Hindawi
Mobile Information Systems
Volume 2022, Article ID 1277765, 15 pages
https://doi.org/10.1155/2022/1277765

Research Article

Performance Evaluation of Simple *K*-Mean and Parallel *K*-Mean Clustering Algorithms: Big Data Business Process Management Concept

Islam Zada,[1] **Shaukat Ali**,[1] **Inayat Khan**,[2] **Myriam Hadjouni**,[3] **Hela Elmannai**,[4] **Muhammad Zeeshan**,[5] **Ali Mohammad Serat**,[6] **and Abid Jameel**[7]

[1]*Department of Computer Science, University of Peshawar, Peshawar, Pakistan*
[2]*Department of Computer Science, University of Buner, Buner, Pakistan*
[3]*Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia*
[4]*Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia*
[5]*Institute of Computing, Kohat University of Science & Technology, Kohat, Pakistan*
[6]*Computer Science Faculty, University of Nangarhar, Jalalabad, Nangarhar, Afghanistan*
[7]*Department of Computer Science & Information Technology, Hazara University, Mansehra, Pakistan*

Correspondence should be addressed to Ali Mohammad Serat; serat_af@yahoo.com

Received 2 April 2022; Accepted 27 May 2022; Published 23 June 2022

Academic Editor: Fazli Wahid

Data is the most valuable asset in any firm. As time passes, the data expands at a breakneck speed. A major research issue is the extraction of meaningful information from a complex and huge data source. Clustering is one of the data extraction methods. The basic *K*-Mean and Parallel *K*-Mean partition clustering algorithms work by picking random starting centroids. The basic and *K*-Mean parallel clustering methods are investigated in this work using two different datasets with sizes of 10000 and 5000, respectively. The findings of the Simple *K*-Mean clustering algorithms alter throughout numerous runs or iterations, according to the study, and so iterations differ for each run or execution. In some circumstances, the clustering algorithms' outcomes are always different, and the algorithms separate and identify unique properties of the *K*-Mean Simple clustering algorithm from the *K*-Mean Parallel clustering algorithm. Differentiating these features will improve cluster quality, lapsed time, and iterations. Experiments are designed to show that parallel algorithms considerably improve the Simple *K*-Mean techniques. The findings of the parallel techniques are also consistent; however, the Simple *K*-Mean algorithm's results vary from run to run. Both the 10,000 and 5000 data item datasets are divided into ten subdatasets for ten different client systems. Clusters are generated in two iterations, i.e., the time it takes for all client systems to complete one iteration (mentioned in chapter number 4). In the first execution, Client No. 5 has the longest elapsed time (8 ms), whereas the longest elapsed time in the following iterations is 6 ms, for a total elapsed time of 12 ms for the *K*-Mean clustering technique. In addition, the Parallel algorithms reduce the number of executions and the time it takes to complete a task.

1. Introduction

Most commercial organizations that generate vast amounts of data do so during their daily operations. These businesses require an easy means to obtain and access their stored data, which necessitates the use of a centralized storage concept known as a database. A database is a collection of data that is compacted and arranged in such a way that it is easy to access, retrieve, manage, and change. Because business analysts require stored data to make business decisions, important information should be extracted utilizing a discovery concept known as data mining, which is also known as knowledge discovery [1, 2].

Today, everything is based on data. People come across vast volumes of data daily and save it for later review or analysis. Because such massive datasets are continuously rising, extracting and mining valuable information using traditional techniques are becoming increasingly difficult [3]. As a computer system can process data in a specific order from a set of facts, numbers, or statistics, it is called data. Companies today are collecting vast volumes of data in a variety of circumstances, formats, and databases.

Clustering is a phenomenon of unsupervised learning, whereas classification is a process of supervised learning. These two methods are frequently employed when extracting data from large databases. The graphical representation of Supervised and Unsupervised Techniques is shown in Figure 1.

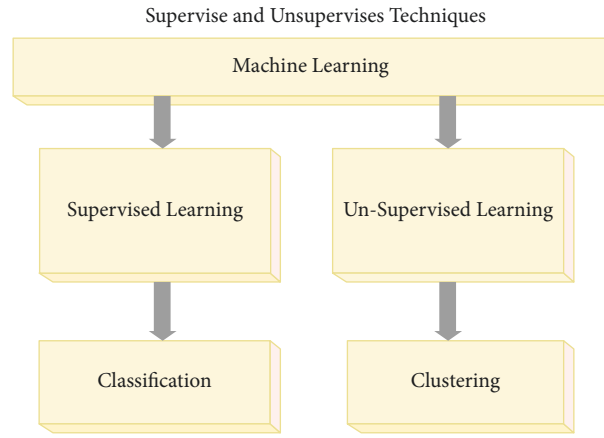


Figure 1: Supervised and unsupervised techniques.

1.1. Classification of Clustering Techniques. The clustering techniques are categorized into four fundamental categories, as illustrated in Figure 2.

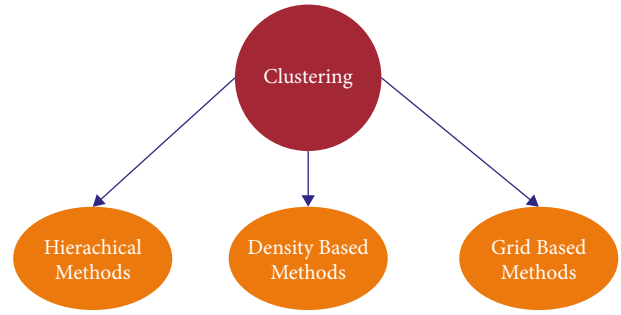


Figure 2: Types of clustering.

A massive amount of data can be difficult to turn into usable information. By using data mining algorithms, researchers can predict and evaluate students' academic progress based on their academic records and forum involvement.

Even though various research has been conducted around the world to evaluate student academic performance, there is a dearth of acceptable studies to examine aspects that can help students improve their academic performance. The goal of this study was to evaluate the factors that influence student academic achievement in Pakistan.

