Available online at www.jpit.az

15 (1)
2024

# Improved parallel Big data clustering based on k-medoids and k-means algorithms

Rasim Alguliyev[a], Ramiz Aliguliyev [b], Lyudmila Sukhostat[c]

[a,b,c] Institute of Information Technology, B. Vahabzade str., 9A, AZ1141 Baku, Azerbaijan

[a] r.alguliev@gmail.com; [b] r.aliguliyev@gmail.com; [c] lsuhostat@hotmail.com

[a] https://orcid.org/0000-0003-1223-7411; [b] https://orcid.org/0000-0001-9795-1694; [c] https://orcid.org/0000-0001-9449-7457

ARTICLE INFO

**Keywords:**

k-means
k-medoids
Big data
Parallel clustering
Batch clustering

ABSTRACT

In recent years, the amount of data created worldwide has grown exponentially. The increase in computational complexity when working with "Big data" leads to the need to develop new approaches for their clustering. The problem of massive data amounts clustering can be solved using parallel processing. Dividing the data into batches helps to perform clustering in a reasonable time. In this case, the reliability of the obtained result for each block will affect the performance of the entire dataset. The main idea of the proposed approach is to apply the k-medoids and k-means algorithms to parallel Big data clustering. The advantage of this hybrid approach is that it is based on the central object in the cluster and is less sensitive to outliers than k-means clustering. Experiments are conducted on real datasets, namely YearPredictionMSD and Phone Accelerometer. The proposed approach is compared with the k-means and MiniBatch k-means algorithms. Experimental results proved that the proposed parallel implementation of k-medoids with the k-means algorithm shows greater accuracy and works faster than the k-means algorithm.

## 1. Introduction

Many studies have been dedicated to Big data clustering (Zhao et al., 2018; Karmitsa et al., 2018). The problem with clustering is that the data requires a lot of computing resources (Ikotun et al., 2023). Researchers propose new clustering methods and extend existing to solve this issue (Zein et al., 2023).

Recently, the k-means algorithm and its modifications have become the research subject in analysing large volumes of data (Ping et al., 2024; Bagirov et al., 2022; Ahmadov, 2023; Aggarwal & Reddy, 2014). So, (Bahmani et al., 2012) proposed an approach, that is easy to implement, non-trivial, and converges reasonably quickly in a small number of iterations (Boutsidis et al., 2010; Jain, 2010). However, this approach is computationally expensive.

Cluster centroids in k-means clustering are typically not data points and may not be used in several applications, such as sparse data in recommender systems, images in computer vision, etc.

Kaufman & Rousseeuw (1990) proposed an alternative algorithm based on k-medoids. In this case, instead of centroids, representative objects called medoids are considered. The advantage of this approach compared to k-means clustering is that it is less sensitive to outliers and is based on the centrality of the cluster. The k-medoids algorithm shows the best results when working with random distance metric and outliers (Han et al., 2001). The most popular of them is PAM (Partitioning Around Medoids) (Lenssen & Schubert, 2024). However, the key problem of this algorithm is the high cost of execution time. Recently, a new k-medoids-based approach called

*Received 15 September 2023, Received in revised form 16 November 2023, Accepted 30 November 2023*
http://doi.org/10.25045/jpit.v15.i1.03

FasterPAM was proposed (Schubert & Rousseeuw, 2021). Unlike previous works, this method, combined with simple uniform random initialization, works quickly and provides high accuracy.

Parallel technology must be used. Dividing the data into batches helps to perform clustering in a reasonable time (Alguliyev et al., 2016; Zhao et al., 2018; Meng et al., 2018). In this case, the reliability of the obtained result for each block will affect the performance of the entire dataset. (Alguliyev et al., 2016). Thus, the MiniBatch k-means algorithm (Sculley, 2010) was proposed as an alternative to the k-means algorithm for clustering large datasets. The advantage of this algorithm is that it reduces computational costs by using not the entire dataset at each iteration but a fixed-size subsample. However, this affects the accuracy of clustering (Zhu et al., 2023).

The purpose of this work is to develop an effective approach to clustering large datasets. The main idea of the proposed algorithm is to apply the algorithms of k-medoids and k-means to the parallel data clustering. The obtained k-medoids are fused into a dataset and subsequently clustered using the k-means algorithm. k-medoids-based clustering ensures that the cluster center is the most central data point. The hybrid approach is evaluated on large, well-known datasets.

