The International Journal of Interdisciplinary Organizational Studies
ISSN: 2324-7649 (Print), ISSN: 2324-7657 (Online)
Volume 19, Issue 1, July-December, 2024

EFCM-FFO: A NOVEL HYBRID ENHANCED FUZZY C-MEANS CLUSTERING BASED FRUIT FLY OPTIMIZATION IN HADOOP MAPREDUCING MODEL

M. M. Kavitha
Research Scholar, Bharathiar University, Coimbatore

Dr. K. Anandapadmanabhan
M.CA., M.Phil., Ph.D., Research Supervisor, Dean, Sri Vasavi College (SF Wing), Erode.

Abstract:

The clustering of Bigdata is a common task in data mining and machine learning. The goal is to group similar data points to identify patterns and relationships in the data. However, clustering large datasets can be computationally expensive and time-consuming. K-Means Clustering is a very powerful and frequently used algorithm for the clustering, it has got its own limitation. Hadoop is a sophisticated framework that facilitates the distributed processing of voluminous datasets across multiple clusters of computers. MapReduce is a programming model that simplifies the processing of large datasets by breaking them down into smaller chunks and processing them in parallel across the cluster. One approach to Enhanced Fuzzy C-Means clustering Bigdata using Hadoop MapReduce is to use a genetic algorithm. A Fruit Fly Optimization (FFO) algorithm finds the wellness of the populace to choose the optimal C values as far as execution time and classification error. The experiments show that the improved algorithm is generally applicable to the clustering of different shape class clusters and larger scale data and has obvious improvement in accuracy and parallel efficiency.Two datasets, namely localization and skin segmentation datasets, are used for theexperimentation and the performance is evaluated regarding two performance evaluation metrics: clusteringaccuracy and DB-index. The maximum accuracy attained by the proposed EFCM-FFO technique is 87.91%and 90% for the localization and skin segmentation datasets, respectively, thus proving its effectiveness inbig data clustering.

Keywords: clustering, big data, MapReduce, Hadoop and skin segmentationddataset.

Introduction:

Science and industrialization have enhanced the capacity for information in practically every subject of research and engineering, as well as in many applications [1]. Speed, diversity, and volume [2] are the three data attributes that big data [3] now possesses. The rate at which data is processed and created based on the applications required is referred to as speed, whilst the kind and type of data is referred to as variety. On identifies data volume for calculating data value [4]. Existing storing and processing technologies may be unable to handle the pace, variety, and volume of bulk data. The term "big data" describes this information. Analysing Big Data is a strategy for discovering important geometric and statistical patterns in enormous datasets. Besides of data storage and access, the massive rise in data causes a variety of processing issues. Since data

collection is expensive, it is critical that the data be utilized properly so that more progress may be achieved by building more efficient algorithms.

With the advancement of technology, the amount of data that is being created and stored on a global level is almost inconceivable and keeps growing. Data volumes processed by many applications crosses the peta-scale threshold and this massive volume of data is termed as big data [5]. It is estimated that the data is growing at a 40% compound annual rate, reaching nearly 45 Zeta bytes by 2020. Big data is an assortment of so large and complex that it becomes difficult to process using conventional database management tools. The challenges include capture, storage, search, sharing, analysis, and visualization. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets [6]. Data-driven decision making as well as the burgeoning demand for data analytics has inspired increasing numbers of scholars as well practitioners to develop and apply clustering algorithms [7].

Generally, there are four categories of clustering methods: Hierarchical method is based on the distance between objects and clusters. The idea of hierarchical methods is that objects are more related to nearby objects rather than the farther objects. A balanced iterative reducing clustering hierarchy (BIRCH) is the well-knownalgorithm in this category. The second category is thepartitioning method, the main idea is that it construct k(k<n) partitions and then evaluate them by some criterion,for example, minimizing the sum of square errors. Typicalalgorithms include k-means and affinity propagationclustering (AP) [8]. Density-based method is the thirdcategory, clusters are dense regions in the data space,separated by regions of lower object density and a cluster isdefined as a maximal set of density-connected points. Thefourth is the model-based method, which is hypothesized foreach of the clusters and tries to find the best fit of that modelto each other; the well-known algorithm is expectationmaximization clustering (EM). However, theconventional data analytics are becoming less suitable for handling the current big data processing.

To meet the challenge of big data applications, researchers proposed new methods and extended standard clustering algorithms to tackle big data which is too large to load it all into the memory for deriving clusters. K-means was extended to hybridize with particle swarm algorithm k-means clustering (CPSO) for enhancing the search ability over high dimensional data by Tang in 2012.Data clustering can be easily cast as a global optimization problem of finding the partition that maximizes/minimizes an objective function. This can be appropriately tackled using meta-heuristics, such as Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO) and so forth. These meta-heuristics exhibit different dynamics leading to distinct strategies that can be effectively combined to handle hard optimization problems such as Clustering large data sets [9].

Contribution:

Today is the world of computer, technologies and computing devices, because new technologies have made computer systems faster and more affordable. Due to the growing number of labs, departments, and institutions high performance parallel systems are required, such as clusters. Because a single computing system has limited computing resources, so to satisfy the increasing computational demands there is one method to utilize computing resources across multiple distributed computing systems. Here a Hybrid Approach for Clustering Big Data by Adopting Fruit Fly Optimization Algorithm is proposed for parallel processing.