Both basic and parallel clustering approaches are constructed and studied in this work to highlight their greatest qualities. Simple *K*-Mean methods have shortcomings, and parallel *k*-mean approaches resolve those weaknesses. The results of parallel *k*-mean techniques are always the same: improved cluster quality, fewer executions, and faster execution times. The outcomes of the Simple *K*-Mean are likewise variable for different iterations or executions; as a result, the number of iterations varies depending on the iterations or executions. In some circumstances, the clustering algorithms' outcomes are always different, and the algorithms separate and identify unique properties of the *K*-Mean Simple clustering algorithm from the *K*-Mean Parallel clustering algorithm. The Parallel *K*-Mean algorithms have been proven to be more efficient than the Simple *K*-Mean algorithms in several tests. Parallel algorithms reduce the number of executions and the amount of time it takes to complete a task.

2. Literature Review

2.1. Simple K-Mean Clustering. J. B. MacQueen was one of the first users of the *K*-means clustering technique, which he introduced in 1967. The most recent research on *K*-Mean clustering is described here, and some of the related work has been published since. The author [4] introduced the Min-Max distance measure. The input dataset is first adjusted, and then initial centroids are chosen at random within the normalized range (0, 1). The distance is estimated using the min-max similarity measure.

Reference [5] divides the entire data collection into unit blocks using the lowest and highest bounds (UB). Following the modification, the items in the datasets are sorted by distance and then separated into subclusters (*k* sets). Each set of data is evaluated by computing the median. Initial centroids are computed using the specified medium, and clusters are built using the design of the initial cluster [6]. This method made use of sorting algorithms, which are more time consuming. The dataset's simplest representation is found by finding the centroids of each unit block.

Simple *K*-Means are algorithms in which the information from each iteration is stored in a data structure, as described in [7, 8]. The recorded information is then utilized in the next iteration. A dynamic *K*-Mean clustering algorithm was introduced in [9]. In the first phase, subdatasets are created on the server side from the provided dataset. By modifying the datasets, items are now sorted by distance and arranged into subclusters (*k* sets).

2.2. Simple Parallel K-Mean Clustering Algorithm. Sanpawat and Alva [2, 10] proposed a parallelized *K*-Mean clustering method. The algorithm uses a (Client-Server) method. Technology, Earth sciences, engineering, social and economic sciences, medical sciences, and life are just a few of the fields that employ clustering.

A parallel *K*-Mean clustering technique is proposed in [6, 11]. Each data point's distance from the next is calculated. The data items that are the furthest away from the origin are segregated from the rest of the dataset and placed in a separate list. For this new list, a threshold value is chosen. For the simultaneous *K*-Mean clustering process, [12] developed the ParaMeans program. They adopt the Basic parallelized *K*-Mean clustering technique for regular laboratory application. ParaMeans is a client-server application that is simple to use and manage.

2.3. Simple and Parallel K-Mean. [13, 14] explain the Simple *K*-Mean clustering technique. The distance between the original centroids and the data items is determined, and each of the data items is given to its proper location. The input dataset is first adjusted, and then initial centroids are chosen at random within the normalized range (0, 1). The min-max similarity measure is used to calculate the distance. (0, 1) The min-max similarity measure [15] is used to calculate the distance. The *K*-Mean algorithm, developed by Singh and Bhatia [16], identifies items with the lowest frequency. The centroids are calculated as the average of each section. All clusters are compiled on the server (received from all clients). Based on the clustering method, the arithmetic means of each cluster are determined. It is efficient and progressive due to the integration of a dynamic load balance technique and the *K*-Mean clustering method in [17, 18]. In this strategy, the main system assigns the client system the same size subdataset [19, 20].

The parallel *K*-mean clustering approach and the basic *K*-mean clustering technique have both been thoroughly investigated. Many academics worked individually on Simple and Parallel *K*-Mean techniques, offering alternative methodologies discussed in Section 2. However, they make no explicit recommendations or suggestions on how to use parallel and simple *k*-mean approaches in any of the domains where they are useful [21–23].

3. Research Methodology

Researchers have created many methods for Simple and Parallel *K*-Mean clustering approaches. Some existing strategies concentrated on sorting the dataset to select initial centroids, while others focused on the random selection of first centroids. When it comes to the Parallel and Simple *K*-Mean techniques, there is no clear understanding of the best approach and which technique should be used in which situation. Researchers looked at, implemented, and evaluated both Parallel and Simple *K*-Mean clustering algorithms to see what qualities they had and how well they performed when applied to these problems in general. The overall research flow is depicted in Figure 3.

3.1. Data Sets. The scores of 10,000 students in two different topics and the attendance of 5000 employees for two months are represented by these two datasets of 10,000 and 5000 integers, respectively. The challenge of randomly selecting initial centroids in *K*-Mean clustering is solved in this paper.

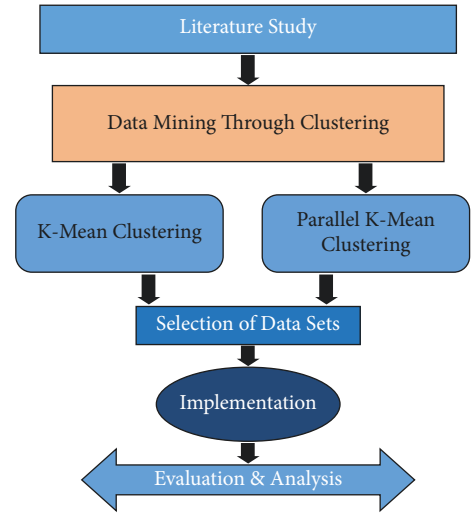


Figure 3: Research flow diagram [24].

These two sets of 10,000 and 5000 integers could represent 10,000 of students' grades in two subjects and five thousand (5000) employees' attendance over the duration of two months, respectively. Table 1 displays a typical representation of these pupils in two distinct disciplines using two different methodologies, while Table 2 illustrates staff/employees attendance.

For these two algorithms, below are some samples of input and output:

(i) **Input**

k: the No. of clusters derived from students and workers' scores in two separate topics and months,

D_k: There are two datasets, each with 10,000 pupils and 5,000 employees.

(ii) **Output**

A set of *k* clusters.

3.2. Method. Simple and parallel approaches are used on these data components individually. The flowchart of the Basic Simple *K*-Mean clustering technique is created using standard UML (Unified Modeling Language) notations.

The differences between the Parallel and *K*-Mean clustering methods are assessed and analyzed using experimental findings from both techniques. These two algorithms use the JAVA with Neat beans as an (IDE) and C++ platforms to execute different execution for varied data ranges and times.

3.2.1. Simple K-Mean Clustering Algorithm. The *K*-Mean clustering approach randomly chooses "*k*" initial centroids. The distances between data items and centroids are calculated using the Euclidean distance function in the second phase. [24, 25] mentions a couple of distance functions.