The rest of the paper is organized as follows. Section 2 describes the literature review. The proposed algorithm for parallel Big data clustering is given in section 3. Section 4 analyses the experimental results, as well as a comparison of the proposed hybrid approach with the k-means and MiniBatch k-means algorithms. Section 5 draws the conclusions and outlines the directions for future research.

## 2. Related work

The volume and speed of data creation have steadily increased in recent years. The volume of global data is expected to grow 10-fold in the next five years. The increase in computational complexity when working with Big data leads to the need to develop new approaches for their clustering. Researchers propose various Big Data clustering methods to solve the curse of dimensionality problem in various applications (Wang et al., 2023; Zhang et al., 2023).

Mussabayev et al. (2023) proposed a parallel scheme based on k-means and k-means++ algorithms for Big data clustering. A comparative analysis between the proposed Big-means algorithm and Euclidean minimum sum-of-squares clustering algorithms was performed. The work's limitations include determining the optimal sample size and runtime.

The LBKC (Lower Bound k-means Clustering) algorithm based on k-means was developed by Zhang et al. (2024) to enable faster clustering of Big Data in multidimensional space. It considers a lower bound on the Euclidean distance to reduce the number of operations performed.

Schubert and Rousseeuw (2021) proposed a modified PAM algorithm, which provides computational acceleration. The found solution reduces the number of iterations. The PAM algorithm allows clustering to be performed with a large number of clusters.

Hu et al. (2023) proposed a k-means clustering algorithm based on the Lévy flight path (Lk-means). It prevents early entry into local optima, making processing large data possible. A limitation of the proposed algorithm is the lack of initial clustering center selection.

Li et al. (2023) developed the SMKKM-KWR (Simple Multiple Kernel k-means with Kernel Weight Regularization) algorithm. An approach based on the fused information of all kernel representations can learn the clustering structure.

Summing up the above works, the main contributions of our work are summarized as follows:

- An approach for large dataset clustering based on k-medoids in combination with the k-means algorithm is proposed to improve clustering accuracy.
- The proposed method can process large datasets in parallel.
- The approach was evaluated on large datasets and compared with other machine learning methods.
- Experiments showed that the proposed parallel implementation of k-medoids with k-means has high accuracy and works faster than the classical k-means algorithm when $k$ is greater than 10.

## 3. Proposed approach

This section describes the proposed approach. Let us introduce the following notation:

$X = \{x_1, x_2, \ldots, x_n\}$ is a set of data points specified in $m$-dimensional space, $l$ is the number of batches $(l < n)$, which is determined by PC, $C = \{C_1, C_2, \ldots, C_k\}$ is a set of clusters, where $C_p^l$ is the $p^{\text{th}}$ cluster of the batch $l$. Here $p = 1,2,\ldots,k$ and $k$ determines the number of clusters.

The dataset is then split into equally sized batches. The optimal package size is determined as follows (Parker & Hall, 2014):

$$Q = \frac{\phi(\alpha) \cdot k^2}{r^2}, \quad (1)$$

$\alpha$ is the desired level of significance, parameter $\nu(\alpha)$ is set by Thompson (1987), and $r$ is the "relative difference."

It is assumed that $\alpha = 0.05$ (with 95% probability), $\phi(\alpha) = 1.27359$ and $r = 0.08$. According to Eq. (1) the package size at $k = 3$ will be equal to $Q = \frac{1.27359 \cdot 3^2}{0.08^2} = 1791$, and at $k = 15$ it will be equal to $Q = \frac{1.27359 \cdot 15^2}{0.08^2} = 44775$. The proposed approach considers all clusters in the dataset.

After partitioning the dataset, the k-medoids algorithm is applied to each partition. The clustering problem is to find medoids $M = \{\mu_1, \mu_2, \ldots, \mu_k\} \subset X$ so that the data points and the medoid (representative data point) are as close as possible:

$$g(x) = \sum_{p=1}^{k} \sum_{x_i \in C_p^l} \left\| x_i - \mu_p^l \right\|^2, \quad (2)$$

where $\|\cdot\|$ is the Euclidean norm in $\Re^m$.

k-medoids-based clustering is performed in parallel before reaching the convergence condition, proving that the cluster center is the most central data point.

