**International Journal on Semantic Web and Information Systems**
Volume 18 • Issue 1

# A Parallel Fractional Lion Algorithm for Data Clustering Based on MapReduce Cluster Framework

Satish Chander, Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, India*
P. Vijaya, Department of Mathematics and Computer Science, Modern College of Business and Science, Muscat, Oman
Praveen Dhyani, Banasthali University, Jaipur Campus, India

## ABSTRACT

This work introduces a parallel clustering algorithm by modifying the existing fractional lion algorithm (FLA). The proposed work replaces the conventional Euclidean distance measure with the Bhattacharya distance measure to newly propose the improved FLA (IMR-FLA). The proposed IMR-FLA is implemented in both the mapper and the reducer in the MapReduce framework to achieve the parallel clustering. The experimentation of the proposed IMR-FLA is done by using six standard databases, namely Pima Indian diabetes dataset, heart disease dataset, hepatitis dataset, localization dataset, breast cancer dataset, and skin segmentation dataset, from the UCI repository. The proposed IMR-FLA has the overall improved Jaccard coefficient value of 0.9357, 0.6572, 0.7462, 0.5944, 0.9418, and 0.8680, for each dataset. Similarly, the proposed IMR-FLA algorithm has outclassed other classifiers' performance with the clustering accuracy value of 0.9674, 0.9471, 0.9677, 0.777, 0.9023, and 0.9585, respectively, for the experimental databases.

## KEYWORDS

Bhattacharya Distance, Big Data, FLA, MapReduce, Parallel Clustering

## 1. INTRODUCTION

The evolution of big data has been the trend nowadays since various communities share data in internet sources. The huge flow of data on the Internet has given rise to various data mining techniques, out of which data clustering and classification have been on the trend. Manual processing of large data from various sources can lead to error, and hence, automation of the data processing scheme is an emerging topic this decade. Big data contains the data from the various domains, and hence, the clustering of the information concerning their domains is necessary for retrieving the information (Gowanlock, *et al.,* 2017). The various applications, such as image segmentation, data mining, biomedical and information retrieval, require data clustering (Rahnema, et al., 2020). Besides, it is widely used in Internet of things (IoT) device applications and related services (B.B. Gupta, and Megha Quamara, 2018). It requires a suitable encryption technique for the authentication of the information sharing (Christian *et al.,* 2021) (Anupama *et al.,* 2021), (C. Yu *et al,* 2018). Analytic processing from the

DOI: 10.4018/IJSWIS.297034 *Corresponding Author

large data domains, such as science and commercial application, possesses various challenges to the parallel clustering schemes due to their computational complexity and storage (Amintoosi, et al., 2020). Clustering of the data improves the knowledge discovery from the large volume of data (Zhou and Yang, 2020). Clustering is one of the important data mining schemes, which helps the user to retrieve the data from a large volume of the data more effectively by considering the load characteristic curves. The clustering technique groups the data belonging to the same cluster by calculating the distance measure (Kaur and Kumar, 2021). The cluster groups formed by the clustering algorithm can be grouped into homogeneous and heterogeneous clusters (Sreedhar, *et al.,* 2017).

Building the clustering algorithms in the parallel stream has significant challenges since parallel processing is necessary to build the multiprocessor hardware system with specialized chips (Tripathi, et al, 2020). It is also widely used in recommendations of students, which motivates further studies (T.T.H. Bui, et al., 2021) (N. T. Hung, 2020) (N.T. Hung, and J.C. Chang, 2019). It makes these algorithms utilize the high-speed computer systems effectively. Literature has classified the clustering schemes as hierarchal clustering and partitional clustering (Xua, et al., 2020). The clustering methods, like squared-error methods, are categorized under the partitional clustering algorithms, while the techniques, such as the Complete-link method and single-link method, are hierarchical clustering. Normally, the partitional clustering algorithm takes the pattern matrix from the data as the training input. From the data with large volumes, it isn't easy to generate the pattern matrix for each incoming data, and hence, parallel processing of the pattern matrix improves the clustering process. In parallel clustering, the distance measure between the data points and the cluster center is calculated. Thus, the similarity between the large volumes of the data is achieved (Sharma and Seal, 2020). Properties, such as continuous streaming and the large volume of the data, can be solved using the MapReduce framework with the clustering algorithm. The MapReduce was developed by (Dean & Ghemawat, 2008) at Google to process large data continuously. Incorporating the MapReduce framework with the clustering algorithm increases the strength of the clustering process and makes the algorithm suitable for automatic parallelism and distribution. Besides, the MapReduce concept makes the clustering algorithm to be fault-tolerant.

The clustering algorithms discussed in the literature merely depend on the distance measure between the various data points and the cluster center. The k-means clustering algorithm (Sreedhar, *et al.,* 2017; Tang, *et al.,* 2017) discussed in the literature was primarily introduced for parallel clustering, which calculates the distance measure between the data points and the cluster for the clustering process. Besides, the k-means clustering algorithm contains some regression distance calculations, making the algorithm complex in the large data environment. In the literature work (Tang, *et al.,* 2017), the Manhattan distance is replaced with the Euclidean distance to calculate the distance measure. Literature has suggested various parallel clustering schemes, such as Hadoop (Chaturbhuj & Chaudhary, 2016), Parallel Clustered Particle Swarm Optimization (Hossain, *et al.,* 2016), etc. Traditional clustering algorithms fail to cluster the large volume data, and hence, literature has introduced the sequential clustering approach based on the traditional k- means algorithm for data clustering. The various challenges involved in the k-means algorithm are no information sharing between the cluster and the high sensitivity. These challenges can be avoided using the hybridization of the optimization algorithms (Rahnema, et al., 2020). Traditional clustering schemes require a set of highly efficient supercomputers to cluster the big data. Besides, the cost of the clustering system and the quality of the clusters suggest that the clustering techniques are not up to the standard (Pima, https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes). Also, the employment of the noniterative clustering approaches, like MapReduce and Hadoop, needs re-clustering during the arrival of new data (Wang, et al., 2020).

The primary intention of this paper is to design a parallel clustering algorithm by modifying the existing Fractional Lion Algorithm (FLA). This paper proposes an improved fractional lion algorithm (IMR-FLA) to conduct a parallel clustering analysis based on the MapReduce cluster framework. Here, the FLA employed in (Chander, *et al.,* 2016) is implemented in the MapReduce framework

for achieving the parallelization. The Euclidean distance measure used for the distance calculation is replaced with the Bhattacharyya distance. The proposed IMR-FLA is placed in both the mapper and the reducer for clustering the systematically selected data samples. Then, the cluster centroids found from each mapper are combined on the reducer side, and further clustering is done to complete the parallelism.

The major contributions of this research for the data clustering in the big data are explained as follows:

- Firstly, the research introduces the parallel data clustering, termed IMR-FLA, with the MapReduce framework by adapting the FLA algorithm for parallelization.
- Secondly, the distance measure calculation done in the FLA is done based on the Bhattacharya distance measure.

The rest of this paper is organized as follows: Section 1 introduces the parallel data clustering scheme, and the various literature suggested for the parallel clustering is discussed in section 2. Section 3 briefs the proposed IMR-FLA for the parallel clustering of the database. The simulation results achieved by the proposed IMR-FLA algorithm are discussed in section 4, and section 5 concludes the research work.

## 2. MOTIVATION

### 2.1 Literature Survey

This section presents the various literary works dealing with the parallel data clustering concept.

Jing *et al.* (2020) devised an immune evolutionary algorithm for medical data clustering in the cloud environment. In this, the developed immune algorithm accurately classifies the data with a reduced error rate. Besides, the performance enhancement is achieved in terms of accuracy, which is low. Kareem *et al.* (2020) devised a hybrid optimization algorithm for the data clustering with the Tabu search algorithm. In this, the Whale optimization algorithm is used for the improvement of the convergence rate. They achieved high quality of clustering with a reduced number of iterations. The system degrades the performance while comparing to the existing techniques while considering some of the datasets. Kotadi and Raju, (2020) devised a data clustering algorithm for large dataset clustering. In this, K++ means clustering is employed for the data clustering. They achieved better grouping of the data compared to the K-means clustering algorithm. The drawback of the system is that it doesn't use any optimization algorithm for clustering. Michele *et al*. (2020) devised a fast and efficient data clustering algorithm for big data. In this, the complex hierarchical clustering algorithm CLUBS+ is employed for better scalability in clustering. They achieved less execution time, but it is not applicable for the spark architecture. Laxmi *et al*. (2020) devised a clustering and indexing of the document for the big data. In this, Non-Negative Matrix Factorization and k-means clustering are employed for the data clustering. The feature extraction is used to enhance the performance. They achieved better accuracy by testing in the MapReduce framework. Here, the optimization is not devised for efficient clustering. M. Gowanlock, D. *et al.* (2017) presented the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) technique for the parallel data clustering of the data present in the scientific stream. In this model, various algorithms were executed in parallel to leverage the exponential growth of the scientific data. The model has less computational overhead for computing a large volume of data but is more sensitive to the distribution of the input dataset. Sreedhar, C. *et al.* (2017) presented the K-Means Hadoop MapReduce (KM-HMR) by integrating the k-means and the Hadoop framework. The algorithm ensured the quality of the clustering process by calculating the maximum intra-cluster and minimum inter-cluster distances. The algorithm is more suited for the large data environment but lacked in the scheduling strategy. Tang, Z., *et al.* (2017)

presented the parallel implementation of the Improved K-Means Algorithm (IMR-KCA) along with the MapReduce framework. The model had computed the multiple clustering centers from various data points, and then the data were clustered based on the cluster centers. The model also had the selection model for effectively calculating the suitable cluster centroid.