During relocation, each data item is relocated to the cluster that has the least amount of space. The earliest clusters are created in this manner. The arithmetic mean of

Table 1: Ten thousand students and their marks in two subjects.

Students	Marks of subj-A	Marks of subj-B
01	84	66
02	74	81
;	;	;
;	;	;
;	;	;
;	86	61
10000	56	76

Table 2: Attendance of employees in two months.

Emp_id	Attendance %age of month_A	Attendance %age of month_B	Total attendance %age
0001	91	76	(91 + 76)/2 = ...
0002	74	89	
0003	86	81	
. . .	. . .	. . .	
. . .	. . .	. . ..	
5000	96	76	

each cluster is then calculated. That cluster's data points are closer to the arithmetic mean. Following that calculation, data points are assigned a cluster based on the arithmetic mean. Until there are no more data points to transfer from one cluster to another, the process is repeated [26].

(1) Steps in the Simple K-Mean Clustering Algorithm. The pseudocode for the basic *K*-Mean clustering approach [14] is shown below:

(2) Flow Chart of Simple K-Mean Algorithm. The flowchart of the basic *K*-Mean method is created using standard UML (Unified Modeling Language) notations, which are depicted in Figure 4 as Simple *k*-mean algorithm's Flow chart.

3.2.2. Parallel K-Mean's Clustering Algorithm. When the dataset is sufficiently large, the space and processing performance requirements for the Simple *K*-Mean clustering approach are the most significant hurdles. The Simple or Basic *K*-Mean clustering technique is parallelized to solve these challenges.

(1) Main Steps of Parallel K-Mean Clustering Algorithm. Three main steps of the Simple Parallel *K*-Mean's algorithm are as follows:

(i) Compilation
(ii) Partition
(iii) Computation

In the first phase, subdatasets are created on the server side from the provided dataset. Each client computer connected to the server receives these subdatasets, which include the number of clusters, "*k*," and starting centroids. Client systems that are affected calculate the clusters and send the results to the server. The process is continued until the clusters do not change.

(2) Flow chart of Parallel K-Mean Clustering Algorithm. The above-mentioned steps are depicted in Figure 5 as a flow chart. The flow chart is created using UML (Unified Modeling Language) standard notations.

4. Results and Discussion

The features of simple *K*-Mean and Parallel *K*-Mean techniques are highlighted in this research. Some existing strategies concentrated on sorting the dataset to select initial centroids, while others focused on the random selection of first centroids.

For the experiments, two datasets of 10,000 and 5000 integers representing students and teachers are chosen at random. The performance of Simple and Parallel clustering methods is tested using these datasets. The experimental results are presented in detail in the following sections of this chapter.

4.1. Experimental Results Analysis. For a dataset of 10,000 and 5000 integer data pieces, both techniques are tested and compared with each other. Using the Simple *K*-Mean clustering technique, both strategies produced positive experimental results. In the next phase, the results of the comparison of the Simple and Parallel algorithms are shown.

4.1.1. Comparison of Parallel and Simple K-Mean Algorithm. A comparison between the Simple and the Parallel *K*-Mean method is performed by considering the number of executions, elapsed time, and cluster quality.

4.1.2. Number of Iterations. The tables and graphs below illustrate the performance of the Parallel and Simple *K*-Mean clustering algorithms for varying numbers of clusters (K).

Table 3 compares the *K*-Mean technique versus the parallel *K*-Mean algorithm for identical datasets and cluster number ($K = 3$). The same dataset (10,000 data points) is used in each run to observe and perceive how the number of executions in the Simple *K*-Mean algorithm changes over time. Because the starting centroids are not produced at

Input: Array $\{a_1, a_2, a_3, \ldots, a_n\}$
a = data points
k = Number of Required Clusters
Output: A set of Clusters
Steps:
(1) Randomly select k data points from dataset D as initial centers.
(2) Calculate the distance between each data point d_i $(1 < I \leq n)$ and all the k clusters C_j $(1 \leq j \leq k)$ and recalculate the cluster center by taking the Arithmetic Mean of each cluster.
(3) Repeat until no change in the center of clusters

Algorithm 1: To Find the clusters by simple K-Mean clustering algorithm.

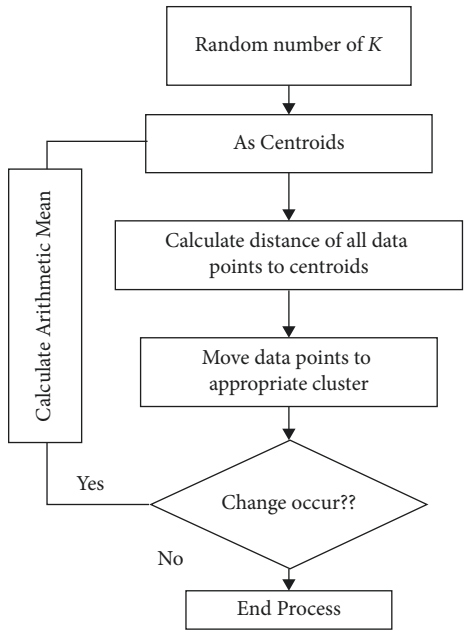


Figure 4: Simple k-mean algorithm's flow chart.

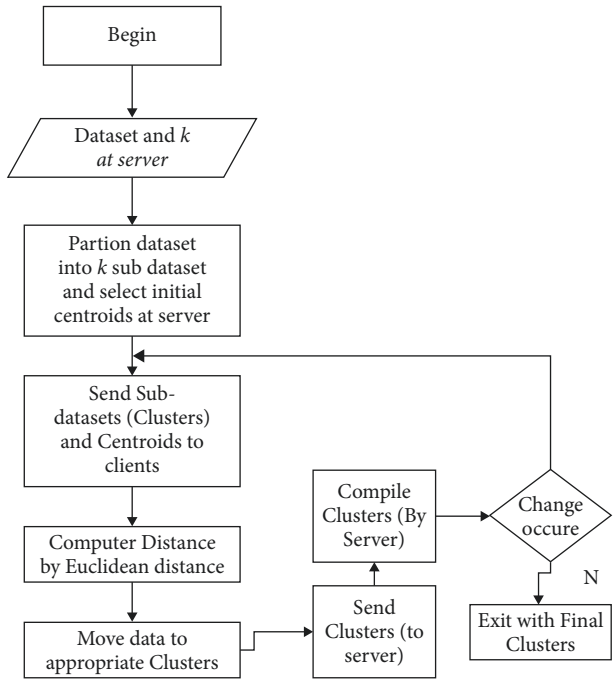


Figure 5: Flow chart of simple parallel K-mean's clustering algorithm.

