Omkaresh Kulkarni*, Sudarson Jena and C. H. Sanjay

Fractional Fuzzy Clustering and Particle Whale Optimization-Based MapReduce Framework for Big Data Clustering

https://doi.org/10.1515/jisys-2018-0117
Received January 31, 2018.

Abstract: The recent advancements in information technology and the web tend to increase the volume of data used in day-to-day life. The result is a big data era, which has become a key issue in research due to the complexity in the analysis of big data. This paper presents a technique called FPWhale-MRF for big data clustering using the MapReduce framework (MRF), by proposing two clustering algorithms. In FPWhale-MRF, the mapper function estimates the cluster centroids using the Fractional Tangential-Spherical Kernel clustering algorithm, which is developed by integrating the fractional theory into a Tangential-Spherical Kernel clustering approach. The reducer combines the mapper outputs to find the optimal centroids using the proposed Particle-Whale (P-Whale) algorithm, for the clustering. The P-Whale algorithm is proposed by combining Whale Optimization Algorithm with Particle Swarm Optimization, for effective clustering such that its performance is improved. Two datasets, namely localization and skin segmentation datasets, are used for the experimentation and the performance is evaluated regarding two performance evaluation metrics: clustering accuracy and DB-index. The maximum accuracy attained by the proposed FPWhale-MRF technique is 87.91% and 90% for the localization and skin segmentation datasets, respectively, thus proving its effectiveness in big data clustering.

Keywords: Big data clustering, fractional theory, TSK clustering, MRF, PSO, WOA.

1 Introduction

One of the popular and significant topics in computing science research is big data [16]. Basically, datasets are composed of several hundred items, and advancements in technologies make it possible to store and process these data sets that are larger in volume. Such kind of data is known as big data, which is a collection of large-sized and complex data sets. Big data includes three characteristics formed by three V's (volume, variety, and velocity). Volume refers to the huge size of the data set, variety infers various types of data, and velocity is new data that accumulates constantly. When one of these three characteristics of data exceeds the capacity of the system in storing, analyzing, and processing, the data becomes big. In recent times, big data is popular with the inclusion of two additional V's such that it can be characterized by five V's, as follows: volume, velocity, variety, veracity, and value. Big data not only concerns with large-sized data, but it is also a new concept that offers a choice to discover new insights into already existing data. Big data is applicable in various fields, such as business, telecommunication, technology, health care, medicine, bioinformatics, e-commerce, science, finance, information search, etc. [14]. Social networking sites like Facebook, Twitter, Instagram, and so on, have billions of users that create several gigabytes of contents in a minute, and retail shops require continuous data collection of their customers [18]. Hence, such "big data" can create "big challenges" [8].

***Corresponding author: Omkaresh Kulkarni,** Research Scholar, Gandhi Institute of Technology and Management, GITAM University, Rudraram Mandal, Sangareddy district, Patancheru, Hyderabad, Telangana 502329, India, e-mail: kulkarniomkaresh@gmail.com
Sudarson Jena: Associate Professor, Department of Computer Science Engineering and Application, Sambalpur University Institute of Information Technology, Sambalpur, Orissa, India
C. H. Sanjay: Distinguished Professor and Dean, GITAM University, Hyderabad, India

Besides dealing with the challenges of data collection, the major issue has been focused on processing these enormous volumes of data. Processing big data using traditional data processing techniques is usually difficult. It is often difficult to use analytics and traditional inference approaches using individual processors due to the dimensionality and massive size of the data [20]. It is essential to use optimum mechanisms for knowledge discovery to handle such data. One of the common knowledge discovery tools used for this purpose is the data mining approach. Clustering is a data mining approach where the data is split into groups such that the objects within each group contain more similarity than with the objects in different groups [14]. The major idea behind the clustering approach is to find the target cluster accurately for every case in the data set. Data clustering is an accepted method in different fields of computer science and other related areas. The clustering methods [4, 13] in big data can be categorized into two: single machine and multiple machine clustering techniques. Nowadays, clustering methods based on multiple machines gain more attention, as they are faster and adaptable to most of the challenges of big data [27]. Big data generates heterogeneous data that is difficult to exploit.

Clustering techniques can be a powerful solution, as they can overcome these challenges. Clustering, also known as unsupervised classification, is the process of classifying a set of data into clusters or groups of homogeneous data such that the elements in each cluster are similar. However, clustering methods also have their limitations in processing big data. One of the main challenges in big data is in offering a clustering technique that can generate a satisfactory quality of clustering within a reasonable time [17]. Hence, strategies, techniques, and architectural models that are presently in use are not suitable for handling big data. Accordingly, the MapReduce programming model [2, 11] has been introduced to deal with large data analytic applications, overcoming the challenges of query processing, analysis, and data modeling. The MapReduce model is generally implemented using Hadoop1, which is a parallel programming structure, to handle issues related to large-scale datasets. The MapReduce framework (MRF) comprises mapper and reducer functions, where the mapper performs filtering and sorting and the reducer does a summary operation to obtain the result. This framework with its simple Hadoop architecture can provide better performance for processing large-scale data if the configuration factors are adjusted accurately [19]. The parallelization process using MapReduce has become an attractive technique due to its programming model, which automatically processes the tasks in parallel, providing load balancing and fault tolerance.

1.1 Novelties of the Paper

In this paper, an MRF, FPWhale-MRF, is proposed for big data clustering by developing two hybrid clustering algorithms: Fractional Tangential-Spherical Kernel (FTSK) and Particle-Whale (P-Whale). Due to the complexity in the mapping of large datasets, a novel clustering algorithm, FTSK, is designed by including fractional theory in the Tangential-Spherical Kernel (TSK) clustering approach [12]. TSK clustering is a clustering approach designed by combining tangential and spherical kernels. This approach can improve the scalability by reducing the mapping problems. For further improvement in the performance of the TSK clustering approach, the fractional theory has been included, which finds the centroids such that the convergence speed is enhanced, thereby improving the performance. The fractional theory is a division of applied mathematics that solves fractional-order equations using Laplace transforms. FTSK is utilized in the mapper to find the cluster centroids for clustering, whereas the reducers use P-Whale that combines Particle Swarm Optimization (PSO) with Whale Optimization Algorithm (WOA) to provide optimal clustering based on the intermediate results obtained from the mappers. The function of a reducer is to combine the local centroids generated by the mappers to find the feasible cluster centroids. In P-Whale, the update process of PSO is modified using WOA, so that the clustering task is made effective and the P-Whale algorithm determines the optimal centroid. With the proposed P-Whale, the reducer clusters the data, where the number of clusters is user-defined. Thus, the proposed MRF with the newly designed clustering algorithms can perform big data clustering effectively.

The major contributions of the proposed FPWhale-MRF used for big data clustering are as follows:

- Introducing FTSK clustering by integrating fractional theory into the TSK clustering algorithm such that the mapper can locate the most appropriate cluster centroids for the clustering.

- Combining PSO with WOA to design a novel algorithm, P-Whale, and utilizing it in the reducer for the optimal clustering of big data, based on the given input, which are the intermediate results of the mappers.
- Designing the FPWhale-MRF clustering technique for big data clustering using FTSK clustering and the P-Whale algorithm in the mapper and reducer functions, respectively, for the effective clustering of data.

The organization of the paper is as follows: The literature survey presenting various techniques of big data clustering is given in Section 2. Section 3 explains the proposed FPWhale-MRF technique developed for big data clustering using the proposed FTSK and P-Whale algorithms adopted in MRF with a suitable block diagram. In Section 4, the results and the comparative analysis to evaluate the performance of the proposed technique are demonstrated. Finally, the paper is concluded in Section 5.

