International Journal of Cognitive Informatics and Natural Intelligence
Volume 15 • Issue 4 • October-December 2021

MapReduce-Based Crow Search-Adopted Partitional Clustering Algorithms for Handling Large-Scale Data

Karthikeyani Visalakshi N., Government Arts and Science College, India
Shanthi S., Kongu Engineering College, India
Lakshmi K., Kongu Engineering College, India

ABSTRACT

Cluster analysis is the prominent data mining technique in knowledge discovery, and it discovers the hidden patterns from the data. The k-means, k-modes, and k-prototypes are partition-based clustering algorithms, and these algorithms select the initial centroids randomly. Because of its random selection of initial centroids, these algorithms provide the local optima in solutions. To solve these issues, the strategy of crow search algorithm is employed with these algorithms to obtain the global optimum solution. With the advances in information technology, the size of data increased in a drastic manner from terabytes to petabytes. To make proposed algorithms suitable to handle these voluminous data, the phenomena of parallel implementation of these clustering algorithms is used with Hadoop MapReduce framework. The proposed algorithms are experimented with large-scale data, and the results are compared in terms of cluster evaluation measures and computation time with the number of nodes.

KEYWORDS

Cluster Analysis, Crow Search Optimization, Hadoop MapReduce, K-Means, K-Modes, K-Prototypes, Large-Scale Data, Parallel Computing

1. INTRODUCTION

Clustering is the unsupervised classification technique that extracts useful knowledge from the data without knowing their class labels. The main objective of clustering is that the data objects within a group are similar to one another and dissimilar from the data objects between the clusters. Clustering can be applied in different application domains such as image processing (Lei, Wang, Peng, & Yang, 2011), bioinformatics (Bhattacharya & De, 2010), document clustering (Jun, Park, & Jang, 2014), information retrieval (Chan, 2008) and healthcare (Güneş, Polat, & Yosunkaya, 2010).

Clustering algorithms are broadly divided into two categories: partitional and hierarchical. The partitional clustering algorithms group the data objects into a predefined number of clusters and the hierarchical clustering algorithms group the data objects on the basis of tree like structure using either the bottom-up or top-down approach. The K-Means, K-Modes and K-Prototypes are partition based clustering algorithms and these algorithms handle the numeric, categorical and mixing of numeric and categorical data objects respectively. K-Means is one of the most widely used partitional clustering algorithms to handle numerical data. This algorithm is extended to handle the categorical, mixed numeric and categorical types of data. These algorithms are called as K-Modes and K-Prototypes (Huang, 1998, 1997). While these algorithms are very fast and simple, they have some drawbacks.

DOI: 10.4018/IJCINI.20211001.oa32

Firstly, the performance heavily depends on the selection of initial centroids and secondly, objective function values contain the local minima. To defeat these issues, various optimization algorithms are proposed and some of those algorithms are literature surveyed. Some of these optimization algorithms include the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Firefly Algorithm (FA) and Cuckoo Search (CS).

Crow Search Algorithm (Askarzadeh, 2016) is a population based meta-heuristic optimization algorithm that stimulates the intelligent behaviour of the crows. Crows protect their leftover food in hidden places and rescue it whenever needed. This algorithm is based on finding the hidden storage position of excess food of crows. Finding a hidden food source of another crow is not an easy task because if a crow finds any one following, it fools the other crow by moving to the random position.

Recent advances in the technologies, the size of data increases everyday and finding useful information from that is a tedious task. To deal with this, recent technologies are incorporated with traditional algorithms to improve performance. Apache Hadoop is an open source framework and it is used for processing large scale data. This framework processes the data objects in a parallel manner using the MapReduce.

MapReduce (Dean & Ghemawat, 2008) is a programming model for processing large scale data. Hadoop stores the data in Hadoop Distributed File System (HDFS) and it is designed to handle the very large data files running on clusters of commodity hardware. MapReduce is a programming model for data processing and these programs are inherently parallel, thus putting very large scale data for analysis into the hands of anyone with enough machines at their disposal. MapReduce works by breaking the processing into two phases: the map phase and the reduce phase. Each phase has key-value pairs as input and output, the types of which may be chosen by the programmer. The programme needs to specify two functions the map function and the reduce function.

The research gap from the existing works are based on either the K-Means, K-Modes and K-Prototypes or the optimization algorithmsare implemented in a Hadoop MapReduce framework or in a Spark for handling very large scale data. Similarly, the global optimization algorithms try to resolve the local optimum insolutions, but they suffer from lowquality results andlow convergence speed, complicated operators, complexstructure and parameter setting issues.

Our motivation of this study is in threefold. First, a three different hybrid clustering algorithms are proposed for choosing the optimal initial centroids and also minimizing the objective function values. Second, the parallel implementation of clustering algorithms is proposed to handle very large scale data. Third, the combination of optimization algorithms and clustering algorithms with MapReduce framework to handle very large scale data efficiently.

The main contribution of this study is to combine the partitional clustering algorithms with nature inspired global optimization algorithm to handle very large scale data more efficiently. In this paper, an improved version of K-Means, K-Modes and K-Prototypes clustering algorithms are proposed. Initially, the CSA is used to obtain the best initial centroids for K-Means, K-Modes and K-Prototype clustering algorithms respectively and also to implement the parallel version of these algorithms using Apache Hadoop MapReduce framework.

This paper is organized as follows: Related works are discussed in Section 2, the partitional clustering algorithms such as K-Means, K-Modes and K-Prototypes clustering algorithms are discussed in Section 3, Crow Search optimization algorithm is discussed in Section 4, Apache Hadoop MapReduce framework is discussed in Section 5, proposed clustering algorithms are presented in Section 6, the experimental results are discussed in Section 7 and the conclude the paper in Section 8.

2. BACKGROUND

Data clustering using CSA and K-Means clustering algorithm is proposed in (Lakshmi, Karthikeyani Visalakshi, & Shanthi, 2018). In this, CSA optimization is integrated with the K-Means algorithm to obtain the optimum initial centroids and similarly obtain the global optimum in clustering solutions.

The experimental results are compared with some existing optimization algorithms based on K-Means clustering algorithm. The proposed clustering algorithm outperforms than existing techniques. The proposed clustering algorithm achieves the minimum fitness values with the rich quality of clusters in solutions. Finally, the authors suggested that each optimization algorithm has its own parameters and it is tedious to fix optimum values for these parameters. Similarly, this algorithm can be extended to automatically determine the optimal number of clusters for datasets.

A new automatic clustering algorithm called Automatic Clustering Based on Data Envelopment Analysis (ACDEA) is proposed in (Balavand, Kashan, & Saghaei, 2018). The prominent contribution of this algorithm is to use anaggregation ofCluster Validity Indices (CVI) and Data Envelopment Analysis (DEA). In this, CSA is utilized to attain the initial centroids for the K-Means clustering algorithm and this algorithm is named as CSAK-Means. The CSAK-Means is executed by varying the number of clusters and the corresponding CVI are computed. These outputs form the Decision Making Units (DMU) and its efficiency is calculated using the DEA method. In DEA, the selection of input and output variables is one of the most important steps. In this study, CVI is considered as input and output variables and the number of clusters as DMUs.

CSA based K-Prototype clustering algorithm is proposed in (Lakshmi, Karthikeyani Visalakshi, Shanthi, & Parvathavarthini, 2018). In this work, CSA is integrated with the K-Prototype clustering algorithm to acquire the best initial centroids for the K-Prototype clustering algorithm and it provides global optimum solutions. The experimental results are compared with some existing optimization algorithms based on K-Prototypes clustering algorithm. The proposed clustering algorithm outperforms than existing techniques.

Parvathavarthini, Karthikeyani Visalakshi, Shanthi, & Mohan (2018) proposed the Hybridized FCM and CSA clustering algorithms namely FCM-Crow Search Algorithm to obtain the global optima in clustering solutions. The proposed algorithm results the minimum fitness values in comparison to existing approaches. The authors concluded that the proposed algorithm can be extended to alter the parameter values. Also, extend this method to hybridize the CSA and Intuitionistic fuzzy c-means clustering algorithm.

CSA based Intuitionistic Fuzzy C-Means (IFCM) clustering algorithm is introduced in (Parvathavarthini, Karthikeyani Visalakshi, Shanthi, & Lakshmi, 2018). In this, the CSA is utilized to to obtain the optimum initial centroids for the IFCM clustering algorithm. The performance of the proposed clustering algorithm has experimented with benchmark and artificial datasets. It provides optimal results in terms of error rate and objective function values.