Table 3: Executions for $K = 3$.

$K = 3$ Iteration/execution	No. of executions done by simple K-mean clustering	No. of executions done by parallel K-mean clustering
1	10	3
2	12	3
3	9	3
4	12	3
5	14	3
6	7	3
7	12	3
8	9	3
9	12	3
10	15	3

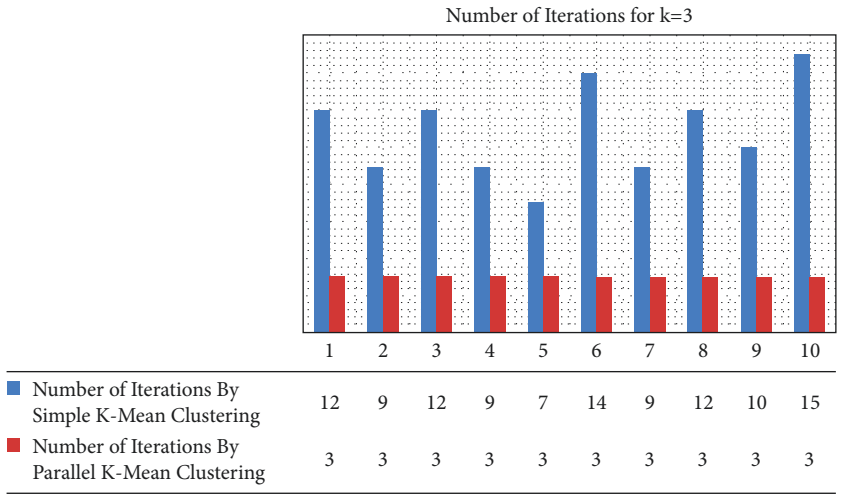


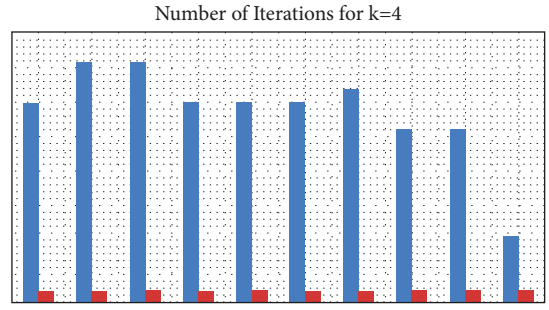
Figure 6: Representation of Table 3.

Table 4: Executions for $K = 4$.

$K = 4$ Iteration/executions	No. of iterations by simple K-mean clustering	No. of iterations by parallel K-mean clustering
1	5	aph1
2	15	1
3	15	1
4	18	1
5	18	1
6	18	1
7	13	1
8	16	1
9	15	1
10	5	1

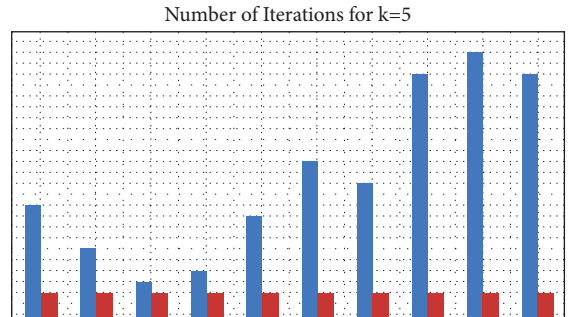
Table 5: Executions for $K = 5$.

$K = 5$ Iteration/execution	No. of executions by simple K-mean clustering	No. of executions by parallel K-mean clustering
1	13	7
2	12	7
3	8	7
4	7	7
5	11	7
6	16	7
7	18	7
8	28	7
9	26	7
10	28	7



	1	2	3	4	5	6	7	8	9	10
Number of Iterations By Simple K-Mean Clustering	15	18	18	15	15	15	16	13	13	5
Number of Iterations By Parallel K-Mean Clustering	1	1	1	1	1	1	1	1	1	1

Figure 7: Graph of Table 4.



	1	2	3	4	5	6	7	8	9	10
Number of Iterations By Simple K-Mean Clustering	14	10	7	8	13	18	16	26	28	26
Number of Iterations By Parallel K-Mean Clustering	6	6	6	6	6	6	6	6	6	6

Figure 8: Graph of Table 5.

Table 6: Iterations for $K = 6$.

$K = 6$ Iteration/execution	No. of executions by simple K-mean clustering	No. of executions by parallel K-mean clustering
1	18	3
2	11	3
3	13	3
4	11	3
5	17	3
6	11	3
7	13	3
8	8	3
9	20	3
10	20	3

random, the number of executions in the Parallel K-Mean method is fixed.

The graph in Table 3 is depicted in Figure 6. With the Parallel K-Mean technique, $k = 3$ means that 3 executions are performed, but in the Simple K-Mean method, it fluctuates from run to run.

According to Tables 4 and 5, the number of parallel K-Mean clustering is lower than the number of Simple K-Mean

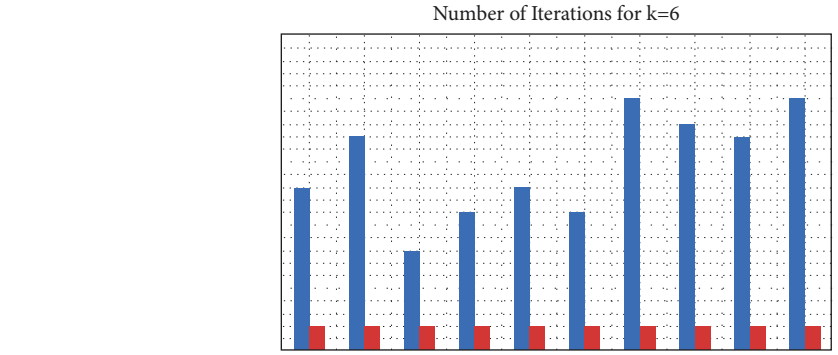


Figure 9: Graph of Table 6. No. of executions for $K = 6$.

Table 7: Iterations for $K = 7$.