The tabular form of the literature review is displayed in Table 1.

### 2.2 Challenges

The various challenges prevailing in the parallel data clustering are enlisted below:

- Big data contains the data from the various domains, and hence, the clustering of the information concerning their domains is necessary for retrieving the information (Gowanlock, *et al.,* 2017).
- The major challenge involved in the parallel data clustering scheme is identifying the similarities and the dissimilarities between the various data points, and the calculated similarity measure is used for the clustering (Sreedhar, *et al.,* 2017).
- The big data from various data streams comply with volume, velocity, variety, veracity, value, and volatility. Thus, the parallel data clustering must precisely measure the cluster similarities to classify the data (Sreedhar, *et al.,* 2017).
- The k-means clustering algorithm (Sreedhar, *et al.,* 2017) contains regression distance calculations, making the algorithm complex in the large data environment.

The major challenge associated with the existing techniques and the conventional FLA algorithm compromises cooperative interaction or competitive interaction. Besides, when data are insufficient, the performance will degrade. Hence, the improved FLA is introduced to overcome the drawbacks faced by the existing systems. The IMP-FLA consists of the Battacharya distance measure, which is widely used for applications related to data clustering because it reduces the computational complexity. Besides, it calculates the similarities between the clusters with less error probability.

## 3. PROPOSED MAPREDUCE CLUSTER FRAMEWORK BASED ON THE IMR-FLA

This section presents the MapReduce cluster framework by adapting the existing FLA for parallelization. Figure 1 depicts the block diagram of the parallel data clustering scheme with the proposed IMR-FLA algorithm. The various domains, such as social networks, scientific data, and medical data, are present in the database $D$. Initially, the database $D$ is classified randomly into $S$ data sources for simplifying the clustering process. Each data source is fed to the mapper of the MapReduce framework. The mapper contains the proposed IMR-FLA algorithm, which finds the suitable cluster centroids for each data cluster, and clusters the data accordingly.

The proposed scheme is built on the MapReduce framework with the parallelization of the IMR-FLA algorithm. The big data arriving from various sources are randomly split into different sources and are provided to the mapper. The mapper performs the clustering of the data sources with the help of the proposed IMR-FLA algorithm. The basic steps involved in the proposed parallel clustering scheme are explained as follows:

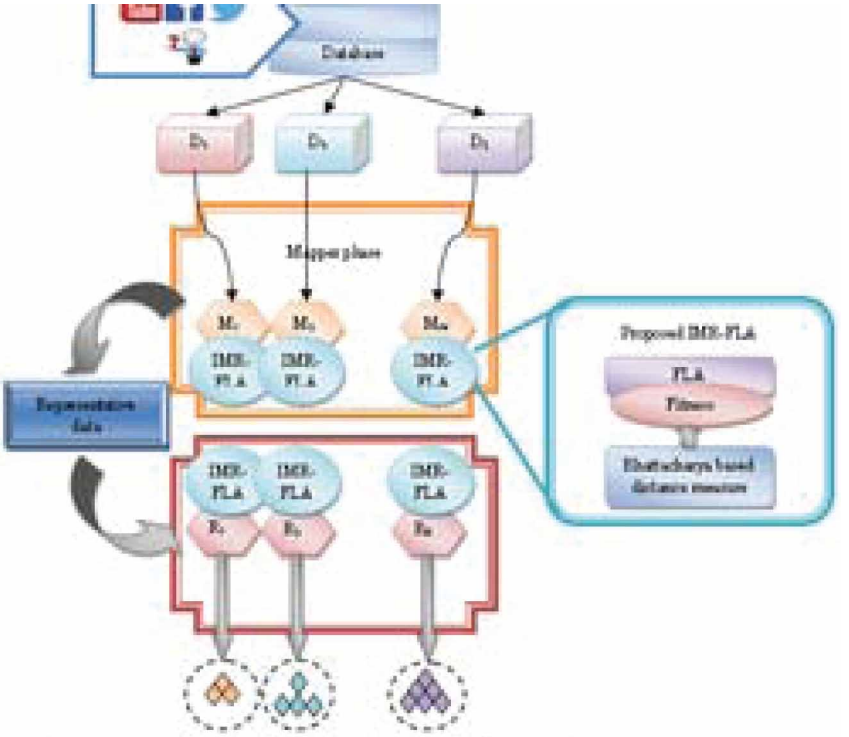
1. Consider the database $D$ contains information from many domains. Since the database $D$ has a large size of $P * Q$, the database is randomly split into $S$ data sources to simplify the clustering process. The data sources are represented as follows:

$$D = \left\{ D_1, D_2, \ldots, D_i, \ldots, D_S \right\} \tag{1}$$

**Table 1. Literature review of existing methods**

| Author | Objective | Contribution | Advantages | Disadvantages | Findings |
|---|---|---|---|---|---|
| (Yu, et al., 2020) | To enhance the accuracy of data clustering. | Modified immune evolutionary method for medical data clustering | Achieved enhanced performance in terms of reduced error rate. | Deep learning is not employed, and accuracy is low. | Accuracy-85% |
| (Ghany, et al., 2020) | To achieve the efficient data clustering. | Whale optimization algorithm with Tabu search(WOATS). | Efficient data clustering and can be used for real-life applications. | For some datasets like survival, the performance of the system degrades. | Silhouette index-0.7435. |
| (Divya and Raju, 2020) | To improve the clustering for the large-scale dataset. | K-means++ clustering | Efficient performance compared to K-means clustering. | None of the optimization technique is employed | Time taken=105.51 sec for k=6. |
| (Ianni, et al., 2020) | To improve the scalabilty of clustering. | complex hierarchical clustering algorithm CLUBS+ | Achieved better scalability | Not suitable for Spark architecture. | Execution time. |
| (Lydia, et al., 2020) | To enhance the parallel clustering. | Non-Negative Matrix Factorization and k-means clustering. | Accuracy is tested using the MapReduce framework. | None of the optimizations of machine learning is used. | Computational time. |
| M. Gowanlock, D. et al. (2017) | To enhance parallel clustering in shared memory. | Density-Based Spatial Clustering of Applications with Noise (DBSCAN) | Less computation overhead. | Highly sensitive to the distribution of the input data. | Execution time. |
| Sreedhar, C. et al. (2017) | To cluster the data in the Hadoop environment. | K-Means Hadoop MapReduce (KM-HMR) by integrating the k-means and the Hadoop framework. | The algorithm ensured the quality of the clustering process by calculating the maximum intra-cluster and minimum inter-cluster distances. | Lacked in the scheduling strategy. | Execution period-3876.23s |
| Tang, Z., et al. (2017) | To achieve redundancy reduction in the map-reduce framework. | Improved k-means clustering algorithm (IMR-KCA) | It effectively calculated the suitable cluster centroid. | Not suitable for high iterations. | Total time and average iteration time. |
| Proposed Method | To achieve parallel data clustering. | Improved Fractional Lion Algorithm (IMP-FLA) | Effectively finds the cluster centroid. | None | Clustering accuracy-96.27%, Jaccard coefficient-94.18% |

**Figure 1. Block diagram of the proposed parallel data clustering with IMR-FLA**




where, $D_i$ refers to the $i^{th}$ data source subjected to the data clustering.

2. In the next step, the data points present in the sources are provided to the mappers. Consider the system has $M$ number of mappers. The proposed IMR-FLA algorithm in each mapper cluster the data source by finding the suitable cluster centroids. The output of each mapper is the optimally found cluster centroid for the data source $D_i$. Consider the IMR-FLA algorithm that clusters the data into $K$ number of clusters. Thus, the output of the mapper $M_i$ is expressed as:

$$M_i = \left\{C_1, C_2, \ldots, C_k, \ldots, C_K\right\} \tag{2}$$

where, $C_k$ refers to the $k^{th}$ cluster centroid provided by the $i^{th}$ mapper.