Parvathavarthini, Karthikeyani Visalakshi, & Shanthi (2019) proposed CSA based Intuitionistic fuzzy clustering approach with neighborhood attraction (CrSA-IFCM-NA) to identify the breast cancer. Instead of selecting the random initial centroids, CSA is applied to choose the best initial centroids. Futthur, this algorithm is applied for detecting the breast cancer. The proposed algorithm achieves the better results in comparison to the state-of-art techniques.

Wu, Huang, & Girsang (2018) introduced the hybrid algorithm based on Whole optimization algorithm (WOA) and Crow search algorithm (CSA) called HWCA. The proposed algorithm results the high quality of clusters in comparison to WOA and CSA. The authors suggested to develop an application of the HWCA algorithm in medical image clustering and to decrease the computation time by using some pattern reduction approaches.

The CSA is modified to resolve the graph coloring problem and it is proposed in (Meraihi, Mahseur, & Acheli, 2020). The binary crow search algorithm is derived from the CSA and the chaotic maps are used to choose the right values for the parameters of CSA. Finally, the Gaussan distribution is adopted to replace the random variables used in updating the position of crows.

CSA is hybridized with Elman Neural Network (ENN) and it is proposed in (Ullah et al, 2020). The proposed model has experimented with leukemiaDNA sequence classification. The performance is experimented with accuracy and mean square error measures.

Ghany, AbdelAziz, Soliman, & Sewisy (2020) proposed the hydridized WOA with TS algorithm called WOATS and it is utilized for data clustering. This technique prevented the WOATS from getting trapped in local optima. Likewise, proved its superiority over multiple original and hybrid swarm based intelligence methods. The authors suggested in future to grouping the massive biological datasets in the new technology called Big data.

Nguyen and Kuo (2019) developed a novel automatic fuzzy clustering usingnon-dominated sorting particle swarm optimization (AFC-NSPSO) algorithm for categorical data. The proposed AFC-NSPSO algorithm can automatically identify the optimal number of clusters and exploit the clustering result with the corresponding selected number of clusters.

Ji, Pang, Zheng, Wang, & Ma (2015)proposed the Artificial Bee Colony (ABC) based K-Modes clustering algorithm namely, ABC-K-Modes algorithm. To avoid K-Modes partitional clustering algorithm prone to local optima, the authors proposed the hybrid Artificial Bee Colony with K-Modes clustering algorithm is proposed. The experimental results revealed that the proposed algorithm is superior to the existing algorithms according to the cluster evaluation measures. The author extend this approach to group the mixed data containing both numeric and categorical attributes. Also, investigate the potential of ABC-K-Modes to social media data. Moreover, explore the other swarm intelligence algorithms for clustering categorical data as well as mixed.

Ji, Pang, Li, He, Feng, & Zhao (2020) proposed the Novel partitional Clustering algorithm based on Cuckoo Search and K-Prototypes (CCS-K-Prototypes) for clustering mixed numeric and categorical data. It finds the global solutions for the different types of attributes.They also suggested the multi-objective optimisation approaches to clustering mixed data and multi-view clustering and deep clustering for mixed data.

Nithya and Arun Prabha (2019) proposed the Lion Optimization based K-Prototypes clustering algorithm. The authors utilize the Lion optimization for finding the best initial centroids for K-Prototypes clustering algorithm. It reaches the global optimum in solutions and also yields the improved results than the traditional K-Prototypes algorithm.

Lu (2019) proposed the Parallel K-means clustering algorithm that is implemented in Hadoop platform. The proposed clustering algorithm reduces the time complexity and improves the accuracy. The selection of data sampling and initial centroids that affecting the final clustering effect. The author suggested the quick selection of sample data and initial clustering centroids needed in further research. This experiment only uses four nodes so that lacking in testing of large-scale data clustering on larger data sets.

A parallel implementation of the k-means clustering algorithm based on MapReduce is introduced in (Zhao, Ma, & He, 2009). In this algorithm, it first computes the initial cluster center in the map phase and then updates the global center in the reduce phase. The efficiency of this method is measured using speedup, scaleup and sizeup measures. The experimental results show that this method is able to cluster large data sets using commodity hardware.

Tao,Xiangwu, & Yefeng (2015) proposed parallel K-Modes clustering algorithm based on MapReduce framework to handle the large scale categorical data. The proposed algorithm has been experimented with US Census data and the speedup measure is utilized to compare its performance. The experimental results show that the parallel k-modes achieves the better speedup ratio to process the large scale categorical data.

A parallel implementation of MapReduce based K-Prototypes clustering algorithm is proposed in (HajKacem, N'cir, & Essoussi, 2015). The K-Prototypes clustering algorithm is used for handling large scale mixed numeric and categorical types of data. This parallelized K-Prototypes algorithm is applied to clustering the big data. The performance of the proposed method is compared in terms of speedup and scaleup measures. This algorithm requires the computation of distances between each of the cluster centers and the data points. But, these distance computations are redundant, because data objects usually stay in the same cluster after some iterations. Similarly, each iteration of K-Prototypes, the whole data set must be read and written to disks and it results inthe high input/

output (I/O) operations. To handle these issues, one-pass accelerated MapReduce-based K-Prototypes clustering algorithm for mixed large scale data (HajKacem, N'cir, & Essoussi, 2019). The proposed algorithm reads and writes the data only once which reduces largely the I/O operations. Similarly, the proposed algorithm is based on a pruning strategy to accelerate the clustering process by reducing the redundant distance computations between cluster centers and data points.

A parallel implementation of the Fuzzy C-Means (FCM) clustering algorithm using the MapReduce model is presented in (Ludwig, 2015). This algorithm consists of two MapReduce jobs, one MapReduce calculates the centroid matrix by iterating over the data records, and the second MapReduce iterates over the data records to update the membership matrix. The accuracy measure is performed to validate the proposed method. Also, speedup is calculated to evaluate the efficiency of the proposed method.

Parallel implementation of MapReduce based DBCURE algorithm called DBCURE-MR is proposed in (Y. Kim, Shim, M. Kim, Lee, 2014). In this, the new density-based clustering algorithm to find clusters with varying densities and parallelizing the algorithm with MapReduce is proposed. The traditional density-based algorithms find each cluster one by one but DBCURE-MR finds several clusters together in parallel. The k-Nearest Neighbour method based on the MapReduce paradigm has been proposed by (Anchalia & Roy, 2014) in order to process a high volume of data in a distributed computing environment.

A parallel implementation of the Tabu Search based k-means clustering algorithm is presented in (Lu, Cao, Rego, & Glover, 2018). Applying the metaheuristic algorithm Tabu Search is implemented in the Spark environment and also obtains the best centroids for the k-means clustering algorithm.

The scaling Genetic Algorithm using MapReduce is presented in (Verma, Llorà, Goldberg, & Campbell, 2009). In this, the transformation of GA into MapReduce primitives to demonstrate its scalability lies on large datasets. It also overcomes the MPI based parallel Genetic Algorithm because it requires knowledge about the machine architecture. Mapreduce based Genetic Algorithm provides the scalability and fault-tolerant applications.

Wang, Yuan, & Jiang (2012) proposed the Parallel K-PSO and this algorithm takes the advantage of PSO to overcomethe local search ability of K-means and makes the K-means in parallel with MapReduce to enhance the processing massive data. The results show that the Parallel K-PSO efficiently handle the large data in comparison to Parallel K-means and Serial K-means.

Parallel implementation of a Glowworm swarm optimization based clustering algorithm is proposed in (Al-Madi, Aljarah, & Ludwig, 2014). In this, the implementation of scalable glowworm swarm optimization clustering (MRCGSO) using MapReduce to handle the big data is used. It uses the Glowworm swarm optimization to formulate the clustering algorithm and also used to take advantage of its ability in solving multimodal problems, which in terms of clustering means find the multiple centroids. MRCGSO uses the MapReduce framework for the parallelization because it provides fault tolerance, load balancing and data locality. The experimental results reveal that the MRCGSO scales very well with increasing the size of data objects and reaches a very close to linear speedup while preserving the quality of clusters.

Banhansakun (2017) implemented the ABC algorithm in MapReduce framework and groups the large data instances with the objective of minimizing the sum of the squared Euclidean distance between each data object and the cluster centroids which it belongs. In this, two main operations are performed, one is updating the cluster centroids and the second one is evaluation of fitness. The cluster centroids are updated based on the ABC algorithm. The results show that the MR-ABC offers a good quality of clustering and it outperforms than other recent techniques. Also it is highly efficient from the perspective of both solution quality and algorithm performance. The author suggested to extend this experiments to be on much larger datasets (terabyte size) with more than hundreds computing nodes. The problem with parameter settings on the proposed algorithm will be also addressed. Furthermore, employ the MR-ABC method in other practical applications such as content based image retrieval.

