Optimization Driven MapReduce Framework for Indexing and Retrieval of Big Data

Hemn Barzan Abdalla[1*], Awder Mohammed Ahmed[2,] and M. A. Al Sibahee[3]
[1] School of Computer Science, Wenzhou-Kean University, Wenzhou, China
[E-mail: habdalla@kean.edu]
[2] Communication Engineering Department, Technical College of Engineering, Sulaimani Polytechnic University, Sulaimani-Iraq,
[E-mail: awder.ahmed@spu.edu.iq]
[3] Department of Computer Science, Shenzen Institute of Huazhong University of Science and Technology, China,
[E-mail: mustafa.a@hust.edu.cn]
*Corresponding author: Hemn Barzan Abdalla

Received October 21, 2019; revised January 3, 2020; accepted March 15, 2020; published May 31, 2020

Abstract

With the technical advances, the amount of big data is increasing day-by-day such that the traditional software tools face a burden in handling them. Additionally, the presence of the imbalance data in big data is a massive concern to the research industry. In order to assure the effective management of big data and to deal with the imbalanced data, this paper proposes a new indexing algorithm for retrieving big data in the MapReduce framework. In mappers, the data clustering is done based on the Sparse Fuzzy-c-means (Sparse FCM) algorithm. The reducer combines the clusters generated by the mapper and again performs data clustering with the Sparse FCM algorithm. The two-level query matching is performed for determining the requested data. The first level query matching is performed for determining the cluster, and the second level query matching is done for accessing the requested data. The ranking of data is performed using the proposed Monarch chaotic whale optimization algorithm (M-CWOA), which is designed by combining Monarch butterfly optimization (MBO) [22] and chaotic whale optimization algorithm (CWOA) [21]. Here, the Parametric Enabled-Similarity Measure (PESM) is adapted for matching the similarities between two datasets. The proposed M-CWOA outperformed other methods with maximal precision of 0.9237, recall of 0.9371, F1-score of 0.9223, respectively.

Keywords: Big data indexing, MapReduce, Sparse FCM, Retrieval, Data clustering

1. Introduction

Big data is a field wherein the data are analyzed for the extraction of useful information by employing conventional data processing applications. The big data is described using 3V's, which includes Volume, Velocity, and Variety. Due to large structures, the advanced applications are highly recommended for processing and to facilitate improved decision making with enhanced discovery and process optimization. The big data is referred to as a dataset, which increases the complexities in collecting crucial data. Data mining means the extraction of useful information from massive databases using computing technologies [13] [12]. Due to the increasing size of the database, an extreme learning machine (ELM) is employed, which executes on the conventional serial environment, which is fast and efficient [10] [9]. The big data mining needs a scalable and effective solution, which is easily accessible to the users [14].

The solution for effective storage of data and management cannot complete the desires of heterogeneous data wherein the amount of data increases repeatedly. This poses the advantage of having a storage system, which effectively enables information search and retrieval. The indexing of big data is based on the solution, which uses a massive parallel computer or machine, which interconnects the CPU's, RAM's, and disk units. The usage of this method yields increased throughput for processing the data, thereby decreases the access time for data replication, with increased availability and reliability [30] [31]. The design of the method or the indexing type can be utilized for distinct processing datasets that are based on queries for processing a large number of queries like range queries, keyword queries, ad-hoc queries, and similarity queries. Thus, the designers are sensitive with respect to the data types that are to be indexed, and the query is selected based on indexes [16]. Indexing is utilized in the big data for performing effective retrieval tasks considering complex and voluminous datasets, along with distributed and scalable storage in cloud computing.

The goal of this paper is to devise a big data indexing model using an optimization algorithm. The big data indexing is progressed using the MapReduce framework that uses the proposed optimization algorithm, named M-CWOA algorithm. The proposed M-CWOA algorithm is designed by combining MBO with the CWOA algorithm. The big data is generated from the distributed sources is subjected to the mapper, wherein the obtained data is clustered based on Sparse FCM. The obtained clusters are further subjected to the reducers for easier retrieval. Then, the input query is used for determining the cluster to which the data resides. Once the cluster is obtained, then again, a query is adapted for determining the relevant data. Finally, the ranking of data is done using the proposed M-CWOA algorithm.

The main contribution of the paper:

- The big data indexing is progressed using the MapReduce framework that uses the proposed optimization algorithm, named M-CWOA algorithm. The proposed M-CWOA algorithm is designed by combining MBO with the CWOA algorithm.
- Designing a novel fitness function: The newly devised fitness function considers the Kendal rank correlation coefficient, Tanimoto similarity, and the Spearman correlation coefficient for big data indexing.

The arrangement of the paper is made as follows: The introductory part depending on the big data indexing, is deliberated in section 1, and the review of eight existing works is elaborated in section 2. In section 3, the proposed method of big data indexing is presented,

and section 4 elaborates on the results of the methods. At last, section 5 illustrated a summary of the research work.

2. Literature review

The eight existing methods based on big data indexing and retrieval is demonstrated in this section.

Aisha Siddiqaet et al.,[1] developed a framework named SmallClient indexing framework based on indexing strategies for ensuring enhanced data retrieval performance of the massive datasets. However, the method failed to utilize Java on Hadoop for solving significant data issues. Ezatpooret, P et al.,[2] developed a MapReduce Enhanced Bitmap Index Guided Algorithm (MRBIG) for dealing with the considerable data issues. However, the method may lead to performance overhead due to different performances with different structures. Limkar, S.V., and Jha, R.K [3] developed a method for parallel constructing the R-Tree based on the variants using Apache Spark framework and IoT Zetta platform. However, this method led to insecure communication due to a lack of security. Wang, L et al. [4] developed a pips Cloud, which combined cloud computing and HPC methods for enabling huge-scaled remote sensing (RS) data in the processing system with on-demand real-time services. Thus, the system overhead has occurred with massive data. Siddiqaet, A et al. [5] designed a model named SmallClient for speeding the query for executing massive datasets. This method failed to use a probabilistic machine learning algorithm with B-Tree indexing for attaining adaptive index creation by predicting query workload and index attribute. Omri, A et al. [6] proposed a Big Uncertainty Web Data Services Indexing Model, which was able to reason in the uncertain data environment. Anyhow, it did not use for big data processing. Kim, M et al. [7] proposed a GPU-aware parallel indexing method called G-tree, which combines the efficiency of the R-tree in low-dimensional space with the massive parallel processing potential of GPUs. It offers stable and consistent performance in the high-dimensional area. It did not support the multiple-query processing on GPUs. Chen, X et al. [8] proposed a DataMed which index, search, and ingest the biomedical data in the repositories. It built an integrated dataset search engine for the biomedical domain. Anyhow, there is no in-depth indexing of the different repositories. The features and challenges of existing and proposed systems are represented in **Table 1**.

Table 1. Features and Challenges of existing and proposed Indexing and Retrieval of Big Data

Authors [Citations]	Methodology	Merits	Demerits
Aisha Siddiqa et al., [1]	SmallClient indexing framework	• Used to enhance the indexing of big datasets.	• Big data issues can't be able to solve.
Ezatpooret, P et al., [2]	MRBIG	• random-distributed missing nodes in the dimensions • Deal with big data issues.	• lead to performance overhead due to different performances with different structures
Limkar, S.V., and Jha, R.K [3]	R-Tree	• Retrieve the data with high speed.	• insecure communication due to lack of security
Wang, L et al.[4]	pips Cloud	• enabling huge-scaled remote sensing data	• Real-time services.