Next, for each batch, the resulting medoids are fused into a dataset, to which k-means is applied in order to perform clustering. The objective function has the following form:

$$f(x) = \sum_{p=1}^{k} \sum_{x_i \in C_p^l} \left\| x_i - O_p \right\|^2 \to \min, \quad (3)$$

$$O_p = \frac{\sum_{x_i \in C_p} x_i}{|C_p|}, \quad p = 1,2,\ldots,k, \quad (4)$$

where $O_p$ is the centroid of the $p^{\text{th}}$ cluster, and $|C_p|$ is the number of data points in the $C_p$ cluster.

Fig. 1 shows the general framework of the proposed clustering approach. We consider a combination of k-medoids and k-means as a clustering method. The algorithm works in parallel and iteratively. The found medoids of each batch are fused and re-clustered using k-means to obtain the resulting centroids.

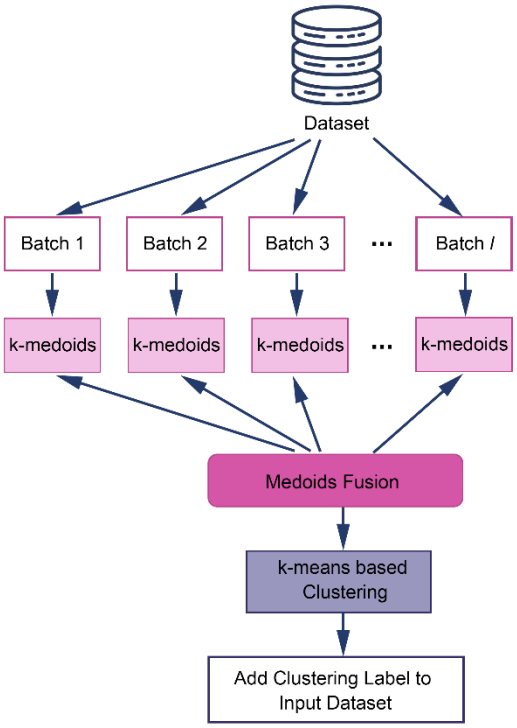



**Fig. 1.** General scheme of the proposed approach

The proposed parallel clustering algorithm is evaluated using various large datasets. They are converted into smaller datasets. We use $k = 2$, $k = 3$, $k = 5$, $k = 10$, and $k = 15$ to cluster the dataset based on batch sizes of 5000, 10000, 15000, and 20000 for different numbers of features. The use of k-medoids and k-means based parallel clustering improves clustering accuracy.

## 4. Experimental results

The sequential version of the k-means algorithm and the proposed approach were implemented on Intel(R) Core(TM) i7-4170HQ with quad-cores running at 2.50 GHz and 8 GB RAM. The experiments were conducted in Python 3.7, and Mini Batch k-means (Sculley, 2010) were implemented in R 3.4.1.

Extensive experiments are conducted on the proposed approach based on k-medoids and k-means and sequential k-means clustering algorithms. The convergence criterion is met after 200 iterations. We used a different number of clusters and various batch sizes to illustrate the impact of both target function value and running

time. YearPredictionMSD and Phone Accelerometer datasets were used (Table 1). They are stored in the UCI machine learning repository (Lichman, 2018). The additional parameters are illustrated in Table 2.

**Table 1.** Description of the experimental datasets

| Dataset | Number of samples | Number of features |
|---|---|---|
| Phone Accelerometer (Stisen et al., 2015) | 1,048,575 | 6 |
| YearPredictionMSD (Bertin-Mahieux et al., 2011) | 515,345 | 90 |

The YearPredictionMSD dataset (Bertin-Mahieux et al., 2011) is a freely available metadata collection from one million popular pieces of music. It contains 515,576 records collected from 1922 to 2011.

The Phone Accelerometer dataset for heterogeneous human activity recognition using smartphones was collected by Stisen et al. (2015). It was designed to compare real-world human activity recognition algorithms. The dataset contains 1,048,575 samples.

**Table 2.** Parameters used during experiments

| Parameter | Value |
|---|---|
| Batch size | 5,000 / 10,000 / 15,000 / 20,000 |
| Number of clusters | 2, 3, 5, 10, 15 |
| Number of iterations | 200 |

FasterPAM (Schubert & Rousseeuw, 2021) was considered as a k-medoids algorithm. The input data is converted into smaller datasets. The algorithms are computed ten times to obtain the average computation time and target function value.

The experiment compares the average computation time and the average objective function value of the proposed approach with the k-means algorithm. Tables 3 and 4 show performance measured based on computation time in seconds relative to the amount of data.