Sinha & Jana (2018) proposed the two phase MapReduce based hybrid clustering algorithm for distributed datasets. The authors proposed the novel clustering algorithm for handling distributed datasets using Genetic Algorithm with Mahalanbi distance and k-means clustering algorithm. The proposed algorithm outperforms than MR-k-means and Parallel k-means. It works only for static datasets. In future, develop a real-time GA based clustering algorithm for dynamic dataset.

Wang & Tsai (2018) proposed Coral Reef Optimization with Substrate Layers (CRO-SL) in a Spark environment. The CRO-SL with Spark isnot faster than k-means, but it is faster than the GKA in solving large-scale or unbalanced datasets. The analysis results of the parameter settings also show that the end results are affected by the parameter settings of CRO. Therefore, how to find out a set of applicable parameters for CRO-based algorithms is still an open issue.

Tripathi, Sharma, & Bala (2018) introduced a novel variant of Grey Wolf Optimizer called Enhanced Grey Wolf Optimizer (EGWO). Further, the proposed EGWO is parallelized in an Apache Hadoop MapReduce framework named named MR-EGWO to process the large-scale datasets. The proposed approach outperforms than k-means and other optimization algorithms. Also, MR-EGWO provides a rich quality of clusters and efficiently handles the large volume of data sets. Recent parallelization tools like Spark may be tested to reduce the computation time of the proposed method. Moreover, the proposed method could be extended on some real-world clustering applications with large datasets like twitter analysis, video analysis, and satellite image analysis.

From the literature survey, it is observed that the CSA is hybridized with various clustering algorithms and are experimented with limited size of datasets. Likewise, the CSA is utilized for resolving the various engineering optimization problems. However, the hybrid CSA and clustering algorithms are not utilized for very large scale data in the distributed environment.

3. PARTITIONAL CLUSTERING ALGORITHMS

The partitional clustering algorithms group the given *N* data instances into predefined *K* number of partitions. The K-Means, K-Modes and K-Prototypes are such partition based clustering algorithms. The K-Modes algorithm is an extension of the K-Means algorithm by done the following modifications: (i) Euclidean distance is replaced with matching dissimilarity measure and (ii) frequency-based approach is utilized to update the centroids. The K-Means algorithm is extended to the K-Prototypes algorithm to group the mixing of numeric and categorical types of data objects. The main objective of these clustering algorithms is to minimize the sum of squared distance between the data instances and centroids values. The main objective function of K-Means, K-Modes and K-Prototypes clustering algorithms are defined in Equation (1).

$$F(U,Z)=\sum_{l=1}^{K}\sum_{i=1}^{N}U_{il}d(U_i,Z_l) \tag{1}$$

U_{il} is an $N \times K$ matrix where each element belongs to 0 or 1 (0£ U_{il} ^{3}1); N is the total number of data objects and *K* is the number of clusters. $d(U_i,Z_l)$ is the distance measure and it is computed between the data objects and centroids objects. The Euclidean distance is computed in K-Means, matching dissimilarity is computed in K-Modes.

The distance is computed for the K-Means clustering algorithm using Equation (2).

$$dis(X^i,Z^j)=\sqrt{\sum_{l=1}^{m^r}(X^{il}-Z^{jl})^2} \tag{2}$$

where m^r is the number of numeric features.

Likewise, computing the distance for K-Modes clustering algorithm by utilizing Equation (3).

$$dis(X^i, Z^j) = \sum_{l=1}^{m^c} \delta(X^{il}, Z^{jl}) \tag{3}$$

where m^c is the number of categorical features and $\delta(X^{il}, Z^{jl})$ is a dissimilarity measure. The $\delta(X^{il}, Z^{jl})$ is computed by using Equation (4).

$$\left(U^{il}, Z^{jl}\right) = \begin{cases} 0, U^{il} = Z^{il} \\ 1, U^{il} \neq Z^{jl} \end{cases} \tag{4}$$

In the same way, the computation of distance for K-Prototypes clustering algorithm by using Equation (5).

$$dis(X^i, Z^j) = \sum_{l=1}^{m^r} (X_{il}^r - Z_{jl}^r)^2 + wt \sum_{l=1}^{m^c} \delta(X_{il}^c, Z_{jl}^c) \tag{5}$$

where X_{il}^r is the numeric data instances, Z_{jl}^r is the numeric centroid values, X_{il}^c is the categorical data instances, Z_{jl}^c is the categorical centroid values and *wt* is the weight for categorical data.

These algorithms are terminated when one of the following conditions are satisfied: (i) The average change in the centroids (ii) The maximum number iterations is reached (iii) No change in the clustership of objects.

While these partitional clustering algorithms are simple and easy to implement, they possess some main issues that include (i) need the number of clusters in advance and (ii) K-Means that handles the numeric data, K-Modes that handles the categorical data and K-Prototypes that handles the mixed numeric and categorical data (iii) the K-Means, K-Modes and K-Prototypes algorithms produce the local optimum in objective function values.

The pseudocode for K-Means, K-Modes and K-Prototypes clustering algorithmsare described in Algorithm 1:

```
Algorithm 1. Pseudocode for partitional Clustering Algorithms
Input: Dataset X, Number of partitions K
Output: K partitions of data
Step 1: Generate the K initial centroids randomly.
Step 2: For each data instance, find the distance with initial
centroids and assign it to the
closest centroids. The distance is computed for K-Means, K-Modes
and K-Prototypes
using (2),(3) and (5) respectively.
Step 3: For updating the centroids,
a. In K-Means, Compute the mean from newly formed clusters by
using the equation (6).
b.  In K-Modes, frequencies of attributesare utilized.
c. In K-Prototypes, mean computation is utilized for numeric and
frequency-based approach for categorical types of data.
Step 4: Repeat the steps 2 and 3 until the convergence criteria
are met.
```

4. CROW SEARCH ALGORITHM

Crows observe other birds where they hide their foods and steal them when those birds leave. Crows follow each other to obtain better food sources but finding food source is not an easy task because if the crow finds anyone following, it tries to fool that by moving to another random position. For example, when the crow *x* is moving to visit its hidden position, the crow *y* follows it. In this stage, two situations occur: in the first situation, the crow *x* does not know about the existence of crow *y* and it moves towards its hidden position. In the second situation, if the crow *x* is aware of crow *y* following, in order to protect its hidden food position, it will direct crow *y* to an unspecified place. Based on this, the new positions are searched in successive iterations and are memorized to obtain the optimum solutions.

In this, crows are searchers, the environment is the search space, and each position is a feasible solution. Food hiding places search space, quality of the food source is fitness function and best food source is the global solution. The main principles of this algorithm are: (i) Crows live in the form flock, (ii) Crows memorize the position of their hiding places, (iii) Crows follow each other to do thievery (iv) Crows protect their caches from being pilfered by a probability.

The *n* is a number of crows with *d* dimensional environment and the position of crow *i* at iteration *iter* in the search space is specified as $x^{i,iter}$. Crows also have a memory to memorize the best history of positions of hiding places and it is specified by $m^{i,iter}$.

Each optimization algorithm has intensification and diversification parameters. Intensification specifies the explored regions thoroughly to find better solutions and diversification specifies the non-explored must be visiting in all regions of the search space. In CSA, intensification is specified in Awareness Probability (AP) and diversification is specified in Flight Length (FL). The pseudocode for the CSA optimization algorithm is shown in Algorithm 2:

Algorithm 2. Pseudocode for Crow Search Algorithm
Input: flock size *n*, number of iterations *max_iter*, flight length *FL* and
awareness probability *AP*
Output: Global best solutions
Step 1: Initialize the flock size n, number of iterations max_iter, Flight length (FL) and Awareness Probability (AP).
Step 2: Initialize the population randomly such that each crow specifies the feasible solution to the problem.
Step 3: Initialize the memory of the crows with the initial position of the crows because initially, crows have no experiences about better positions.
Step 4: Evaluate the initial position of the crows.
Step 5: While it < max_iter
a. for i =1:n
i. Generate a new position by randomly choose one of the crows and follows it to discover the new position.
ii. If the crow does not find anyone following, the crow will access the hiding place of that crow by using the Equation (6)

$$x^{it+1} = x^{i,it} + r_i \times fl^{i,it} \times (m^{i,it} - x^{i,it}) \tag{6}$$

iii. Else fool the following crow by choosing the random position.
b. End for
c. Check the feasibility of the new positions.

```
d. Evaluate the new position of the crows.
e. Update the memory of the crows using the Equation (7)
```

$$m_i^{it+1} = \begin{cases} x_i^{(it+1)} \ \ if\ f(x_i^{(it+1)})\ is\ better\ than\ f(m_i^{it}) \\ m_i^{it} \ \ otherwise \end{cases} \tag{7}$$

```
Step 6: End while
```

5. APACHE HADOOP MAPREDUCE FRAMEWORK

Apache Hadoop is a Java based open source framework and it allows the processing of massive amounts of data that are distributed across multiple computers using simple programming models. It is designed to scale up from the single servers to thousands of machines and each offers the local computation and storage. It works in an environment that provides distributed storage and computation across the clusters of computers.