Siddiqaet, A et al.[5]	SmallClient	• speeding the query for executing massive datasets	• lack of predicting query workload • prediction of index attribute
Omri, A et al., [6]	Big Uncertainty Web Data Services	• able to reasoning in the uncertain data environment	• big data processing
Kim, M et al., [7]	G-tree	• combine the efficiency of the R-tree in low-dimensional space with the massive parallel processing potential of GPUs. • stable and consistent performance in high-dimensional space	• multiple-query processing on GPUs.
Chen, X et al., [8]	DataMed	• index, search, and ingest the biomedical data in the repositories. • Build an integrated dataset search engine for the biomedical domain.	• No in-depth indexing of the different repositories.
Proposed Method	M-CWOA	• data clustering is done based on the Sparse FCM algorithm • combines the clusters generated by the mapper • The two-level query matching is performed for determining the requested data. • PESM is adapted for matching the similarities between two datasets.	•

2.2. Challenges

• In [1], a SmallClient indexing framework was devised using the petrinets for handling the massive datasets. However, the method reduces the big data performance and needs incessant efforts in enhancement.

•Incomplete or partial data is a kind of multi-dimensional dataset, which poses random-distributed missing nodes in the dimensions. It is complex to retrieve information from these kinds of datasets when it turns out to be significant. Thus the determination of top-k dominant values is a major challenge [2].

•Indexing methods are effective for current datasets, but inefficient with big data. The indexing size and indexing time are essential for massive datasets, and thus it is unfeasible to tolerate long delays in uploading data and data search operations [5].

•Indexing in the context of data-warehouse is a significant limitation due to the massive volume of the processed data and the number of candidate attributes, which is very large. The major challenge that exists is in indexing the uncertain data and data sources [6].

•The issues of the curse of dimensionality while handling massive datasets is a significant challenge. Many techniques were devised for enhancing the performance of queries in huge dimensional space by hierarchical indexing with the R-tree. Despite the curse of dimensionality challenge, the conventional datasets worsen considerably as the dimensionality of the datasets maximizes [7].

•The massive size of data brings enormous complexities in analytic data applications. The acquisition of knowledge and making decisions from the growing voluminous data creates many problems in organizations. Moreover, the read or write performance or data access and on-demand file creation are significant challenges faced in the design and processing of data. Other challenges of big data include data security, data acquisition, analysis, storage, and management.

3. Proposed M-CWOA for Big Bata Indexing Using the MapReduce Framework

This section illustrates the proposed technique named M-CWOA for big data indexing performed in the MapReduce framework. Each mapper from the MapReduce framework employs a clustering algorithm called Sparse FCM [23] for clustering the obtained big data. The centroids generated from the mapper are merged and fed as an input to the reducer. In reducer, again, the collected data is clustered based on sparse FCM in order to determine the optimal clusters. Here, two-level query matching is performed for the retrieval. In first level query matching, when an input query is generated then, the cluster relevant to the query is retrieved. For the obtained cluster, again, a query is made for getting the relevant data from the cluster. As per **Fig. 1**, if suppose during the two-level matching C_1 matches at level-1, then, in the level-2 matching, the data attributes within C_3 are compared based on the PESM measure such that the highly relevant data is retrieved. Finally, the ranking of data is performed using the newly designed optimization algorithm named M-CWOA, which is designed by integrating MBO [22] and CWOA [21]. **Fig. 1** elaborates on the big data indexing method using the MapReduce framework with the proposed M-CWOA algorithm.

The MapReduce framework provides high processing power as the number of servers is processed parallel in the mapper phase that poses the ability for parallel processing. The feasibility and the execution time of the big data are improved using the MapReduce framework such that the big input data is partitioned into different subsets of data, and the individual mapper processes each subset to produce the required output. Assume that the big input data is represented as, B with different number of attributes that are represented as

$$B = \left\{ b_{e,f} \right\};\ \ (1 \le e \le E);\ \ (1 \le f \le F) \tag{1}$$

where $b_{e,f}$ represents the data in the big data B specifying f^{th} attribute of the e^{th} data, E indicates the total number of data points, and F represents the total number of attributes for each data point. The mapper phase employs a mapper function and the sparse FCM algorithm for generating the optimal clusters. The significance of clustering is that the highly significant

clusters are chosen in order to ensure dimensional reduction and to maximize the classification accuracy.

The big data is fed to the mapper in the MapReduce framework, wherein the parallel processing of the big data is enabled. At first, the big data is split into different subsets of data, and is represented as,

$$b_{e,f} = \{P_g\}; \quad (1 \le g \le G); \tag{2}$$

where G denote total subsets of generated data.

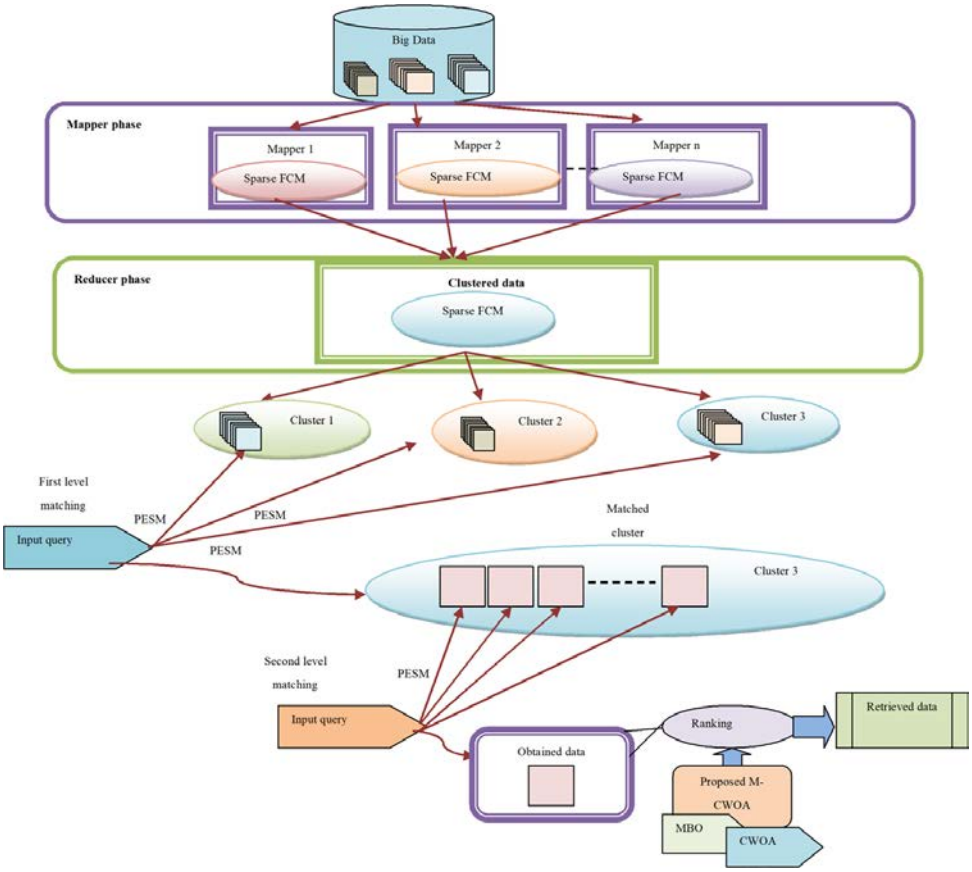

Fig. 1. Block diagram of the proposed M-CWOA based big data indexing

3.1. First level data clustering in the mapper phase using Sparse FCM

In first level data clustering, the clustering of data is performed on the mapper using the big partitioned data. The total subsets of data are equal to the total number of the mapper in the mapper phase that is formulated as,

$$M = \{M_1, M_2, ...M_g, ..., M_G\} \tag{3}$$

Where M_g indicates g^{th} mapper, and G indicates the total number of the mapper. The input of the g^{th} mapper is given as,

$$P_g = \{W_{x,y}\}; (1 \le x \le X); (1 \le y \le Y) \tag{4}$$

Where Y is the total data point in the g^{th} subset, and X is total attributes in g^{th} the subset, and $W_{x,y}$ is the data in g^{th} the subset. Each mapper M_g takes P_g as input and performs clustering using a sparse FCM algorithm for selecting the clusters from all the mapper. In the mapper phase, the significant data clustering is performed with the centroids for the big data $b_{e,f}$ that is devised using the sparse FCM [23]. The Sparse FCM handles the massive data and possesses the ability to select the highly significant and effective cluster centroid. The purpose of the proposed Sparse FCM is to compute the optimal centroid in the mappers of the MapReduce framework. Consider the data matrix as $D_c = a_{kl} \in \Im^{e \times f}$, which e is the total number of data points in the data and f is the total number of attributes. The Sparse FCM algorithm employs the distance metric to compute the cluster centroid, and the total number of centroids is based on the user and is hence, predefined. The cluster centroids are given as,

$$r = \{r_1, r_2, ..., r_d, ..., r_h\} \tag{5}$$

Where h refers to the total number of cluster centroids and r_d id the d^{th} cluster centroid. The cluster centroids are the highly significant data points that are highly essential to group the big data such that the processing and analysis using the big data become less complicated and take less time. The algorithmic steps of the proposed Sparse FCM are given below.