$K = 7$ Iteration/execution	No. of executions by simple K-mean clustering	No. of executions by parallel K-mean clustering
1	12	9
2	10	9
3	9	9
4	10	9
5	19	9
6	11	9
7	14	9
8	18	9
9	21	9
10	6	9

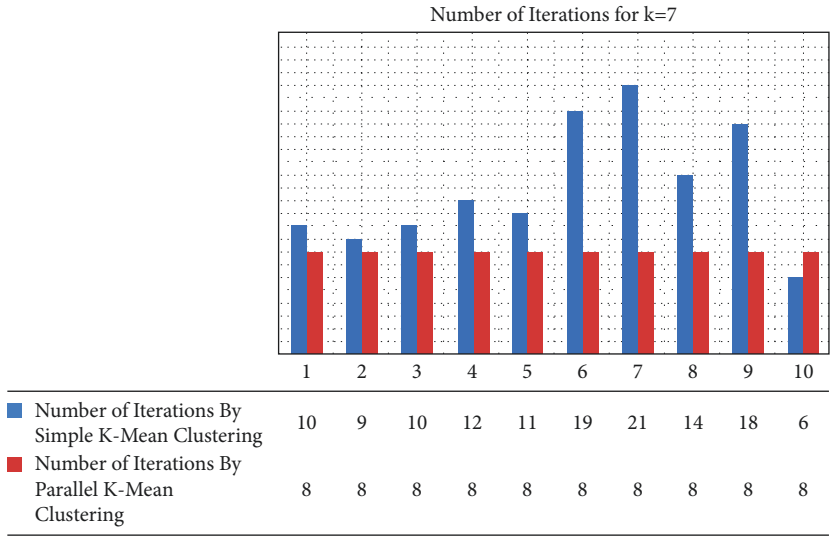


Figure 10: Graph of Table 7. The number of iterations for $K = 7$.

Table 8: Elapsed time for $K = 3$.

$K = 3$ Iteration/execution	Simple K-mean clustering elapsed time in ms	Parallel K-mean clustering elapsed time in ms
1	13.9	6.9
2	19.7	7.9
3	14.1	9.4
4	20.3	9.4
5	20.3	9.4
6	18.7	9.4
7	18.0	7.9
8	14.1	8.4
9	14.7	8.4
10	18.7	9.3

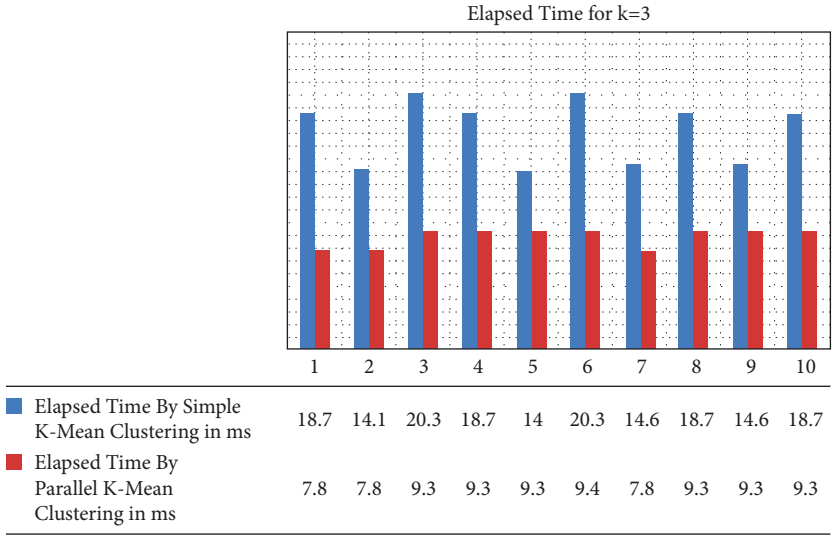


Figure 11: Graph of Table 8. Elapsed time for $K = 3$.

Table 9: Elapsed time for $K = 4$.

$K = 4$ Iteration/execution	Simple K-mean clustering elapsed time in ms	Parallel K-mean clustering elapsed time in ms
1	17.5	8.7
2	14.8	8.7
3	14.8	8.7
4	18.2	8.3
5	19.8	7.3
6	14.9	7.3
7	17.5	7.4
8	17.5	7.1
9	17.5	7.1
10	10.9	7.2

clustering which is represented in Figures 6 and 7, respectively.

According to Table 5, the number of parallel K-Mean clustering is lower than the number of Simple K-Mean clustering which is represented in Figure 7, respectively.

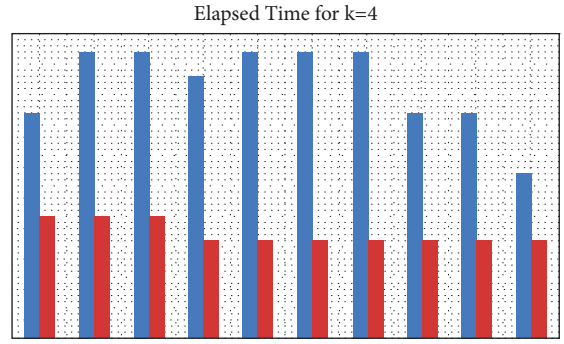
As shown in Table 5, there are fewer executions of the K-Mean clustering method using the parallel approach, as $k = 5$ is fixed, which is given in Figure 8.

Table 6 shows the fixed and lower No. of iterations for the Parallel and Simple K-Mean clustering methods for $k = 6$, which is depicted in Figure 9.

The number of times the Parallel and Simple K-Mean algorithms were run for $k = 7$ is shown in Table 7 and presented in Figure 10.

4.2. Elapsed Time. For varied numbers of clusters, the following tables and graphs show the elapsed time of the Simple and Parallel K-Mean clustering methods (K).