3. In the next step, the output of each mapper is combined to generate the representative data, given as follows:

$$Y = \left\{M_1, M_2, \ldots, M_i, \ldots, M_M\right\} \tag{3}$$

where, the term $M_i$ refers to the output of the $i^{th}$ mapper,

4. The representative data $Z$ serves as an input to the reducer, and consider there are $R$ reducers in the system, given as $I = \left\{I_1, I_2, \ldots, I_r, \ldots, I_R\right\}$. The reducer utilizes the IMR-FLA algorithm

**Figure 2. Solution encoding**

for the clustering process. Each reducer finds the similarity between the cluster centroids from the mapper and further refines the clustering process to find the new cluster centroids.

5. Then, based on the new optimal centroids from the IMR-FLA algorithm, the data clustering is done. The final clustered data is represented as:

$$O = \left\{O_1, O_2, \ldots O_r, \ldots, O_R\right\} \tag{4}$$

where, the term $O_r$ represents the data present in the $r^{th}$ reducer. The final output $O$ has a size equal to the size of the database $D$. Hence, the output of the $r^{th}$ reducer is represented as:

$$O_r = \left\{G_1, G_2, \ldots G_k, \ldots, G_K\right\} \tag{5}$$

where, $G_k$ refers to the $k^{th}$ cluster of $r^{th}$ reducer.

## 3.1 Proposed IMR-FLA

This section presents the algorithmic description of the existing FLA algorithm. This work uses the existing FLA algorithm for building the proposed IMR-FLA algorithm for the parallel clustering of the database. The existing FLA algorithm is the modified form of the Lion algorithm with fractional calculus. As the FLA algorithm derives the characteristics of the behavior of the Lion for the solution update, the solution requires a male Lion, a female, and two nomadic lions for obtaining the required optimal solution. The following subsections describe the algorithm in detail.

### *3.1.1 Solution Encoding*

The proposed IMR-FLA found the appropriate cluster centroid points from the data, and based on the optimal cluster centroids, the clustering of the data is done. The IMR-FLA finds $K$ cluster centroids to cluster the data in the database $D_i$ and the solution vector of the IMR-FLA is represented in Figure 2. The solution can be represented as a vector of dimension $1 \times K$, from which the proposed IMR-FLA finds the optimal centroids, depending on the fitness function.

### *3.1.2 Fitness Calculation: Bhattacharya Based Distance Measure*

In the existing work, the distance parameter used in the fitness function used the Euclidean distance measure. The Euclidean distance measures the distance between the centroids of the clusters and the data points, but the Euclidean distance measures data points present in other clusters. For achieving parallel clustering, it is necessary to consider the distance measure between the data points within and the other clusters. Thus, this work considers the Bhattacharya-based distance measure (Nielsen & Boltz, 2011). The proposed IMR-FLA algorithm utilized the Bhattacharya based distance measure for the fitness evaluation, and the expression for the fitness measure is expressed as follows:

$$F = \sum_{k=1}^{K} \sum_{\substack{j=1 \\ j \in k}}^{D_i} J(d_j, C_k) \tag{6}$$

where, $J(d_j, C_k)$ refers to the Bhattacharya-based distance measure $D_i$ represents the $i^{th}$ data source provided for the clustering. The following equation gives the expression for the distance measure:

$$J(d_j, C_k) = -\ln\left(B\left(d_j, C_k\right)\right) \tag{7}$$

where, the term $B$ refers to the Bhattacharya coefficient, and its value can be expressed as:

$$B\left(d_j, C_k\right) = \sum_{s \in j.k} \sqrt{d_j(s).C_k(s)} \tag{8}$$

where, $d_j$ and $C_k$ refer to the data points and the cluster centroids present in the data source $D_i$.

### *3.1.3 Algorithmic Description*

The algorithmic steps of the existing FLA algorithm are briefed as follows:

1. **Generating the pride for the solution constraint:** As the behavior of the Lion inspires the lion algorithm, the FLA algorithm considers the solution as finding the best/suitable male Lion, female Lion, and nomad lion. Hence, in the initial step, the suitable pride with the male Lion, female Lion, and the nomad lion is generated, and it is represented as follows:

$$Z = \left\{Z^U, Z^V, Z^W\right\} \tag{9}$$

where, the terms $Z^U$, $Z^U$, and $Z^U$ are the solutions representing the male Lion, female Lion, and nomad lion, respectively. The length of the solution vector will be the total cluster centroid for clustering the database.

2. **Calculating the fertility of the male Lion and the female lioness:** The evaluation of the fertility of the male and the female Lion is required to avoid the solution from converging at the local optima. The fertility rate of both the male and the female Lion tends to provide new solutions. The fertility rate of the solution representing the male Lion can be defined through the laggardness rate. Similarly, the fertility of the female Lion solutions is evaluated with the parameters such as laggardness rate and the sterility rate. The choice of the fertility evaluations is made irrespective of gender, and the fertility of the nomad lion is neglected here. Thus, based on the fertility rate of the female Lion, the female lion count is updated, and it is calculated as follows:

$$Z_c^{V+} = \begin{Bmatrix} Z_d^{V+}; if\ c = d \\ Z_c^V; otherwise \end{Bmatrix} \tag{10}$$

where, $c$ and $d$ define the vector for updating the female lion count, and the value of the $Z_d^{V+}$ depends on the following expression:

$$Z_d^{V+} = \min\left[Z_c^{\max}, \max(Z_d^{\min}, \nabla_d\right] \tag{11}$$

where, $\Delta_d$ defines the update function for the female count, and its value depends on the random integer $u_1$ and $u_2$. The expression for the update function $\Delta_d$ is defined in the following equation,

$$\nabla_d = Z\left[x_d^V + (0.1u_2 - 0.05)\left(Z_d^U - u_1 Z_d^V\right)\right] \tag{12}$$

The vector elements $c$ and $d$ defined for calculating the fertility rate of the female Lion are defined under the solution vector. The random integers $u_1$ and $u_2$ further refine the female lion update and its value range between 0 and 1.

3. **Performing the crossover and mutation for updating the solution:** The evolutionary optimization approaches utilized the crossover and the mutation operations to find the new solutions. In the FLA algorithm performing the crossover and the mutation operations yield the four new cubs:
   a. **Crossover:** In the crossover operation, the new solutions are identified by applying the crossover operator, and this value depends on the crossover probability. The crossover operation provides the new solution formerly represented as the lion cubs. The expression for the lion cub formation is expressed as:

$$Z^G(cubs) = L \circ Z^U + \overline{L} \circ Z^V \tag{13}$$

where, the term $L$ represents the crossover mask, and the length of the crossover mask varies from 1 to 4, and the operator $\circ$ indicates the Hadamard product.

   b. **Mutation:** Then, the output from the crossover operation is provided to the mutation to refine the new solution further. The mutation process also has the mutation probability for identifying the new solutions, and the solutions from the mutation are expressed as $Z^Q$.
   c. **Clustering the solution based on the gender class:** Here, the solution from the mutation is subjected to the clustering since the solution contains both the male and the female lion cubs. Thus, the male and female cubs present in the solution are represented as $Z^{U\_Q}$ and $Z^{V\_Q}$.
4. **Defining the growth of the lion cubs:** The solutions represented by the lion cubs can be updated by defining the parameters, such as growth rate, and the cub growth function is the mutation function.
5. **Updating the pride of the Lion based on fractional calculus:** The existing work utilizes the fractional theory for updating the pride of the male Lion. The FLA (Chander, *et al.,* 2016) had utilized the first-order fractional calculus for defining the pride of the male Lion. The pride of the Lion is generated through the calculation of the fitness, and if the fitness of the male solutions at the iteration $\left(t+1\right)$ and $t$ is the same, then the solution is updated as follows:

$$Z_{t+1}^U = Z_t^U \tag{14}$$

$$X_{l+1}^M - X_l^M = 0 \tag{15}$$

In the above expression, the solution derivation of the first-order calculus is applied, and it is expressed as follows:

$$H^{\alpha}\left[Z_{t+1}^{U}\right]=0 \tag{16}$$

The application of the fractional calculus for the pride generation yields the following updated expression:

$$Z^{U}=\alpha Z_{t}^{U}+\frac{1}{2}\alpha Z_{t-1}^{U} \tag{17}$$

where, the term $\alpha$ refers to the fractional order.

6. **Solution update through the territorial defense:** Besides the male and the female Lion, the solution space also has two nomadic lions, represented as $Z_1^W$ and $Z_2^W$, respectively. The survival of the male Lion in the solution depends on the fitness of both nomadic lions. The expression for the territorial defense is defined as follows:

$$N_2^W=\exp\left(\frac{w_2}{\max\left(w_1,w_2\right)}\right)\frac{\max\left(f\left(Z_1^W\right),f\left(Z_2^W\right)\right)}{f\left(Z_2^W\right)} \tag{18}$$

where, $w_1$ and $w_1$ indicate the Euclidean distance from the nomad lion 1 & 2 and the male Lion, respectively.