The Apache Hadoop framework base is composed of the following modules: (i) Hadoop Common: contains libraries and utilities which are needed for the Hadoop modules, (ii) Hadoop Distributed File System (HDFS): this is a distributed file system that stores data on commodity machines, (iii) Hadoop YARN: it is a platform that is responsible for managing computing resources in clusters and used for scheduling users' applications and (iv) Hadoop MapReduce: it is a programming model for handling large-scale data processing.

MapReduce is the Java based open source framework for large scale data processing. It processes large data by distributing and running the data in clusters of commodity hardware. Hadoop possesses the Hadoop Distributed File System (HDFS) and it stores the very large data files. MapReduce process the data in the form of <key, value> pairs. MapReduce is divided into two phases: Map phase and Reduce phase.

The map/reduce framework will split the input into fixed-size chunks and each chunk is executed in the map function for each record in the chunk. The map script takes some input data and maps it to <key, value> pairs according to your specifications. The purpose of the map script is to model the data into <key, value> pairs for the reducer to aggregate. The map task results are writing output to a local disk of the respective node and not in the HDFS. Also, the output of Map is intermediate and it is processed by reduce tasks to produce the final output.The reduce script takes a collection of <key, value> pairs and "reduces" them according to the user. The reduce script to simply sum the values of the collection of <key, value> pairs which have the same key.

MapReduce is a programming abstraction that hides the underlying complexity of distributed data processing. The traditional systems would bring the data to the processing unit and process it. As the data grew huge, transferring these data to the processing unit involved the following issues: (i) moving huge data to process is costly and deteriorates the network performance, (ii) the processing takes more time as the data process in a single unit becomes the bottleneck and (iii) master node can get overburdened and may fail.

Nowadays, the MapReduce allows overcoming the above obstacles by bringing the processing unit to the data. MapReduce technology has the following advantages: (i) It is very cost effective because it moves the processing unit to the data, (ii) The processing time is also reduced as all the nodes are working in parallel and (iii) Every node gets a part of the data to process and there is no chance of a node getting overburdened.

6. MAPREDUCE BASED CROW SEARCH ADOPTED PARTITIONAL CLUSTERING ALGORITHMS

The proposed MapReduce based Crow Search optimization with K-Means, K-Modes and K-Prototypes clustering algorithms are implemented in two phases. In the first phase, the Crow search optimization algorithm is hybridized with the partitional clustering algorithms such as CSAK-Means, CSAK-Modes and CSAK-Prototypes to obtain the best centroids. The pseudocode for the first phase is shown in Algorithm 3.

Algorithm 3. Pseudocode for CSA adopted partitional clustering algorithms
Input: flock size *n*, number of iterations *max_iter*, Flight Length *FL* and
Awareness Probability *AP*, Dataset *X*, Number of partitions *K*
Output: K number of partitions
Step 1: Initialize the flock size n, number of iterations max_ iter, Flight Length (FL) and Awareness Probability (AP), Number of Clusters K.
Step 2: Initialize the population randomly such that each row specifies the cluster centroids.
Step 3: Initialize the memory of the crows with the initial population.
Step 4: Compute the fitness using equation (1) and evaluate the initial population.
Step 5: While it <max_iter
a. for i =1:n
i. Generate a new position by randomly choose one of the crows and follows it to discover the new position.
ii. If the crow does not find anyone following, the crow will access the hiding place of that crow by using the Equation (6).
iii. Else fool the following crow by choosing the random position.
b. End for
c. Check the feasibility of the new positions
d. Evaluate the new position of the crows
e. Update the memory of the crowsusing the Equation (7).
Step 6: End while
Step 7: obtain the best centroids

In the second phase, the data and best centroids files are uploaded in the Hadoop Distributed File System and run the MapReduce based partitional clustering algorithms with best centroids obtained in the first phase. The pseudocode for the second phase is shown in Algorithm 4.

Algorithm 4: Pseudocode for MapReduce based CSA adopted partitional clustering algorithms
Algorithm 4.1: Map Function
Step 1: In the setup function, search and read the centroids from the distributed cache.
Step 2: In the Map function,
a. Input the <key, values>, the key is the by default takes the line offset and value is the line content.
b. Read and construct the instances from values.
c. Initialize index = -1, min_distance = MAX_VALUE
d. for i=0 to centroids.length
i. In K-Means, Compute theEuclidean distance between the data and

```
the centroids.
ii. In K-Modes, frequencies of attributes are utilized to compute
the distance.
iii. In K-Prototypes, Euclidean distance is utilized for numeric
data and a frequency-based approach is utilized for categorical
data.
if dist<min_distance
{
min_distance = dist;
index = i;
}
e. end for
Step 3: Output the <key, values>, key is the cluster index and
values is the instances.
Algorithm 4.2: Reduce Function
Step 1: Input the <key, values> from the mapper function.
Step 2: Initialize, sum = 0, count =0
Step 3: While(values)
{
Collect each instance from values
Sum the different dimensions of instances
Count++
}
a. For updating the centroids
i. In K-Means, Compute the new centroids by dividing the sum of
instances by count.
ii. In K-Modes, frequencies of attributes are computed.
iii. In K-Prototypes, mean computation is utilized for numeric
data and a frequency based approach is utilized for categorical
data.
Step 4: end while
Step 5: Output the <key,values>, the key is the cluster number and
values is the new
centroids.
```

7. EXPERIMENTAL SETUP AND RESULTS

The experiments were performed on Hadoop cluster running with the hadoop version 2.7.4 and Java 1.8. It consists of three machines and each machine has Pentium Core i3 (2.93 GHZ) and 2 GB RAM. The operating system for each machine is Ubuntu 14.04. In the Crow Search algorithm, the values for Flight Lengthand Awareness Probability are 2 and 0.1 respectively. To evaluate the performance of the proposed Parallel CSAK-Means, Parallel CSAK-Modes and Parallel CSAK-Prototypes clustering algorithms, four datasets for each algorithm are used and these datasets are derived from the UCI Machine Learning Repository (Asuncion, & Newman, 2007). These datasets are scaled up to about 10^7 records for each baseline dataset (Banharnsakun 2017). The description of each dataset is shown in Table 1.

7.1 Cluster Evaluation Measures

Cluster evaluation is an important task because it evaluates the goodness of clustering solutions. It measures the quality of clusters quantitatively. The cluster evaluation can be performed by using two

major approaches, namely internal and external measures to assess the performance of the clustering algorithms. The internal measures are used to evaluate the results of the clustering algorithm without any prior knowledge of the data objects. The external measures are used to assess the results of the clustering algorithm by comparing the predicted class labels with pre-specified class label information of the data objects. The good quality results obtained from the cluster validation indices specify the effectiveness of the clustering algorithms.

To evaluate the performance of the proposed method, Silhouette, F-Measure, Rand Index and Purity measures are utilized. The Silhouette is an internal measure and F-Measure, Rand Index, Purity are external measures. In addition, the Analysis Of Variance (ANOVA) statistical test is performed to experiment with the significant differences among the clustering algorithms. Likewise, the execution time is computated to prove the efficiency of the proposed Parallel CSAK-Means, Parallel CSAK-Modes and Parallel CSAK-Prototypes clustering algorithms.

Table 1. Dataset Details

K-Means			
Dataset	**No. of instances**	**No. of attributes**	**No. of classes**
Iris	10,000,050	4	3
Wine	10,000,040	13	3
CMC	10,000,197	9	3
Vowel	10,000,822	3	6

7.1.1 Silhouette Index

The Silhouette index(Rousseeuw, 1987) is an internal measure that assesses how the data objects belong to their own clusters compared to other clusters. Its value ranges from -1 to $+1$, where a high value indicates that the object is well grouped to its own cluster. This measure combines both the cohesion and the separation. It is calculated using Equation (8):

$$Sil\left(X^i\right) = \frac{b\left(X^i\right) \text{ - } a\left(X^i\right)}{max\{a(\mathrm{X}^i), b(\mathrm{X}^i)\}} \tag{8}$$

where $Sil(X^i)$ is the Silhouette value of the data instance X^i, $a(X^i)$ is the average dissimilarity of X^i to all the other data instances within the same group and $b(X^i)$ is the average dissimilarity of X^i with all the other data instances in different groups.

