on the elements allocated to the slots called as cluster group. This step is repeated until no element or objects altered [21–24].

Process Flow

The stages of the K-means clustering set of rules are (Figure 1):

1. Set k centroids, for ease select randomly m1, m2, …, mk.
2. Recurrence until all centroids retain constant:
   Assign an element xi to the cluster Sj, according to:
   $$S_j = \left\{ x_i \left\| x_i - x_j \right\|^2 \le \| x_i - m_t \|^2, 1 \le t \le k \right\}$$
3. Update the center of Sj, namely, mj.

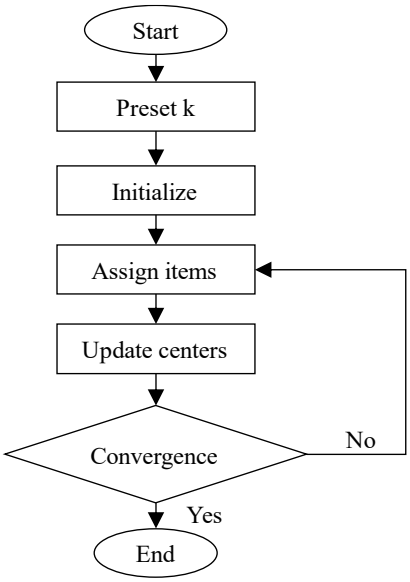

Figure 1. Flow-chart of K-means clustering set of rules.

Assigning Elements or Objects from a Dataset to their Closest Reference Point is Known as the Centroid Method

To determine the closest center of mass, we need a measure of proximity that accurately defines "closeness" based on the specific data being analyzed. In Euclidean space, the commonly used metric is Euclidean (L2) distance, whereas for document comparisons, trigonometric function similarity is often more appropriate. For example, Manhattan (L1) distance is often used for geometer knowledge, whereas the Jaccard live is usually utilized for documents [25].

The similarity measures used in the K-means algorithm are generally straightforward because the algorithm continuously determines the similarity between each data point and its respective centroid. However, in certain situations, such as when working with low-dimensional Euclidean space, it is possible to bypass numerous similarity calculations, which can greatly enhance the algorithm's efficiency. Bisecting K-means is another approach that accelerates K-means by reducing the number of similarities computed.

We have overturned the k-means methodology, and some new features like cluster calculation, density fixation are added. The main thing is that we have also given a bit modification to the distance formula to overcome the fact that the previously used Euclidean function took longer time.

Number of clusters formed is controlled depending upon the sample, due to the fact that quantity of samples does not lie into a particular cluster group range. This control is also required because no fixed data is available with time and development time shall also be increased and then the number of slots shall be required at every iteration of algorithm. The two equations used in this work are as:

I: *when samples <10,000,00 then,*

$$r=\sqrt{n}/3 \quad (1)$$

Where n= number of sample points, r= cluster
No. of Iterations= ${}^{n}C_{r}$

II: *when samples >10,000,00 then,*

$$r=n^{0.4} \quad (2)$$

Centroid formation: We have two distinct formulas for different ranges of clusters since we must determine the number of ideal clusters of data items that do not fall within the local optima range. Here, below are the example describing the formulas that work for all range of sets of data:

- *Example 1:* Let we have a set having data elements and objects not more than 10,00,000.
  Datasets n=15678
  Number of clusters formed $r=\sqrt{n}/3$ is 41
  If we use formula II in Eq. (2), i.e. $r=n^{0.4}$ is 47
  Hence 41 is minimum number of ideal data clusters where we can group elements.
- *Example 2:* If set of data elements is greater than10,00,000. So here,
  Datasets n=2222222,
  Number of clusters are formed $r=\sqrt{n}/3$ is 496
  If we use formula II in Eq. (2), i.e. $r=n^{0.4}$ is 345
  Hence 345 is minimum number of ideal groups or clusters found by the equation $r=n^{0.4}$

RESULTS

This section presents the potential outcomes derived from the simulation of the compiled code based on the experimental work conducted above.

First Iteration

Figure 2 shows Iteration having 10 data points in two dimensions. However, the algorithm calculated the number of centroids that comes out to be 2 clusters, with a set of 2 centroids.

- *Data elements:* 10

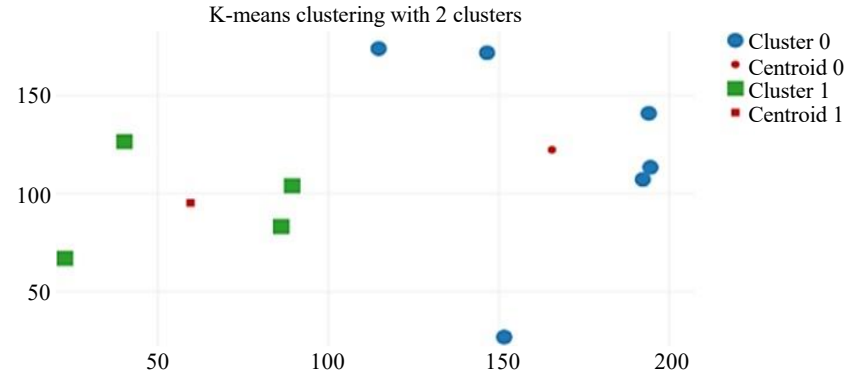

Figure 2. 2D cluster graph on 10 data points.