7. **Solution update through the territorial takeover:** Here, the solution update occurs when the cub's fitness is greater than the male Lion's. After the takeover, the previous male Lion's sterility rate is considered zero, and the solution counts through the increment of the solution generation count with one.
8. **Termination:** The algorithm gets terminated in the final iteration, and at the end of the iteration, the FLA provides the optimal solution.

### 3.2 IMR-FLA: Adapting FLA to the Parallel Data Clustering

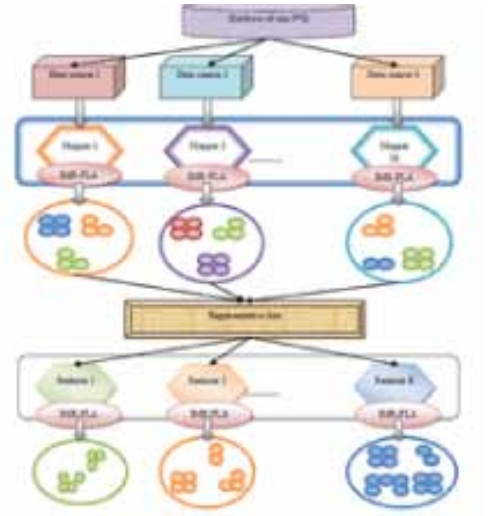
This section presents the proposed IMR-FLA algorithm for the parallel clustering of the database $D$. The proposed IMR-FLA algorithm is the parallelization of the existing FLA, and it is used along with the MapReduce framework for parallel computation of each data source. Figure 3 presents the architecture of the proposed IMR-FLA algorithm with the MapReduce framework for the classification of big data. The proposed IMR-FLA algorithm is placed in both the mapper and the reducer phase of the MapReduce framework to achieve the parallel clustering process. The proposed IMR-FLA framework is set at both the mapper and reducer, and thus, the architecture performs two levels of clustering.

Various blocks involved in the proposed parallel clustering framework and their description is given as follows.

#### *3.2.1 Data Sources for the Clustering*

The database subjected to the parallel clustering is subdivided into $S$ data sources to reduce the system's complexity. The proposed system establishes each mapper for the individual data sources. Thus, consider that the system has $M$ mappers and $R$ reducers for the parallel clustering. The data sources are given in parallel to each mapper for the clustering.

**Figure 3. The architecture of the MapReduce cluster framework based on proposed the IMR-FLA**



### *3.2.2 Working of the Mapper*

The $M$ number of mappers present in the system is provided with the IMR-FLA algorithm for finding the optimal cluster centroid. The mapper phase contains the proposed IMR-FLA algorithm for clustering the data in each data source, and the clustering is done based on the optimal cluster centroids from the proposed algorithm.

### *3.2.3 Generating the Representative Data*

The clustered information from each mapper is gathered together to form the representative data. The generated representative data has the size of $X * Y$ , and it serves as the input to the reducer of the proposed system.

### *3.2.4 Working of the Reducer*

The proposed parallel data clustering system contains a total of $R$ reducers. The reducer helps in identifying similar data points among the various clusters obtained through the mapper phase. The clustered data points present in the representative data are further subjected to clustering with the help of the IMR-FLA algorithm. Finally, the new optimal centroids from the IMR-FLA in the reducer find the final clustered data. Thus, the parallel clustering process using the proposed IMR-FLA shows that the size of the clustered data from $R$ reducers is equivalent to that of the database $D$ .

The description of the flow of the parallel data clustering process based on the proposed IMR-FLA is briefed in Algorithm 1. The database provided as the input to the proposed scheme is randomly subjected to division and provided as a smaller-sized data source for each mapper of the model. The proposed IMR-FLA clustering algorithm.

## 4. RESULTS AND DISCUSSION

This section presents the experimental results of the proposed IMR-FLA algorithm, and the results of the proposed approach are compared with the other techniques implemented in the parallel framework. The simulation of the proposed IMR-FLA algorithm is analyzed with the six datasets taken from the UCI machine repository.

**Algorithm 1. Pseudocode of the proposed IMR-FLA for parallel clustering**

| Sl. no | Algorithm of the proposed IMR-FLA |
|---|---|
| 1 | **Input: database** |
| 2 | **Output: data clusters** |
| 3 | **Begin** |
| 4 | Initialize the mapper |
| 5 | Initialize the reducer |
| 6 | Randomly split the database into $S$ sources |
| 7 | **// IMR-FLA algorithm** |
| 8 | **Begin** |
| 9 | **For** each data points |
| 10 | Find the fitness measure |
| 11 | Calculate the optimal centroid |
| 12 | Find the clusters based on the optimal centroid |
| 13 | **End for** |
| 14 | **Return** clusters |
| 15 | **End** |
| 16 | **// Mapper phase** |
| 17 | **Begin** |
| 18 | **For** each data source |
| 19 | Provide the data source as training information for each mapper |
| 20 | Call the IMR-FLA |
| 21 | Find the clustered output of each mapper |
| 22 | **End for** |
| 23 | **End** |
| 24 | Generate the representative data |
| 25 | **// Reducer phase** |
| 26 | **Begin** |
| 27 | **For** each cluster centroid in the representative data |
| 28 | Provide the representative data as training information for each reducer |

*continued on following page*

**Algorithm 1. Continued**

| Sl. no | Algorithm of the proposed IMR-FLA |
|---|---|
| 29 | Call the IMR-FLA algorithm |
| 30 | Find the clustered output of the each reducer |
| 31 | **End for** |
| 32 | **End** |
| 33 | **Return** the data clusters |
| 34 | **End** |

## 4.1 Experimental Setup

The experimentation is done in the MATLAB tool, and the system contains the configurations of Windows 10 OS, Intel I3 processor, and 4 GB RAM.

### *4.1.1 Database Description*

The experimentation of the proposed IMR-FLA algorithm for the parallel data clustering utilized six standard databases from the UCI machine repository. The database used are Pima Indian diabetes dataset (Pima, https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes), Heart disease dataset (Heart, http://archive.ics.uci.edu/ml/datasets/heart+Disease), Hepatitis dataset (Hepatitis, https://archive.ics.uci.edu/ml/datasets/hepatitis), localization dataset (Localization, https://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity),breast cancer dataset (Breast, http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29), and Skin segmentation dataset (Skin, https://archive.ics.uci.edu/ml/datasets/skin+segmentation). The description of these databases are briefed as follows:

- **Pima Indian diabetes dataset:** The Pima Indian diabetes database contains 768 instances and eight attributes, and besides the data attributes, it also contains the missing values. The database is collected from several young women patients of age 21 and has various test results.
- **Heart disease dataset:** The heart disease dataset is obtained from the UCI machine repository, which contains information about heart patients. The database contains a collection of 303 instances categorized under 75 attributes. Heart disease in the individual patient is categorized through the integer value of 0 to 4.
- **Hepatitis dataset:** The hepatitis dataset has data about the presence of hepatitis disease in the patient. The presence of the disease is collected from various data samples through different tests. The test results of the patients are grouped in the hepatitis dataset. The hepatitis dataset has 155 instances and 19 attributes.
- **Localization dataset:** The localization dataset is collected to recognize the person's actions, and it is done with the use of several body sensor tags worn by samples in their leg, hand, wrist, etc. The localization dataset has a collection of 164860 instances and eight number attributes. Besides, the dataset does not have any missing attributes.
- **Breast cancer dataset:** The breast cancer dataset is used for diagnostic purposes, and the information about the diagnosis is obtained through the feature selection from the image. The database is multivariate and does not have many missing values. The information present in the database is real and, thus, has no integer values. The dataset has 569 instances and 32 attributes in total.

- **Skin dataset:** The skin segmentation dataset present in the UCI machine repository is one of the large databases, and hence, more suitable for the parallel clustering process. The skin segmentation dataset has 245057 instances under four attributes. The instances are collected from persons of different age groups and skin color.

### 4.1.2 Evaluation Metrics

The evaluation of the proposed scheme is done with the metrics, such as clustering accuracy and the Jaccard coefficient, and these metrics evaluate the efficiency of the data clusters obtained. The expression for the evaluation metrics is explained below:

- **Clustering accuracy:** The clustering accuracy defines the closeness of the clustered data obtained from the clustering algorithm and the available ground information. The expression for the clustering accuracy is expressed as follows:

$$Clustering\ \ accuracy = \frac{\left( \sum_{v=1}^{K} \max_{k-\{1,2,\ldots,K\}} \left\{ 2 \frac{\left| T_v \cap O_{rk} \right|}{\left| T_v + \left| O_{rk} \right| \right\|} \right\} \right)}{K} \tag{19}$$

where, the term $T_v$ indicates the ground value and the term $O_{rk}$ indicate the output of the $r^{th}$ reducer.