7.1.2 Rand Index

The Rand Indexis an external measure for finding the similarity between actual labels and predicted labels (Rand, 1971). This measure has the value between 0 and 1; 0 indicates that the real and predicted data clusters do not match any pair of points and 1 indicates that the groups are precisely the same. The RI is calculated using Equation (9):

$$Rand\ \ Index = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

The term *TP* means True Positive and it is the count of similar data objects in the same cluster. The term *TN* means True Negative and it is a count of dissimilar data objects in different clusters. The term *FP* means False Positive and it is the count of dissimilar data objects in the same cluster. The term *FN* means False Negative and it is the count of similar data objects in different clusters.

7.1.3 F-Measure

F-Measure is an external measure to obtain the accuracy of the clustering results (Van Rijsbergen, 1979). It measures the quality of clusters by finding the harmonic mean of precision and recall. Its value ranges from 0 to 1, where a high value indicates that the object is grouped well to its own cluster. It is computed using Equation (10):

$$FMeasure = 2 * \frac{\mathrm{Pr}\,ecision * \mathrm{Re}\,call}{\mathrm{Pr}\,ecision + Recall} \tag{10}$$

Precision is obtained as the number of correct positive predictions divided by the total number of positive predictions. The best precision is 1, whereas the worst is 0. It is specified in Equation (11):

$$\mathrm{Pr}\,ecision = \frac{TP}{TP + FP} \tag{11}$$

The recall is calculated as the number of correct positive predictions divided by the total number of positives. The best sensitivity is 1 whereas the worst is 0. It is calculated using Equation (12):

$$\mathrm{Re}\,call = \frac{TP}{TP + FN} \tag{12}$$

7.1.4 Purity

The Purity measures the homogeneity of the data instances in generated clusters (Ghany, AbdelAziz, Soliman, &Sewisy, 2020). The bigger the purity value the better the quality of the clustering algorithm. The Purity measure is computed using Equation (13)

$$Purity = \frac{1}{n} \times \sum_{i=1}^{K} \max_i \left(\left| O_i \cap G_j \right| \right) \tag{13}$$

where n is the number of data objects in the dataset, G represents thegenerated clusters $G = \{g_1, g_2, \ldots, g_K\}$, and L represents the original centers $O = \{o_1, o_2, \ldots, o_K\}$.

7.1.5 One-Way Anova

ANOVA is a statistical measure that determines whether there is a significant difference between the performances of clustering algorithms.The significance level α is used in hypothesis tests to determine the accept or reject the hypothesis. The *p-value* is smaller than the α value (i) the null hypothesis will be rejected and (ii) conclude that all the values taken for analysis are not equal. It is strong evidence to reject the null hypothesis. The large *p-value* in comparison to α, (i) the null hypothesis will be accepted and (ii) conclude that all the values taken for analysis are equal. Therefore, the smaller the *p-value* implies the more significant results.

In this study, the silhouette values of clustering algorithms at different iterations are computed and the one-way ANOVA test is applied to those values to analyze the performance of the proposed clustering algorithms. The null hypothesis (H_0) means that the performances of all clustering algorithms are equal and the alternative hypothesis (H_1) means that the performance of all clustering algorithms is different.

7.2 Results and Discussion

Table 2 shows the experimental results of the proposed MapReduce based CSA adopted K-Means clustering algorithm. The silhouette values obtained from various iterations for the Parallel CSAK-Means algorithm show that the proposed clustering algorithm outperforms than Parallel K-Means and Parallel PSOK-Means for all data sets. It is observed that the results of the Silhouette, F-Measure, Rand Index and Purity reveal that the Parallel CSAK-Means are higher than the Parallel K-Means and Parallel PSOK-Means clustering algorithms. The Parallel CSAK-Means takes minimum computation time to obtain the rich quality of clustering solutions.

Table 2. Performance analysis ofproposed Parallel CSAK-Means algortihm

Dataset	Measure	Parallel K-Means	Parallel PSOK-Means	Parallel CSAK-Means
Iris	Silhouette	0.6565	0.6902	0.7346
	Rand Index	0.6756	0.7956	0.9600
	F-Measure	0.6704	0.7799	0.9414
	Purity	0.8800	0.9000	0.9400
	Comp.Time(secs)	17434	9348	3148
Wine	Silhouette	0.6945	0.7323	0.7646
	Rand Index	0.6667	0.6816	0.8315
	F-Measure	0.4837	0.5125	0.7603
	Purity	0.7078	0.7134	0.7415
	Comp.Time(secs)	7021	5448	2538
CMC	Silhouette	0.5792	0.6399	0.6426
	Rand Index	0.5547	0.5718	0.6004
	F-Measure	0.2926	0.3271	0.3952
	Purity	0.3971	0.4005	0.4297
	Comp.Time(secs)	11483	3269	2818
Vowel	Silhouette	0.5358	0.5532	0.5671
	Rand Index	0.6919	0.7030	0.7405
	F-Measure	0.5892	0.6240	0.6926
	Purity	0.4592	0.4626	0.4730
	Comp.Time(secs)	16059	3550	1773

To examine the implementation of the Parallel CSAK-Means algorithm, the obtained experimental results are compared with existing techniques such as ParallelK-Means (PKMeans), Parallel K-PSO, MapReduce based Artitficial Bee Colony (MR-ABC), and MapReduce based Enhanced Grey Wolf Optimizer (MR-EGWO), hybrid Whale Optimization and Tabu Search Algorithm (WOATS) algorithms. Table 3 shows the comparison of proposed MapReduce based CSA adopted K-Means clustering algorithm with the existing parallel implementation of clustering algorithms. The proposed parallel CSAK-Means outperforms than all other methods under comparison while the parallel K-means have given the least performance among all the considered methods. From this, it can be concluded that the proposed parallel CSAK-Means can be used for the clustering of large datasets efficiently.

Table 3. Comparative analysis of proposed Parallel CSAK-Means with existing algorithms

Author	Techniques	Dataset	Results
Zhao et al (2009)	PKMeans	Iris Wine CMC Vowel	FMeasure 0.667 0.298 0.482 0.586
Wang et al (2012)	Parallel K-PSO	Iris Wine CMC Vowel	FMeasure 0.785 0.324 0.517 0.627
Banhansakun (2017)	MR-ABC	Iris Wine CMC Vowel	FMeasure 0.842 0.387 0.718 0.643
Tripati et al (2018)	MR-EGWO	Iris Wine CMC Vowel	FMeasure 0.846 0.391 0.733 0.635
Ghany et al. (2019)	WOATS	Magic Electricity Poker CoverType	Purity 0.75 0.68 0.67 0.72
Proposed	**Parallel CSAK-Means**	Iris Wine CMC Vowel Magic Electricity Poker CoverType	FMeasure 0.9414 0.3952 0.7603 0.6926 Purity 0.82 0.70 0.72 0.75

The Silhouette, F-Measure, Purity, Rand Index and Computation time of Parallel CSAK-Modes clustering algorithm are shown in Table 4. The average silhouette values obtained from various iterations for the Parallel CSAK-Modes algorithm show that the proposed clustering algorithm outperforms than Parallel K-Modes and Parallel PSOK-Modes for all data sets. Likewise, the Parallel CSAK-Modes takes minimum computation time to obtain the rich quality of clustering solutions. It is observed from the above results, the overall performance of Parallel CSAK-Modes provides high quality clusters in minimum computational time.

Table 5 shows the comparison of proposed MapReduce based CSA adopted K-Modes clustering algorithm with the existing clustering algorithms. The existing clustering algorithms consider for comparative analysis includes Artificial Bee Colony based K-Modes (ABC-K-Modes) and *Automatic Fuzzy Clustering Using Non-Dominated Sorting Particle Swarm Optimization Algorithm for Categorical Data* (AFC-NSPSO) algorithms. The proposed parallel CSAK-Modes algorithm outperforms than all other methods under comparison while the Parallel CSAK-Modes have given

Table 4. Performance analysis of proposed Parallel CSAK-Modes algorithm

Dataset	Measure	Parallel K-Modes	Parallel PSOK-Modes	Parallel CSAK-Modes
Soybean	Silhouette	0.4282	0.4308	0.4752
	Rand Index	1.0000	1.0000	1.0000
	F-Measure	1.0000	1.0000	1.0000
	Purity	1.0000	1.0000	1.0000
	Comp.Time(secs)	8129	5023	2087
Mushroom	Silhouette	0.3640	0.3734	0.3827
	Rand Index	0.7707	0.8560	0.8929
	F-Measure	0.7931	0.8670	0.9004
	Purity	0.7656	0.8099	0.8577
	Comp.Time(secs)	20306	6474	5121
Congressional Voting	Silhouette	0.5149	0.5457	0.5609
	Rand Index	0.7793	0.8322	0.8759
	F-Measure	0.8019	0.8420	0.8832
	Purity	0.8459	0.8643	0.8643
	Comp.Time(secs)	9367	4547	2261
Car Evaluation	Silhouette	0.0675	0.0724	0.0821
	Rand Index	0.7092	0.7295	0.7610
	F-Measure	0.4890	0.5818	0.5909
	Purity	0.3115	0.3460	0.3712
	Comp.Time(secs)	8654	4078	2204

the highest performance among all the considered methods. From this, it can be concluded that the proposed parallel CSAK-Modes can be used for the clustering of large datasets efficiently.