2 Literature Survey

Different techniques based on the MapReduce model used in the literature for big data clustering are explained in this section, stating their drawbacks along with the challenges discovered.

2.1 Clustering Techniques

Xia et al. [24] designed an MRF using the Parallel Three-Phase K-means (Par3PKM) algorithm depending on the Hadoop technology. The algorithm utilized a modified distance measure with K-means algorithm for the initialization. To improve the optimized K-means algorithm, the MRF was employed to cluster large-scale taxi trajectory data. The Par3PKM algorithm could perform clustering with better efficiency and higher scalability. However, it suffers from the drawback of missing kernel space for the distance measurement.

Traganitis et al. [20] had presented two variants, sketch and validate, of kernel-based K-means clustering for big data clustering. The former was based on batch processing, whereas the latter was the sequential one used to provide efficiency in computation. The algorithm can perform clustering of data efficiently; however, the failure to consider the defined MRF is the limitation.

Vadivel and Raghunath [23] designed an approach for hierarchical clustering to group big data based on the MRF. The clustering task utilized feature selection based on co-occurrence depending on the distributed architecture that shuffled the results obtained from the mappers according to the queues. Even though the method offers less computational time, it takes considerable time for the computation in the merging step for hierarchical clustering.

Fries et al. [7] had extended the state-of-the-art projected clustering algorithm P3C by investigating its solutions for large-scale data sets that were in high-dimensional spaces. The authors proved that the original model of the algorithm was not appropriate to handle large datasets. Hence, they designed a MapReduce-based implementation, known as the P3C+-MR algorithm, by providing the changes required in the basic clustering model. It has better scalability provided by the MRF, whereas the curse of dimensionality becomes a challenging process.

Akthar et al. [1] had modified the K-means clustering algorithm for big data clustering by taking the optimized centers according to the data dimensional density. The modification of the algorithm was based on the basic idea of selecting the optimal “K” data points that are in the highly dense areas as the initial centers, so that the data points outside the specified areas were eliminated from the computation of final clusters. Even though the algorithm provides better results, it has several limitations, as follows: (i) the results of the algorithm are compared only with a single algorithm, which is not adequate for effective performance analysis, and (ii) computation of distance measure takes considerable time. The clustering techniques have the following limitations: fixed number of clusters can make it difficult to predict the K-value; they do not work well in non-globular clusters; and different initial partitions can result in different final clusters.

2.2 Optimization Techniques

Hans et al. [9] developed a technique to implement clustering based on Genetic Algorithm (GA) in a parallel manner using Hadoop MapReduce by extending the coarse-grained parallel model of GAs. Thereby, the authors could perform a two-phase clustering process on the given data set based on the MapReduce architecture utilized. However, the technique requires further improvement in accuracy and speed. The optimization techniques have drawbacks, such as being time consuming and requiring more repetition.

2.3 Supervised Learning Techniques

With the utilization of a series of optimizations, Chen et al. [5] had developed the Parallel Semi-supervised Extreme Learning Machine (PASS-ELM) algorithm using the MapReduce model to enhance the performance of SS-ELM. The design of PASS-ELM was based on the Approximate Adjacent Similarity Matrix (AASM) algorithm, which utilized the Locality-Sensitive Hashing method to compute AASM, thereby reducing the complexity and the storage space required. However, the requirement for several optimizations to enhance the efficiency of the algorithm is a drawback.

Kamal et al. [10] presented a distributed clustering approach based on imbalance data reduction with the K-nearest neighbor classification method. The major contribution of the approach was to demonstrate real training data sets by reducing the number of instances such that the data classification could be performed quickly. The difficulties during data reduction were managed by an MRF, which was developed based on several clusters of automated contents with different algorithms. The approach can handle large datasets with better speedup, low reduction time, and less storage capacity. The supervised learning techniques have drawbacks, such as being inaccurate and unable to detect emerging and unknown anomalies.

2.4 Challenges

The methods discussed in the literature survey pose various challenges, which the proposed technique tries to overcome. Some of the challenges found in big data clustering are as follows:

– In Ref. [20], the clustering task is carried out using the kernel space, which is not applicable to the determination of clusters that are non-linear. As the accuracy of the algorithm in clustering depends on those clusters, it is a major issue in clustering big data.

  The proposed technique for big data clustering is applicable for the determination of all types of clusters, as it takes the advantages of fractional theory, which is used to model the non-linearity.
– One of the main challenges in big data is in data clustering, due to the large size and varieties of data to be considered. The common tools available for the processing of such data are not effective, even if various computer clusters are employed. Hence, it is necessary to find a better approach for clustering and handling large datasets [14].

  The proposed FTSK clustering approach handles the large datasets. FTSK is designed by including fractional theory in the TSK clustering approach. The TSK clustering approach can improve scalability by reducing the mapping problems.
– Another challenge in big data is its complexity, which increases with increasing amount of data. Usual techniques managing relational database tools cannot provide a satisfactory outcome meeting all the requirements [27].

  The proposed FPWhale-MRF technique offers satisfactory results, as it takes the advantages of both the FTSK and P-Whale algorithms.
– Increasing demands for data is another issue, which can be managed only by increasing the capacity and performance of the techniques utilized, within the resources provided.

  The proposed method manages the big data as it utilizes the MRF, which performs clustering by processing the partitioned data in parallel. Thus, it is possible to handle large-scale datasets using the MapReduce model, by sharing a task on several cluster nodes.

- MapReduce models can handle big data issues to a great extent. However, the techniques utilized in Refs. [7, 23, 24] are not effective without additional processes to enhance their performances in terms of computational time and accuracy.

  The proposed MRF offers good performance in terms of computational time and accuracy.

3 Proposed MRF Using Fractional P-Whale-Based Clustering for Big Data Clustering

This section presents the proposed technique of FPWhale-MRF, used for big data clustering based on two-hybrid clustering algorithms adopted in the MRF. Each mapper in the MRF utilizes a novel clustering algorithm, FTSK, which finds the cluster centroids using tangential and spherical kernels combined with fractional theory. The centroids obtained by the mappers are merged and fed as input to the reducers, where the proposed P-Whale is employed to determine the optimal clusters for data clustering. The P-Whale algorithm is developed by integrating WOA into PSO, and is used in clustering to estimate the centroids for the final cluster. The block diagram of the proposed technique of clustering big data is depicted in Figure 1.

3.1 Proposed Technique of Big Data Clustering Using FPWhale-MRF

In this subsection, the proposed FPWhale-MRF technique designed for big data clustering is described. MRF, which is a programming model, is composed of mappers and reducers. It performs clustering by processing the partitioned data in parallel. Thus, it is possible to handle large-scale datasets using the MapReduce model, by sharing a task on several cluster nodes. In FPWhale-MRF, the mappers, which receive the partitioned data