Step 1: Initialization: The primary step in the proposed Sparse FCM is the initialization of the data points in the big data that is expressed as,

$$w = w_1^o = w_2^o = ... = w_f^o = \frac{1}{f} \tag{6}$$

Step 2: Update the partition matrix: Consider r as the cluster center and fix the attribute weights as, w such that $\mathrm{E}(\Im)$ is minimized if and only if,

$$R_{k,d} = \begin{cases} \dfrac{1}{C_d} & ; \quad if\ \mathrm{E}_{k,d} = 0\ and\ C_d = card\ \{r : \mathrm{E}_{kd} = 0\} \\ 0 & ; \quad if\ \mathrm{E}_{kd} \neq 0\ but\ \mathrm{E}_{k\alpha} = 0\ for\ some\ \alpha,\ \alpha \neq d \\ \dfrac{1}{\sum_{\alpha=1}^{h} \left(\dfrac{\mathrm{E}_{kd}}{\mathrm{E}_{\alpha d}} \right)^{\left(\frac{1}{\beta - 1} \right)}} & ; \quad Otherwise \end{cases} \tag{7}$$

where, $card(h)$ specifies the cardinality of a set h, $R_{k,d}$ is the degree of membership of the k[th] object belonging to the d[th] fuzzy cluster, β is the fuzzy index of fuzzy membership E_{kd} is the distance measured in the Sparse FCM algorithm, α denotes the number of features, and C_d is the cardinality of cluster center r when $\mathrm{E}_{kd} = 0$. When $\mathrm{E}_{k,d} = 0\ and\ C_d = card\ \{r : \mathrm{E}_{kd} = 0\}$ then $R_{k,d}$ is the inverse of C_d, and $\mathrm{E}_{kd} \neq 0\ but\ \mathrm{E}_{k\alpha} = 0\ for\ some\ \alpha,\ \alpha \neq d$ then the value for $R_{k,d}$ is 0, otherwise inverse of

$\sum_{\alpha=1}^{h}\left(\frac{\mathrm{E}_{kd}}{\mathrm{E}_{\alpha d}}\right)^{\left(\frac{1}{\beta-1}\right)}$. The distance measured in the Sparse FCM algorithm is given as,

$$\mathrm{E}_{kd}=\sum_{d=1}^{h} w_l\left(a_{kl}-r_{dl}\right)^2 \tag{8}$$

where a_{kl} is the data point, w_l is the weight of cluster data, and r_{dl} is the cluster centroid. The distance can be calculated by the sum of the square of the difference between the data point, and cluster centroid is multiplied with the weight.

Step 3: Update the cluster centers: Assume r and $\Im$ be fixed and $\mathrm{E}(t)$ is minimized if

$$r_{dl}=\begin{cases}0 & ; \quad if\ r_l=0\\ \dfrac{\sum_{k=1}^{e} R_{k,d}^{\beta}.a_{kl}}{\sum_{i=1}^{p} R_{k,d}^{\beta}} & ; \quad if\ r_l\neq 0\end{cases} \tag{9}$$

Where, $k=1,...,e$, $l=1,...,f$, and β is the weighted exponent, which is responsible for handling the degree of membership sharing among the fuzzy clusters. The dissimilarity measure is indicated $\Im$. The proposed update rule of cluster centroid is derived as follows:

The proposed update rule of the Sparse FCM algorithm obtains the cluster centroids effectively with reasonable accuracy. The Sparse FCM applies to the significant data clustering that holds the ability to solve the data with variable features and missing data.

$$\frac{\sum_{l=1}^{f}\left|w^{*}-w_l^{o}\right|}{\sum_{l=1}^{f}\left|w_l^{o}\right|}<10^{-4} \tag{10}$$

Thus, the obtained clusters are given as,

$$C=\{C_1,C_2,\mathrm{K}\ ,C_q,\mathrm{K}\ C_s\} \tag{11}$$

Where s indicates total clusters. Thus, the cluster centroids are obtained using the Sparse FCM, and the number of the centroids is based on the user-set count, and the centroids group the clusters such that the data points in a cluster group exhibit similar characteristic, whereas the data points between the clusters exhibit dissimilar characteristics.

3.2. Second level data clustering in the reducer phase using Sparse FCM

The cluster centroids determined using Sparse FCM in the individual mappers are combined to form the intermediate clusters that form the input to the reducer R. In reducer, again the sparse.

FCM is adapted for the clustering of the merged data. Thus, the clustered data generated in the reducer is given as,

$$C^{'} = \{C_1^{'}, C_2^{'}, \text{K}\ C_q^{'}, \text{K}\ , C_s^{'}\} \tag{12}$$

The steps for executing the sparse FCM are illustrated in section 3.1.

3.3 Query matching for retrieving relevant data

Consider the input query Q, which is done to match the data belonging to the cluster $C_q^{'}$. The obtained cluster $C_q^{'}$ is matched with the query Q for second-level matching. The matched data P_g is finally retrieved.

Due to the massive size of data, the complexities in managing and handling the data increases in an exponential manner. Thus, the processing of big data is needed, and the replica of the database should be kept in the main memory for yielding faster data processing, but it can be done by adapting effective indexing techniques, which could retrieve the data with less cost and time. The processing of queries is done to retrieve the data in an efficient manner in order to speed up the retrieval process. Many query processing techniques are supported by indexing methods for ensuring effective data retrieval. Here, two-level query matching is carried out for retrieving the data from the massive database. In first level query matching, the input query is generated to the retrieve the relevant cluster to which the data resides using the PESM [24] similarity measure whereas, in second-level query matching, the query is given to each cluster for retrieving the exact data from the cluster using the PESM similarity measure. Finally, the ranking of the data is done using the newly designed optimization algorithm named M-CWOA, which is obtained by integrating the MBO [22] and CWOA [21].

3.4. Proposed M-CWOA for ranking big data

In this section, the proposed technique, named M-CWOA, for big data ranking performed in a MapReduce framework. Each mapper from the MapReduce framework employs a clustering algorithm named Sparse FCM for clustering the obtained big data. The clusters generated from the mapper are merged and fed as an input to the reducer. In reducer, again, the obtained data is clustered based on sparse FCM in order to determine the optimal clusters. Here, second-level query matching is performed for the final data retrieval. In first level query matching, when an input query is generated, then the cluster relevant to the query are retrieved. For the obtained cluster, again, a query is generated for obtaining the relevant data from the cluster. Finally, the ranking of data is performed using the newly designed optimization algorithm named M-CWOA, which is designed by combining MBO [22] and CWOA [21]. The solution encoding, fitness function, and the proposed M-CWOA algorithm are illustrated in the subsections.

3.4.1. Solution encoding

The solution encoding presents the simplest view of representing the algorithm designed for finding the clusters. Here, the solution is the cluster centroid, which is initialized in random, depending on the intermediate clusters produced by the mappers. The clusters are merged in the reducer and again clustered for performing the data clustering. Thus, the solution is a vector whose size is equivalent to the number of clusters and the data. Based on the fitness evaluated, the cluster centroids can be determined optimally using the M-CWOA algorithm.

Here, the relevant data is determined by ranking the data using the proposed M-CWOA algorithm. Consider $C_1^{'}, C_2^{'}, \mathrm{K}, C_q^{'}, \mathrm{K}\ C_s^{'}$ being the clusters in which the data resides. The required data is retrieved from the selected cluster, which consists of several data subsets which range from $1 \le e \le E$. **Fig. 2** illustrates the solution for ranking the data

.