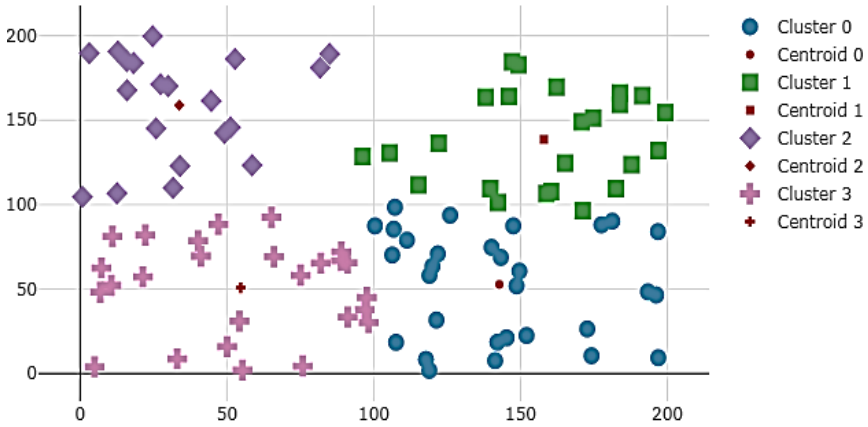

Figure 3. 2D cluster graph representation for over 100 data points.

- *Dimension:* 2
- *Cluster formed:* 2
- *Centroids:* (59.6535910928749, 122.29868592805366); (95.12803406026305, 165.5778410030529)

Second Iteration

Figure 3 shows the resultant graph obtained in Second iteration having 100 data points in two dimensions. The centroids obtained from algorithm that comes out to be 4 clusters, having 4 centroids sets.

- *Data elements:* 100
- *Dimensions:* 2
- *Clusters formed:* 4

Third Iteration

Figure 4 shows the resultant graph obtained for 2-dimensional data in iteration having 500 data. According to the method, there are eight clusters with eight centroids.

- *Data points:*500
- *Dimensions:* 2
- *Clusters formed:* 8

Fourth Iteration

Figure 5 shows the resultant graph obtained for 3-dimensional data in iteration having 10 data points. The centroid estimated by algorithm comes out to be 2 clusters having 2 centroids.

- *Data elements:* 10
- *Dimensions:* 3
- *Resultant slots or cluster:* 8

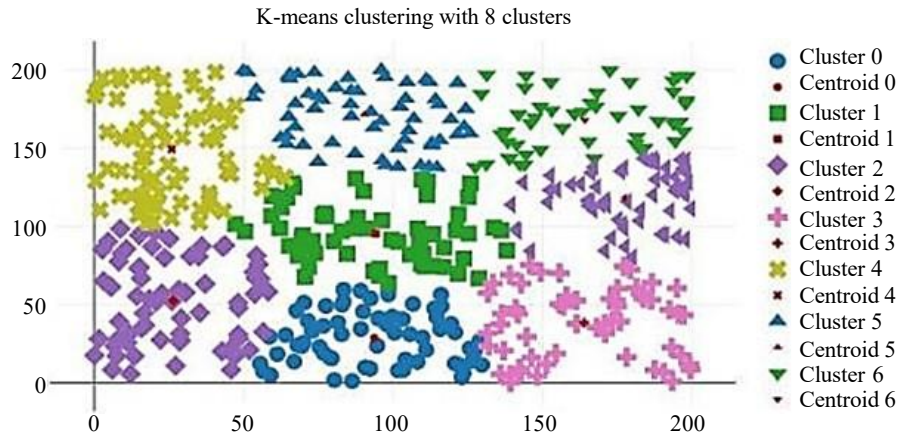

Figure 4. Cluster graph representation with large data.