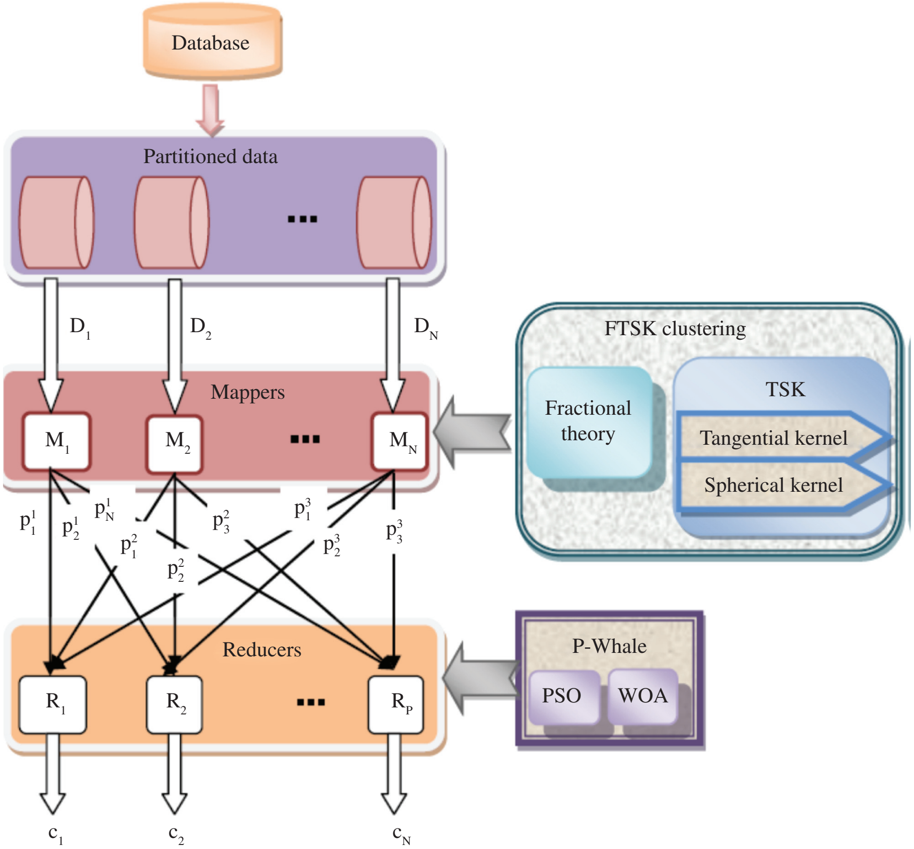

Figure 1: Block Diagram of the Proposed FPWhale-MRF.

as input, find the clustering centroids using the FTSK clustering algorithm. Meanwhile, the reducer functions use the P-Whale algorithm for the optimal selection of centroids for the clustering. Let B denote the database having m number of data with n attributes, as represented below.

$$B = \{b_{k,l}\};\ 0 \le k \le m;\ 0 \le l \le n. \tag{1}$$

The data $b_{k,l}$ in the database is split into a finite number, which is equal to the number of mappers in the MRF. The partitioned data is given by

$$b_{k,l} = \{D_q\};\ 1 \le q \le N, \tag{2}$$

where N is the total number of mappers. Let N number of mappers in the MRF be represented as

$$M = \{M_1,\ M_2,\ \ldots,\ M_q,\ \ldots,\ M_N\};\ 1 \le q \le N. \tag{3}$$

Hence, the input to the q^{th} mapper can be given as

$$D_q = \{d_{r,l}\};\ 1 \le r \le m_q;\ 1 \le l \le n, \tag{4}$$

where $d_{r,l}$ is the partitioned data given to the q^{th} mapper for processing and m_q is the number of data in the q^{th} mapper. Each mapper maps its input data based on the number of clusters defined by the user and generates the intermediate results. Thus, the outcome of each mapper that depends on the data pair and the attributes represents the cluster centroid. All N number of mappers generates output of size $m \times n$, represented as follows:

$$I = \{p_1 || p_2 \ldots || p_N\}, \tag{5}$$

where p indicates the mapper outcome, which represents the cluster centroid.

The generated output from the mapper is given as input to the set of reducers, represented as

$$R = \big\{R_1,\ R_2,\ \ldots,\ R_j,\ \ldots R_P\big\};\ 1 \le j \le P, \tag{6}$$

where P is the number of reducers. The reducers merge the resulting cluster centroids obtained from the mappers to produce the final clusters, as given below:

$$c_j = \left\{ \begin{matrix} c^j_{x,l}\,;\ 1 \le x \le C_j \\ 1 \le l \le n \\ 1 \le j \le P \end{matrix} \right\}, \tag{7}$$

where C_j is the size of the cluster assigned to the j^{th} reducer and $c^j_{x,l}$ represents the cluster in the j^{th} reducer.

3.2 Proposed FTSK-Based Mapper Function for Centroid Estimation

The mapper function designed using the proposed FTSK clustering algorithm is explained in this section. Due to the complexity in the mapping of large datasets, a novel clustering algorithm is designed by including fractional theory in the TSK clustering approach [12]. TSK clustering is a clustering approach designed by combining tangential and spherical kernels. This approach can improve the scalability by reducing the mapping problems. For further improvement in the performance of the TSK clustering approach, the fractional theory has been included. Hence, every mapper in the MRF finds the cluster centroids based on the proposed FTSK clustering algorithm, which will be discussed below.

Initially, the cluster centers are chosen in random from the data given as input. Let p_i be the selected cluster center, after which the partitioned data is applied to the cluster center depending on the similarity measured. In the distance matrix computed, the column denotes the data, while the row indicates the cluster. For the first column, the distance matrix is obtained by calculating the Euclidean distance between the first centroid and the data. Similarly, for the second column, Euclidean distance is measured between the second centroid and the data. Then, the tangential and spherical distances are computed at iteration zero between the cluster centroids and the data. The data is clustered by minimizing the distance measures based on the tangential and spherical kernels. The tangential kernel-based distance matrix computation is expressed as

$$K_{r,i}^{T} = \frac{E^{T}(d_r,\ p_i)}{\sum_{i=1}^{N^C} E^{T}(d_r,\ p_i)}, \tag{8}$$

where $E^{T}(d_r,\ p_i)$ represents the tangential Euclidean distance between the data d_r and the cluster centroid p_i, and N_C is the number of clusters. Meanwhile, the spherical kernel-based distance measure is computed as follows:

$$K_{r,i}^{S} = \frac{E^{S}(d_r,\ p_i)}{\sum_{i=1}^{N^C} E^{S}(d_r,\ p_i)}, \tag{9}$$

where $E^{S}(d_r, p_i)$ represents the spherical Euclidean distance between the data and the cluster centroid. The tangential and spherical Euclidean distances are measured as

$$E^{T}(d_r,\ p_i) = \tanh\left(\alpha||d_r^{T} - p_i|| + a\right), \tag{10}$$

$$E^{S}(d_r,\ p_i) = 1 - \frac{3}{2}\left(\frac{||d_r^{T} - p_i||}{\sigma}\right) + \frac{1}{2}\left(\frac{||d_r^{T} - p_i||}{\sigma}\right)^{2}, \tag{11}$$

where tanh represents the hyperbolic tangent, α represents the slope, $||d_r^{T} - p_i||$ is the distance measure, a represents the constant, and σ represents the standard deviation. The centroids for each cluster are measured using these kernel functions. The procedure of distance measurement and the computation of new cluster centroids continue until the convergence condition is reached. Hence, the cluster centroids are formulated based on both tangential and spherical kernels as

$$p_{i(t+1)}^{T} = \frac{1}{m_i}\sum_{\substack{r=1\\ r\in i}}^{m_i}\left(\frac{K_{r,i}^{T} \times d_r}{\sum_{i=1}^{N^C} K_{r,i}^{T}}\right), \tag{12}$$

$$p_{i(t+1)}^{S} = \frac{1}{m_i}\sum_{\substack{r=1\\ r\in i}}^{m_i}\left(\frac{K_{r,i}^{S} \times d_r}{\sum_{i=1}^{N^C} K_{r,i}^{S}}\right), \tag{13}$$

where m_i is the number of data in the i^{th} mapper. Combining both kernels, the TSK clustering approach increases the clustering accuracy by finding the cluster centroids as

$$p_{i(t+1)}^{F} = \frac{1}{2}\left(p_{i(t+1)}^{T} + p_{i(t+1)}^{S}\right), \tag{14}$$