**Table 3.** Experimental results for Phone Accelerometer

| Number of clusters | k-means | | Batch size | Proposed approach | | | |
|---|---|---|---|---|---|---|---|
| | $f$ | $T$ | | $f$ | % | $T$ | % |
| $k$ =2 | $4.1942 \times 10^{10}$ | 5.22 | 5000 | $4.1949 \times 10^{10}$ | -0.02 | 3.17 | +64.67 |
| | | | 10000 | $4.1943 \times 10^{10}$ | -0.00 | 3.19 | +64.64 |
| | | | 15000 | $4.1942 \times 10^{10}$ | 0.00 | 4.20 | +24.29 |
| | | | 20000 | $4.1941 \times 10^{10}$ | +0.00 | 4.84 | +7.85 |
| $k$ =3 | $2.5774 \times 10^{10}$ | 6.99 | 5000 | $2.5774 \times 10^{10}$ | 0.00 | 5.15 | +35.73 |
| | | | 10000 | $2.5773 \times 10^{10}$ | +0.00 | 5.32 | +31.39 |
| | | | 15000 | $2.5773 \times 10^{10}$ | +0.00 | 7.08 | -1.27 |
| | | | 20000 | $2.5772 \times 10^{10}$ | +0.00 | 7.12 | -1.83 |
| $k$ =5 | $1.5673 \times 10^{10}$ | 22.41 | 5000 | $1.5719 \times 10^{10}$ | -0.29 | 7.56 | +196.43 |
| | | | 10000 | $1.5704 \times 10^{10}$ | -0.20 | 8.02 | +179.43 |
| | | | 15000 | $1.5697 \times 10^{10}$ | -0.15 | 9.89 | +126.59 |
| | | | 20000 | $1.5673 \times 10^{10}$ | 0.00 | 10.08 | +122.32 |
| $k$ =10 | $7.8899 \times 10^{9}$ | 160.91 | 5000 | $7.9963 \times 10^{9}$ | -1.33 | 26.02 | +518.41 |
| | | | 10000 | $7.9850 \times 10^{9}$ | -1.19 | 36.35 | +342.67 |
| | | | 15000 | $7.9140 \times 10^{9}$ | -0.30 | 84.39 | +90.67 |
| | | | 20000 | $7.8905 \times 10^{9}$ | -0.00 | 100.50 | +60.11 |
| $k$ =15 | $5.2791 \times 10^{9}$ | 339.77 | 5000 | $5.4125 \times 10^{9}$ | -2.46 | 71.89 | +372.62 |
| | | | 10000 | $5.4051 \times 10^{9}$ | -2.33 | 108.72 | +212.52 |
| | | | 15000 | $5.3928 \times 10^{9}$ | -2.11 | 157.37 | +115.91 |
| | | | 20000 | $5.3803 \times 10^{9}$ | -1.88 | 270.85 | +25.45 |

The tables also show the relative improvement (in %) of the proposed approach compared to k-means. According to Table 3, improvement of the proposed approach is observed for different values and batch sizes for Phone Accelerometer. Compared to k-means, the proposed approach

showed a slight increase in the average objective function value with the number of clusters $k=5$, $k=10$, and $k=15$. When $k=3$, the average value of the objective function was less than of k-means, and when the batch size was 5000, it coincided with k-means. However, the average clustering execution time compared to the second one was 35.73%.

**Table 4.** Experimental results for YearPredictionMSD

| Number of clusters | k-means | | Batch size | Proposed approach | | | |
|---|---|---|---|---|---|---|---|
| | $f$ | $T$ | | $f$ | % | $T$ | % |
| $k$ =2 | 1.0415x10$^9$ | 44.88 | 5000 | 1.0202x10$^9$ | +2.09 | 15.99 | +180.68 |
| | | | 10000 | 1.0202x10$^9$ | +2.09 | 16.03 | +179.98 |
| | | | 15000 | 1.0199x10$^9$ | +2.12 | 17.12 | +162.15 |
| | | | 20000 | 1.0195x10$^9$ | +2.16 | 18.59 | +141.48 |
| $k$ =3 | 9.6949x10$^8$ | 136.82 | 5000 | 9.6975x10$^8$ | -0.03 | 18.51 | +639.17 |
| | | | 10000 | 9.6959x10$^8$ | -0.01 | 21.34 | +541.14 |
| | | | 15000 | 9.6338x10$^8$ | +0.63 | 24.57 | +456.86 |
| | | | 20000 | 9.5843x10$^8$ | +1.15 | 32.80 | +317.13 |
| $k$ =5 | 9.0560x10$^8$ | 327.77 | 5000 | 9.0296x10$^8$ | +0.29 | 55.01 | +495.83 |
| | | | 10000 | 9.0290x10$^8$ | +0.30 | 62.26 | +426.45 |
| | | | 15000 | 9.0283x10$^8$ | +0.31 | 64.57 | +407.62 |
| | | | 20000 | 9.0247x10$^8$ | +0.35 | 72.58 | +351.60 |
| $k$ =10 | – | – | 5000 | 8.2988x10$^8$ | – | 55.04 | – |
| | | | 10000 | 8.2970x10$^8$ | – | 104.40 | – |
| | | | 15000 | 8.2817x10$^8$ | – | 155.65 | – |
| | | | 20000 | 8.2757x10$^8$ | – | 205.27 | – |
| $k$ =15 | – | – | 5000 | 7.9720x10$^8$ | – | 58.16 | – |
| | | | 10000 | 7.9689x10$^8$ | – | 107.84 | – |
| | | | 15000 | 7.9462x10$^8$ | – | 160.49 | – |
| | | | 20000 | 7.9400x10$^8$ | – | 206.51 | – |