- **Jaccard coefficient:** The Jaccard coefficient finds the similarity and diversity between the data samples and identifies whether the data sample belongs to the cluster group. The expression for the Jaccard coefficient is expressed as:

$$Jaccard\ \ coefficient = (AA) / (AA + AE + EA) \tag{20}$$

where, $AA$ refers to the data points of the same cluster and the same group, the term $AE$ refers to the data point of the same cluster and a different group, and the term $EA$ refers to the data point of the different cluster and the same group.

### 4.1.3 Comparative Models

The comparison of the proposed IMR-FLA algorithm is made with the existing works, such as Fast Fuzzy C-Means algorithm (FFCM) (Cai, *et al.,* 2007), Fuzzy C-means algorithm (FCM) (Cai, *et al.,* 2007), k-means algorithm (Tang, *et al.,* 2017), k-Medoids algorithm (Park & Jun, 2009), and Lion Optimization Algorithm (LOA) (Yazdani & Jolai, 2016). The description of these existing works is given as follows:

- **Fast fuzzy C-means algorithm (FFCM):** The FFCM algorithm incorporates the generalized approach with the standard FCM algorithm for the clustering approach. Here, the parallelization of the FFCM is achieved using the FFCM in the MapReduce.
- **Fuzzy C-means algorithm (FCM):** The FCM algorithm is one of the traditional clustering algorithms, which uses the fuzzy approach along with the c-means algorithm for the clustering. The FCM algorithm has overhead issues when used for the clustering of the big data.
- **K-Means algorithm:** The k-means algorithm is one of the supervised approaches for clustering the database. The k-means algorithm implemented in the MapReduce faces the complexity issues since it allows the regression calculation of the distance parameter.

- **k-Medoids algorithm:** The k-Medoids algorithm is the improvement over the k-means algorithm. The k-Medoids algorithm performs the clustering by finding the medoid value, and medoid depends on the distance matrix.
- **Lion optimization algorithm (LOA):** LOA algorithm is nature inspired optimization algorithm for the optimization. The algorithm selects the suitable centroid through the optimization process.

## 4.2 Comparative Analysis

The comparative analysis of the proposed work is done by varying the number of clusters (K), and the analysis is done for the various databases. The performance of each comparative model is measured against the performance of the proposed IMR-FLA algorithm.

### *4.2.1 Comparative Analysis Using the Pima Indian Diabetes Database*

Figure 4 presents the comparative analysis of each model against the proposed IMR-FLA algorithm for the Pima Indian diabetes database. Figure 4.a. shows the performance of the models based on the clustering accuracy for varying numbers of clusters (K). For K value of 2, the existing FFCM, FCM, k-means, k-Medoids, and the LOA achieved the clustering accuracy value of 0.9609, 0.9609375, 0.95442, 0.8567, and 0.93619, respectively. Besides, the proposed IMR-FLA algorithm has a better clustering accuracy value of 0.96744 for the value of K=2 in the Pima Indian diabetes database. Figure 4.b presents the analysis of the comparative models based on the Jaccard coefficient for the Pima Indian diabetes database. The existing FFCM, FCM, k-means, k-Medoids, and the LOA models achieved the Jaccard coefficient value of 0.867586, 0.867586, 0.588464, 0.915453, and 0.540403, respectively, while the proposed IMR-FLA algorithm has a better Jaccard coefficient than the existing algorithms with the value of 0.93571 for $K=2$. From the above analysis, it is clear that the proposed IMR-FLA outperformed other existing techniques in terms of clustering accuracy and Jaccard coefficient.

### *4.2.2 Comparative Analysis Using the Heart Disease Database*

Figure 5 presents the comparative analysis of each comparative model against the proposed IMR-FLA algorithm using the Heart disease database. Figure 5.a shows the performance of the models based on the clustering accuracy for the varying K values. For the K value of 2, the existing FFCM, FCM,

**Figure 4a. Comparative analysis using the Pima Indian diabetes database based on clustering accuracy**

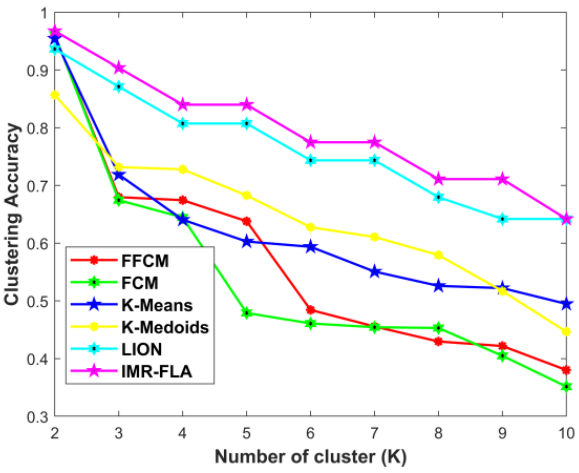
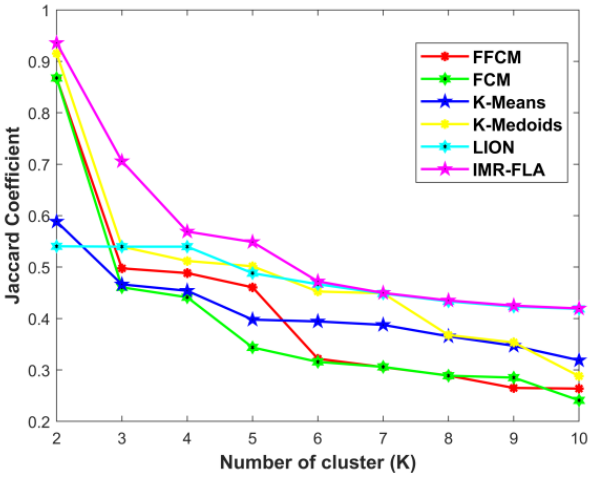


**Figure 4b. Jaccard coefficient**




k-means, k-Medoids, and the LOA achieved the clustering accuracy value of 0.90429, 0.752475, 0.924092, 0.854785, and 0.924092, respectively. Besides, the proposed IMR-FLA algorithm has a better clustering accuracy value of 0.947195 for K=2 in the heart disease database. Figure 5.b presents the analysis of the comparative models based on the Jaccard coefficient for the Heart disease database. The existing FFCM, FCM, k-means, k-Medoids, and the LOA models achieved the Jaccard coefficient value of 0.640562, 0.606809, 0.367234, 0.606809, and 0.333872, respectively, while the proposed IMR-FLA algorithm has a better Jaccard coefficient than the existing algorithms with the value of 0.657238 for $K=2$. From figure 5 shown below, for the small clustering size, the enhanced performance is obtained for both the metrics clustering accuracy and Jaccard coefficient.

**Figure 5a. Comparative analysis using the heart disease database based on clustering accuracy**

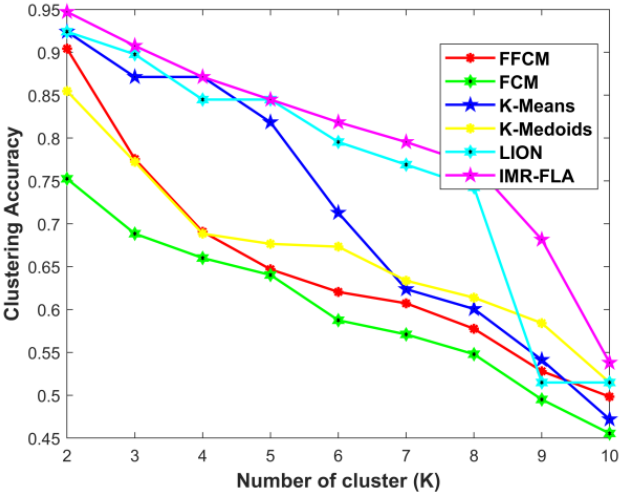
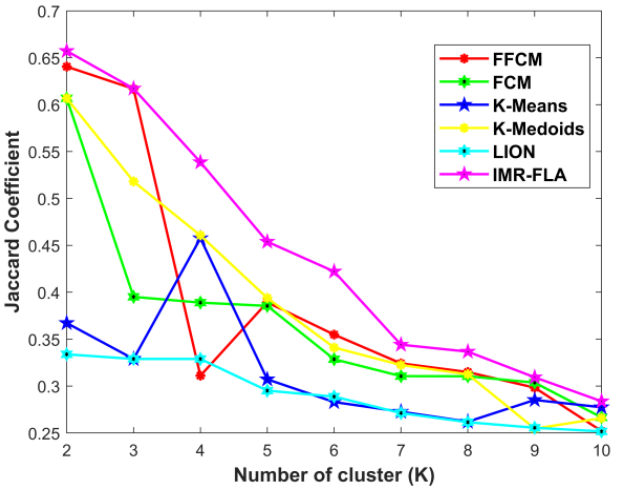