where p_i^{T} is the cluster centroid obtained based on the tangential kernel and p_i^{S} is the cluster centroid based on the spherical kernel. In the proposed FTSK clustering algorithm, fractional calculus [3] is incorporated to find the centroids such that the convergence speed is enhanced, thereby improving the performance. Fractional calculus is a division of applied mathematics that solves fractional-order equations using Laplace transforms. Accordingly, Eq. (14) can be rearranged by subtracting the cluster centroid estimated at current iteration as

$$p_{i(t+1)}^{F} - p_{i(t)}^{F} = \frac{1}{2}\left(p_{i(t+1)}^{T} + p_{i(t+1)}^{S}\right) - p_{i(t)}^{F}, \tag{15}$$

where $p^F_{i(t)}$ is the centroid estimated at iteration t. Considering the left-hand side of the above equation as a derivative of order γ, Eq. (15) can be represented as follows:

$$D^{\gamma}\left[p^F_{i(t+1)}\right] = \frac{1}{2}\left(p^T_{i(t+1)} + p^S_{i(t+1)}\right) - p^F_{i(t)}, \tag{16}$$

where γ takes a value between 0 and 1 and hence considers the first four terms of the derivative. Thus, Eq. (15) becomes

$$\begin{aligned} p^F_{i(t+1)} = {} & \gamma p^F_{i(t+1)} + \frac{1}{2}\gamma p^F_{i(t-1)} + \frac{1}{6}\gamma(1-\gamma)p^F_{i(t-2)} \\ & + \frac{1}{24}\gamma(1-\gamma)(2-\gamma)p^F_{i(t-3)} + \frac{1}{2}\left(p^T_{i(t+1)} + p^S_{i(t+1)}\right) - p^F_{i(t)}. \end{aligned} \tag{17}$$

Rearranging the equation, the cluster centroid estimated by the proposed FTSK clustering algorithm is given by

$$\begin{aligned} p^F_{i(t+1)} = {} & p^F_{i(t+1)}[\gamma - 1] + \frac{1}{2}\gamma p^F_{i(t-1)} + \frac{1}{6}\gamma(1-\gamma)p^F_{i(t-2)} \\ & + \frac{1}{24}\gamma(1-\gamma)(2-\gamma)p^F_{i(t-3)} + \frac{1}{2}\left(p^T_{i(t+1)} + p^S_{i(t+1)}\right), \end{aligned} \tag{18}$$

where $p^F_{i(t-2)}$ and $p^F_{i(t-3)}$ are the cluster centroids measured at iterations $(t-2)$ and $(t-3)$, respectively. Hence, the mapper function calculates the cluster centroids based on Eq. (18) formulated using the proposed FTSK clustering algorithm so that the performance of clustering can be improved from that of TSK clustering [12].

3.3 Proposed P-Whale-Based Centroid Estimation for Big Data Clustering

This section illustrates the reducer function, which is executed using the proposed P-Whale algorithm. In P-Whale, the update process of PSO is modified using WOA, so that the clustering task is made effective. The function of a reducer is to combine the local centroids generated by the mappers to find the feasible cluster centroids. The reducer utilizes the intermediate results produced by the mappers, given as

$$I = \left\{I^{y,l}\right\};\ 1 \le y \le N^C \times N;\ 1 \le l \le n, \tag{19}$$

where N_C is the number of clusters and $I^{x,l}$ is the intermediate result generated by N mappers. Processing I obtained from the mapper, the reducer finds the cluster, as defined below:

$$R(I) = c_j, \tag{20}$$

where c_j is the number of clusters in the j^{th} reducer. Thus, the reducer generates the output given by the following function:

$$R_P = \left\{\begin{matrix} p_{i,l};\ 1 \le q \le N \\ \quad\ \ 1 \le i \le N^C_q \\ \quad\ \ 1 \le l \le n \end{matrix}\right\}, \tag{21}$$

where N^C_q is the number of clusters in the q^{th} mapper. This calculation is done using the proposed P-Whale algorithm that generates the optimal cluster centroids for the clustering based on the fitness defined using the DB-index. Thus, with the proposed P-Whale, the reducer clusters the data, where the number of clusters is user-defined. PSO [26] is a stochastic optimization method that mimics the social behavior of fish schooling, whereas WOA [15] is a nature-inspired algorithm developed based on the hunting behavior of humpback whales. Integrating WOA in the update process of PSO, the performance of the proposed algorithm in clustering can be improved. Adopting P-Whale, the reducer processes the data, which is represented in vector form, to find the clusters.

3.3.1 Solution Representation

The solution encoding presents the simplest view of representing the proposed P-Whale algorithm designed for finding the clusters in the proposed FPWhale-MRF. Here, the solution is the cluster centroid, which is initialized in random, depending on the intermediate data produced by the mappers. Thus, the solution is a vector, whose size is equivalent to the number of clusters and the data. Based on the fitness evaluated using the DB-index, the cluster centroids can be determined optimally using the P-Whale algorithm.

3.3.2 Fitness Evaluation

The fitness function, which decides the quality of the solution, is designed using a DB-index [6], similar to that utilized in Ref. [12]. The DB-index measures the similarity between the clusters and is suitable for clustering algorithms that depend on distance conditions. The fitness function is formulated based on the DB-index as

$$DB = \frac{1}{c_j}\sum_{e=1}^{c_j} F_e, \tag{22}$$

where F_e is a function that selects the maximum similarity value measured between the clusters, as defined by the following equation:

$$F_e = \max_{f \neq e} S_{e,f}, \tag{23}$$

where $S_{e,f}$ is the similarity measure that measures the similarity between the clusters based on the Euclidean distance measured between the clusters and is represented as

$$S_{e,f} = \frac{A_e + A_f}{E_{e,f}}, \tag{24}$$

where A_e and A_f are the measures of scattering in two clusters and $E_{e,f}$ is the Euclidean distance between the two cluster matrices. The lower the distance between the cluster centroid and the data points, the greater is the performance of clustering:

$$E_{e,f} = \left\| L_e - L_f \right\|, \tag{25}$$

where L_e and L_f denote the centroids of the two clusters. Based on the data given to the cluster, A_e is computed with respect to the Euclidean distance between the data and the cluster:

$$A_e = \frac{1}{m^e}\sum_{e=1}^{m^e} \left\| I^{y,l} - L_e \right\|, \tag{26}$$

where m^e is the number of data points associated with L_e. Even though the distance between the centroids and the data points needs to be minimum, the distance between two centroids has to be maximum for better clustering.

3.3.3 Proposed P-Whale Algorithm

The proposed P-Whale algorithm, designed for the optimal selection of cluster centroids to perform clustering, is summarized in this section. The P-Whale algorithm is designed by modifying the update process of PSO using WOA. In PSO [26], a number of particles interact to find the optimal solutions in the search space, where the particles learn based on the personal best and global best solutions. This way of learning may lead to premature convergence, which can be solved using WOA, which has better convergence behavior with local optima avoidance. The steps involved in the proposed P-Whale algorithm are described below.

I. Initialization
The foremost step is the random initialization of the swarm population with a number of solutions, represented as

$$H = \{H_1,\ H_2,\ \ldots,\ H_s,\ \ldots,\ H_z\};\ 1 \leq s \leq z, \tag{27}$$

where H_s represents the position of the s^{th} solution, such that the dimension of each solution is $1 \times J$ and z is the number of swarm particles.

II. Fitness Calculation
Once the population is initialized, the fitness of all the solutions is computed using the fitness function formulated in Section 3.3.2. The solution having the minimum fitness value is considered as the best solution. Thus, the algorithm selects the personal best and the global best solution, represented as G_{pb} and G_{gb}, respectively.