- We employ a Fruit Fly Optimization Algorithm (FFO) to tackle the local optimality problem in Enhanced Fuzzy C-Means clustering technique and to produce tighter and more cohesive clusters based on incorporating criteria for these characteristics in the objective function.
- We design parallel processing scheme for EFCM-FFO mechanisms that implemented in Hadoop environment for reducing the execution time.
- Mapper approach generates the population for given data set for clustering. The reducer approach finds the fitness of the population to select the optimal clusters in terms of execution time and classification error.
- Through extensive experimental results, we evaluate the performance of the proposed EFCM-FFO scheme.

Literature Survey:

This section presents the literary assessment, in which different approaches for the classification of enormous data in works are discussed along with problems. Also included in this section is an analysis of any flaws. They developed a distribution-based nearest neighbor classification model based on clusters for the purpose of doing various analyses more quickly and presented it in [10]. Map Reduced Method was established to execute process employing computing mechanism sample for optimizing the sampling process. The problem with dimensions continued to exist throughout the procedure, despite the fact that the approach delivered excellent accuracy decrease rates. Utilizing the method of multiplier optimization, the authors of [11] presented Echo State Networks as a means of carrying out neighborhood exchanges between components that are located closer together. In addition, training patterns for new nodes were not required in any way. The experiment performed on synthetic data showed improved performance; nonetheless, adding weights without first taking into consideration inaccurate value estimates is a drawback.

In this study, we present a method known as social group optimization (SGO), which is an optimization strategy that uses populations [12]. It derives its motivation from the notion of social conduct shown by humans in the process of resolving a difficult issue. In this article, a flowchart is used to describe both the conceptual framework of the SGO algorithm as well as its mathematical formulation.

The clever actions that are shown by groups of insects or animals in nature have ensured the continued existence of their species over the course of thousands of years. In this study, a new swarm intelligence method for addressing optimization problems dubbed the social group entropy optimization (SGEO) algorithm is developed [13]. SGEO stands for the social group entropy optimization. The primary contributions of this research are the social group model, the status optimization model, and the entropy model. These models serve as the foundation for the algorithm that has been suggested.
Identifying the best solutions to technical applied issues is necessary due to the presence of financial and physical restrictions; yet global optimization algorithms are unable to provide these solutions [14]. It is required to move between known various local and global solutions in order to achieve optimization that is both precise and quick. In the work [15] that was done recently, a social group optimization, or SGO, was offered as a means of tackling issues involving multimodal functions and data clustering.

Malignant is now one of the most prevalent forms of severe cancer in the human population. Melanomas are cancers that begin in the skin. As a direct result of this, there is a growing need for methods that are both automated and resilient in order to provide accurate and prompt clinical identification and detection of skin cancer [16]. A social group optimization (SGO) assisted automated technique was created for the purpose of analyzing dermoscopy pictures for signs of skin melanoma in this present body of work.

The Social Group Optimization Algorithm, is a meta-heuristic optimization method that was presented in the year 2016 for the purpose of tackling issues involving global optimization [17]. In the research that has been published, it has been shown that SGO is successful in comparison to other optimization techniques.

Proposed Method:

Preliminaries:

MapReduce Paradigm:

MapReduce [6] is a programming paradigm used for processing large data sets. It was developed by Google to handle the large-scale distributed data processing in a parallel and fault-tolerant manner. MapReduce divides a large data set into smaller parts and processes them in parallel across multiple computers in a cluster. The MapReduce paradigm consists of two main phases: the map phase and the reduces phase. In the map phase, the input data is processed and converted into intermediate key-value pairs. The map function takes input data and produces a set of intermediate key-value pairs. The key-value pairs are then shuffled and sorted based on the keys, and sent to the reduce phase. In the reduce phase, the intermediate key-value pairs are combined based on their keys to produce the final output. The reduce function takes a set of intermediate key-value pairs with the same key and produces a set of output key-value pairs.

The MapReduce [8] paradigm also includes a partitioning function that determines how the intermediate key-value pairs are distributed across the reducers. The partitioning function ensures that all key-value pairs with the same key are sent to the same reducer. The advantages of MapReduce include scalability, fault tolerance, and ease of programming. MapReduce can handle large-scale data processing across multiple machines in a cluster, and it automatically handles failures by re-executing failed tasks on other machines. MapReduce also provides a simple programming model for developers to write distributed data processing applications.

MapReduce [12] is commonly used in big data processing frameworks such as Apache Hadoop, Apache Spark, and Apache Flink. These frameworks provide high-level APIs for developers to write MapReduce programs, and handle the details of distributed data processing and fault tolerance.

Hadoop MapReduce:

Apache Hadoop [14] MapReduce is a distributed data processing framework that is designed to handle large data sets in a distributed and parallel manner. It is a key component of the Apache Hadoop ecosystem and allows for the distributed processing of data across a large number of nodes. MapReduce framework consists of two main components: Map and Reduce. The Map component takes input data and transforms it into a set of intermediate key value pairs. The Reduce component takes the output of the Map component, processes it, and produces the final output.