**Figure 5b. Jaccard coefficient**




### *4.2.3 Comparative Analysis Using the Hepatitis Database*

Figure 6 presents the comparative analysis of the comparative model against the proposed IMR-FLA algorithm using the Hepatitis database. Figure 6.a shows the performance of the models based on the clustering accuracy for different K values. For K=2, the existing FFCM, FCM, k-means, k-Medoids, and the LOA had achieved the clustering accuracy value of 0.909677, 0.78709, 0.929032, 0.935484, and 0.967742, respectively. However, the proposed IMR-FLA algorithm has a better clustering accuracy value of 0.967742 for the value of K=2 in the hepatitis database. Figure 6.b presents the analysis of the comparative models based on the Jaccard coefficient for the Hepatitis database. The existing FFCM, FCM, k-means, k-Medoids, and the LOA models achieved the Jaccard coefficient value of 0.509845, 0.67089, 0.69196, 0.731472, and 0.509845, respectively, while the proposed

**Figure 6a. Comparative analysis using the heart disease database based on clustering accuracy**

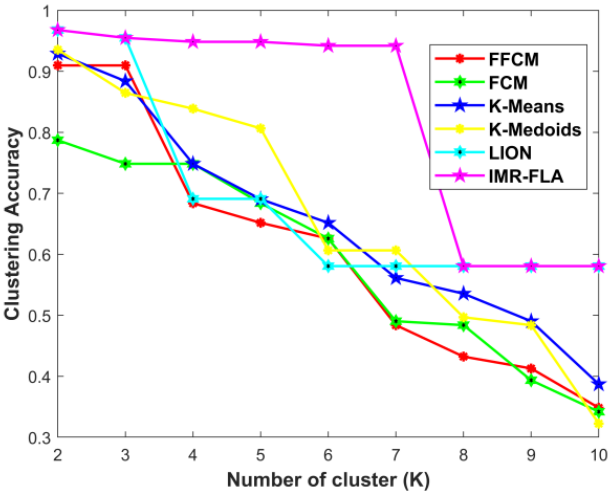
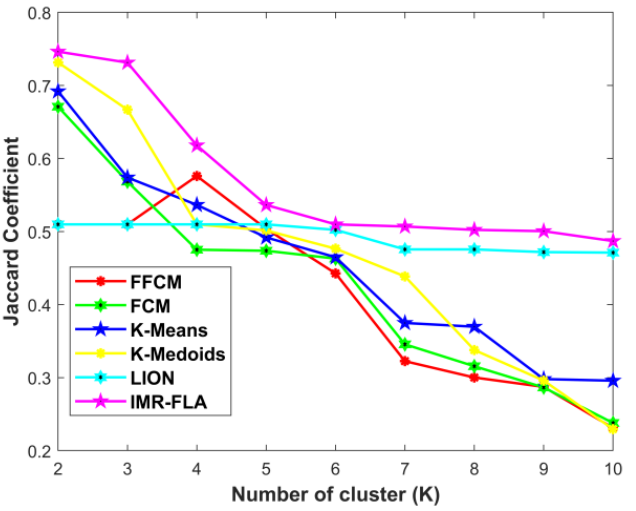


**Figure 6b. Jaccard coefficient**




IMR-FLA algorithm has a better Jaccard coefficient than the existing algorithms with the value of 0.746267 for $K$ =2. Hence the proposed methodology has improved performance compared to the existing techniques.

### *4.2.4 Comparative Analysis Using the Localization Database*

Figure 7 presents the comparative analysis of each model against the proposed IMR-FLA algorithm for the localization database. Figure 7.a shows the performance of the models based on the clustering accuracy for varying K values. For K=2, the existing FFCM, FCM, k-means, k-Medoids, and the

**Figure 7a. Comparative analysis using the Localization database based on clustering accuracy**

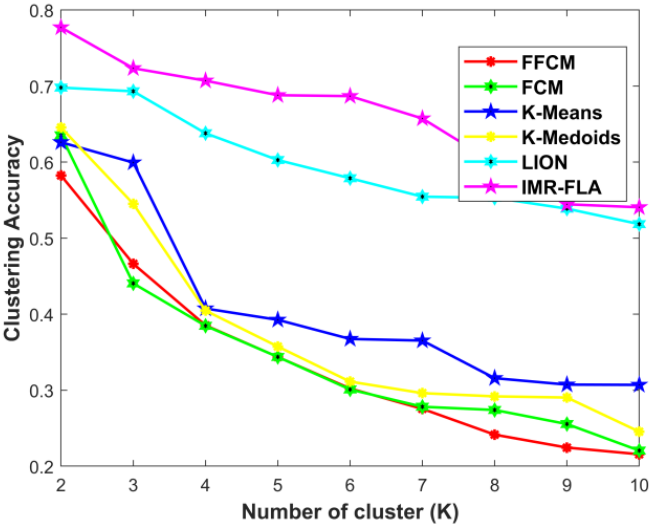
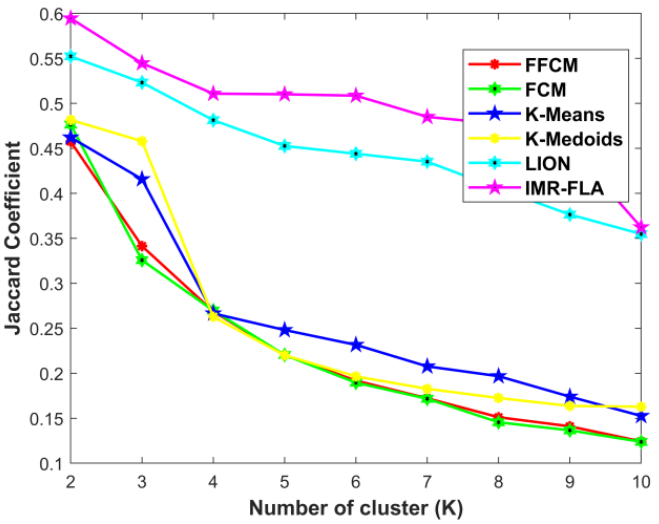


**Figure 7b. Jaccard coefficient**




LOA achieved the clustering accuracy value of 0.582474, 0.635233, 0.626137, 0.645543, and 0.697999, respectively, while the proposed IMR-FLA algorithm has a better clustering accuracy value of 0.777138, in the localization database. Figure 7.b presents the analysis of the comparative models based on the Jaccard coefficient for the localization database. The existing FFCM, FCM, k-means, k-Medoids, and the LOA models achieved the Jaccard coefficient value of 0.457502, 0.476416, 0.462176, 0.481643, and 0.552325, respectively, whereas the proposed IMR-FLA algorithm has a better Jaccard coefficient than the existing algorithms with the value of 0.59449 for $K$ =2. Thus, the proposed methodology outperformed other existing techniques in terms of performance metrics.

### *4.2.5 Comparative Analysis Using the Breast Cancer Database*

Figure 8 presents the comparative analysis of each model against the proposed IMR-FLA algorithm using the Breast cancer database. Figure 8.a shows the performance of the models based on the clustering accuracy for the varying number of clusters (K). For the K value of 2, the existing FFCM, FCM, k-means, k-Medoids, and the LOA achieved the clustering accuracy value of 0.764354, 0.862042, 0.902312, 0.902312, and 0.76435, respectively. Besides, the proposed IMR-FLA algorithm has a better clustering accuracy value of 0.902312 for the same value of K in the breast cancer database. Figure 8.b presents the analysis of the comparative models based on the Jaccard coefficient for the Breast cancer database. The existing FFCM, FCM, k-means, k-Medoids, and the LOA models had achieved the Jaccard coefficient value of 0.77657, 0.930369, 0.639735, 0.77657, and 0.639735, respectively, while the proposed IMR-FLA algorithm has a better Jaccard coefficient than the existing algorithms with the value of 0.941852 for $K$ =2. Thus, in terms of the performance metrics, the proposed methodology outperformed other existing techniques.