III. Whale-Based Update Process
The update process of PSO involves the velocity and position updates. The velocity assigned to the s^{th} particle can be updated based on the personal and the global best solutions together with the velocity computed at current iteration t, as given below:

$$v_s(t+1) = Wv_s(t) + k_1h_1\big(G_{pb} - H_s(t)\big) + k_2h_2\big(G_{gb} - H_s(t)\big), \tag{28}$$

where W is the inertia weight, k_1 and k_2 are the acceleration rates, h_1 and h_2 are two numbers chosen in random between 0 and 1, $v_s(t)$ is the velocity at iteration t, and $H_s(t)$ is the position of the s^{th} particle at the current iteration. Based on the velocity update, the position of the s^{th} particle can be updated as follows:

$$H_s(t+1) = H_s(t) + v_s(t+1), \tag{29}$$

where $v_s(t+1)$ is the velocity at $(t+1)^{\text{th}}$ iteration. The P-Whale algorithm is introduced by modifying the above equation using the position update equation of WOA, represented as

$$H_s(t+1) = X'e^{ug}\cos(2\pi g) + G_{gb}, \tag{30}$$

where the distance measure X' is given by $X' = \big|G_{gb} - H_s(t)\big|$, u is a constant, g is a random number ranging from -1 to 1, and the best solution in WOA is replaced by the global best solution G_{gb}. The competitive performance of WOA ensures the effectiveness of the proposed algorithm. Equation (30) is rearranged as

$$G_{gb} = H_s(t+1) - X'e^{ug}\cos(2\pi g). \tag{31}$$

Substituting Eq. (31) in the velocity update equation of PSO, given in Eq. (28):

$$v_s(t+1) = Wv_s(t) + k_1h_1\big(G_{pb} - H_s(t)\big) + k_2h_2\big(H_s(t+1) - X'e^{ug}\cos(2\pi g) - H_s(t)\big). \tag{32}$$

This newly obtained velocity update equation is substituted in the position update equation [Eq. (29)], as expressed below

$$H_s(t+1) = H_s(t) + Wv_s(t) + k_1h_1\big(G_{pb} - H_s(t)\big) + k_2h_2\big(H_s(t+1) - X'e^{ug}\cos(2\pi g) - H_s(t)\big), \tag{33}$$

$$H_s(t+1) = H_s(t) + Wv_s(t) + k_1h_1\big(G_{pb} - H_s(t)\big) + k_2h_2H_s(t+1) - k_2h_2\big(X'e^{ug}\cos(2\pi g) + H_s(t)\big), \tag{34}$$

$$H_s(t+1) - k_2h_2H_s(t+1) = H_s(t) + Wv_s(t) + k_1h_1\big(G_{pb} - H_s(t)\big) - k_2h_2\big(X'e^{ug}\cos(2\pi g) + H_s(t)\big), \tag{35}$$

Table 1: Pseudocode of the P-Whale Algorithm.

Proposed P-Whale algorithm

```
Input: Intermediate data
Output: Global best solution $G_{gb}$
Parameters: $t$ → iteration, $v_s(t)$ → velocity at iteration $t$, $G_{gb}$ → global best solution, $G_{pb}$ → personal best solution
Begin
    Initialize the random population
    Assign the velocity $v_s(t)$ to each particle in the population
    While ($t <$ max_ $t$)
        for each particle
            Evaluate the fitness using Eq. (22)
            Determine $G_{pb}$ and $G_{gb}$
            Update the velocity $v_s(t + 1)$ using Eq. (32)
            Update the position of the particle $H_s(t + 1)$ using Eq. (38)
        end for
        Determine the best solution by replacing the existing solution based on the fitness function
    end while
    $t = t + 1$
    Return $G_{gb}$
Terminate
```

$$H_s(t+1)[1-k_2h_2] = H_s(t) + Wv_s(t) + k_1h_1\left(G_{pb} - H_s(t)\right) - k_2h_2\left(X'e^{ug}\cos(2\pi g) + H_s(t)\right), \tag{36}$$

$$H_s(t+1) = \frac{1}{[1-k_2h_2]}\left[H_s(t) + Wv_s(t) + k_1h_1\left(G_{pb} - H_s(t)\right) - k_2h_2\left(X'e^{ug}\cos(2\pi g) + H_s(t)\right)\right], \tag{37}$$

$$H_s(t+1) = \frac{1}{Z}[H_s(t) + d(t)], \tag{38}$$

where $Z = [1 - k_2h_2]$ and $d(t) = Wv_s(t) + k_1h_1\left(G_{pb} - H_s(t)\right) - k_2h_2\left(X'e^{ug}\cos(2\pi g) + H_s(t)\right)$. Equation (38) forms the position update equation of the proposed P-Whale algorithm, which improves the performance of the algorithm in clustering.

IV. Determining the Best Solution

After the position update, the new solutions generated are evaluated using the same fitness function. Among the fitness-evaluated solutions, the solution with the minimum fitness replaces the existing one and thus becomes the optimal solution.

V. Termination

The above steps are repeated until the number of iterations t reaches the maximum number of counts within which the optimal solution for clustering can be determined.

The pseudocode of the proposed P-Whale algorithm is presented in Table 1.

4 Results and Discussion

The results of the proposed FPWhale-MRF technique used for big data clustering are demonstrated in this section with the experimental setup and comparative analysis.

4.1 Experimental Setup

The experiment is carried out in a system operated using Windows 10 with the following configurations: RAM, 2 GB; system type, 64-bit Operating System (OS); and processor, Intel Pentium. The proposed technique is implemented using the JAVA software tool (Sun Microsystems, Oracle Corporation, Redwood City, CA, USA). The number of mappers and reducers used for the experimentation is six and seven, respectively.

Dataset Description: The number of datasets utilized for the experimentation is two, namely the localization dataset (dataset 1) [21] and the skin segmentation dataset (dataset 2) [22], taken from UCI Machine Learning Repository. The first dataset includes data obtained from various activities recorded from five different people who wore four tags: ankle left, ankle right, chest, and belt. The number of instances in the dataset is 164,860, and every instance represents a localization data for each tag. It consists of eight attributes, which can be used to identify the tag. The second is the skin segmentation dataset, which is built by sampling the R, G, and B values, generating skin and non-skin dataset from the FERET and PAL databases. This includes four attributes and 245,057 instances, with 50,859 skin samples and 194,198 non-skin samples.

4.2 Comparative Techniques

The performance of the proposed FPWhale-MRF is compared with four existing techniques, such as (i) Multiple Kernel and the Swarm-Based Map-Reduce Framework (MKS-MRF) [12], (ii) K-means-MRF [24], (iii) Fuzzy C-means-MRF (FCM-MRF) [25], and (iv) kernel fuzzy C-means-MRF (KFCM-MRF) [28]. Clustering is performed based on these existing techniques considering MRF in each technique for processing the big data. The performance of these techniques is evaluated using two performance evaluation metrics and compared in the comparative analysis.

4.3 Performance Evaluation Measures

The comparison of the performance of the comparative techniques is based on two evaluation metrics: DB-index, which is explained in Section 3.3.2, and clustering accuracy, which is defined as

$$ACC = \frac{1}{m}\sum_{i=1}^{N^C} \max_{j=1}^{C^L}\left(c_i \cap c_j^l\right), \tag{39}$$

where m is the number of data, N_C is the number of clusters, C^L is the number of classes, c_i denotes the i^{th} cluster, and c_j^l denotes the j^{th} class.

4.4 Evaluation of Performance

This section illustrates the performance evaluation of the proposed technique evaluated using the measures, accuracy, and DB-index, in the two datasets.