A comparison between the Simple K-Mean algorithm and parallel K-Mean algorithm can be found in Table 8 for $K = 3$, which is depicted in Figure 11. Parallel K-Mean

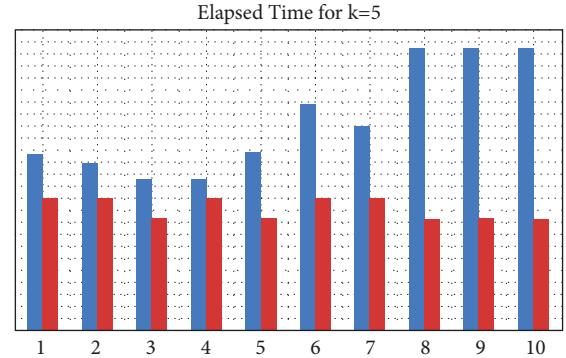


	1	2	3	4	5	6	7	8	9	10
Elapsed Time By Simple K-Mean Clustering in ms	14.6	18.8	18.7	17.1	18.7	18.8	18.7	14.6	14.6	10.9
Elapsed Time By Parallel K-Mean Clustering in ms	7.8	7.8	7.8	6.2	6.2	6.2	6.3	6.2	6.2	6.3

Figure 12: Graph of Table 9. Elapsed time for $K = 4$.

Table 10: Elapsed time for $K = 5$.

$K = 5$ Iteration/execution	Simple K-mean clustering elapsed time in ms	Parallel K-mean clustering elapsed time in ms
1	12.6	9.4
2	12.1	9.4
3	12.6	9.4
4	23.5	11.0
5	17.2	10.9
6	18.8	9.4
7	14.7	10.9
8	12.6	11.0
9	23.5	11.0
10	23.5	9.4



	1	2	3	4	5	6	7	8	9	10
Elapsed Time By Simple K-Mean Clustering in ms	14.6	14	12.5	12.5	14.6	18.7	17.1	23.4	23.4	23.4
Elapsed Time By Parallel K-Mean Clustering in ms	10.9	10.9	9.3	10.9	9.3	10.9	10.9	9.4	9.3	9.3

Figure 13: Graph of Table 10. Elapsed time for $K = 5$.

Table 11: Elapsed time for $K = 6$.

$K = 6$ Iteration/execution	Simple K-mean clustering elapsed time in ms	Parallel K-mean clustering elapsed time in ms
1	20.12	7.7
2	20.4	14.1
3	14.7	14.1
4	20.0	9.6
5	20.2	7.7
6	20.12	7.7
7	26.6	7.9
8	18.6	9.4
9	18.9	9.5
10	23.5	9.5

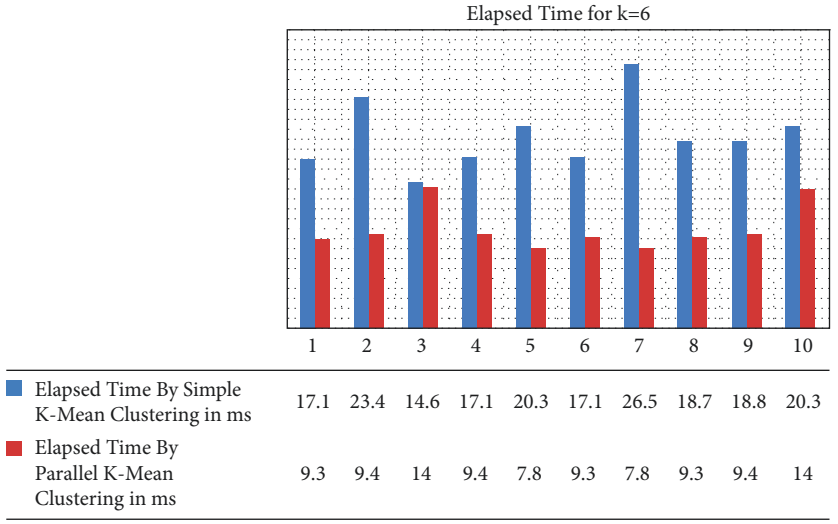


Figure 14: The graphical representation of Table 11. Elapsed time for $K = 6$.

Table 12: Elapsed time for $K = 7$.

$K = 7$ Iteration/execution	Simple K-mean clustering elapsed time in ms	Parallel K-mean clustering elapsed time in ms
1	12.7	11.0
2	14.1	14.6
3	20.4	12.2
4	17.3	14.1
5	12.7	11.1
6	12.6	14.2
7	14.7	14.6
8	23.5	11.1
9	14.2	12.6
10	20.3	12.6

clustering consumes less time for each iteration than Simple K-Mean clustering.

The parallel K-Mean method takes less time than the Simple K-Mean method at different runs or executions.

According to Table 9, the Parallel K-Mean Clustering method takes about half the time as the Simple K-Mean clustering method for $k = 4$, which is presented in Figure 12.

Comparing the Parallel K-Mean clustering algorithm to the Simple K-Mean algorithm for $k = 5$, Table 10 compares the elapsed time of both methods, which is depicted in Figure 13.

Table 11 shows the elapsed time of the Parallel and Simple K-Mean algorithms for $k = 6$ and is given in Figure 14, while Table 12 shows elapsed time for $k = 7$, respectively.

4.3. Cluster Quality. The next section compares the cluster quality of the Simple K-Mean and Parallel K-Mean methods given in Tables 13 and 14, represented in Figure 16 and in Figure 17, respectively.

Table 13 displays the outcomes of numerous runs or executions of the same data collection of 10,000 data items.

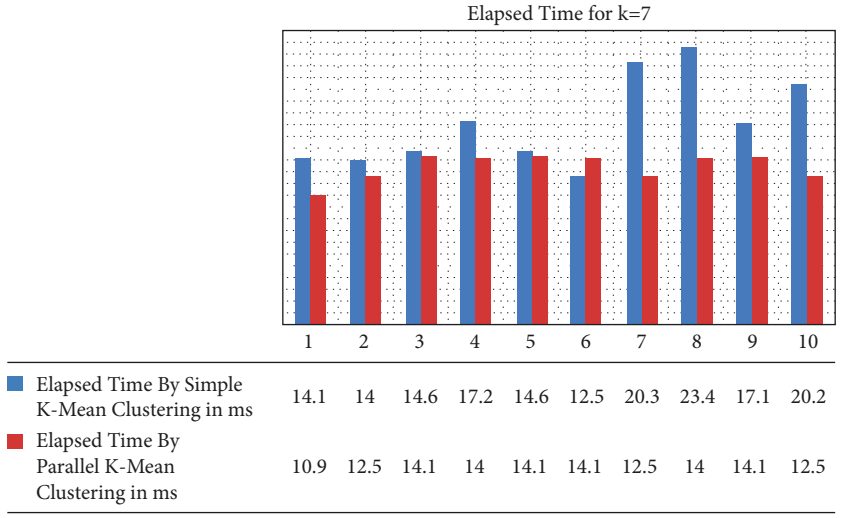


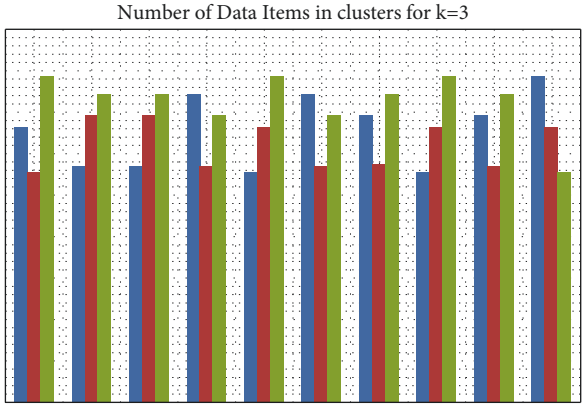
Figure 15: Graph of Table 12. Elapsed Time for $K = 7$.