### *4.2.6 Comparative Analysis Using the Skin Segmentation Database*

Figure 9 presents the comparative analysis of each existing model against the proposed IMR-FLA algorithm for the skin segmentation database. Figure 9a shows the performance of the models based on the clustering accuracy for different cluster sizes. For K=2, the existing FFCM, FCM, k-means, k-Medoids, and the LOA achieved the clustering accuracy value of 0.919032, 0.736821, 0.714991, 0.892525, and 0.69982, respectively. However, the proposed IMR-FLA algorithm has a better

**Figure 8a. Comparative analysis using the Breast cancer database based on clustering**

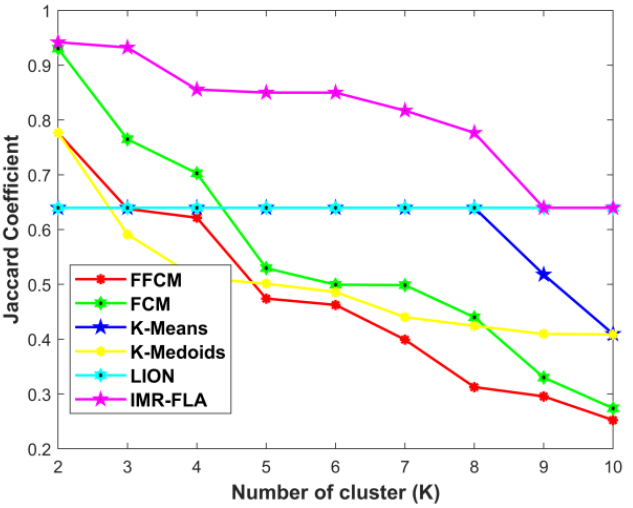
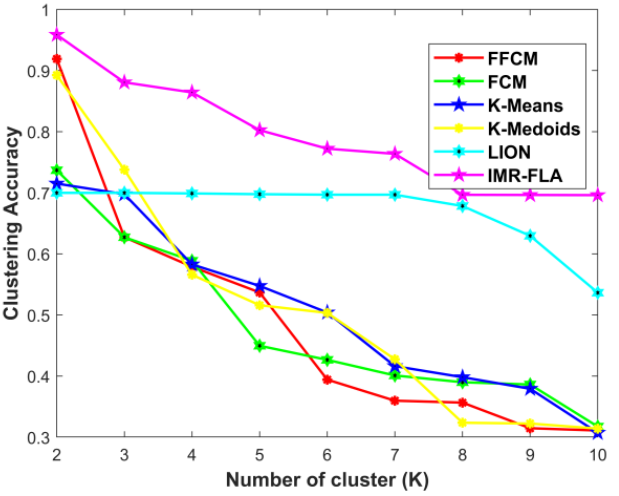



**Figure 8b. Accuracy Jaccard coefficient**




clustering accuracy value of 0.958502 for the value of K=2 in the skin segmentation database. Figure 9b presents the analysis of the comparative models based on the Jaccard coefficient for the skin segmentation database. The existing FFCM, FCM, k-means, k-Medoids, and the LOA models achieved the Jaccard coefficient value of 0.772705, 0.443872, 0.553779, 0.777623, and 0.548769, respectively, while the proposed IMR-FLA algorithm has better Jaccard coefficient than the existing algorithms with the value of 0.868072 for $K$ =2.

## 4.3 Discussion

Table 2 presents the comparative discussion of the proposed IMR-FLA model based on the clustering accuracy. The comparative discussion shows that the proposed IMR-FLA model has improved

**Figure 9a. Comparative analysis using the Skin database based on clustering accuracy**

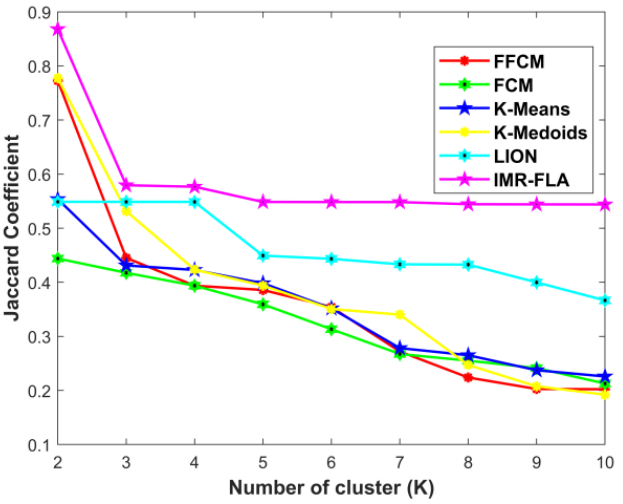



**Figure 9b. Jaccard coefficient**

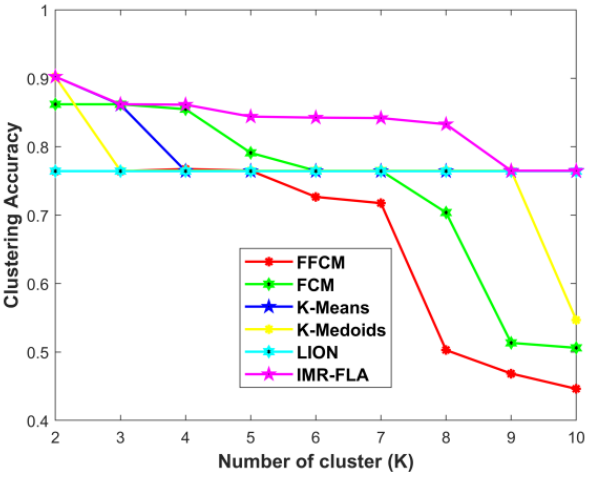



**Table 2. Comparative discussion based on clustering accuracy**

| Database | Analysis based on clustering accuracy | | | | | |
|---|---|---|---|---|---|---|
| | FFCM | FCM | k-means | k-Medoids | LOA | IMR-FLA |
| Pima Indian diabetes | 0.96098 | 0.9609 | 0.9544 | 0.8567 | 0.9361 | 0.9674 |
| Heart disease | 0.904 | 0.7524 | 0.9240 | 0.8547 | 0.9240 | 0.9471 |
| Hepatitis | 0.90967 | 0.7870 | 0.9290 | 0.9354 | 0.9677 | 0.9677 |
| Localization | 0.5824 | 0.6352 | 0.6261 | 0.6455 | 0.6979 | 0.7771 |
| Breast cancer | 0.7643 | 0.8620 | 0.9023 | 0.9023 | 0.7643 | 0.9023 |
| Skin segmentation | 0.9190 | 0.7368 | 0.7149 | 0.8925 | 0.6998 | 0.9585 |

performance over the other comparative techniques while simulating under the different databases. For the Pima Indian diabetes database, the proposed IMR-FLA model has achieved improved clustering accuracy of 0.9674. In contrast, the existing FFCM, FCM, k-means, k-Medoids, and the LOA algorithm achieved lower clustering accuracy values of 0.96098, 0.9609, 0.9544, 0.8567, and 0.9361, respectively. Similarly, for heart disease, hepatitis, localization, breast cancer, and the skin segmentation database, the proposed IMR-FLA algorithm has outclassed other classifier's performance with the clustering accuracy value of 0.9471, 0.9677, 0.777, 0.9023, and 0.9585, respectively.

Table 3 presents the comparative discussion of each model for varying databases based on the Jaccard coefficient. The Jaccard coefficient values of each model state the similarity between the data points and the cluster group. Hence, the model, which attained the higher value of the Jaccard coefficients, can be declared the better model. The discussion shows that the proposed model has achieved better performance than the existing works with the higher Jaccard coefficient while simulating in different databases. The proposed IMR-FLA algorithm has the overall improved Jaccard coefficient value of 0.9357, 0.6572, 0.7462, 0.5944, 0.9418, and 0.8680 for the Pima Indian diabetes, Heart disease, Hepatitis, localization, Breast cancer database, and the skin segmentation database, respectively.

In the K-means clustering algorithm, when the number of central points increases, the redundant distance calculations also increase; hence, the computational complexity increases (Tang et al, 2017). Compared to FCM, the FFCM clustering algorithm is faster due to the reduced number of iterations. However, it requires more spatial information; hence the memory requirement is high (Cai et al., 2007). The k-medoids method is more efficient compared to the k-means clustering algorithm. Still, the initial medoids selection has to be done very carefully; else it degrades the system's performance (Park and Jun, 2009). Besides, the Lion optimization algorithm has a better global convergence rate and optimal solution selection by avoiding local minima. However, the proposed method outperformed the state of art techniques regarding Jaccard coefficient and accuracy for all six different datasets. The IMP-FLA is the improved FLA for the parallel data clustering in the MapReduce framework for efficiency enhancement. It classifies the data based on the similarities between them. Moreover, the computational complexity is also reduced with the IMP-FLA algorithm. Hence, the overall performance of the system is improved in terms of the Jaccard coefficient and accuracy.