4.4.1 Accuracy Analysis

The analysis based on accuracy in the five comparative techniques performed using the datasets, skin segmentation and localization, is explained in this subsection using Figure 2. Figure 2 shows the accuracy

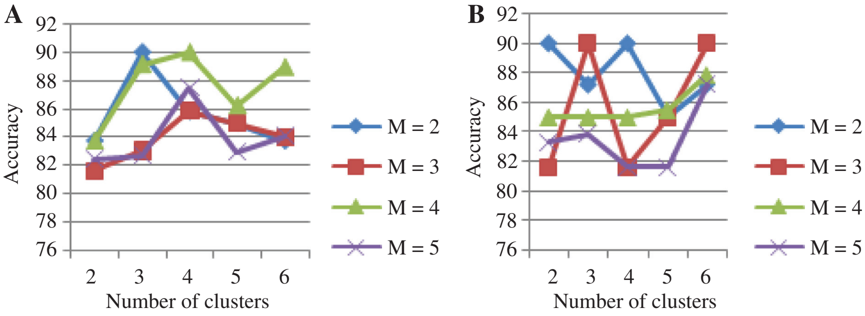

Figure 2: Accuracy Analysis Using (A) Dataset 1 and (B) Dataset 2.

analysis using Dataset 1 and Dataset 2. In Figure 2A, the resulting graph of accuracy analysis for dataset 1 is shown by plotting the accuracy for various mappers, denoted here as M, against the number of clusters varied from 2 to 6. Here, the maximum clustering accuracy of 90% is produced for $M = 2$ and 4, when the cluster size is 3 and 4, respectively. For a number of clusters of 6, the maximum accuracy possible is 88.95%, which is 1.18% less than the maximum accuracy produced. The accuracy analysis plot for dataset 2 is sketched out in Figure 2B. When $M = 2$, the accuracy obtained for the number of clusters 2 is 90%, which reduces to 83.33% when $M = 5$. Increasing the number of clusters to 6, the maximum accuracy of 90% is attained for $M = 3$.

4.4.2 DB-Index Analysis

Figure 3 presents the results of analysis based on the DB-index for the two datasets in the comparative techniques. The accuracy analysis for dataset 1 is given in Figure 3A. The lower the DB-index, the greater is the clustering performance. Here, the minimum value computed is 6.24 for $M = 4$ when the number of clusters is 2. When the number of clusters is 6, for $M = 3$, the DB-index value increases to a peak value of 365.98, which reduces to 91.79 for $M = 5$. In Figure 3B, the accuracy analysis for the second dataset is plotted. As $M = 2$, 3, 4, and 5, the DB-index measured is 19.24, 15.9, 5.85, and 10.96, for the number of clusters fixed as 2. The minimum DB-index obtained using the proposed FPWhale-MRF technique is 5.85. When the number of clusters is kept 6, the DB-index produced is 83.63, 172.53, 49.83, and 85.44, respectively, for $M = 2$, 3, 4, and 5.

4.5 Comparative Analysis

To evaluate the level of performance of the proposed technique with the existing techniques, the comparative analysis is performed. The analysis is done based on the accuracy and the DB-index using the two datasets.

4.5.1 Using Dataset 1

The comparative analysis made in the proposed technique and the four existing techniques using dataset 1 is depicted in Figure 4. Figure 4A presents the result of analysis based on accuracy using the first dataset by varying the number of clusters. When the number of clusters is 2, the accuracy obtained using the existing MKS-MRF, K-means-MRF, and FCM-MRF is the same, 75.58%, while that in FPWhale-MRF is 87.91%. As the number of clusters is increased to 5, the accuracy attained by the proposed technique is 85%, whereas 82.43% is the maximum accuracy produced by the existing MKS-MRF. The DB-index values analyzed using the comparative techniques with dataset 1 are shown in Figure 4B. Minimum value is observed in FPWhale-MRF for all the cluster sizes considered. When 10.83 is the minimum DB-index provided by FPWhale-MRF for the cluster size of 3, MKS-MRF, K-means-MRF, FCM-MRF, and KFCM-MRF have a DB-index of 29.79, 172.68, 187.72, and 185.26, respectively. Thus, from the analysis using dataset 1, i.e. localization data, the proposed FPWhale-MRF is observed to have the maximum performance than the other techniques compared.

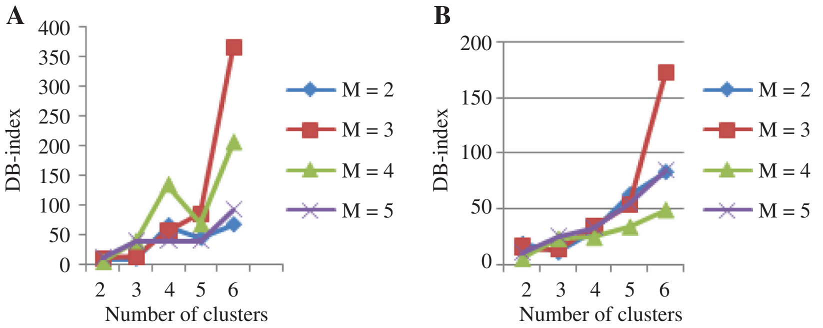

Figure 3: DB-Index Analysis Using (A) Dataset 1 and (B) Dataset 2.

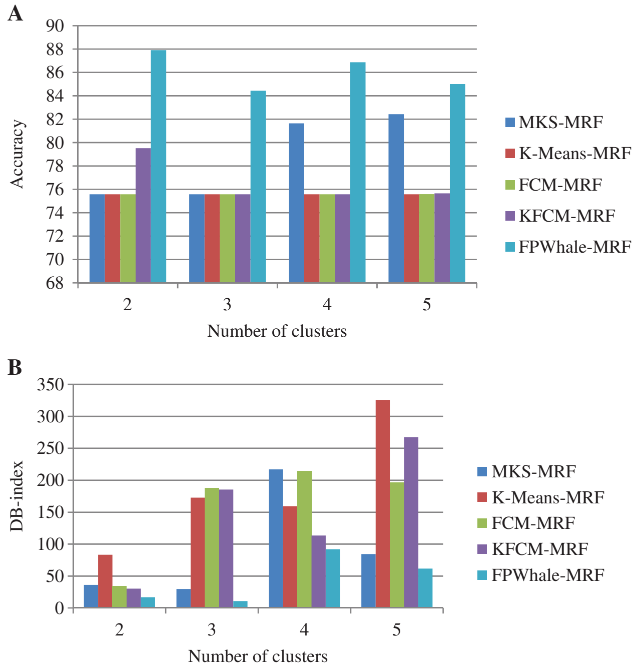

Figure 4: Comparative Analysis Using Dataset 1: (A) Accuracy and (B) DB-Index.

4.5.2 Using Dataset 2

In Figure 5, the comparative analysis result obtained in the five considered techniques using dataset 2 is sketched out. In the accuracy analysis graph shown in Figure 5A, the maximum accuracy produced by the proposed FPWhale-MRF is 90%, for the number of clusters of 2. In the same instant, the accuracy obtained

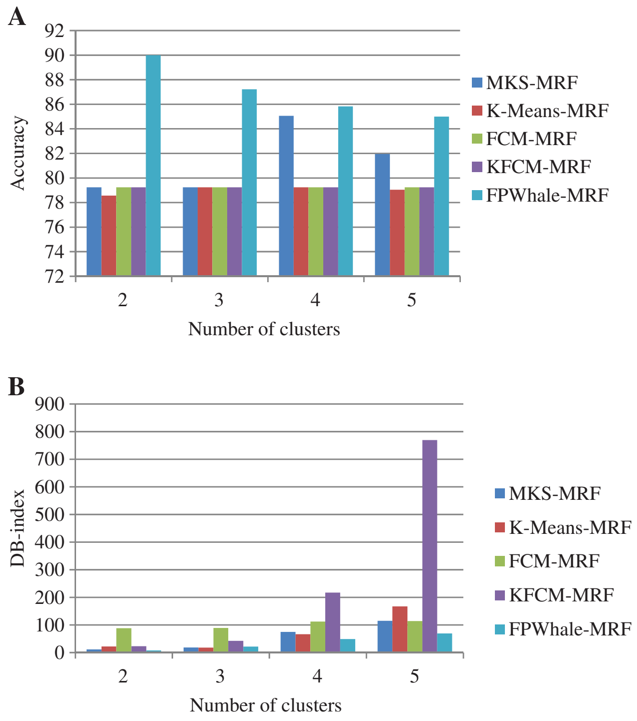

Figure 5: Comparative Analysis Using Dataset 2: (A) Accuracy and (B) DB-Index.

using the existing MKS-MRF, FCM-MRF, and KFCM-MRF is 79.24%. Meanwhile, K-means-MRF has a clustering accuracy of 78.57%. The analysis based on DB-index using dataset 2 is depicted in Figure 5B, where the minimum value is achieved by the proposed technique, with a DB-index of 7.73, for two clusters. For the same case, the minimum DB-index produced among the existing techniques is 12.01 by MKS-MRF. Hence, from the results of the analysis, the proposed FPWhale seems to have better performance than the other techniques considered for the comparison.