Table 13: Cluster quality of simple K-mean clustering for $K = 3$.

S.#	No. of iterations	Elapsed time in ms	No. of data items in cluster# 1	No. of data items in cluster# 2	No. of data items in cluster# 3	Total no. of data items
01	12	18.7	3312	2764	3924	10000
02	9	14.1	2838	3456	3706	10000
03	12	20.3	2838	3456	3706	10000
04	9	18.7	3706	2838	3456	10000
05	7	14.0	2764	3312	3924	10000
06	14	20.3	3706	2838	3456	10000
07	9	14.6	3456	2838	3706	10000
08	12	18.7	2764	3312	3924	10000
09	10	14.6	3456	2838	3706	10000
10	15	18.7	3924	3312	2764	10000

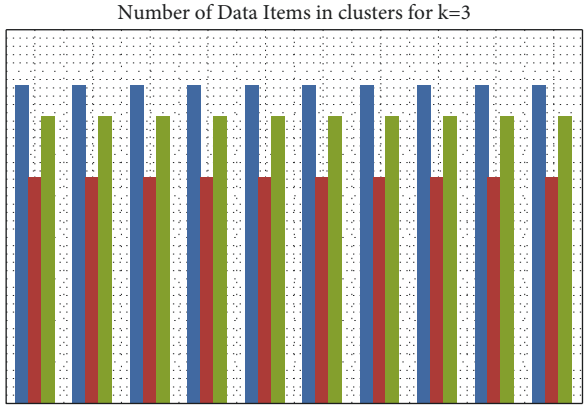
Table 14: Cluster quality of parallel K-mean clustering for $K = 3$.

S.#	No. of executions	Elapsed time in ms	No. of data items in cluster# 1	No of data items in cluster# 2	No. of data items in cluster# 3	Total no. of data items
01	3	7.8	3822	2722	3456	10000
02	3	7.8	3822	2722	3456	10000
03	3	9.3	3822	2722	3456	10000
04	3	9.3	3822	2722	3456	10000
05	3	9.3	3822	2722	3456	10000
06	3	9.4	3822	2722	3456	10000
07	3	7.8	3822	2722	3456	10000
08	3	9.3	3822	2722	3456	10000
09	3	9.3	3822	2722	3456	10000
10	3	9.3	3822	2722	3456	10000



	1	2	3	4	5	6	7	8	9	10
# of Data items in Cluster 1	3312	2838	2838	3706	2764	3706	3456	2764	3456	3924
# of Data items in Cluster 2	2764	3456	3456	2838	3312	2838	2838	3312	2838	3312
# of Data items in Cluster 3	3924	3706	3706	3456	3924	3456	3706	3924	3706	2764

Figure 16: Graph of Table 13. Simple K-mean's cluster quality for $K = 3$.



	1	2	3	4	5	6	7	8	9	10
# of Data items in Cluster 1	3822	3822	3822	3822	3822	3822	3822	3822	3822	3822
# of Data items in Cluster 2	2722	2722	2722	2722	2722	2722	2722	2722	2722	2722
# of Data items in Cluster 3	3456	3456	3456	3456	3456	3456	3456	3456	3456	3456

Figure 17: Graph of Table 14. Parallel K-mean's cluster quality for $K = 3$.

Table 14 shows the same results for the same 10,000 data items over numerous runs or executions for the same dataset.

5. Conclusion

The current technique's fundamental flaw is that it produces various results for the same data. Both basic and parallel clustering approaches are constructed and studied in this work to highlight their greatest qualities. Simple K-Mean methods have shortcomings, and parallel k-mean approaches resolve those weaknesses. The results of parallel k-mean techniques are always the same: improved cluster quality, fewer executions, and faster execution times. The outcomes of the Simple K-Mean are likewise variable for different iterations or executions; as a result, the number of iterations varies depending on the iterations or executions. In some circumstances, the clustering algorithms' outcomes are always different, and the algorithms separate and identify unique properties of the K-Mean Simple clustering algorithm from the K-Mean Parallel clustering algorithm. The Parallel K-Mean algorithms have been proven to be more efficient than the Simple K-Mean algorithms in several tests. Parallel algorithms reduce the number of executions and the amount of time it takes to complete a task. Experiments have shown that Parallel algorithms outperform the Simple K-Mean algorithm by a wide margin. The findings of the Parallel techniques are also consistent; however, the Simple K-Mean technique assembles different outcomes with each iteration or execution. In addition, the Parallel techniques reduce overall iterations and elapsed time [27].

6. Future Work

A technique for *K*-Mean clustering that works for many types of data should be developed in the future. When dealing with categorical data, e.g., a method should perform better. The process of selecting a "*k*" number of clusters is still in progress. The user should input the number of clusters in the upgraded framework. To choose "*k*," which denotes the number of clusters, sophisticated procedures might be used. Although the Parallel *K*-Mean approach has only been tested on integer-type data, it might be extended to text-type data, such as English words. Clustering datasets that include many keywords results in the same keywords being assigned to the same groups or clusters. To search for certain terms in a document, a search engine based on the expanded *K*-Mean clustering technique can be introduced.

Data Availability

The authors have added the available data to support the findings of this study that are included within the article.

Disclosure

The paper is a part of the Research Project and Masters in Software Engineering thesis. This paper is based on the second objective of our project, while one paper is already submitted in the same journal, which was based on the first objective of our master thesis [26].

Conflicts of Interest

All the authors declare no conflicts of interest.

Authors' Contributions

Each author has worked equally.