Table 4 shows the complexity of big data clustering using k-means, where the results are presented only for $k=2$, $k=3$, and $k=5$. However, clustering using the proposed approach also made obtaining results with the number of clusters $k>10$ possible.

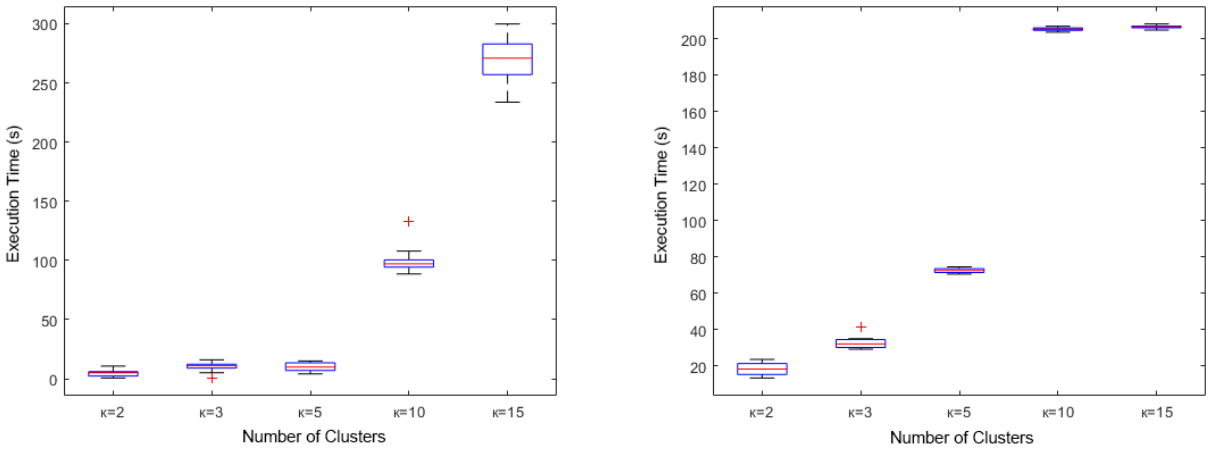
Table 4 shows that the average objective function value of the proposed approach relative to the k-means algorithm improved due to the use of k-medoids and amounted to ~0.95%. At the same time, the average execution time of the proposed algorithm is significantly lower.

Thus, the CPU implementation computational time is significantly reduced for a larger dataset, indicating that the proposed approach effectively solves this issue.

Fig. 2 shows the runtime evaluation of the proposed approach on two datasets using box plots. Each "whisker" box summarizes the results of ten consecutive runs of the proposed approach with a batch size equal to 20,000.

For each dataset, we observe that the execution time of the proposed approach tends to increase with the number of clusters k. The latter corresponds to the Phone Accelerometer dataset (Fig. 2(a)), in which increasing k slows the clustering time. It is more clearly visible when $k>5$. In addition, close clustering performance is observed for the proposed algorithm ($k=10$ and $k=15$) on the YearPredictionMSD dataset (Fig. 2(b)).

Fig. 3 shows the results of the influence of the number of iterations on the objective function value and the execution time for the two considered datasets. The results show that, with the right choice of batch size, the proposed approach based on k-medoids and k-means can be faster than the sequential k-means algorithm (Fig. 3(b, d)).