The results of the Silhouette, Rand Index, F-Measure, Purity and Computation Time of proposed Parallel CSAK-Prototypes clustering algorithm are shown in Table 6. The Silhouette values obtained from various iterations for the Parallel CSAK-Prototypes algorithm outperforms Parallel K-Prototypes and Parallel PSOK-Prototypes for all data sets. It is observed that the results of Rand Index, F-Measure and Purity reveal that the Parallel CSAK-Prototypes are higher than the Parallel K-Prototypes and Parallel PSOK-Prototypes clustering algorithms. The overall performance of Parallel CSAK-Prototypes provides high quality of clusters. Likewise, the time taken for computation is smaller than Parallel K-Prototypes and Parallel PSOK-Prototypes clustering algorithms.

Table 7 shows the comparative analysis of proposed Parallel CSAK-Prototypes clustering algorithm with existing clustering algorithms. The results are compared with MapReduce based K-Prototypes (MR-KP), Lion Optimization based K-Prototypes algorithm and Artificial Bee Colony based K-Prototypes clustering algorithm (ABC-K-Modes). It is observed that the results reveal that the Parallel CSAK-Prototypes provides high quality of clustering solutions in comparison to existing clustering algorithms.

In order to show the statistically significant differences between these clustering algorithms, the statistical one-way ANOVA test is performed on Silhouette values. Table 7 shows the results of ANOVA with the significance level of 5%.For the 5% significance level, the p-value for all datasets is less than 0.05 and this implies that the alternate hypothesis is accepted. So that the proposed Parallel version of CSA adopted K-Means, K-Modes and K-Prototypes clustering algorithms have statistically significant differences.

The execution time of proposed clustering algorithms in three Hadoop nodes are presented in Figures 1 to 3. These algorithms break the input data into smaller chunks such that they fit into the main memory of the computing nodes. The proposed algorithms work efficiently for default block size. These algorithms are run on large scale data up to ten lakhs. In this, the size of dataset is kept constant and increases the number of nodes and moreover, the proposed clustering algorithms are scaled well with these large datasets and also have the optimized initial centroids.

Table 5. Comparative analysis of proposed Parallel CSAK-Modes with existing algorithms

Author	Techniques	Dataset	Measure	
Ji et al (2015)	ABC-K-Modes	Zoo	***Accuracy***	***Rand Index***
		Breast Cancer	0.9307	0.9766
		Soybean	0.9399	0.8869
		Lung Cancer	1.0000	1.0000
		Mushroom	0.6563	0.6530
		Dermatology	0.8964	0.8114
			0.8361	0.9073
Nguyen & Kuo(2019)	AFC-NSPSO	Breast Cancer	***Purity***	
		Soybean	0.981±0.022	
		Mushroom	0.987±0.034	
		Vote	0.898±0.045	
		Zoo	0.958±0.037	
		Tic-tac-toe	0.953±0.021	
		Lymphography	0.652±0.043	
		Splice	0.622±0.056	
			0.737±0.028	
Proposed	**Parallel CSAK-Modes**	Zoo	***Accuracy***	***Rand Index***
		Breast Cancer	0.9561	0.9861
		Soybean	0.9583	0.9053
		Lung Cancer	1.0000	1.0000
		Mushroom	0.6875	0.6943
		Dermatology	0.8577	0.8929
			0.8512	0.9365
		Breast Cancer	***Purity***	
		Soybean	0.9854	
		Mushroom	1.0000	
		Vote	0.8577	
		Zoo	0.8643	
		Tic-tac-toe	0.9561	
		Lymphography	0.6600	
		Splice	0.6312	
			0.7564	

8. CONCLUSION

An efficient parallel implementation of Crow Search based K-Means, K-Modes and K-Prototypes clustering algorithms are proposed in this work. These algorithms group the numeric, categorical and mixing of numeric and categorical data respectively. The Parallel CSAK-Means efficiently handles the very large scale numeric data and similarly, provides better clusters than its Parallel K-Means and Parallel PSOK-Means clustering algorithms. The Parallel CSAK-Modes algorithm is capable of handling the large scale categorical data and outperforms the Parallel K-Modes and Parallel PSOK-Modes clustering algorithms. The Parallel CSAK-Prototypes efficiently handle the large scale mixed numeric and categorical data and similarly, provides better clusters than the Parallel K-Prototypes and Parallel PSOK-Prototypes clustering algorithms. These proposed clustering algorithms are compared with existing clustering algorithms and the results are comparatively outperforms than the existing algorithms. These proposed clustering algorithms takes fewer computational time to achieve the high quality of clusters.

Our proposed clustering algorithms are partition based clustering algorithms that handle the very large scale numeric, categorical and mixing of these data by distributing the data between the clusters of commodity hardware. Similarly, these proposed algorithms choose the optimal initial centroids

Table 6. Performance analysis of proposed Parallel CSAK-Prototypes algorithm

Dataset	Measure	Parallel K-Prototypes	Parallel PSOK-Prototypes	Parallel CSAK-Prototypes
Heart	Silhouette	0.5214	0.5412	0.5804
	Rand Index	0.6111	0.6370	0.6519
	F-Measure	0.6123	0.6298	0.6426
	Purity	0.6296	0.6704	0.6925
	Comp.Time(secs)	5795	2859	2597
Hepatitis	Silhouette	0.5398	0.5456	0.5936
	Rand Index	0.6387	0.7290	0.7484
	F-Measure	0.6672	0.7311	0.7456
	Purity	0.6839	0.7097	0.7161
	Comp.Time(secs)	8490	6381	1408
German Credit	Silhouette	0.6106	0.6312	0.6543
	Rand Index	0.5990	0.6330	0.6960
	F-Measure	0.5502	0.6674	0.6921
	Purity	0.6513	0.6690	0.6710
	Comp.Time(secs)	5312	3122	1677
Credit Approval	Silhouette	0.6362	0.6514	0.6870
	Rand Index	0.6391	0.6623	0.6855
	F-Measure	0.6398	0.6677	0.6785
	Purity	0.6783	0.6043	0.6377
	Comp.Time(secs)	7110	1751	1531

Table 7. Comparative analysis of proposed Parallel CSAK-Prototypes with existing algorithms

Author	Techniques	Dataset	Measure		Proposed Parallel CSAK-Prototypes	
HajKacem et al (2015)	MR-KP	Chess	**Purity**		**Purity**	
		CoverType	0.53		0.55	
		KDD1M	0.59		0.63	
			0.66		0.69	
Nithya, G. S., & Prabha, K. A. (2019).	Lion Optimization based K-Prototype Clustering Algorithm	Australian Credit Approval	**Rand Index**		**Rand Index**	
		German Credit	0.62		0.6377	
		Hepatitis	0.57		0.6960	
		Post Operative Patient	0.74		0.7484	
		Stat log Heart	0.47		0.5630	
			0.72		0.7512	
Albalawi et al, (2019)	ABC k-prototypes	Dermatology	***Accuracy***	***Rand Index***	***Accuracy***	***Rand Index***
		Zoo	0.7288	0.8169	0.7327	0.8470
		Sponge	0.8311	0.8629	0.8611	0.8911
		Covertype	0.6289	0.8459	0.6570	0.8932
		Adult	0.4953	0.5889	0.5129	0.6025
			0.7591	0.5008	0.8100	0.6135
Jinchao Ji et al (2020)	CCS-K-Prototypes	Zoo	**Accuracy**	**Rand Index**	**Accuracy**	**Rand Index**
		Heart disease1	0.888	0.901	0.9000	0.9201
		Heart disease2	0.648	0.680	0.6925	0.6519
		Credit Approval	0.812	0.694	0.8705	0.7165
		Soybean	0.796	0.674	0.6377	0.6855
		Breast Cancer	0.989	0.988	1.0000	1.0000
			0.958	0.919	0.9755	0.9345

Table 8. Results of one-way ANOVA statistical test for the proposed clustering algorithms

Parallel CSAK-Means		Parallel CSAK-Modes		Parallel CSAK-Prototypes	
Dataset	**p-value**	**Dataset**	**p-value**	**Dataset**	**p-value**
Iris	0.0025	Soybean	0.0191	Heart	0.0411
Wine	0.0472	Mushroom	0.0303	Hepatitis	0.0490
CMC	0.0495	Congressional Voting	0.0424	German Credit	0.0420
Vowel	0.0178	Car Evaluation	0.0397	Credit Approval	0.0273