In Hadoop MapReduce [14], data is stored in a distributed file system called Hadoop Distributed File System (HDFS). The HDFS is designed to handle large data sets and provides a fault-tolerant storage layer for Hadoop MapReduce jobs.

The Hadoop MapReduce [17][18] programming model is based on the concept of functional programming. The Map and Reduce components are functions that operate on the input data and produce output data. The Map function is executed in parallel on multiple nodes, with each node processing a subset of the input data. The Reduce function is also executed in parallel on multiple nodes, with each node processing a subset of the intermediate key-value pairs produced by the Map function.

One of the key advantages of Hadoop MapReduce is its scalability. It can be used to process large amounts of data by distributing the workload across multiple nodes in a cluster. This makes it possible to process large data sets that cannot be processed on a single machine. It also provides fault tolerance, meaning that if a node fails during the processing of a job, the framework will automatically re-run the job on another node to ensure that it is completed successfully. It has become a popular framework for large-scale data processing, and it has been used in a wide range of applications, including web search, image processing, and machine learning.

Enhanced Fuzzy C-Mean's approach:

In IFCM, the probability value of every data point is assigned for the corresponding CC depending on the data point and the clustering distance. IFCM also yields exceptional results in cases when the data overlap exists. While computing time and precision are required, it also needs the execution of many iterations, and Eudoxus' Euclidean distance assesses uneven weight. As a result, this may be accomplished by combining CNN with encoder-decoder [19].

Let us consider the dataset$Z=\{z1,z2,\ldots,zq\}$ with cluster set$X=\{x1,x2,\ldots,xp\}$ and probability set W={wkl|1≤k≤e,1≤1≤p}.

We advance this FCM as an Optimized Auto-Encoder

$$min: \sum_{k=1}^{e} \sum_{l=1}^{p} w_{kl}^{o} \|z_l - x_k\|^2$$
$$\sum_{l=1}^{e} w_{kl} = 1, w_{kl} \geq 0 \quad (1)$$

In order to overcome limitations, we introduce an Advanced FCM approach

$$L_o(W,X) = \sum_{k=1}^{e} \eta_i \sum_{k=1}^{p} (1 - u_{kl}^{o})^{\circ} + \sum_{k=1}^{e} \sum_{k=1}^{p} w_{kl}^{m} \|z_l - z_i\|^2 \quad (2)$$

Optimizing this gives:

$$x_k = \sum_{l=1}^{p} w_{kl}^{o} z_l / \sum_{l=1}^{p} w_{kl}$$

Membership matrix

$$w_{kl} = \left(1 + \left(\frac{e_{kl}}{\eta_k}\right)^{-1/(o-1)}\right)^{-1}$$

Here ekl is the distance from cluster to membership matrix.

In general, FCM techniques utilise the unsupervised analysis to place unknown data components in the best appropriate cluster or organization. Through the data, it accumulates N- (biggest) clusters. This technique assigns every data element to one of N classes based on its functionality and distance. Routine statistics components are grouped according to similarities and a centroid is created for each cluster reflecting function values. Since each centroid is linked to a cluster component, choosing the most exact and suitable centroid is critical. Calculating the mean in

cluster seems to be an iterative method that takes as inputs the number of N clusters and the data items andgroups them based on their similarity [20]. It works by starting with a random centroid and updating it based on each detail. The sections that follow discuss IFCM clustering.

i. Assigning Data: Data gathering is the first step in clustering. The initial cluster centroid produced is one-of-a-kind. Each fact is assigned to its nearest centroid using Euclidean, Manhattan, and City-block distances. Using Cj as the jth centroids of C, each element x may be assigned to a cluster using the equation below.

$$\operatorname{argmin} \operatorname{dist}(C_j, x)^2$$

In the equation given above, dist(.) represents distance (L2). Suppose si represents a collection of data points allocated to each centroid of a cluster, and then it should be updated repeatedly.

ii. Updating Centroid Distance: By calculating the mean of all associated data items, IFCM updates centroids repeatedly. In principle

$$C_i = \frac{1}{|S_i|} \sum x_i$$

This approach is repeated until the termination conditions are satisfied. We employed a feature-adaptive (or performance-adaptive) halting scenario, as opposed to typical IFCM algorithms, that use iterations as stopping criteria. Since we developed SSGO for IFCM clustering, which cluster all data components depending on best model characteristics (i.e., similarity or distance metrics with the greatest centroid selection), our recommended SSGO is set termination criterion to obtain optimum overall performance.

Fruit Fly Optimization Algorithm:

Fruit fly optimization algorithm (FOA) is an optimization algorithm that simulates the foraging behavior of fruit fly swarms. In FOA, a fruit fly searches for food by continuously updating the position. The parameters of the fruit fly optimization algorithm are simple in structure and easy to adjust. If the number of fruit flies is N_f, the positions of fruit flies are X_{axis} and Y_{axis}. The basic fruit fly optimization algorithm update iteration formula is described in the following equation:

$$\begin{cases} x_{j,t_f} = X_{\text{axis},t_f} + R_{t_f} \times \text{rand}, \\ y_{j,t_f} = Y_{\text{axis},t_f} + R_{t_f} \times \text{rand}, \end{cases}$$

where j is the fruit fly serial number. $j \in \{1, 2, \cdots, N_f\}$. t_f is the fruit fly dimension. $t_f \in \{1, 2, \cdots, d\}$. rand is the random number. $\text{rand} \in [0, 1]$. is the search radius of the t_f dimension of the fruit fly.