(a) (b)

**Fig. 2.** Performance evaluation of the proposed approach with different numbers of clusters on datasets: (a) Phone Accelerometer and (b) YearPredictionMSD

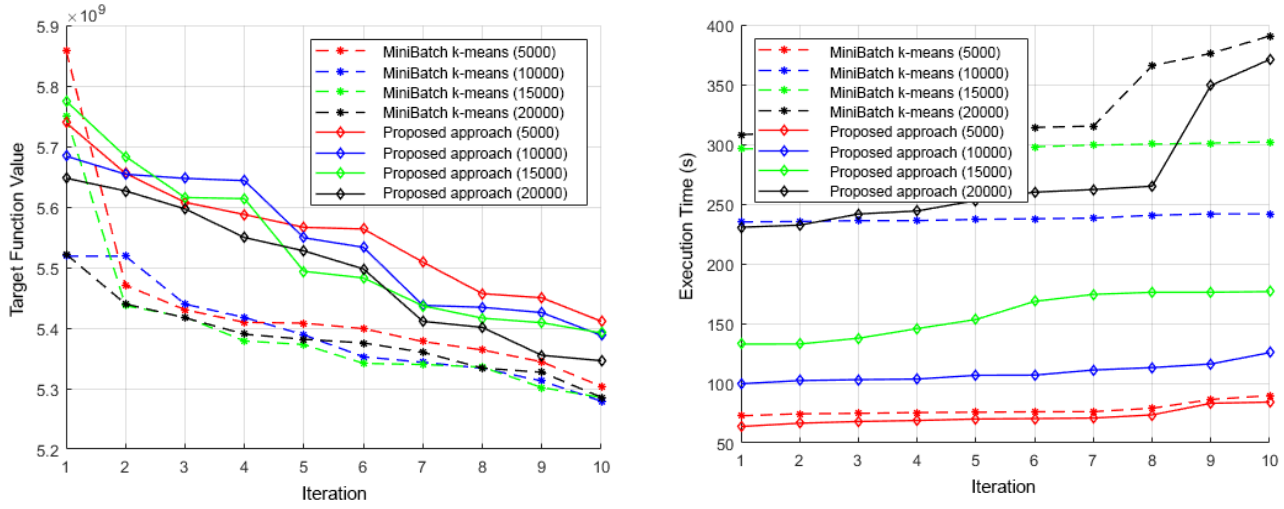
We noticed that for batch sizes equal to 10,000 and 15,000, the execution time of the proposed algorithm is significantly reduced compared to MiniBatch k-means. However, the execution time of the algorithms is almost the same for a batch size of 5,000 (Phone Accelerometer dataset). Moreover, for the YearPredictionMSD dataset, a significant reduction in execution time is observed with an increase in the number of iterations using the proposed approach compared to MiniBatch k-means.

The objective function value of the proposed algorithm shows a sharp decrease for various batch sizes compared to MiniBatch k-means for the YearPredictionMSD dataset. This is more clearly visible with a batch size 20,000 (Fig. 3(c)).

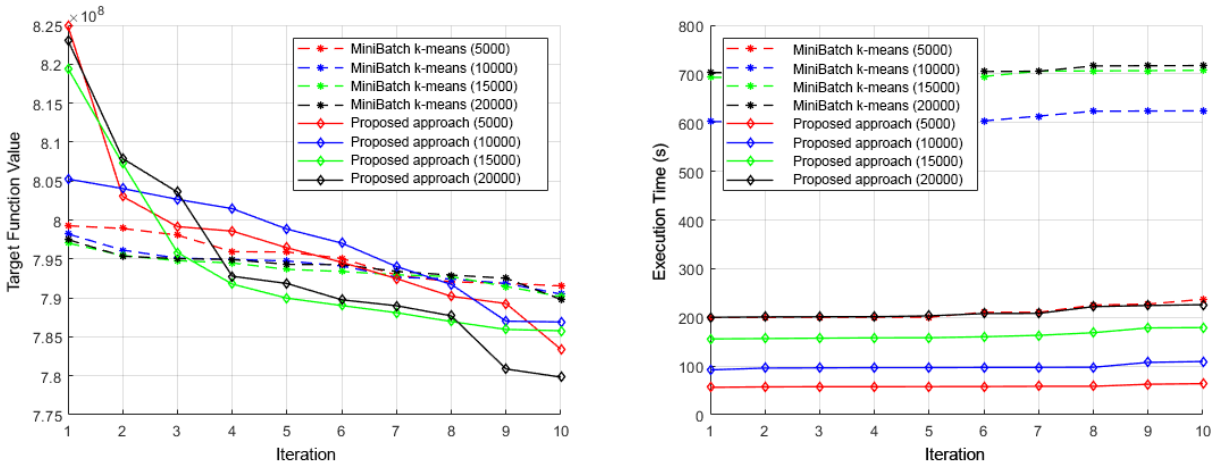
So, comparison with the MiniBatch k-means algorithm proved the effectiveness of the proposed approach for batch sizes of 5000, 10000, 15000, and 20000 (Sculley, 2010).