Acknowledgments

The authors would like to thank Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R193), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. This research work was supported by the University of Nangarhar, Jalalabad Afghanistan, and University of Peshawar, Pakistan.

References

[1] M. K. Bhetwal, "Data warehouse and business intelligence: comparative analysis of OLAP tools," *Regis University*, vol. 2011, p. 69, 2011.

[2] F. Marozzo, L. Belcastro, and P. Trunfio, "Big data analysis on clouds," in *Handbook of Big Data Technologies*, pp. 101–142, Springer, 2017.

[3] Q. Cai, H. Zhang, W. Guo et al., "MemepiC: towards a unified in-memory big data management system," *IEEE Transactions on Big Data*, vol. 5, no. 1, pp. 4–17, 2018.

[4] A. Sharma and R. Dhir, "A wordsets based document clustering algorithm for large datasets," in *Proceedings of the Methods and Models in Computer Science, 2009. ICM2CS 2009. Proceeding of International Conference on*, pp. 1–7, IEEE, New Delhi, India, 14 December 2009.

[5] S. Morris and C. Coronel, "Database Systems: Design, Implementation, & Management: Cengage Learning," 2016, https://www.cengage.co.in/.

[6] X. J. Tan, N. Mustafa, M. Y. Mashor et al., "Understanding domain knowledge in initialization method for *K*-mean clustering algorithm in medical images," in *Proceedings of the 6th International Conference on Electrical, Control and Computer Engineering*, pp. 805–817, Springer, Singapore, 09 March 2022.

[7] P. S. Gamare and G. A. Patil, "Efficient clustering of web documents using hybrid approach in data mining," *International Research Journal of Engineering and Technology (IRJET)*, vol. 02, no. 05, pp. 2395–0056, 2015.

[8] S. Ren and A. Fan, "*K*-means clustering algorithm based on coefficient of variation," in *Proceedings of the Image and Signal Processing (CISP), 2011 4th International Congress on*, pp. 2076–2079, IEEE, Shanghai, China, 15-17 October 2011.

[9] M. Sharma, G. N. Purohit, and S. Mukherjee, "Information retrieves from brain MRI images for tumor detection using hybrid technique *K*-means and artificial neural network (KMANN)," in *Networking Communication and Data Knowledge Engineering*, pp. 145–157, Springer, New York, US, 2018.

[10] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang, "Parallel spectral clustering in distributed systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 568–586, 2011.

[11] I. A. Maraziotis, S. Perantonis, A. Dragomir, D. Thanos, and K Nets, "*K*-Nets: clustering through nearest neighbors networks," *Pattern Recognition*, vol. 88, pp. 470–481, 2019.

[12] M. J. Reddy and B. Kavitha, "Clustering the mixed numerical and categorical dataset using similarity weight and filter method," *International Journal of Database Theory and Application*, vol. 5, pp. 121–134, 2012.

[13] C. Mary and S. K. Raja, "Refinement of clusters from *K*-means with ant colony optimization," *Journal of Theoretical and Applied Information Technology*, vol. 6, 2009.

[14] J. Wang and X. Su, "An improved *K*-Means clustering algorithm," in *Proceedings of the Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, pp. 44–46, IEEE, Xi'an, China, 27 May 2011.

[15] R. Taha, S. Alshakrani, and A. Alqaddoumi, "Implementing parallel computing to enhance the performance of *K*-mean algorithm," in *Proceedings of the 2021 International Conference on Data Analytics for Business and Industry (ICDABI)*, pp. 140–143, IEEE, Sakheer, Bahrain, 25 October 2021.

[16] R. V. Singh and M. Bhatia, "Data clustering with modified *K*-means algorithm," in *Proceedings of the Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, pp. 717–721, IEEE, Chennai, India, 03 June 2011.

[17] Y. Zhang, Z. Xiong, J. Mao, and L. Ou, "The study of parallel *K*-means algorithm," in *Proceedings of the Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, pp. 5868–5871, IEEE, Dalian, China, 21 June 2006.

[18] C. Brecher, W. Bleck, J. Feldhusen et al., "Multi-technology platforms (MTPs)," in *Integrative Production Technology*, pp. 369–513, Springer, New York, US, 2017.

[19] R. Ramya, S. Iyengar, K. Venugopal, and L. Patnaik, "Feature extraction and duplicate detection for text mining: a survey," *Global Journal of Computer Science and Technology*, vol. 16, no. 5, pp. 0975–4172, 2017.

[20] Y. Wei, X. Zhang, Y. Shi et al., "A review of data-driven approaches for prediction and classification of building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 1027–1047, 2018.

[21] S. Bourouis and N. Bouguila, "Unsupervised learning using expectation propagation inference of inverted beta-liouville mixture models for pattern recognition applications," *Cybernetics & Systems*, vol. 2022, pp. 1–25, 2022.

[22] S. Ali, S. Baseer, I. A. Abbasi, B. Alouffi, W. Alosaimi, and J. Huang, "Analyzing the interactions among factors affecting cloud adoption for software testing: a two-stage ISM-ANN approach," *Soft Computing*, vol. 2022, 2022.

[23] S. Ali, I. A. Abbasi, E. E. Mustafa, F. Wahid, and J. Huang, "Practitioner's view of the success factors for software outsourcing partnership formation: an empirical exploration," *Empirical Software Engineering*, vol. 27, no. 2, p. 52, 2022.

[24] E. Y. Cheu, C. Keongg, and Z. Zhou, "On the two-level hybrid clustering algorithm," in *Proceedings of the International conference on artificial intelligence in science and technology*, pp. 138–142, Springer, Zakopane, Poland, 7 June 2004.

[25] S. Mehrotra and S. Kohli, "Comparative analysis of *K*-Means with other clustering algorithms to improve search result," in *Proceedings of the Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on, 2015*, pp. 309–313, IEEE, Greater Noida, India, 08 October 2015.

[26] R. Shang, B. Ara, I. Zada, S. Nazir, Z. Ullah, and S. U. Khan, "Analysis of simple *K*-mean and parallel *K*-mean clustering for software products and organizational performance using education sector dataset," *Scientific Programming*, vol. 2021, Article ID 9988318, 2021.

[27] A. Sinha and P. K. Jana, "A hybrid MapReduce-based *k*-means clustering using genetic algorithm for distributed datasets," *The Journal of Supercomputing*, vol. 74, no. 4, pp. 1562–1579, 2018.