Since the food location is not known, it is necessary to calculate the distance Dist_j between the current individual position of the fruit fly with serial number j and the origin and then calculate the taste concentration judgment value Sm_j. Sm_j is the reciprocal of the distance Dist_j, and the specific calculation formula is described in the following equation:

$$\begin{cases} \text{Dist}_j = \sqrt{x_j + y_j} \\ Sm_j = \dfrac{1}{\text{Dist}_j}. \end{cases}$$

The fruit fly optimization algorithm determines the merit by its flavor concentration value, which is calculated as described in the following equation:

$$\text{Smell}_j = f_s(Sm_j),$$

where Smell_j is the taste concentration function value of the jth individual fruit fly and f_s is the formula for calculating the taste concentration value.

Parallel Processing Scheme for EFCM-FFO Mechanisms in MapReduce Hadoop:

In this paper we intend to propose a Hadoop MapReduce hybrid Enhanced Fuzzy C-means clustering algorithm and a Fruit Fly Optimization (FFO) algorithm to manage vast data set. The mean value estimations of data indicated in clusters are utilized to measure cluster the quality of being similar to something.

Then Map Reduce approach is used to calculate the distance between the information focused simultaneously. The main advantage of this algorithm is its less mathematical complexity. As we are using the map reduce approach fast convergence is possible. Accuracy and wide application is another advantage of this algorithm. MBFO is not largely affected by the size and non- linearity of the problem.

When grouping expansive data sets, parallel handling is most appropriate to Hadoop MapReduce because of its ability to separate extensive data sets into lumps and to store them in laborer nodes. The hybrid EFCM-FFO calculation is adequate to expansive data sets however for a decent clustering algorithm tending to substantial data sets, cluster efficiency and Euclidean distance based cluster measures ought to be viewed as separated from execution times and ought to be scaled up to meet input data estimate prerequisites.

For large data set, the processing and result generation process takes a considerable amount of time and our proposed hybrid EFCM-FFO approach speeds up the process within a reasonable execution time, the proposed parallel processing scheme is depicted in figure 1.

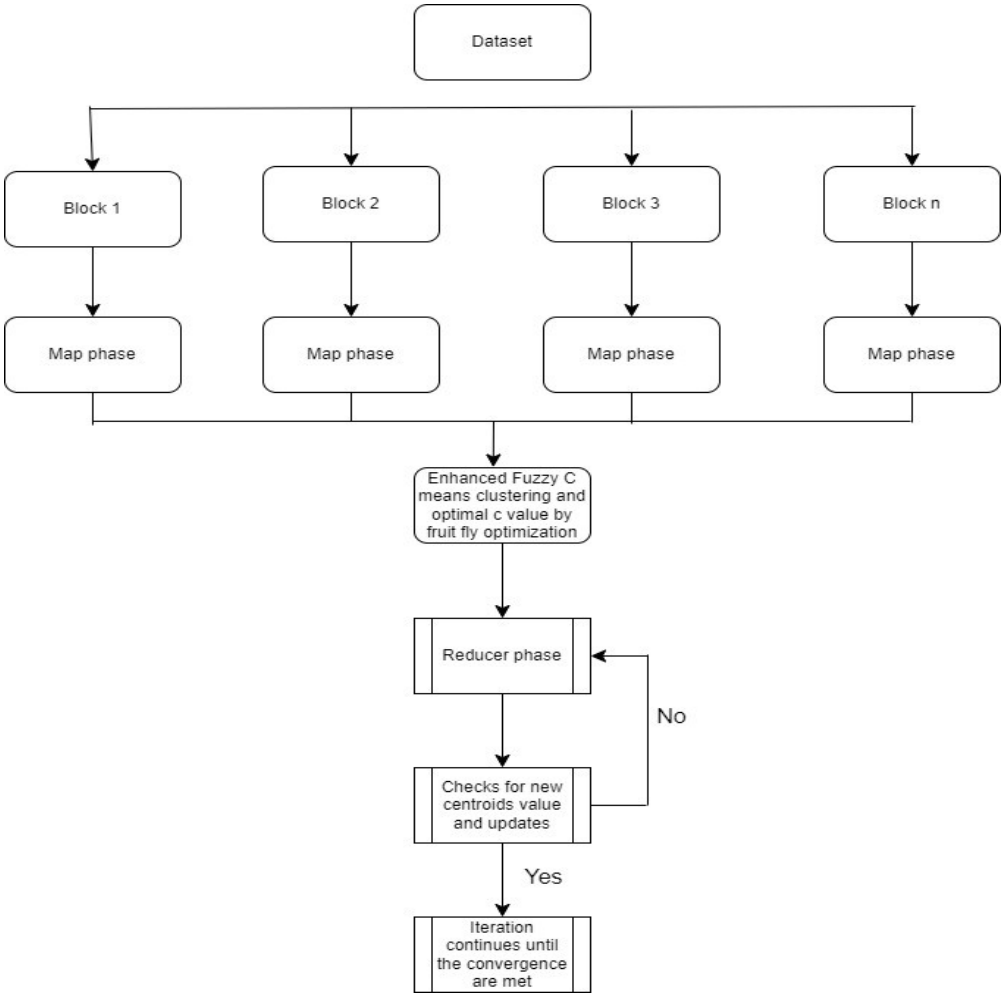

Figure: Parallel processing scheme of proposed EFCM-FFO

Steps Involved in our Proposed Approach:

- The map phase stores input data sets in a MapReduce Hadoop which is utilized to relegate information objects or indicates the nearest focus and which decreases works in updating determined centers dependent on the Euclidean distance.
- EFCM-FFO focuses on the MapReduce implementation of standard Fuzzy C-means, which serves as a framework for parallelization problems (e.g., clustering) which involves two major phases: a map phase and a reduce phase.
- The MapReduce job involves splitting a data set into parts of a fixed sized referred to as chunks.

- The map phase figures the separations between each item and each group and allots each object to its closest cluster. One map task is made for each input split and is executed by map abilities for each record of the data/information split.
- Initial C data points as cluster centers were selected optimally by means of FFO algorithm and update the centroid value for each and every iteration.
- In the map phase the data file with cluster centers form a key value and the distance between the clusters were computed by means of Euclidean distance.
- The objects among similar cluster are sent to reducer stage. The reducer stage computes the new group by combining and finds centroids for the following MapReduce work.
- In order to improve the clustering process efficiency, we make use of an FFO algorithm and the large data sets described in Data sets.
- Figure 2 portrays the general progression of EFCM-FFO. Group centroids delivered toward the end of an underlying cycle are put away in an old cluster record and are verified for the presence of new cluster centroids with every iteration process.
- In mapping phase, assigning the data points to the nearest clusters. In the combine phase clustered data is gone through optimization for the centroids of the clusters. Then check it out with the old one and update.
- At the point when new group centroid esteems are gotten, new cluster centroid esteems are refreshed in another document and the quantity of cycles is expanded by one.
- This procedure is rehashed until no more changes in group centroid esteems are found, and this state is alluded to as convergence. The last yield clusters are warehoused in an output file record.
- The outcomes have been dissected dependent on different sizes of data sets; better outcomes have been accomplished with expanding data set size.

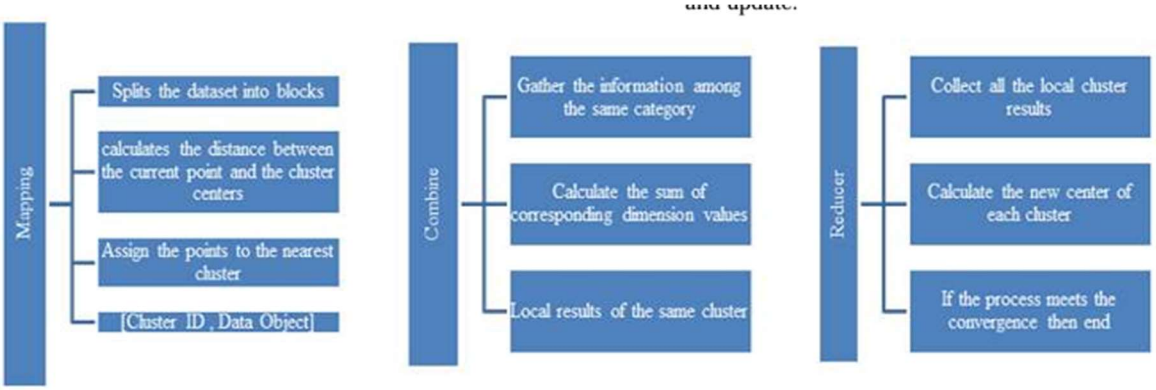

Figure 2: General progression of EFCM-FFO

Implementation and analysis:

The Hadoop cluster is a special parallel computational cluster that includes a master node and several slave nodes. For a given data set, a file is split into numerous components equal to the block size set for the Hadoop MapReduce cluster. Several experiments were conducted on various numbers of files in the data sets to evaluate the quality and scalability of our proposed algorithms.

To evaluate the performance of the EFCM-FFO algorithm, we compared it to other algorithms [21] (e.g., standard K-means, KM-HMR, BFO,PSO). Here we simulated the data set in [22] for performance analysis. The KDD-99 training data set consists of 4 292 637 instances of 41-dimensional vectors and is labelled data that specifies the attack type (normal or attack). KDD-99 has 22simulated attack types, which fall in one of the following four categories.

- Denial of service attack
- Users to root attack
- Remote to local attack
- Probing attack

The experiments were carried out on a system with Intel Core i3 CPU M 380 @ 2.53 Ghz and 4GB RAM running Window 8 Professional 64-bit operating system. We have performed experiments to various clustering algorithms and file sizes for the given data set. We selected this data set due to the reason that clustering makes it easier for grouping of attacks according to their similarities.