Figure 1. The execution time of Iris, Wine, CMC and Vowel datasets with 3 Hadoop cluster nodes

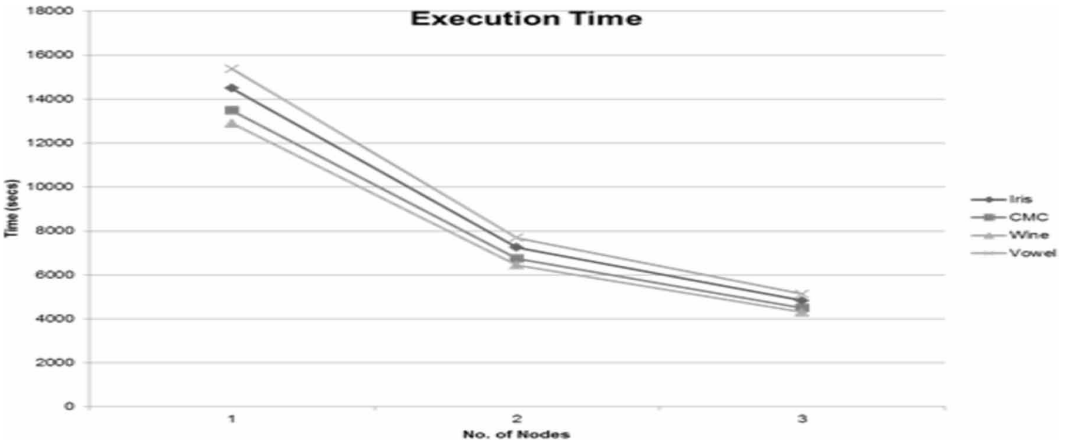


Figure 2. The execution time of Soybean, Mushroom, Congressional Voting and Car Evaluation datasets with 3 Hadoop cluster nodes

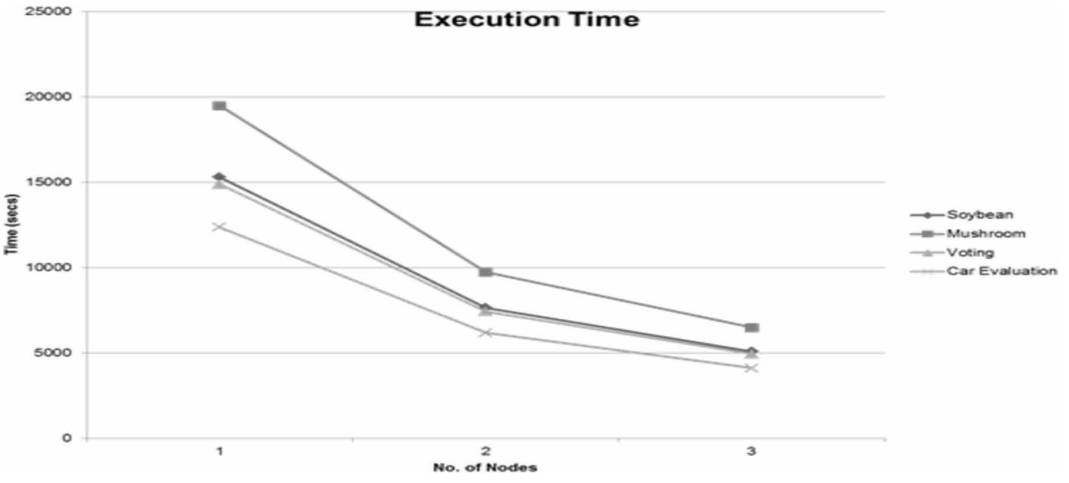
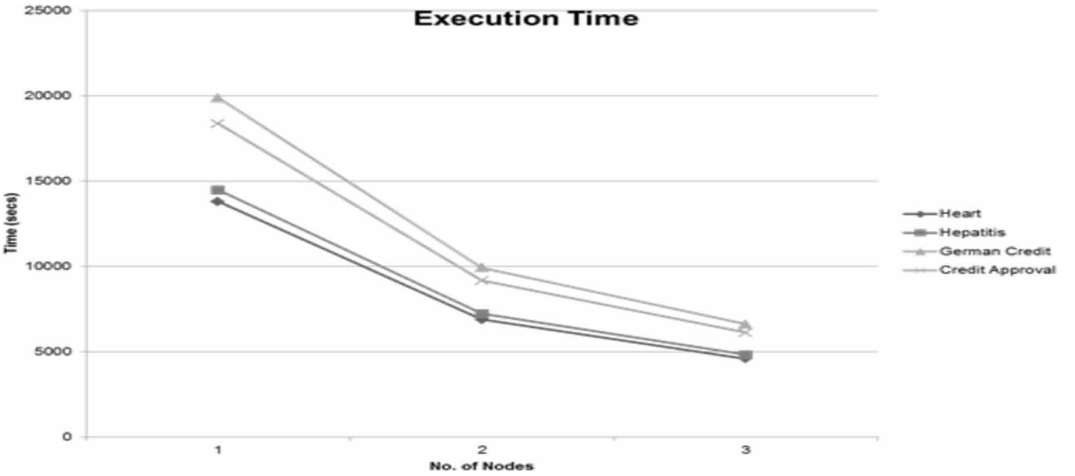


Figure 3. The execution time of Heart, Hepatitis, Credit Approval and German Credit datasets with 3 Hadoop cluster nodes



using Crow Search optimization that exempted from the local optima and also minimize the objective function values. These models can be utilized to create a real time applications.

Our proposed clustering algorithms are experimented only with nearly ten lakhs data objects. The MapReduce based Crow Search with the objective function of K-Means, K-Modes and K-Prototypes may handle the Bigdata more efficiently.

The proposed model is an initial stage for handling very large scale data in a Hadoop MapReduce framework. In future, this model can be implemented in a cloud environment with Bigdata. Similarly, this model is tested with synthetic data sets that are derived from the benchmark datasets and the number of nodes is limited to three and in future, it can be extended to more number of nodes and utilized for real time environment.

In future, the recent metaheuristic optimization algorithms may be substituted in the place of Crow Search Algorithm and its efficiency and effectiveness can be compared with this. Additionally, any of the techniques for finding the number of clusters in dataset can be incorporated with this model.

REFERENCES

Al-Madi, N., Aljarah, I., & Ludwig, S. A. (2014, December). Parallel glowworm swarm optimization clustering algorithm based on MapReduce. In *2014 IEEE Symposium on Swarm Intelligence* (pp. 1-8). IEEE. doi:10.1109/SIS.2014.7011794

Anchalia, P. P., & Roy, K. (2014, January). The k-nearest neighbor algorithm using the MapReduce paradigm. In *2014 5th International Conference on Intelligent Systems, Modelling and Simulation* (pp. 513-518). IEEE. doi:10.1109/ISMS.2014.94

Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures*, *169*, 1–12. doi:10.1016/j.compstruc.2016.03.001

Asuncion, A., & Newman, D. (2007). UCI machine learning repository. The University of California, School of Information and Computer Science.

Balavand, A., Kashan, A. H., &Saghaei, A. (2018). Automatic clustering based on Crow Search Algorithm-Kmeans (CSA-Kmeans) and Data Envelopment Analysis (DEA). *International Journal of Computational Intelligence Systems, 11*(1), 1322-1337.

Banharnsakun, A. (2017). A MapReduce-based artificial bee colony for large-scale data clustering. *Pattern Recognition Letters*, *93*, 78–84. doi:10.1016/j.patrec.2016.07.027

Ben HajKacem, M. A., N'cir, C.-E. B., & Essoussi, N. (2019). One-pass MapReduce-based clustering method for mixed large scale data. *Journal of Intelligent Information Systems*, *52*(3), 619–636. doi:10.1007/s10844-017-0472-5

Bhattacharya, A., & De, R. K. (2010). Average correlation clustering algorithm (ACCA) for grouping of co-regulated genes with a similar pattern of variation in their expression values. *Journal of Biomedical Informatics*, *43*(4), 560–568. doi:10.1016/j.jbi.2010.02.001 PMID:20144735

Chan, C. C. H. (2008). Intelligent spider for information retrieval to support mining-based price prediction for online auctioning. *Expert Systems with Applications*, *34*(1), 347–356. doi:10.1016/j.eswa.2006.09.031

Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107–113. doi:10.1145/1327452.1327492

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

Güneş, S., Polat, K., & Yosunkaya, Ş. (2010). Efficient sleep stage recognition system based on EEG signal using k-means clustering based feature weighting. *Expert Systems with Applications*, *37*(12), 7922–7928. doi:10.1016/j.eswa.2010.04.043

HajKacem, M. A. B., N'cir, C. E. B., & Essoussi, N. (2015). Parallel K-prototypes for Clustering Big Data. In Computational Collective Intelligence (pp. 628-637). Springer.