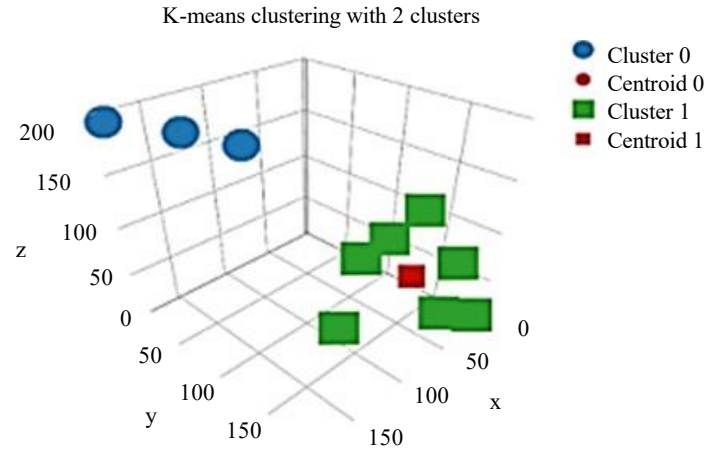

Figure 5. 3D cluster graph representation or fover 10 data points.

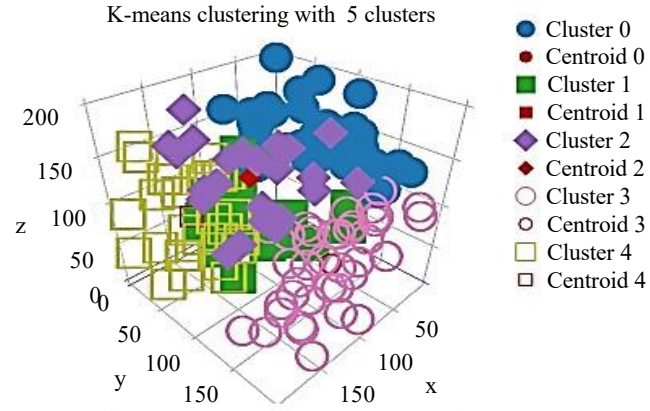

Figure 6. 3D cluster graph representation for over 150 data points.

Fifth Iteration

Figure 6 shows the result generated in iteration 5 for three-dimensional data sample consisting of 150 data points in three dimensions. However, the algorithm calculated the number of centroids that comes out to be 5 clusters, with a set of 5 centroids.

- *Data points:* 150
- *Dimensions:* 3
- *Clusters formed:* 5

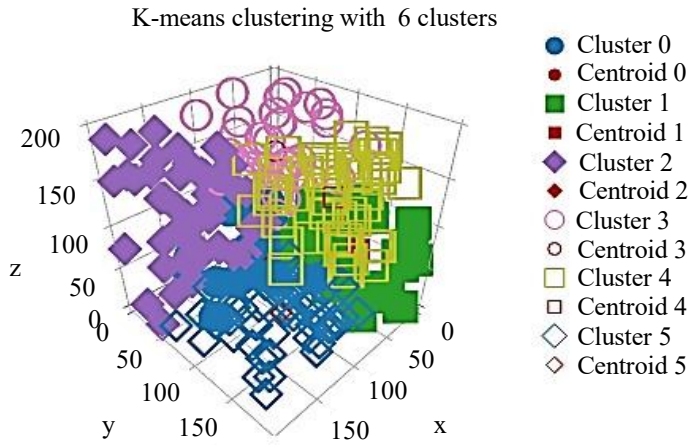

Figure 7. 3D cluster graph representation for over 250 data points.

Sixth Iteration

Figure 7 shows the result generated in iteration 6 for three-dimensional data sample consisting of 250 data points in three dimensions. However, the algorithm calculated the number of centroids that comes out to be 6 clusters, with a set of 6 centroids.

- *Data points:* 250
- *Dimensions:* 3
- *Clusters formed:* 6

CONCLUSION

Based on the experimental results and compiled code analysis, it is evident that identifying the optimal cluster and selecting the appropriate centroid play a vital role in the proposed work. The proposed implementation of new formula on widely used algorithm as k-means enhances the algorithm performance and makes it ideal for clustering purpose. The challenge we faced here is that the algorithm gets slow as when the clusters go above 20 lacs and requirement of the system also goes to high level.

After the experimental setup we can conclude that K-means Clustering technique could be very crucial when we need to fragment data on large scale and also to analyze it. Clustering technique can be very handy while analyzing big data, and to convert data into a relatable data is a big plus.

This research offers several benefits, including the automatic determination of cluster numbers and the ability to form clusters using both two-dimensional and three-dimensional sample points. Also, the two dimensional sample points could go up to 20 lakh sample points without an issue, though for more than 20 lakhs, the system requirements go bit high and are not easy to compile as they make take some time as well.

REFERENCES

1. MacQueen J. Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. 1967; 1: 281–297. Available from: https://www.scirp.org/reference/referencespapers?referenceid=1866605
2. Singh M, Meenakshi Bansal. A survey on various k-means algorithms for clustering. International Journal of Computer Science and Network Security (IJCSNS). 2015 Jun 1; 15(6): 60–65.
3. Lamirel JC, Dugué N, Cuxac P. New efficient clustering quality indexes. In 2016 IEEE International joint conference on neural networks (IJCNN). 2016 Jul 24; 3649–3657.
4. Joshi K, Gupta H, Chaudhary P, Sharma P. Survey on different enhanced K-means clustering algorithm. International Journal of Engineering Trends and Technology (IJETT). 2015; 27(4): 178–182.