It can be concluded that the proposed approach proved the reduction of the average objective function value compared to k-means, which, in the case of the YearPredictionMSD dataset, only shows results when $k<10$. It also achieves more accurate results and reduces clustering time compared to MiniBatch k-means.

The results are acceptable because the proposed parallel algorithm based on k-medoids and k-means can quickly achieve efficient results and requires less computational time.




(a) (b)




(c) (d)

**Fig. 3.** Performance comparison of the proposed approach and MiniBatch k-means for different numbers of iterations on the Phone Accelerometer (a, b) and YearPredictionMSD (c, d) datasets

## 5. Conclusion and future work

This study proposed an approach for parallel clustering of large datasets based on the k-medoids and k-means algorithms to improve the clustering accuracy. Comparison of the obtained medoids for the considered datasets with other algorithms proved the superiority of the proposed approach for the best potential medoids calculation. The approach also has the advantage of taking into account the entire large dataset. The distance between medoids and data points was experimentally optimized and calculated using a hybrid approach. Experiments have shown that the proposed parallel implementation of k-medoids together with k-means is an order of magnitude accurate and works faster compared to the classical k-means algorithm when $k=10$ and $k=15$. In contrast, the k-means algorithm does not work when the number of clusters is $k>10$ for a large dataset. The proposed approach can perform efficient clustering of large datasets in a short time and can be applied in various applications.

In the future, it is planned to develop an approach for more accurately determining the batch size, which will improve the quality of the solution and the speed of the massive data clustering algorithm convergence.

## Acknowledgement

This work was supported by the Science Foundation of the State Oil Company of Azerbaijan Republic (SOCAR) (Contract No. 01LR-EF/2024).

## References

Aggarwal, C. C. & Reddy, C. K. (2014). Data clustering: algorithms and applications. New York: CRC Press. https://doi.org/10.1201/9781315373515

Ahmadov, E. Y. (2023). Comparative analysis of k-means and fuzzy c-means algorithms on demographic data using the PCA method. Problems of Information Technology, 14(1), 15-22. https://doi.org/10.25045/jpit.v14.i1.03

Alguliyev, R., Aliguliyev, R., Karimov, R., & Bagirov, A. (2016). Batch clustering algorithm for big data sets. In 10th IEEE International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, October 2016 (pp. 79–82). https://doi.org/10.1109/ICAICT.2016.799165

Bagirov, A. M., Taheri, S., & Ordin, B. (2022). An adaptive *k*-medians clustering algorithm. Problems of Information Technology, 13(2), 3-15. https://doi.org/10.25045/jpit.v13.i2.01

Bahmani, B., Moseley, B., Vattani, A., Kumar, R., & Vassilvitskii, S. (2012). Scalable k-means++. Proceedings of the VLDB Endowment, 5(7), 622-633. https://doi.org/10.14778/2180912.2180915

Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B., & Lamere, P. (2011). The Million Song Dataset. 2011 International Society for Music Information Retrieval Conference (ISMIR) (pp. 591-596).

Boutsidis, C., Drineas, P., & Mahoney, M. W. (2009). Unsupervised feature selection for the k-means clustering problem. 22nd International Conference on Neural Information Processing Systems (NIPS), Vancouver, Canada, December 2009 (pp. 153-161).

Han, J., Kamber, M., & Tung, A. K. H. (2001). Spatial clustering methods in data mining: A survey. In H.J. Miller, J. Han (Eds), Geographic data mining and knowledge discovery (pp. 1-29).