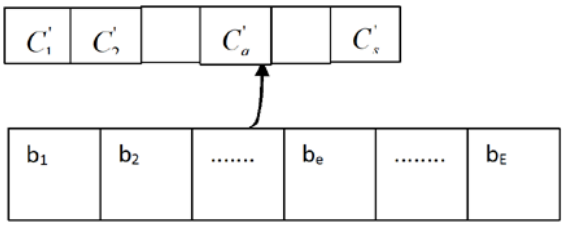

Fig. 2. Solution encoding for data ranking

3.4.2. Fitness function

The fitness function, which decides the quality of the solution, is designed using three similarity measures, namely the Kendal rank correlation coefficient, Tanimoto similarity measure, and Spearman correlation coefficient. The fitness function measures the similarity between the clusters and is suitable for clustering algorithms. The fitness function is formulated based on similarity measures, which are formulated as,

$$Fitness = \{Z + T + U\} \tag{13}$$

where Z is the Kendal rank correlation coefficient, T refers to the Tanimoto similarity, and U is the Spearman correlation coefficient.

(i) Kendal rank correlation coefficient

The Kendall rank correlation coefficient is utilized for measuring the ordinal association between the two measured quantities. It is also known as Kendall's tau coefficient.

Consider $(A_1, B_1), (A_2, B_2), \mathrm{K}, (A_v, B_v)$, representing a set of observations of the variable A and B such that all values of a_y and b_y are unique. The pair (A_y, B_y) and (A_t, B_t) where $y < t$ is termed as concordant if and only if ranks of both elements are satisfied that is if both $a_y > a_t$ and $b_y > b_t$; or else if both $a_y < a_t$ and $b_y < b_t$. They are termed as discordant pairs if $a_y > a_t$ and $b_y < b_t$, or else if both $a_y < a_t$ and $b_y > b_t$. If $a_y = a_t$ and $b_y = b_t$, then, the pair is neither discordant nor concordant. The Kendal rank coefficient is formulated as follows,

$$Z = \frac{N_c - N_d}{v(v-1)/2} \tag{14}$$

where N_c represents the count of concordant pairs, N_d indicates the number of discordant pairs, $v(v-1)/2$ refers to the binomial coefficient for the number of ways to select two items from v items.

(ii) Tanimoto similarity

The Tanimoto [25] metric is the metric utilized for determining the similarity through the

overlap or by the intersection of a set. This metric is defined as the ratio of intersecting set to the union set for computing the similarity. The Tanimoto similarity measure is computed as follows,

$$T = \frac{\sum_{\gamma}^{\kappa} \eta_\gamma \rho_\gamma}{\sum_{\gamma=1}^{\kappa} \eta_\gamma^2 + \sum_{\gamma=1}^{\kappa} \rho_\gamma^2 - \sum_{\gamma=1}^{\kappa} \eta_\gamma \rho_\gamma} \tag{15}$$

Where, η_γ and ρ_γ indicates the two vectors, and κ is the total number of observations where $1 \le \gamma \le \kappa$.

(iii) Spearman correlation coefficient

The Spearman correlation coefficient is also referred to as the Pearson correlation coefficient amongst the rank variables. For sample size, L, the L raw scores $(X_j^{'}, Y_j^{'})$ are transformed to RX_j, RY_j and is given by,

$$U = \mathrm{I}_{RX,RY} = \frac{Cov(R_X, R_Y)}{\sigma_{RX}\sigma_{RY}} \tag{16}$$

Where, I indicates the Pearson correlation coefficient, $Cov(R_X, R_Y)$ indicate the covariance of rank variables, σ_{RX}, σ_{RY} is the standard deviation of rank variables.

3.4.3. Proposed M-CWOA algorithm

The obtained data are ranked based on the proposed M-CWOA algorithm. The proposed M-CWOA algorithm is designed for the optimal selection of cluster centroids to perform clustering is summarized in this section. The proposed M-CWOA algorithm is designed by modifying the update process of MBO using CWOA. The MBO [22] algorithm is inspired by the migration behavior of monarch butterflies for optimizing the global optimization tasks. However, the MBO algorithm may lead to premature convergence, which can be solved using CWOA that has better convergence behavior with local optima avoidance. The CWOA is designed by introducing chaos in the WOA algorithm. The CWOA algorithm is well known for its speed, which helps the optimization techniques in exploring the search space in a vigorous manner [21]. Thus, integrating CWOA with the MBO algorithm leads to an optimal global solution, with high performance.

The steps involved in the proposed M-CWOA algorithm are described as follows:

Step 1. Initialization: The initialization of the population P is given as follows:

$$P = \{P_1, P_2, \mathrm{K}, P_o, \mathrm{K}, P_n\} \tag{17}$$

Where N is the total population, and P_o represents the o^{th} solution.

Step 2: Evaluation of fitness function: The fitness of the solution is computed based on the formula shown in equation (13). The fitness of the individual solutions is computed, and the solution that attains the maximum value of fitness is selected as the optimal solution.

Step 3: Determination of update position: To enhance the algorithmic performance, the CWOA algorithm is being utilized. Thus, the update equation based on the CWOA [21] is

given as,

$$P_{i,v}(\vartheta+1)=\begin{cases}\dfrac{P(\vartheta)}{0.7} & ;P(\vartheta)<0.7\\ \dfrac{10}{3}(1-P(\vartheta)) & ;P(\vartheta)\geq 0.7\end{cases} \tag{18}$$

To increase the speed of convergence rate, the MBO algorithm is being utilized. The MBO algorithm utilizes the behavior of butterflies for providing optimal performance while solving the optimization issues, and less parameter is needed for fine-tuning. As per the MBO algorithm [22], the solution update is expressed as,

$$P_{i,v}(\vartheta+1)=P_{i,v}(\vartheta)+\mu(dP_v-0.5) \tag{19}$$

Where dP_v represents the walk step of butterfly i, μ refers to the weighting factor, $P_{i,v}(\vartheta)$ is the current position of the butterfly, and $P_{i,v}(\vartheta+1)$ is the next position of the butterfly.

Above equation is rewritten as follows:

$$P_{i,v}(\vartheta)=P_{i,v}(\vartheta+1)-\mu(dP_v-0.5) \tag{20}$$

Thus, the solution update of the proposed M-CWOA algorithm after substituting equation (20) in equation (18) is expressed as,

$$P_{i,v}(\vartheta+1)=\frac{P_{i,v}(\vartheta+1)-\mu(dP_v-0.5)}{0.7} \tag{21}$$

$$P_{i,v}(\vartheta+1)=\frac{P_{i,v}(\vartheta+1)}{0.7}-\frac{\mu(dP_v-0.5)}{0.7} \tag{22}$$

$$P_{i,v}(\vartheta+1)-\frac{P_{i,v}(\vartheta+1)}{0.7}=\frac{-\mu(dP_v-0.5)}{0.7} \tag{23}$$

$$P_{i,v}(\vartheta+1)\left(1-\frac{1}{0.7}\right)=\frac{-\mu(dP_v-0.5)}{0.7} \tag{24}$$

$$P_{i,v}(\vartheta+1)\left(\frac{-0.3}{0.7}\right)=\frac{-\mu(dP_v-0.5)}{0.7} \tag{25}$$

$$P_{i,v}(\vartheta+1)=\frac{\mu(dP_v-0.5)}{0.3} \tag{26}$$

Similarly, the update equation of proposed M-CWOA algorithm can be written by substituting equation (20) in second equation of the CWOA algorithm given in (18) and is expressed as,

$$P_{i,v}(\vartheta+1)=\frac{10}{3}\left(1-P_{i,v}(\vartheta)\right) \tag{27}$$

$$P_{i,v}(\vartheta+1)=\frac{10}{3}\left(1-P_{i,v}(\vartheta+1)+\mu(dP_v-0.5)\right) \tag{28}$$

$$P_{i,v}(\vartheta+1)=\frac{10}{3}-\frac{10}{3}P_{i,v}(\vartheta+1)+\frac{10}{3}\mu(dP_v-0.5) \tag{29}$$

$$P_{i,v}(\vartheta+1)+\frac{10}{3}P_{i,v}(\vartheta+1)=\frac{10}{3}\left(1+\mu(dP_v-0.5)\right) \tag{30}$$

$$P_{i,v}(\vartheta+1)\frac{13}{3}=\frac{10}{3}\left(1+\mu(dP_v-0.5)\right) \tag{31}$$

$$P_{i,v}(\vartheta+1)=\frac{10}{13}\left(1+\mu(dP_v-0.5)\right) \tag{32}$$

Equation (33) forms the position update equation of the proposed M-CWOA algorithm, which improves the performance of the algorithm in big data indexing and retrieval.