All of the proposed algorithms were run with attack data sets of different sizes taken from the above data set to generate effective clustering results. This data set was used to investigate and explore documents to conduct a cluster analysis of unstructured data with better execution times.

Results and Discussion:

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

Dataset Description:

The number of datasets utilized for the experimentation is two, namely the localizationdataset (dataset 1) [21] and the skin segmentation dataset (dataset 2) [22], taken from UCI Machine LearningRepository. The first dataset includes data obtained from various activities recorded from five different peoplewho 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 beused to identify the tag. The second is the skin segmentation dataset, which is built by sampling the R, G,and Bvalues, generating skin and non-skin dataset from the FERET and PAL databases. This includes fourattributes and 245,057 instances, with 50,859 skin samples and 194,198 non-skin samples.

Comparative Techniques:

The performance of the proposed EFCM-FFO is compared with four existing techniques, such as KM-MBFO and FCM-PSO [28]. Clustering is performedbased on these existing techniques considering MRF in each technique for processing the big data. The performanceof these techniques is evaluated using two performance evaluation metrics and compared in thecomparative analysis.

Performance Evaluation Measures

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

$$ACC = \frac{1}{m}\sum_{t=1}^{N^c} \max_{J=1}(c_t \cap c_j^i)$$

where *m* is the number of data, *NC* is the number of clusters, *CL* is the number of classes, *ci* denotes the *i*thcluster, and *clj*denotes the *j*th class.

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.

Accuracy Analysis:

The analysis based on accuracy in the three comparative techniques performed using the datasets, skin segmentationand localization, is explained in this subsection using Figure 3. Figure 3 shows the accuracyanalysis using Dataset 1 and Dataset 2. InFigure 3A, the resulting graph of accuracy analysis for dataset 1is shown by plotting the accuracy for various mappers, denoted here as *M*, against the number of clustersvaried from 2 to 6. Here, the maximum clustering accuracy of 90% is produced for *M* = 2 and 4, when thecluster 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 sketchedout in Figure 3B. When $M = 2$, the accuracy obtained for the number of clusters 2 is 90%, which reduces to83.33% when $M = 5$. Increasing the number of clusters to 6, the maximum accuracy of 90% is attained for$M = 3$.

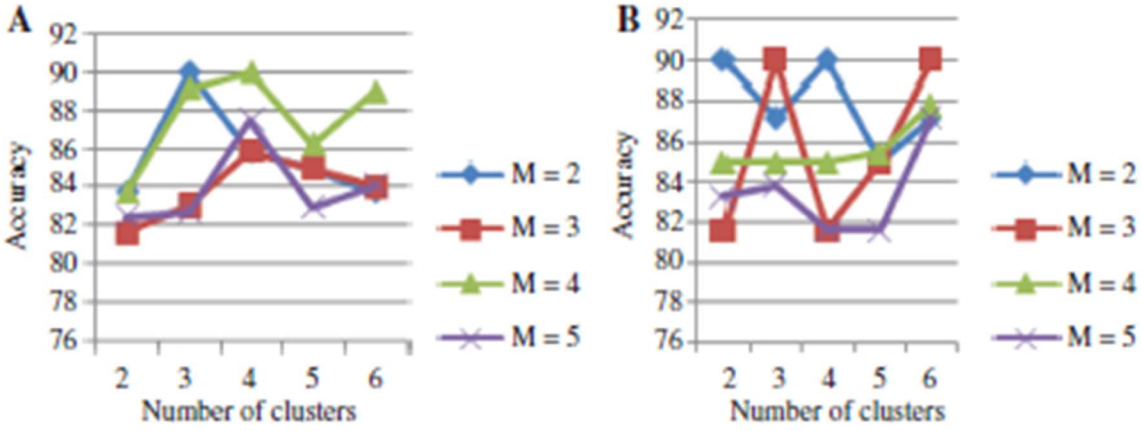

Figure 3: Accuracy Analysis Using (A) Dataset 1 and (B) Dataset 2.

DB-Index Analysis:

Figure 4 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 4A. The lower the DB-index, the greater is theclustering performance. Here, the minimum value computed is 6.24 for $M = 4$ when the number of clustersis 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 4B, 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 EFCM-FFO technique is 5.85. When the number ofclusters 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.

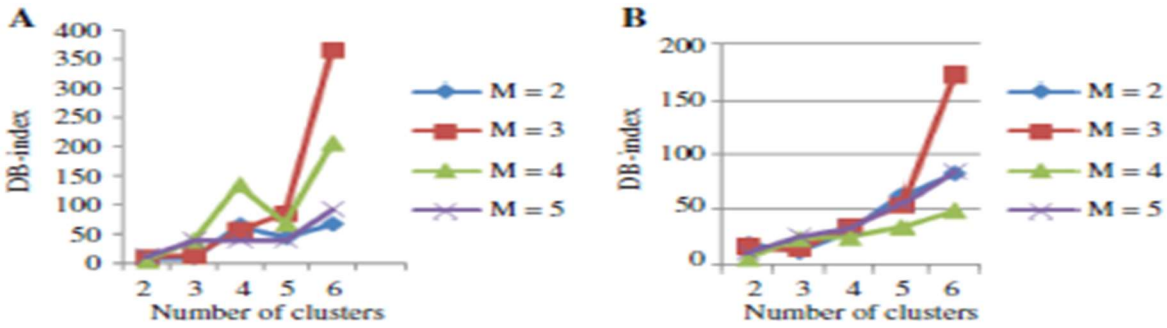

Figure 4: DB-Index Analysis Using (A) Dataset 1 and (B) Dataset 2.