## 5. CONCLUSION

This work presents a parallel clustering algorithm suitable for the data mining schemes. The proposed IMR-FLA algorithm improves the existing FLA algorithm, and the distance between the data objects is calculated based on the Bhattacharya distance measure. The proposed scheme is implemented in the MapReduce framework to achieve parallel clustering. The experimentation

**Table 3. Comparative discussion based on Jaccard coefficient**

| Database | Analysis based on Jaccard coefficient | | | | | |
|---|---|---|---|---|---|---|
| | FFCM | FCM | k-means | k-Medoids | LOA | IMR-FLA |
| Pima Indian diabetes | 0.86758 | 0.8675 | 0.5884 | 0.9154 | 0.5404 | 0.9357 |
| Heart disease | 0.6405 | 0.6068 | 0.3672 | 0.6068 | 0.3338 | 0.6572 |
| Hepatitis | 0.5098 | 0.6708 | 0.6919 | 0.7314 | 0.5098 | 0.7462 |
| Localization | 0.4575 | 0.47641 | 0.4621 | 0.4816 | 0.5523 | 0.5944 |
| Breast cancer | 0.776 | 0.9303 | 0.6397 | 0.7765 | 0.6397 | 0.9418 |
| Skin segmentation | 0.7727 | 0.4438 | 0.5537 | 0.7776 | 0.5487 | 0.8680 |

of the proposed work is implemented in the MATLAB tool and various standard databases, such as Pima Indian diabetes dataset, Heart disease dataset, Hepatitis dataset, localization dataset, breast cancer dataset, and Skin segmentation dataset. The simulation of the proposed IMR-FLA is done by varying the number of clusters. The proposed IMR-FLA algorithm has the overall improved Jaccard coefficient value of 0.9357, 0.6572, 0.7462, 0.5944, 0.9418, and 0.8680 for the Pima Indian diabetes, Heart disease, Hepatitis, localization, Breast cancer database, and the skin segmentation database, respectively. Similarly, for the Pima Indian diabetes, heart disease, hepatitis, localization, breast cancer, and the skin segmentation database, the proposed IMR-FLA algorithm has outclassed other classifier's performance with the clustering accuracy value of 0.9674, 0.9471, 0.9677, 0.777, 0.9023, and 0.9585, respectively.

## REFERENCES

Amintoosi, H., Torshiz, M. N., Forghani, Y., & Alinejad, S. (2020). An efficient storage-optimizing tick data clustering model. *Turkish Journal of Electrical Engineering and Computer Sciences*, *28*.

Breast Cancer Wisconsin (Diagnostic) Dataset. (n.d.). http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29

Bui, Jambulingam, Amin, & Hung. (2021). Impact of COVID-19 pandemic on franchise performance from franchisee perspectives: the role of entrepreneurial orientation, market orientation and franchisor support. *Journal of Sustainable Finance & Investment*, 1-19.

Cai, W., Chen, S., & Zhang, D. (2007). Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. *Pattern Recognition*, *40*(3), 825–838.

Chander, S., Vijaya, P., & Dhyani, P. (2016). Fractional Lion Algorithm-An Optimization Algorithm for Data Clustering. *Journal of Computational Science*, *12*(7), 323–340.

Divya, K., & Raju, V. P. (2020). K-means++ Clustering Using MapReduce Framework for Large Datasets. *International Journal of Computer Science and Mobile Computing*, *9*(10).

Dubes, R., & Jain, A. K. (1980). Clustering methodologies in exploratory data analysis. *Advances in Computers*, *19*, 113–228.

Esposito, C., Ficco, M., & Gupta, B. B. (2021). Blockchain-based authentication and authorization for smart city applications. *Information Processing & Management*, *58*(2), 2021.

Ghany, K.K.A., Aziz, A.M.A., Soliman, T.H.A., & Sewisy, A.A.E-M. (2020). A hybrid modified step Whale Optimization Algorithm with Tabu Search for data clustering. *Journal of King Saud University - Computer and Information Sciences*.

Gowanlock, M., Blair, D. M., & Pankratius, V. (2017). Optimizing Parallel Clustering Throughput in Shared Memory. *IEEE Transactions on Parallel and Distributed Systems*, *28*(9), 595–2607.

Gupta, & Quamara. (2018). An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols. *Concurrency and Computation*.

Heart Disease Dataset. (n.d.). http://archive.ics.uci.edu/ml/datasets/heart+Disease

Hepatitis Dataset. (n.d.). https://archive.ics.uci.edu/ml/datasets/hepatitis

Hung & Chang. (2019). Preliminary Investigation of the Current Situation and Influencing Factors of International Students in Taiwan under the Background of New Southbound Policy. *Taiwan Educational Review, 8*(2).

Hung, N. T. (2020). A Model of International Students' Choice: A Mixed-Methods Study. *International Virtual Conference on Public Administration, Social Science & Humanities (ICPASH-2020)*.

Ianni, M., Masciari, E., Mazzeo, G. M., Mezzanzanica, M., & Zaniolo, C. (2020). Fast and effective Big Data exploration by clustering. *Future Generation Computer Systems*, 102.

Kaur, A., & Kumar, Y. (2021). *A new metaheuristic algorithm based on water wave optimization for data clustering*. Evolutinary Intelligence.

Localization Data for Person Activity Dataset. (n.d.). https://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity

Lydia, E.L., Moses, G.J., Varadarajan, V.k., Nonyelu, F., Maseleno, A., Perumal, E., & Shankar, K. (2020). Clustering And Indexing Of Multiple Documents Using Feature Extraction Through Apache Hadoop On Big Data. *Malaysian Journal of Computer Science, Big Data, and Cloud Computing Challenges, 1*.

Mishra, A., Gupta, N., & Gupta, B. B. (2021). Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller. *Telecommunication Systems*.

Nielsen, F., & Boltz, S. (2011). The Burbea-Rao and Bhattacharyya Centroids. *IEEE Transactions on Information Theory*, *57*(8), 5455–5466.

Park, H. S., & Jun, C. H. (2009). A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications*, *36*(2), 3336–3341.

Pima Indians Diabetes Dataset. (n.d.). https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes

Rahnema, N., & Gharehchopogh, F. (2020). An improved artificial bee colony algorithm based on whale optimization algorithm for data clustering. *Multimedia Tools and Applications*, 79.

Sharma, K. K., & Seal, A. (2020). Clustering analysis using an adaptive fused distance. *Engineering Applications of Artificial Intelligence*, 96.

Skin Segmentation Dataset. (n.d.). https://archive.ics.uci.edu/ml/datasets/skin+segmentation

Sreedhar, C., Kasiviswanath, N., & Reddy, P. C. (2017). Clustering large datasets using K-means modified inter and intra clustering (KM-I2C) in Hadoop. *Journal of Big Data*, *4*(1), 27.

Tang, Z., Liu, K., Xiao, J., Yang, L., & Xiao, Z. (2017). A parallel kmeans clustering algorithm based on redundance elimination and extreme points optimization employing MapReduce. *Concurrency and Computation*.

Tripathi, A. K., Sharma, K., Bala, M., Kumar, A., Menon, V. G., & Bashir, A. K. (2020). A Parallel Military Dog based Algorithm for Clustering Big data in Cognitive Industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, *17*(3).

Wang, Y., Wang, D., Zhang, X., Pang, W., Miao, C., Tan, A., & Zhou, Y. (2020). McDPC: Multi-center density peak clustering. *Neural Computing & Applications*, 1–19.

Xua, Q., Zhanga, Q., Liub, J., & Luo, B. (2020). Efficient synthetical clustering validity indexes for hierarchical clustering. *Expert Systems with Applications*, 151.

Yazdani, M., & Jolai, F. (2016). Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. *Journal of Computational Design and Engineering*, *3*(1), 24-36.

Yu, C., Li, J., Li, X., Ren, X., & Gupta, B. B. (2018). Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram. *Multimedia Tools and Applications*, *77*, 4585–4608.

Yu, J., Li, H., & Liu, D. (2020). Modified Immune Evolutionary Algorithm for Medical Data Clustering and Feature Extraction under Cloud Computing Environment. *Journal of Healthcare Engineering*.

Zhou, K., & Yang, S. (2020). Effect of cluster size distribution on clustering: A comparative study of k-means and fuzzy c-means clustering. *Pattern Analysis & Applications*, *23*, 455–466.

*Satish Chander working as Assistant Professor in Computer Science and Engineering Department, Birla Institute of Technology, Mesra, Ranchi, India. He has more than 20 years of teaching and research experience. His research interests include Data Mining, Machine Learning and Big Data Analytics.*

*P. Vijaya is working as Assistant Professor in the Department of Computer Science and Mathematics at Modern College of Business and Sciences, Oman. She has more than 20 years of teaching and research experience. Her research interests include Data Mining, Machine Learning, and Big Data Analytics. She is a life member of Indian Society for Technical Education (ISTE), Computer Society of India (CSI).*

*Praveen Dhyani is presently Honorary Professor of Computer Science at Banasthali Vidyapith (a deemed to be university), India. Earlier, Prof. Dhyani was Director of RAK (UAE) and Muscat (Oman) Centres of Birla Institute of Technology, MESRA. He held key academic and administrative positions in BITS Pilani and BIT, MESRA. He was instrumental in commencing, establishing and heading national and international centers of BIT, MESRA, at Jaipur, Bahrain, Muscat, and RAK (UAE). He has supervised doctoral research and has authored research papers, technical reports, and international conference proceedings in diverse fields of computer science. His R&D accomplishments include development and national and international exhibit of robot, development of electronics devices to aid foot drop patients, and development of voice operated wheelchair.*