4.6 Convergence Analysis

Figure 6 shows the convergence analysis of the proposed technique for dataset 1 and dataset 2. The analysis is performed for various iterations (1–100). Figure 6A shows the convergence analysis of the proposed technique for dataset 1. When the number of iteration is 20, the DB-index of the proposed method is 98.36, which gradually decreases when the number of iterations increases. At 100th iteration, the proposed method has a DB-index of 10.83, which is smaller than the DB-index of the existing methods. Figure 6B shows the convergence analysis of the proposed technique for dataset 2. The proposed method has the minimum DB-index of 7.73 at the 100th iteration, which is lower than the DB-index of the existing methods.

4.7 Analysis Based on Computational Time

Table 2 shows the computational time of the proposed method and the existing methods. From the table, it can be seen that the proposed method has a minimum computational time of 5 s, while the existing methods, such as K-means-MRF, FCM-MRF, KFCM-MRF, and MKS-MRF, have computational times of 8, 7, 6, and 6.5 s, respectively.

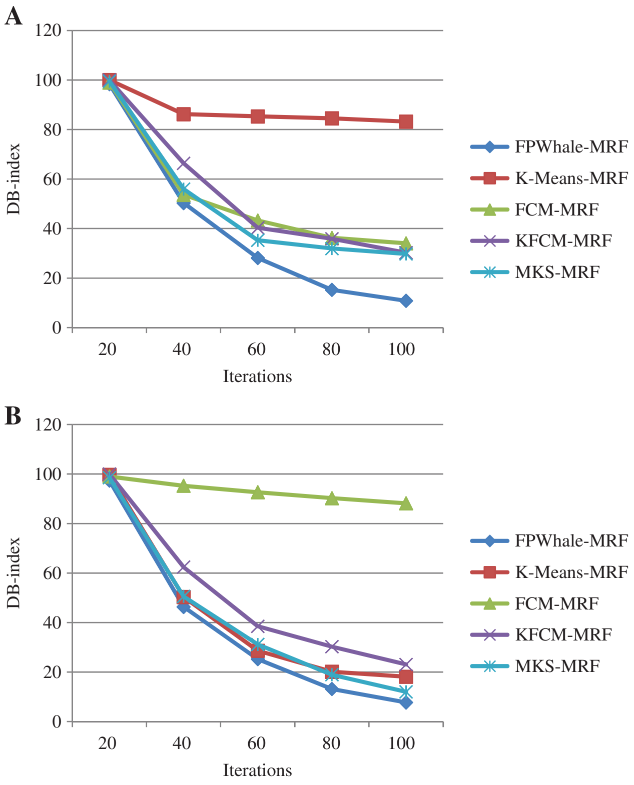

Figure 6: Comparative Analysis Using (A) Dataset 1 and (B) Dataset 2.

Table 2: Computational Time of the Comparative Methods.

Methods	Computational time (s)
K-means-MRF	8
FCM-MRF	7
KFCM-MRF	6
MKS-MRF	6.5
FPWhale-MRF	**5**

The bold value represents best performance.

4.8 Discussion

Based on the comparative analysis made in Section 4.5, a discussion is carried out regarding the maximum performance measures in MKS-MRF, K-means-MRF, FCM-MRF, KFCM-MRF, and FPWhale-MRF, as presented in Table 3.

Table 3 lists the maximum performance obtained by the comparative techniques for the two datasets by varying the cluster size from 2 to 5. For dataset 1, when MKS-MRF had an accuracy of 82.43%, the proposed FPWhale-MRF could provide 87.91% accuracy. For the same dataset, the minimum DB-index measured using the existing technique is 29.79 by MKS-MRF. Meanwhile, FPWhale-MRF has only 10.83 as the DB-index. On the analysis using dataset 2, the accuracy and the DB-index produced by the existing MKS-MRF are 85.06% and 12.01, whereas those in K-means-MRF are 79.24% and 18.11, respectively. In the meantime, the proposed FPWhale-MRF could obtain an accuracy of 90% and a DB-index of 7.73.

Table 4 shows the minimum performance attained by the comparative techniques for the two datasets by varying the cluster size from 2 to 5. For dataset 1, the minimum accuracy attained by the proposed FP-Whale-MRF is 84.44, while the minimum accuracy attained by the other existing methods is 75.58. For the same dataset, the maximum DB-index measured using the proposed technique is 91.86, which is smaller than the DB-index of other existing techniques. On the analysis using dataset 2, the accuracy and the DB-index produced by the existing MKS-MRF are 79.24 and 115.09, whereas those in K-means-MRF are 78.57 and 167.56, respectively. In the meantime, the proposed FPWhale-MRF could obtain an accuracy of 85% and a DB-index of 69.47.

Table 3: Performance Comparison Based on Maximum Performance.

Methods	Dataset 1		Dataset 2	
	Accuracy	DB-Index	Accuracy	DB-Index
MKS-MRF	82.43	29.79	85.06	12.01
K-means-MRF	75.58	83.2	79.24	18.11
FCM-MRF	75.59	34.04	79.24	88.17
KFCM-MRF	79.52	30.27	79.24	23.07
FPWhale-MRF	**87.91**	**10.83**	**90**	**7.73**

The bold values represent best performance.

Table 4: Performance Comparison Based on Minimum Performance.

Methods	Dataset 1		Dataset 2	
	Accuracy	DB-Index	Accuracy	DB-Index
MKS-MRF	75.58	216.91	79.24	115.09
K-means-MRF	75.58	325.92	78.57	167.56
FCM-MRF	75.58	214.35	79.24	114.36
KFCM-MRF	75.58	267.28	79.24	769.56
FPWhale-MRF	**84.44**	**91.86**	**85**	**69.47**

The bold values represent best performance.

Table 5: Performance Comparison Based on the Mean Performance.

Methods	Dataset 1		Dataset 2	
	Accuracy	DB-Index	Accuracy	DB-Index
MKS-MRF	78.81	91.79	81.37	55.23
K-means-MRF	75.58	185.25	79.03	68.59
FCM-MRF	75.58	158.02	79.24	101.01
KFCM-MRF	76.59	149.08	79.24	263.14
FPWhale-MRF	**86.06**	**45.33**	**87.01**	**37.04**

The bold values represent best performance.

Table 5 lists the mean performance obtained by the comparative techniques for the two datasets by varying the cluster size from 2 to 5. For dataset 1, when MKS-MRF has an accuracy of 78.81%, the proposed FPWhale-MRF has 86.06% accuracy. For the same dataset, the DB-index measured using the existing MKS-MRF technique is 91.79. Meanwhile, FPWhale-MRF has only 45.33 as the DB-index. On the analysis using dataset 2, the accuracy and the DB-index produced by the existing MKS-MRF are 81.37% and 55.23, whereas those in K-means-MRF are 79.03% and 68.59, respectively. In the meantime, the proposed FPWhale-MRF obtains an accuracy of 87.01% and a DB-index of 37.04.