Comparative Analysis:

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

Using Dataset 1:

The comparative analysis made in the proposed technique and the four existing techniques using dataset 1is depicted in Table 1. Table 1 presents the result of analysis based on accuracy using the first datasetby varying the number of clusters. When the number of clusters is 1, the accuracy obtained using the existingKM-MBFO and FCM-PSO is the same, 75.58%, while that in EFCM-FFO is 87.91%. Asthe number of clusters is increased to 5, the accuracy attained by the proposed technique is 95%, whereas82.43% is the maximum accuracy produced by the existing FCM-PSO.

Table 1: Comparative Analysis of Accuracy using Dataset 1

No. of Clusters	KM-MBFO	FCM-PSO	EFCM-FFO
1	75.58	75.58	87.91
2	76.21	77.32	88.65
3	78.36	78.87	91.72
4	79.45	79.65	93.35
5	80.62	82.43	95

The DB-index values analyzed usingthe comparative techniques with dataset 1 are shown in Table 2. A minimum value is observed in EFCM-FFO for all the cluster sizes considered. When 10.83 is the minimum DB-index provided by EFCM-FFOfor the cluster size of 3, KM-MBFO and FCM-PSO have a DB-index of 172.68 and 185.26, respectively. Thus, from the analysis using dataset 1, i.e. localization data, the proposedEFCM-FFO is observed to have the maximum performance than the other techniques.

Table 2: Comparative Analysis of DB-Index using Dataset 1

No. of Clusters	KM-MBFO	FCM-PSO	EFCM-FFO
1	68.42	56.35	8.56
2	74.91	64.58	9.45
3	85.26	72.68	10.38
4	92.35	78.36	12.65
5	98.45	84.65	15.9

Using Dataset 2

In Table 3, the comparative analysis result obtained in the three considered techniques using dataset 2 issketched out. In the accuracy analysis graph shown in Table 3, the maximum accuracy produced by theproposed EFCM-FFOis 90%, for the number of clusters of 1. In the same instant, the accuracy obtainedusing the existing KM-MBFO is78.24%. Meanwhile, FCM-PSO has a clusteringaccuracy of 79.57%.

Table 3: Comparative Analysis of Accuracy using Dataset 2

No. of Clusters	KM-MBFO	FCM-PSO	EFCM-FFO
1	78.24	79.57	90
2	80.12	81.47	92.64
3	83.65	84.54	94.87
4	85.72	86.92	95.63
5	86.95	87.63	97.84

The analysis based on DB-index using dataset 2 is depicted in Table 4, where theminimum value is achieved by the proposed technique, with a DB-index of 7.73, for onecluster. For the samecase, the minimum DB-index produced among the existing techniques is 12.01 by KM-MBFO. Hence, from theresults of the analysis, the proposed EFCM-FFO seems to have better performance than the other techniquesconsidered for the comparison.

Table 4: Comparative Analysis of Accuracy using Dataset 2

No. of Clusters	KM-MBFO	FCM-PSO	EFCM-FFO
1	12.01	10.45	7.73
2	14.23	12.56	8.26
3	15.87	13.93	9.65
4	17.65	15.12	10.87
5	18.91	17.64	12.72

The proposed hybrid approach could be a useful primitive for handling Big data sets. Two phases are involved in the EFCM-FFO: Hadoop and MapReduce. Hadoop and MapReduce store the Big data sets and are the primary storage systems used by the Hadoop application. Hadoop and MapReduce is designed to enable high throughput, which enables parallel computation of Input Split blocks. MapReduce phase processes these blocks or splits them into maps and reduces tasks. The efficient outcome achieved by our proposed approach is depicted below.

Table 5 shows the execution times of the proposed algorithms for document sets of the dataset. As the number of documents to be clustered increases, the time dedicated exceeds for the proposed

EFCM-FFO solution. It is noted that the execution time of 600 files is about 1.7 min. Parallel processing is essential in processing large volumes of data. In this experiment, we only considered some documents of a subset of the data set.

Table 5: Execution Time

No. of Files	Execution time
200	0.1
400	0.9
600	1.7
800	2.9
1000	4.5

In table 6 and 7, the comparative analysis result obtained in the three considered techniques using dataset 1 and dataset 2 issketched out. Table 6 and 7 compares the performance of the proposed model to that of the FCM-PSO and KM-MBFO when applied to the attack data set. Enhanced Fuzzy C-means worked well with the data sets studied over a reasonable amount of time due to inherent levels of data parallelism.