$$P_{i,v}(\vartheta+1)=\begin{cases}\dfrac{\mu(dP_v-0.5)}{0.3} & ;P_{i,v}(\vartheta)<0.7\\ \dfrac{10}{13}\left(1+\mu(dP_v-0.5)\right); & P_{i,v}\geq 0.7\end{cases} \tag{33}$$

Step 4: Ranking the solutions based on the fitness: The solutions are ranked based on the fitness and the solution that ranked the highest, forms the best solution.
Step 5: Terminate: The iteration is continued for the maximum number of iterations and terminates upon the generation of the optimal global solution.

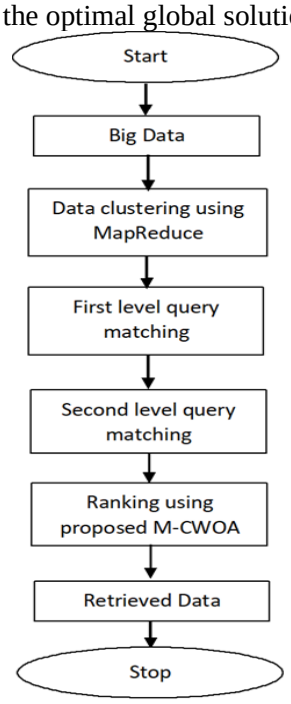

Fig. 3. Flowchart for proposed M-CWOA based big data indexing

The flowchart for the proposed M-CWOA based big data clustering is represented in **Fig. 3**. The big data can be clustered using the MapReduce framework, and then based on the first and second input query, the relevant data can be retrieved. Then, the ranking can be done by using the proposed M-CWOA and based on that, and the required data can be retrieved.

4. Results and Discussion

This section elaborates on the results obtained by the existing method to prove the effectiveness of the proposed method by performing a comparative analysis using four datasets. The analysis is performed based on precision, recall, and F1-Score metric.

4.1 Setup of the experiment

The experimentation is performed in MATLAB tool using Windows 10 OS with Intel i3 processor and 2GB RAM.

4.2 Dataset description

The four datasets are taken for experimentation, which is illustrated in this section.

i) Breast cancer Data set

Breast Cancer Data Set [26] is donated by Ming Tan and Jeff Schlimmer using the domain offered by the Oncology Institute that has appeared from the machine learning literature. The dataset contained 201 instances of one class and 85 instances of another class. The instances are elaborated using nine attributes, which are either linear or nominal. Some of the attributes are age, tumor-size, irradiate, node-caps, and so on. The number of web hits attained by the dataset is around 405043 and is multivariate in nature.

ii) Breast cancer Wisconsin data set

The Breast Cancer Wisconsin Data Set [27] is licensed under CC BY-NC-SA 4.0 and is used for predicting if the cancer is malignant or benign. The features are calculated using a digitized image of a fine needle of a breast mass. The dataset elaborates on the features of the cell nuclei present in the image. Some of the attributes include id, which is used for id number, diagnosis (M = malignant, B = benign), radius means which computes the mean of distances, texture_mean, which evaluates the standard deviation of gray-scale values and so on. This dataset is applicable in three domains, like healthcare, mathematics, and cancer diagnosis.

iii) Cardiotocography

The Cardiotocography Data set [28] consists of measurements of fetal heart rate(FHR) and uterine contraction(UC) features on cardiotocography classified by expert obstetricians database with multivibrator characteristics. The number of instances is 2126, with 23 attributes. The number of web hits attained by the dataset is 146930. The area of the dataset is life. The associated task of the dataset is classification.

iv)Statlog heart disease

The Statlog (Heart) Data Set [29] is a kind of heart disease database with multivariate characteristics. The tasks associated with the dataset are classification. The number of instances is 270, with 13 attributes. The characteristic of attributes is either categorical or real. The number of web hits attained by the dataset is 192944. Some of the attributes include age, chest pain type, resting blood pressure, and so on.

4.3 Performance metrics

The metrics adapted for analyzing the methods include precision, recall, and F1-Score.

4.3.1) Precision: Precision refers to the highest level of exactness, which is given as,

$$\Pr ecision = \frac{TP}{TP + FP} \tag{34}$$

4.3.2) Recall: Recall refers to the ratio of the true positive concerning the addition of true positive and false negative and is formulated as,

$$\mathrm{Re}\, call = \frac{TP}{TP + FN} \tag{35}$$

4.3.3) F1-score: Itis defined as the measure of accuracy, which is used to compute the value using precision and recall. It is defined by,

$$F1 - score = \frac{2 * precision * recall}{precision + recall} \tag{36}$$

4.4. Comparative analysis

The analysis is performed using the existing SmallClient indexing [1], Top-K Dominance indexing [2], parallel indexing [3], and proposed M-CWOA based on precision, recall, and F1-Score. The analysis is performed by varying the number of retrievals from 5 to 25.

4.4.1 Analysis using Breast cancer dataset

Fig. 3 illustrates the analysis of methods based on precision, recall, and F1-Score parameter using the breast cancer Data Set. The analysis based on the F1-Score parameter is portrayed in **Fig. 3a**. When the number of data retrieval is 5, the corresponding F1-Score values computed by existing SmallClient indexing, Top-K Dominance indexing, parallel indexing, and proposed M-CWOA are 0.828, 0.863, 0.863, and 0.96, respectively. Likewise, when the number of data retrieval is 25, the corresponding F1-Score values computed by existing SmallClient indexing, Top-K Dominance indexing, parallel indexing, and proposed M-CWOA are 0.610, 0.630, 0.666, and 0.731, respectively. The analysis based on the precision parameter is portrayed in **Fig. 3b**. When the number of data retrieval is 5, the corresponding precision values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.893, 0.897, 0.901, and 0.96, respectively. Likewise, when the number of data retrieval is 25, the corresponding precision values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.62, 0.62, 0.636, and 0.72, respectively. The analysis based on the recall parameter is portrayed in **Fig. 3c**. When the number of data retrieval is 5, the corresponding recall values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.817, 0.817, 0.849, and 0.96, respectively. Likewise, when the number of data retrieval is 25, the corresponding recall values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.706, 0.706, 0.706, and 0.726, respectively.