5. Qi J, Yu Y, Wang L, Liu J. K*-means: An effective and efficient k-means clustering algorithm. In 2016 IEEE international conferences on big data and cloud computing (BDCloud), social computing and networking (SocialCom), sustainable computing and communications (SustainCom) (BDCloud-SocialCom-SustainCom). 2016 Oct 8; 242–249.
6. Ikotun AM, Ezugwu AE, Abualigah L, Abuhaija B, Heming J. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Inf Sci. 2023 Apr 1; 622: 178–210.
7. Tang R, Fong S, Yang XS, Deb S. Integrating nature-inspired optimization algorithms to K-means clustering. In IEEE Seventh International Conference on Digital Information Management (ICDIM 2012). 2012 Aug 22; 116–123.
8. Arthur D, Vassilvitskii S. k-means++: the advantages of careful seeding. Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms; 2007 Jan 7–9; New Orleans, Louisiana. Philadelphia (PA): Society for Industrial and Applied Mathematics; 2007. p. 1027–35.
9. Likas A, Vlassis N, Verbeek JJ. The global k-means clustering algorithm. Pattern Recognit. 2003 Feb 1; 36(2): 451–61.
10. Ackermann MR, Märtens M, Raupach C, Swierkot K, Lammersen C, Sohler C. Streamkm++ a clustering algorithm for data streams. J Exp Algorithmics. 2012 May 22; 17: 2.4.
11. Kane A, Nagar J. Determining the number of clusters for a k-means clustering algorithm. Indian J Comput Sci Eng. 2012 Oct; 3(5): 670–2.
12. Naz H, Saba T, Alamri FS, Almasoud AS, Rehman A. An improved robust fuzzy local information k-means clustering algorithm for diabetic retinopathy detection. IEEE Access. 2024 Apr 22; 12: 78611–78623.
13. Dorigo M, Birattari M, Stutzle T. Ant colony optimization artificial ants as a computational intelligence technique. IEEE Comput Intell Mag. 2006 Nov; 1(4): 28–39.
14. Dubey A, Choubey AP. A systematic review on k-means clustering techniques. Int J Sci Res Eng Technol. 2017 Jun; 6(6): 624–627.
15. Handhayani T, Wasito I. Fully unsupervised clustering in nonlinearly separable data using intelligent kernel k-means. In 2014 IEEE International Conference on Advanced Computer Science and Information System. 2014 Oct 18; 450–453.
16. Sun Y, Liu G, Xu K. A k-means-based projected clustering algorithm. In 2010 IEEE Third International Joint Conference on Computational Science and Optimization. 2010 May 28; 1: 466–470.
17. Blum C, Sampels M. When model bias is stronger than selection pressure. In International conference on parallel problem solving from nature. Berlin, Heidelberg: Springer Berlin Heidelberg; 2002 Sep 7; 893–902.
18. Blum C, Sampels M. Ant colony optimization for FOP shop scheduling: a case study on different pheromone representations. In Proceedings of the IEEE 2002 Congress on Evolutionary Computation; CEC'02 (Cat. No. 02TH8600). 2002 May 12; 2: 1558–1563.
19. Khadem EA, Nezhad EF, Sharifi M. Data Mining: Methods & Utilities. Researcher. 2013; 5(12): 47–59.
20. Ghosh S, Dubey SK. Comparative analysis of k-means and fuzzy c-means algorithms. Int J Adv Comput Sci Appl. 2013; 4(4): 35–39.
21. Sun H, Wang S, Jiang Q. FCM-based model selection algorithms for determining the number of clusters. Pattern Recognit. 2004 Oct 1; 37(10): 2027–37.
22. Li MJ, Ng MK, Cheung YM, Huang JZ. Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters. IEEE Trans Knowl Data Eng. 2008 Nov 30; 20(11): 1519–34.
23. Awad FH, Hamad MM. Improved k-means clustering algorithm for big data based on distributed smartphoneneural engine processor. Electronics. 2022 Mar 11; 11(6): 883.
24. Zubair M, Iqbal MA, Shil A, Chowdhury MJ, Moni MA, Sarker IH. An improved K-means clustering algorithm towards an efficient data-driven modeling. Ann Data Sci. 2024 Oct; 11(5): 1525–44.
25. Nazeer KA, Sebastian MP. Improving the Accuracy and Efficiency of the k-means Clustering Algorithm. In Proceedings of the world congress on engineering. London, UK: Association of Engineers; 2009 Jul 1; 1: 1–3.