Table 6: Comparative results based on Execution Time using dataset 1

No. of Files	KM-MBFO	FCM-PSO	EFCM-FFO
200	12	10	3
400	18	15	5
600	26	18	7
800	42	22	10
1000	83	25	12

Table 7: Comparative results based on Execution Time using dataset 2

No. of Files	KM-MBFO	FCM-PSO	EFCM-FFO
200	10	8	2
400	12	14	4
600	15	19	6
800	23	27	9
1000	36	39	11

EFCM-FFO performed better in terms of execution time, as clusters formed in less time than when using the other existing clustering techniques. The average lengths of time used for each cluster indicate that with parallelization, the proposed clustering algorithm was more efficient than the single node standard K-means clustering method.

Conclusion:

Clustering high-dimensional data is one of the strongest problems since it deals with several numbers of attributes ina given data set. This paper proposes the Big data set in Hadoop environment which can be stated as an example of high dimensional data. Enhanced Fuzzy C-Means Clustering with Fruit Fly optimization is considered with the goal of producing quality clusters. This paper also deals the C value optimization using a Fruit Fly Optimization Algorithm that is used in enhancing the Hadoop performance and optimizes the execution time to meet the requirement of handling data sets. The experiment is performed using two datasets, localization andskin segmentation, and the results are compared with that of the existing techniques, such as KM-MBFO and FCM-PSO. The performance of the proposed technique is evaluated using twometrics, clustering accuracy and DB-index. EFCM-FFO could attain the maximum accuracy of 87.91% and90%for the localization and skin segmentation datasets,whereas that in the existing FCM-PSO is 82.43%and84.65%, respectively. Therefore, it can be concluded that the proposed EFCM-FFO technique can performbig data clustering effectively with maximum clustering accuracy compared with the existing comparativetechniques.

References:

[1]. Elham Azhir, Mehdi Hosseinzadeh, Faheem Khan, and Amir Mosavi: Performance Evaluation of Query Plan Recommendation with Apache Hadoop and Apache Spark. *10*(19), 3517(2022)
[2]. Deepak Kumar, Vijay Kumar Jha: An improved query optimization process in big data using ACO-GA algorithm and HDFS map reduce technique. Springer Science + Business Media, LLC, part of Springer Nature (2020)
[3] Chandra Shekhar Gautam1, Pandey* (2019)" A REVIEW ON GENETIC ALGORITHM MODELS FOR HADOOP MAPREDUCE IN BIG DATA" IJSRS ISSN:0976-3031, Vol.13, Issue-03(E), Pageno771-775, June 2022.
[4] Pasquale Salza a,∗ , Filomena Ferrucci. Speed up genetic algorithms in the cloud using software containers. (2019)
[5] Mahajan, D., Blakeney, C., Zong, Z.: Improving the energy efficiency of relational and NoSQL databases via query optimizations. Sustainable Computing: Informatics and Systems 22, 120–133 (2019).
[6] Song, J., Ma, Z., Thomas, R., Ge, Yu.: Energy efficiency optimization in big data processing platform by improving resources utilization. Sustainable Computing: Informatics and Systems 21, 80–89 (2019)

[7] Panahi, V.; Navimipour, N.J. Join query optimization in the distributed database system using an artificial bee colony algorithm and genetic operators. *Concurr.Comput. Pract. Exp.* 2019, *31*, e5218.
[8] Naik, A., & Satapathy, S. C. (2021). A comparative study of social group optimization with a few recent optimization algorithms. Complex & Intelligent Systems, 7, 249-295.
[9] Venkateswarlu, Y., Baskar, K., Wongchai, A., Gauri Shankar, V., Paolo Martel Carranza, C., Gonzáles, J. L. A., & Murali Dharan, A. R. (2022). An Efficient Outlier Detection with Deep Learning-Based Financial Crisis Prediction Model in Big Data Environment. Computational Intelligence and Neuroscience, 2022.
[10] Jayasri, N. P., & Aruna, R. (2022). Big data analytics in health care by data mining and classification techniques. ICT Express, 8(2), 250-257.
[11] Mehta, B. B., & Rao, U. P. (2022). Improved l-diversity: Scalable anonymization approach for privacy preserving big data publishing. Journal of King Saud University-Computer and Information Sciences, 34(4), 1423-1430.
[12] Jain, D. K., Boyapati, P., Venkatesh, J., & Prakash, M. (2022). An intelligent cognitive-inspired computing with big data analytics framework for sentiment analysis and classification. Information Processing & Management, 59(1), 102758.
[13] Ali, S. A. G., Al-Fayyadh, H. R. D., Mohammed, S. H., & Ahmed, S. R. (2022, June). A Descriptive Statistical Analysis of Overweight and Obesity Using Big Data. In 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1-6). IEEE.
[14] Chidambaram, S., &Gowthul Alam, M. M. (2022). An integration of archerfish hunter spotted hyena optimization and improved ELM classifier for multicollinear big data classification tasks. Neural Processing Letters, 54(3), 2049-2077.
[15] Ramakrishnan, U., &Nachimuthu, N. (2022). An Enhanced Memetic Algorithm for Feature Selection in Big Data Analytics with MapReduce. Intelligent Automation & Soft Computing, 31(3).
[16] A. Banharnsakun, A MapReduce-based artificial bee colony for large-scale data clustering, *Pattern Recognit. Lett.* **93**(2017), 78–84.
[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