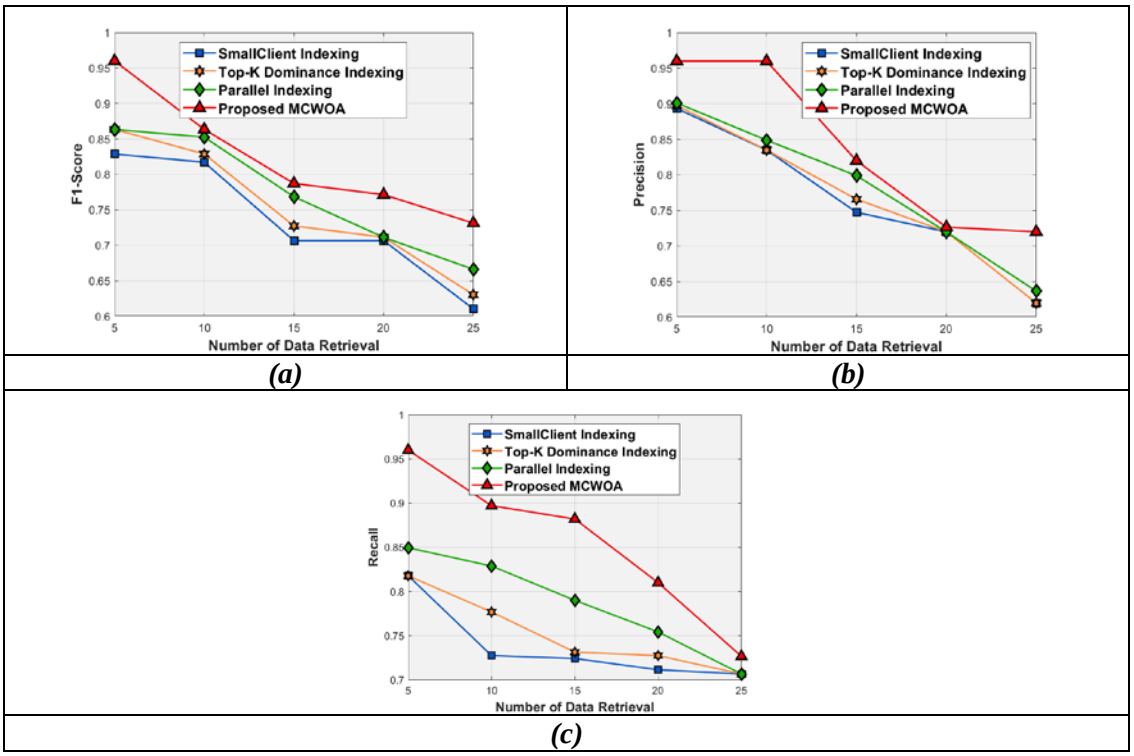

Fig. 3. Analysis of methods using Breast cancer dataset a) F1-Score b) Precision c) Recall

4.4.2 Analysis using Breast cancer wins dataset

Fig. 5 illustrates the analysis of methods based on precision, recall, and F1-Score parameter using breast cancer wins Data Set. The analysis based on the F1-Score parameter is portrayed in **Fig. 4a**. When the number of data retrieval is 5, the corresponding F1-Score values computed by existing SmallClient indexing, Top-K Dominance indexing, parallel indexing, and proposed M-CWOA are 0.931, 0.935, 0.931, and 0.954, respectively. Likewise, when the number of data retrieval is 25, the corresponding F1-Score values computed by existing SmallClient indexing, Top-K Dominance indexing, parallel indexing, and proposed M-CWOA are 0.826, 0.830, 0.744, and 0.829, respectively. The analysis based on the precision parameter is portrayed in **Fig. 4b**. When the number of data retrieval is 5, the corresponding precision values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.882, 0.945, 0.886, and 0.945, respectively. Likewise, when the number of data retrieval is 25, the corresponding precision values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.790, 0.790, 0.735, and 0.805, respectively. The analysis based on the recall parameter is portrayed in **Fig. 4c**. When the number of data retrieval is 5, the corresponding recall values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.936, 0.930, 0.936, 0.950, respectively. Likewise, when the number of data retrieval is 25, the corresponding recall values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.771, 0.794, 0.740, 0.833, respectively.

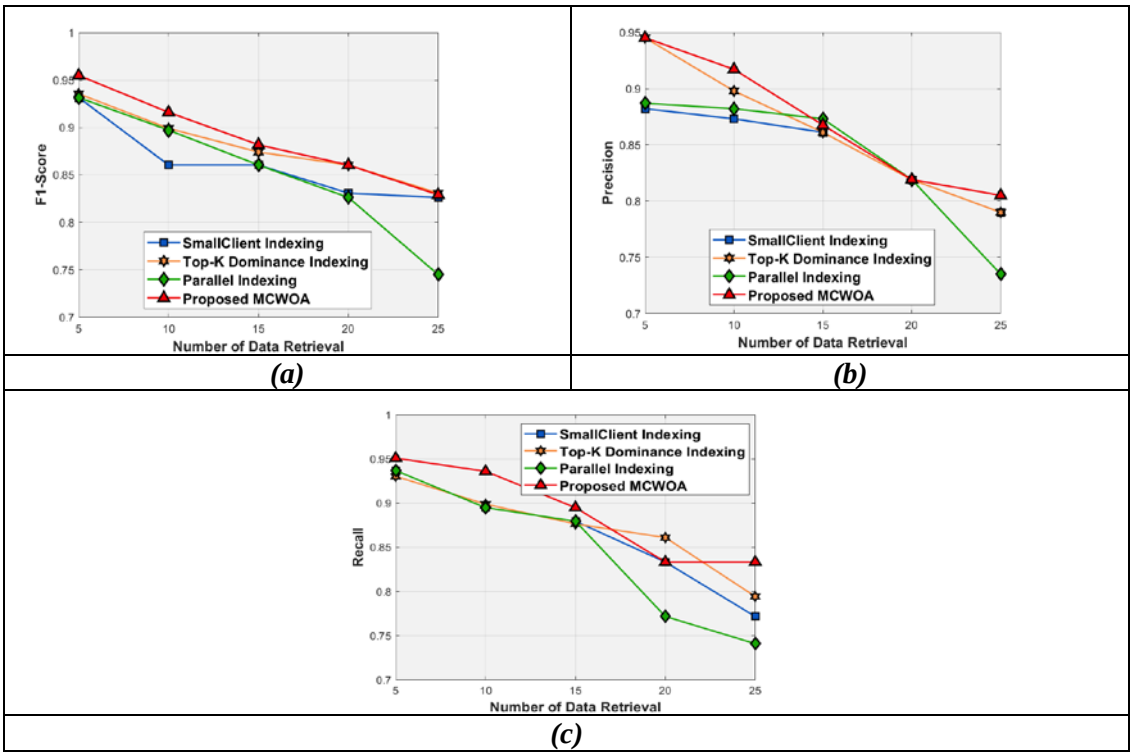

Fig. 4. Analysis of methods using Breast cancer wins dataset a) F1-Score b) Precision c) Recall

4.4.3 Analysis using Cardiotocography dataset

Fig. 6 illustrates the analysis of methods based on precision, recall, and F1-Score parameter using cardiotocography Data Set. The analysis based on the F1-Score parameter is portrayed in **Fig. 5a**. When the number of data retrieval is10, the corresponding F1-Score values computed by existing SmallClient indexing, Top-K Dominance indexing, parallel indexing, and proposed M-CWOA are0.8824, 0.8577, 0.8548, and 0.9223, respectively. Likewise, when the number of data retrieval is 25, the corresponding F1-Score values computed by existing SmallClient indexing, Top-K Dominance indexing, parallel indexing, and proposed M-CWOA are 0.8391, 0.8402, 0.7523, and 0.8408, respectively. The analysis based on the precision parameter is portrayed in **Fig. 5b**. When the number of data retrieval is 10, the corresponding precision values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.8982, 0.8245, 0.8358, and 0.9237, respectively. Likewise, when the number of data retrieval is 25, the corresponding precision values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.8009, 0.8009, 0.7537, and 0.8032, respectively. The analysis based on the recall parameter is portrayed in **Fig. 5c**. When the number of data retrieval is10, the corresponding recall values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.9322, 0.8427, 0.8521, and 0.9371 respectively. Likewise, when the number of data retrieval is 25, the corresponding recall values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are0.8143, 0.8076, 0.7671, and 0.81121, respectively.