Huang, Z. (1997, February). Clustering large data sets with mixed numeric and categorical values. In *Proceedings of the 1st Pacific-Asia conference on knowledge discovery and data mining,(PAKDD)* (pp. 21-34). Academic Press.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, *2*(3), 283–304. doi:10.1023/A:1009769707641

Ji, J., Pang, W., Li, Z., He, F., Feng, G., & Zhao, X. (2020). Clustering Mixed Numeric and Categorical Data With Cuckoo Search. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 30988–31003. doi:10.1109/ACCESS.2020.2973216

Ji, J., Pang, W., Zheng, Y., Wang, Z., & Ma, Z. (2015). A novel artificial bee colony based clustering algorithm for categorical data. *PLoS One*, *10*(5), e0127125. doi:10.1371/journal.pone.0127125 PMID:25993469

Jun, S., Park, S. S., & Jang, D. S. (2014). Document clustering method using dimension reduction and support vector clustering to overcome sparseness. *Expert Systems with Applications*, *41*(7), 3204–3212. doi:10.1016/j.eswa.2013.11.018

Kim, Y., Shim, K., Kim, M. S., & Lee, J. S. (2014). DBCURE-MR: An efficient density-based clustering algorithm for large data using MapReduce. *Information Systems*, *42*, 15–35. doi:10.1016/j.is.2013.11.002

Lakshmi, K., Visalakshi, N. K., & Shanthi, S. (2018). Data clustering using K-Means based on Crow Search Algorithm. *Sadhana*, *43*(11), 190. doi:10.1007/s12046-018-0962-3

Lakshmi, K., Visalakshi, N. K., Shanthi, S., & Parvathavarthini, S. (2018). Clustering mixed datasets using k-prototype algorithm based on crow-search optimization. In *Developments and Trends in Intelligent Technologies and Smart Systems* (pp. 191–210). IGI Global.

Lei, L., Wang, T., Peng, J., & Yang, B. (2011). Image dimensionality reduction based on the intrinsic dimension and parallel genetic algorithm. *International Journal of Cognitive Informatics and Natural Intelligence*, *5*(2), 97–112. doi:10.4018/jcini.2011040106

Lu, W. (2019). Improved K-Means Clustering Algorithm for Big Data Mining under Hadoop Parallel Framework. *Journal of Grid Computing*, 1–12. doi:10.1007/s10723-019-09503-0

Lu, Y., Cao, B., Rego, C., & Glover, F. (2018). A Tabu Search based clustering algorithm and its parallel implementation on Spark. *Applied Soft Computing*, *63*, 97–109. doi:10.1016/j.asoc.2017.11.038

Ludwig, S. A. (2015). MapReduce-based fuzzy c-means clustering algorithm: Implementation and scalability. *International Journal of Machine Learning and Cybernetics*, *6*(6), 923–934. doi:10.1007/s13042-015-0367-0

Meraihi, Y., Mahseur, M., & Acheli, D. (2020). A Modified Binary Crow Search Algorithm for Solving the Graph Coloring Problem. *International Journal of Applied Evolutionary Computation*, *11*(2), 28–46. doi:10.4018/IJAEC.2020040103

Nguyen, T. P. Q., & Kuo, R. J. (2019). Automatic Fuzzy Clustering Using Non-Dominated Sorting Particle Swarm Optimization Algorithm for Categorical Data. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 99721–99734. doi:10.1109/ACCESS.2019.2927593

Nithya, G. S., & Prabha, K. A. (2019). A lion optimization based k-prototype clustering algorithm for mixed data. *System*, *6*(02).

Parvathavarthini, S., Karthikeyani Visalakshi, N., & Shanthi, S. (2019). Breast Cancer Detection using Crow Search Optimization based Intuitionistic Fuzzy Clustering with Neighborhood Attraction. *Asian Pacific Journal of Cancer Prevention: APJCP*, *20*(1), 157–165. doi:10.31557/APJCP.2019.20.1.157 PMID:30678427

Parvathavarthini, S., Karthikeyani Visalakshi, N., Shanthi, S., & Lakshmi, K. (2018). Crow-search-based intuitionistic fuzzy C-means clustering algorithm. In *Developments and Trends in Intelligent Technologies and Smart Systems* (pp. 129–150). IGI Global.

Parvathavarthini, S., Visalakshi, N. K., Shanthi, S., & Mohan, J. M. (2018). Crow search optimization based fuzzy C-means clustering for optimal centroid initialization. *Taga J Graphic Technol*, *14*, 3034–3035.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, *66*(336), 846–850. doi:10.1080/01621459.1971.10482356

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*(4), 53–65. doi:10.1016/0377-0427(87)90125-7

Sinha, A., & Jana, P. K. (2018). A hybrid MapReduce-based k-means clustering using genetic algorithm for distributed datasets. *The Journal of Supercomputing*, *74*(4), 1562–1579. doi:10.1007/s11227-017-2182-8

Tao, G., Xiangwu, D., & Yefeng, L. (2015, February). Parallel K-modes algorithm based on MapReduce. In *2015 Third International Conference on Digital Information, Networking, and Wireless Communications (DINWC)* (pp. 176-179). IEEE. doi:10.1109/DINWC.2015.7054238

Tripathi, A. K., Sharma, K., & Bala, M. (2018). A novel clustering method using enhanced grey wolf optimizer and mapreduce. *Big Data Research, 14*, 93-100.

Ullah, R., Khan, A., Abid, S. B. S., Khan, S., Shah, S. K., & Ali, M. (2020). Crow-ENN: An Optimized Elman Neural Network with Crow Search Algorithm for Leukemia DNA Sequence Classification. In Mobile Devices and Smart Gadgets in Medical Sciences (pp. 173-213). IGI Global.

Van Rijsbergen, C. J. (1979). *Information retrieval*. Butterworth-Heinemann.

Verma, A., Llorà, X., Goldberg, D. E., & Campbell, R. H. (2009, November). Scaling genetic algorithms using mapreduce. In *2009 Ninth International Conference on Intelligent Systems Design and Applications* (pp. 13-18). IEEE. doi:10.1109/ISDA.2009.181

Wang, J., Yuan, D., & Jiang, M. (2012, November). Parallel k-pso based on mapreduce. In *2012 IEEE 14th International Conference on Communication Technology* (pp. 1203-1208). IEEE.

Wang, Y. C., & Tsai, C. W. (2018, October). An efficient coral reef optimization with substrate layers for clustering problem on Spark. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 2814-2819). IEEE. doi:10.1109/SMC.2018.00479

Wu, Z. X., Huang, K. W., & Girsang, A. S. (2018, November). A Whole Crow Search Algorithm for Solving Data Clustering. In *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)* (pp. 152-155). IEEE. doi:10.1109/TAAI.2018.00040

Zhao, W., Ma, H., & He, Q. (2009, December). Parallel k-means clustering based on mapreduce. In *IEEE International Conference on Cloud Computing* (pp. 674-679). Springer.

N. Karthikeyani Visalakshi is working as a Assistant Professor in the Department of Computer Science, Government Arts and Science College, Kangeyam. She completed her Ph. D. degree in the area of Distributed Data Mining. She has got 26 years of teaching experience and 16 years of research experience. She has published 32 research papers in various National and International Journals. She received Best Oral Presentation award in 4 th International Science Congress (ISC-2014) at Pacific University, Udaipur, Rajasthan, organized by International Science Congress Association. She is a reviewer for two International Journals. Her areas of interests include Distributed Data Mining, Clustering, Image Processing, Rough sets, Fuzzy logic and Intuitionistic Fuzzy sets.

S. Shanthi received her BSc Degree in Computer Science, MCA degree in Computer Applications, Master of Philosophy in Computer Science from Bharathiar University, Coimbatore, India in the year 1993, 1996 and 2004 respectively and ME degree in Computer Science and Engineering from Anna University, Chennai, India in the year 2006. She completed her PhD degree in 2015 in Computer Science and Engineering at Anna University, Chennai, India. She is presently working as an Assistant Professor (SLG) in the Department of Computer Applications, Kongu Engineering College, Tamil Nadu, India. Her area of interest includes, Data Mining, Image Processing, Pattern Recognition and Soft Computing.

K. Lakshmi received her MCA from Shrimathi Indira Gandhi College, Trichy, India in 2003 and M.Phil from Bharathiyar University, Trichy in 2004. She is a Research Scholar at the Department of Computer Applications, Kongu Engineering College, Perundurai, India. Her research interest includes clustering algorithms, Knowledge Discovery in Databases.