Hu, H., Liu, J., Zhang, X., & Fang, M. (2023). An Effective and Adaptable K-means Algorithm for Big Data Cluster Analysis. Pattern Recognition, 139, 109404. https://doi.org/10.1016/j.patcog.2023.109404

Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Information Sciences, 622, 178-210. https://doi.org/10.1016/j.ins.2022.11.139

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. Pattern Recognition Letters, 31(8), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011

Karmitsa, N., Bagirov, A.M., & Taheri, S. (2018). Clustering in large data sets with the limited memory bundle method, Pattern Recognition, 83, 245–249. https://doi.org/10.1016/j.patcog.2018.05.028

Kaufman, L. & Rousseeuw, P. J. (1990). Finding groups in data: An introduction to cluster analysis. New York: Wiley.

Lenssen, L. & Schubert, E. (2024). Medoid silhouette clustering with automatic cluster number selection. Information Systems, 120, 102290. https://doi.org/10.1016/j.is.2023.102290

Li, M., Zhang, Y., Liu, S., Liu, Z., & Zhu, X. (2023). Simple multiple kernel k-means with kernel weight regularization. Information Fusion, 100, 101902. https://doi.org/10.1016/j.inffus.2023.101902

Lichman, M. (2018). University of California. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml Accessed 7 July 2018.

Mussabayev, R., Mladenovic, N., Jarboui, B., & Mussabayev, R. (2023). How to Use K-means for Big Data Clustering? Pattern Recognition, 137, 109269. https://doi.org/10.1016/j.patcog.2022.109269

Meng, Y., Liang, J., Cao, F., & He, Y. (2018). A new distance with derivative information for functional k-means clustering algorithm, Information Sciences, 463-464, 166-185. https://doi.org/10.1016/j.ins.2018.06.035

Parker, J. K. & Hall, L. O. (2014). Accelerating fuzzy-c means using an estimated subsample size. IEEE Transactions on Fuzzy Systems, 22(5), 1229-1244. https://doi.org/10.1109/TFUZZ.2013.2286993

Ping, Y., Li, H., Hao, B., Guo, C., & Wang, B. (2024). Beyond k-Means++: Towards better cluster exploration with geometrical information. Pattern Recognition, 146, 110036. https://doi.org/10.1016/j.patcog.2023.110036

Sculley, D. (2010). Web-scale k-means clustering. 2010 ACM International conference on World Wide Web (WWW), North Carolina, USA, April 2010 (pp. 1177-1178). https://doi.org/10.1145/1772690.1772862

Schubert, E. & Rousseeuw, P. J. (2021). Fast and eager k-medoids clustering: O(k) runtime improvement of the PAM, CLARA, and CLARANS algorithms, Information Systems, 101, 101804. https://doi.org/10.1016/j.is.2021.101804

Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T. S., Kjærgaard, M. B., Dey, A., Sonne, T., & Jensen, M. M. (2015). Smart devices are different: assessing and mitigating mobile sensing heterogeneities for activity recognition. In ACM Conference on Embedded Networked Sensor Systems (SenSys), Seoul, South Korea, November 2015 (pp. 127-140). https://doi.org/10.1145/2809695.2809718

Thompson, S. K. (1987). Sample size for estimating multinomial proportions, The American Statistician. 41, 42-46. https://doi.org/10.2307/2684318

Wang, Y., Krishna Saraswat, S., & Elyasi Komari, I. (2022). Big data analysis using a parallel ensemble clustering architecture and an unsupervised feature selection approach. Journal of King Saud University - Computer and Information Sciences, 35(1), 270-282. https://doi.org/10.1016/j.jksuci.2022.11.016

Zein, A. A., Dowaji, S., & Al-Khayatt, M. I. (2023). Clustering-based method for big spatial data partitioning. Measurement: Sensors, 27, 100731. https://doi.org/10.1016/j.measen.2023.100731

Zhao, W. L., Deng, C. H., & Ngo, C. W. (2018). k-means: A revisit. Neurocomputing, 291, 195-206. https://doi.org/10.1016/j.neucom.2018.02.072

Zhang, H., Li, J., Zhang, J., & Dong, Y. (2024). Speeding up k-means clustering in high dimensions by pruning unnecessary distance computations. Knowledge-Based Systems, 284, 111262. https://doi.org/10.1016/j.knosys.2023.111262

Zhang, J., Wolfram, D., & Ma, F. (2023). The impact of big data on research methods in information science. Data and Information Management, 7(2), 100038. https://doi.org/10.1016/j.dim.2023.100038

Zhu, X., Sun, J., He, Z., Jiang, J., & Wang, Z. (2023). Staleness-Reduction Mini-Batch K-Means. IEEE Transactions on Neural Networks and Learning Systems, 1-13. https://doi.org/10.1109/TNNLS.2023.3279122