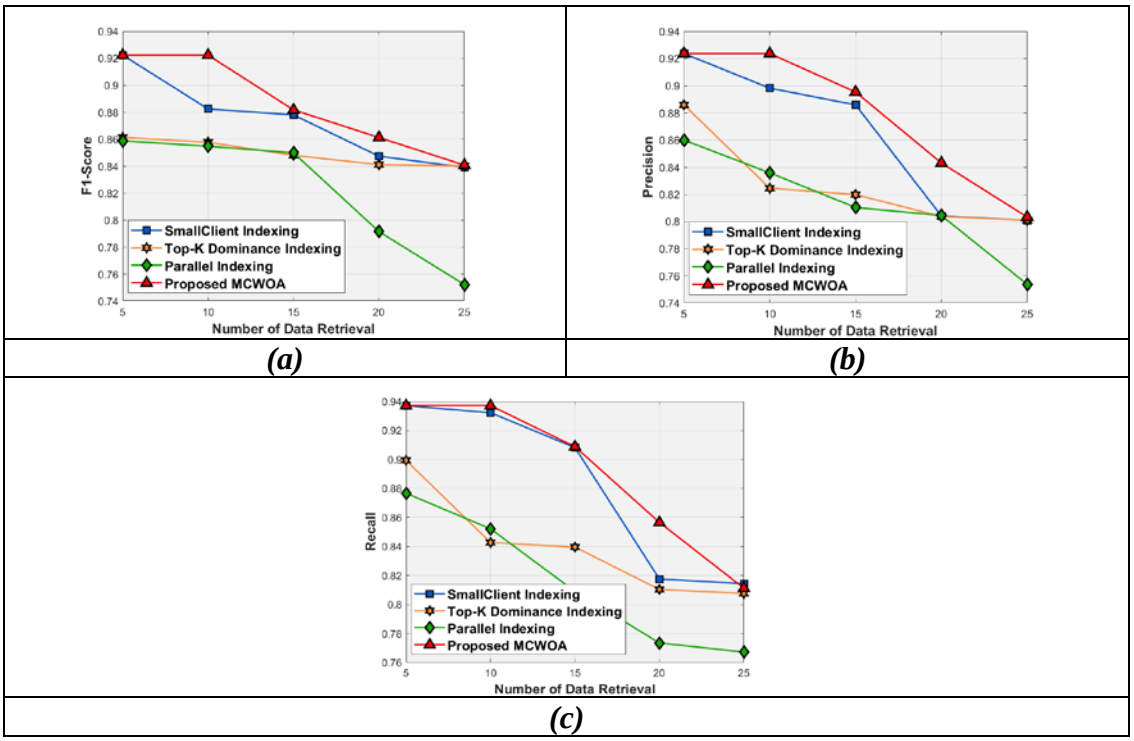

Fig. 5. Analysis of methods using Cardiotocographydataset a) F1-Score b) Precision c) Recall

4.4.4 Analysis using Statlog heart disease dataset

Fig. 7 illustrates the analysis of methods based on precision, recall, and F1-Score parameter using Statlog heart disease Data Set. The analysis based on the F1-Score parameter is portrayed in **Fig. 6a**. When the number of data retrieval is 5, the corresponding F1-Score values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.951, 0.945, 0.951, and 0.957, respectively. Likewise, when the number of data retrieval is 25, the corresponding F1-Score values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.757, 0.798, 0.757, and 0.839, respectively. The analysis based on the precision parameter is portrayed in **Fig. 6b**. When the number of data retrieval is 5, the corresponding precision values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.962, 0.970, 0.954, and 0.974, respectively. Likewise, when the number of data retrieval is 25, the corresponding precision values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.767, 0.808, 0.767, and 0.844, respectively. The analysis based on the recall parameter is portrayed in **Fig. 6c**. When the number of data retrieval is 5, the corresponding recall values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.957, 0.926, 0.957, and 0.963, respectively. Likewise, when the number of data retrieval is 25, the corresponding recall values computed by existing SmallClient indexing, Top-K Dominance indexing, Parallel indexing, and proposed M-CWOA are 0.770, 0.807, 0.770, and 0.851, respectively.

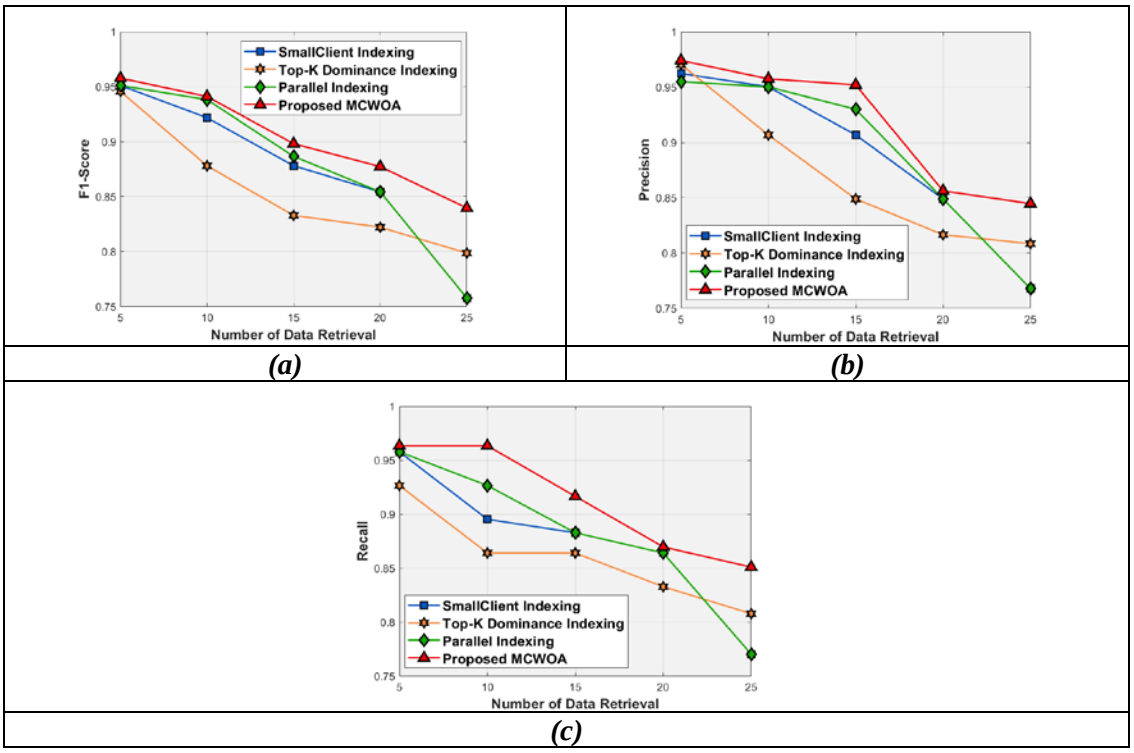

Fig. 6. Analysis of methods using Statlog heart disease dataset a) F1-Score b) Precision c) Recall

4.5. Comparative discussion

Table 2 illustrates the analysis of methods based on F1-Score, precision, and recall measure using five datasets. The maximal F1-Score is acquired by proposed M-CWOA with an F1-Score value of 0.9223, whereas the F1-Score value attained by the existing methods like SmallClient indexing, Top-K Dominance indexing, Parallel indexing are 0.8824, 0.8613 and0.8588, respectively. The maximal precision is measured by the proposed M-CWOA with precision value as 0.9237whereas the precision values of existing methods like SmallClient indexing, Top-K Dominance indexing, parallel indexing, is 0.9227, 0.8859, and 0.8599, respectively. The maximal recall is computed by proposed M-CWOA with recall value as0.9371whereas the existing methods like SmallClient indexing, Top-K Dominance indexing, Parallel indexing, is0.9369, 0.8993, and 0.8767, respectively.