5 Conclusion

In this paper, a technique for big data clustering is presented using MRF, named FPWhale-MRF, based on two clustering algorithms, FTSK and P-Whale. FTSK adopted in the mapper is developed by the integration of fractional calculus in the TSK clustering algorithm to find the cluster centroids. Meanwhile, the reducers contain an optimization-based clustering algorithm, P-Whale, developed by modifying PSO using WOA, for optimal clustering. Thus, the proposed FPWhale-MRF technique performs big data clustering effectively using the proposed clustering algorithm. The experiment is performed using two datasets, localization and skin segmentation, and the results are compared with that of the existing techniques, such as MKS-MRF, K-means-MRF, FCM-MRF, and KFCM-MRF. The performance of the proposed technique is evaluated using two metrics, clustering accuracy and DB-index. FPWhale-MRF could attain the maximum accuracy of 87.91% and 90% for the localization and skin segmentation datasets, whereas that in the existing MKS-MRF is 82.43% and 85.06%, respectively. Therefore, it can be concluded that the proposed FPWhale-MRF technique can perform big data clustering effectively with maximum clustering accuracy compared with the existing comparative techniques.

Bibliography

[1] N. Akthar, M. V. Ahamad and S. Khan, Clustering on big data using Hadoop MapReduce, in: *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, Jabalpur, pp. 789–795, IEEE, Piscataway, NJ, USA, 2015.

[2] A. Banharnsakun, A MapReduce-based artificial bee colony for large-scale data clustering, *Pattern Recognit. Lett.* **93** (2017), 78–84.

[3] P. R. Bhaladhare and D. C. Jinwala, A clustering approach for the *l*-diversity model in privacy preserving data mining using fractional calculus-bacterial foraging optimization algorithm, *Adv. Comput. Eng.* **2014**, (2014), 1–12.

[4] N. Bharill, A. Tiwari and A. Malviya, Fuzzy based scalable clustering algorithms for handling big data using Apache Spark, *IEEE Trans. Big Data* **2** (2016), 339–352.

[5] C. Chen, K. Li, A. Ouyang and K. Li, A parallel approximate SS-ELM algorithm based on MapReduce for large-scale datasets, *J. Parallel Distrib. Comput.* **108** (2017), 85–94.

[6] D. L. Davies and D. W. Bouldin, A cluster separation measure, *IEEE Trans. Pattern Anal. Mach. Intell.* **1** (1979), 224–227.

[7] S. Fries, S. Wels and T. Seidl, Projected clustering for huge data sets in MapReduce, in: *Proceedings of International Conference on Extending Database Technology (EDBT)*, Athens, Greece, pp. 49–60, OpenProceedings.org, Konstanz, Germany, 2014.

[8] K. Grolinger, M. Hayes, W. A. Higashino, A. L'Heureux, D. S. Allison and M. A. M. Capretz, Challenges for MapReduce in big data, in: *2014 IEEE World Congress on Services*, Anchorage, AK, pp. 182–189, IEEE, Piscataway, NJ, USA, 2014.

[9] N. Hans, S. Mahajan and S. N. Omkar, Big data clustering using genetic algorithm on Hadoop Mapreduce, *Int. J. Sci. Technol. Res.* **4** (2015), 58–62.

[10] S. Kamal, S. H. Ripon, N. Dey, A. S. Ashour and V. Santhi, A MapReduce approach to diminish imbalance parameters for big deoxyribonucleic acid dataset, *Comput. Methods Progr. Biomed.* **131** (2016), 191–206.

[11] H. Ke, P. Li, S. Guo and M. Guo, On traffic-aware partition and aggregation in MapReduce for big data applications, *IEEE Trans. Parallel Distrib. Syst.* **27** (2016), 818–828.

[12] O. Kulkarni and S. Jena, MKS-MRF: a multiple kernel and a swarm-based MapReduce framework for big data clustering, *Int. Rev. Comput. Softw.* **11** (2016).

[13] D. Kumar, J. C. Bezdek, M. Palaniswami, S. Rajasegarar, C. Leckie and T. C. Havens, A hybrid approach to clustering in big data, in: *IEEE Trans. Cybernet.* **46** (2016), 2372–2385.

[14] O. Kurasova, V. Marcinkevicius, V. Medvedev, A. Rapecka and P. Stefanovic, Strategies for big data clustering, in: *Proceedings of IEEE 26th International Conference on Tools with Artificial Intelligence*, Limassol, pp. 740–747, IEEE, Piscataway, NJ, USA, 2014.

[15] S. Mirjalili and A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.* **95** (2016), 51–67.

[16] F. Pulgar-Rubio, A. J. Rivera-Rivas, M. D. Pérez-Godoy, P. González, C. J. Carmona and M. J. del Jesus, MEFASD-BD: multi-objective evolutionary fuzzy algorithm for subgroup discovery in big data environments – a MapReduce solution, *Knowl.-Based Syst.* **117** (2017), 70–78.

[17] H. Rehioui, A. Idrissi, M. Abourezq and F. Zegrari, DENCLUE-IM: a new approach for big data clustering, *Proc. Comput. Sci.* **83** (2016), 560–567.

[18] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah and T. Herawan, Big data clustering: a review, in: *Proceedings of International Conference on Computational Science and Its Applications, ICCSA 2014*, pp. 707–720, 2014.

[19] H. Singh and S. Bawa, A MapReduce-based scalable discovery and indexing of structured big data, *Future Gen. Comput. Syst.* **73** (2017), 32–43.

[20] P. A. Traganitis, K. Slavakis and G. B. Giannakis, Sketch and validate for big data clustering,*IEEE J. Select. Topics Signal Process.* **9** (2015), 678–690.

[21] UCI Machine Learning Repository, Localization data for person activity data set, Available at: https://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity, Accessed 24 March, 2018.

[22] UCI Machine Learning Repository, Skin segmentation data set, Available at: https://archive.ics.uci.edu/ml/datasets/skin+segmentation, Accessed 24 March, 2018.

[23] M. Vadivel and V. Raghunath, Enhancing Map-Reduce framework for big data with hierarchical clustering, *Innov. Res. Comput. Commun. Eng.* **2** (2014), 490–498.

[24] D. Xia, B. Wang, Y. Li, Z. Rong and Z. Zhang, An efficient MapReduce-based parallel clustering algorithm for distributed traffic subarea division, *Discrete Dynam. Nat. Soc.* **2015** (2015) Article ID 793010, 18 pp.

[25] Q. Yu and Z. Ding, An improved fuzzy C-means algorithm based on MapReduce, in: *Proceedings of 8th International Conference on Biomedical Engineering and Informatics (BMEI)*, Shenyang, China; IEEE, Piscataway, NJ, USA, 2015.

[26] M. Yuwono, S. W. Su, B. D. Moulton and H. T. Nguyen, Data clustering using variants of rapid centroid estimation, *IEEE Trans. Evol. Comput.* **18** (2014), 366–377.

[27] B. Zerhari, A. A. Lahcen and S. Mouline, Big data clustering: algorithms and challenges, in: *Proceedings of International Conference on Big Data, Cloud and Applications BDCA'15*, May 2015.

[28] H. Zhu, Y. Guo, M. Niu, G. Yang and L. Jiao, Distributed SAR image change detection based on Spark, in: *Proceedings of IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS)*, Milan, Italy; IEEE, Piscataway, NJ, USA, 2015.