Table 2. Comparative analysis

Dataset	Metric	SmallClient indexing	Top-K Dominance indexing	Parallel indexing	Proposed M-CWOA
Using breast cancer Data Set	F1-Score	0.610	0.630	0.666	0.731
	Precision	0.62	0.62	0.636	0.72
	Recall	0.706	0.706	0.706	0.726
Using breast cancer wins Data Set	F1-Score	0.826	0.829	0.744	0.829
	Precision	0.790	0.790	0.735	0.805
	Recall	0.771	0.794	0.740	0.833

Using Cardiotocography Data Set	F1-Score	0.8824	0.8613	0.8588	0.9223
	Precision	0.9227	0.8859	0.8599	0.9237
	Recall	0.9269	0.8993	0.8767	0.9371
Using statlog heart disease	F1-Score	0.757	0.798	0.757	0.839
	Precision	0.767	0.808	0.767	0.844
	Recall	0.770	0.807	0.770	0.851

5. Conclusion

The paper deals with the proposed big data indexing and retrieval that aimed at meeting the rising demands of high volume, high velocity, high value, high veracity, and huge variety. The big data streaming is performed using the MapReduce framework such that the data from the distributed sources is handled parallel at the same time. Initially, the data is split and sent to different mappers wherein each mapper performs data clustering using Sparse FCM. The obtained clusters are merged and fed to the reducer wherein, again, clustering is done to separate the data packets into different clusters. The two-level query is adapted for determining the relevant data. The first level query finds the cluster to which the data belongs, and the second level query finds the exact data that the user requested. The ranking of data is done using the proposed M-CWOA algorithm, which is designed by integrating the CWOA and MBO algorithm. The final output from the MapReduce framework is the relevant data. The analysis of the methods confirms that the proposed method outperformed the existing methods with the precision, recall, and F1-Score with values 0.9237, 0.9371, and 0.9223, respectively.

Acknowledgment

This work is partially supported through a research grant from Wenzhou-Kean University-College of Science with Sulaimani Polytechnic University (SPU) - Technical College of Engineering.

References

[1] Aisha Siddiqa, Ahmad Karim, Victor Chang, “Modeling SmallClient indexing framework for big data analytics,” *Journal Supercomputing*, vol. 74, pp. 5241-5262, 2018. Article (CrossRef Link).

[2] P. Ezatpoor, J. Zhan, J. M. Wu, and C. Chiu, “Finding Top- k Dominance on Incomplete Big Data Using MapReduce Framework,” *IEEE Access*, vol. 6, pp. 7872-7887, 2018. Article (CrossRef Link).

[3] Limkar, S.V. and Jha, R.K., “A novel method for parallel indexing of real time geospatial big data generated by IoT devices,” *Future Generation Computer Systems*, vol.97, pp.433-452, 2019. Article (CrossRef Link).

[4] Wang, L., Ma, Y., Yan, J., Chang, V. and Zomaya, A.Y., “pipsCloud: High performance cloud computing for remote sensing big data management and processing,” *Future Generation Computer Systems*, vol.78, pp.353-368, 2018. Article (CrossRef Link).

[5] Siddiqa, A., Karim, A., and Chang, V., “SmallClient for big data: an indexing framework towards fast data retrieval,” *Cluster Computing*, vol.20, no.2, pp.1193-1208, 2017. Article (CrossRef Link).

[6] Omri, A., Benouaret, K., Omri, M.N. and Benslimane, D., "Toward a New Model of Indexing Big Uncertain Data," in *Proc. of the 9th International Conference on Management of Digital EcoSystems*, pp. 93-98, November 2017. Article (CrossRef Link).
[7] Kim, M., Liu, L., and Choi, W., "A GPU-Aware Parallel Index for Processing High-Dimensional Big Data," *IEEE Transactions on Computers*, vol.67, no.10, pp.1388-1402, 2018. Article (CrossRef Link).
[8] Chen, X., Gururaj, A.E., Ozyurt, B., Liu, R., Soysal, E., Cohen, T., Tiryaki, F., Li, Y., Zong, N., Jiang, M. and Rogith, D., "DataMed–an open source discovery index for finding biomedical datasets," *Journal of the American Medical Informatics Association*, vol.25, no.3, pp.300-308, 2018. Article (CrossRef Link).
[9] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, J. M. Benítez, and F. Herrera, "Nearest Neighbor Classification for High-Speed Big Data Streams Using Spark," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2727-2739, October 2017. Article (CrossRef Link).
[10] M. Duan, K. Li, X. Liao and K. Li, "A Parallel Multi classification Algorithm for Big Data Using an Extreme Learning Machine," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2337-2351, June 2018. Article (CrossRef Link).
[11] W. Lin, Z. Wu, L. Lin, A. Wen and J. Li, "An Ensemble Random Forest Algorithm for Insurance Big Data Analysis," *IEEE Access*, vol. 5, pp. 16568-16575, 2017. Article (CrossRef Link)
[12] Ramírez-Gallego, S., García, S., Benítez, J.M. and Herrera, F., "A distributed evolutionary multivariate discretizer for big data processing on apache spark," *Swarm and Evolutionary Computation*, vol. 38, pp. 240-250, 2018. Article (CrossRef Link).
[13] Karim, M.R., Cochez, M., Beyan, O.D., Ahmed, C.F. and Decker, S., "Mining maximal frequent patterns in transactional databases and dynamic data streams: a spark-based approach," *Information Sciences*, vol. 432, pp.278-300, 2018. Article (CrossRef Link).
[14] A. Koliopoulos, P. Yiapanis, F. Tekiner, G. Nenadic and J. Keane, "A Parallel Distributed Weka Framework for Big Data Mining Using Spark," in *Proc. of 2015 IEEE International Congress on Big Data, New York, NY*, pp. 9-16, 2015. Article (CrossRef Link).
[15] Gani, A., Siddiqa, A., Shamshirband, S. and Hanum, F., "A survey on indexing techniques for big data: taxonomy and performance evaluation," *Knowledge and information syste*ms, vol.46, no.2, pp.241-284, 2016. Article (CrossRef Link).
[16] Adamu, F.B., Habbal, A., Hassan, S., Malaysia, U.U., Cottrell, R.L., White, B., Abdullah, I. and Malaysia, U.U, "A survey on big data indexing strategies," in *Proc. of 4th International Conference on Internet Applications, Protocol and Services (NETAPPS2015)*, 2015. Article (CrossRef Link).
[17] Aguilera M K, Golab W, Shah M A, "A practical scalable distributed b-tree," in *Proc. of the VLDB Endowment*, vol.1, no.1, pp.598–609, 2008. Article (CrossRef Link).
[18] Wu S, Wu K L, "An indexing framework for efficient retrieval on the cloud," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, pp.8, 2009.
[19] Cheng J, Ke Y, Fu AW-C, Yu JX, "Fast graph query processing with a low-cost index," *VLDB J*, vol.20, no.4, pp.521–539, 2011. Article (CrossRef Link).
[20] Wu K, Shoshani A, StockingerK , "Analyses of multi-level and multi-component compressed bitmap indexes," *ACM Trans Database Syst*, vol.35, no.1, pp.1–52, 2010. Article (CrossRef Link).
[21] Kaur, G. and Arora, S., "Chaotic whale optimization algorithm," *Journal of Computational Design and Engineering*, vol.5, no.3, pp.275-284, 2018. Article (CrossRef Link).
[22] Wang, G.G., Deb, S., Zhao, X. and Cui, Z., "A new monarch butterfly optimization with an improved crossover operator," *Operational Research*, vol.18, no.3, pp.731-755, 2018. Article (CrossRef Link).
[23] Chang, X., Wang, Q., Liu, Y. and Wang, Y., "Sparse Regularization in Fuzzy c-Means for High-Dimensional Data Clustering," *IEEE transactions on cybernetics*, vol.47, no.9, pp.2616-2627, 2016. Article (CrossRef Link).

[24] PoonamYadav, "Case Retrieval Algorithm Using Similarity Measure and Fractional Brain Storm Optimization for Health Informaticians," *The International Arab Journal of Information* Technology, Vol. 16, No. 2, March 2019.
[25] Sergyan, S., "Color histogram features based image classification in content-based image retrieval systems," in *Proc. of 6th International Symposium on Applied Machine Intelligence and Informatics, IEEE*, pp. 221-224, 2008. Article (CrossRef Link).
[26] Breast Cancer Data Set, Accessed on August 2019. https://archive.ics.uci.edu/ml/datasets/breast+cancer
[27] Breast Cancer Wisconsin (Diagnostic) Data Set.
[28] Cardiotocography Database, Accessed on September 2019. https://archive.ics.uci.edu/ml/datasets/Cardiotocography-database
[29] Statlog (Heart) Data Set, Accessed on August 2019. http://archive.ics.uci.edu/ml/datasets/statlog+(heart)
[30] Sumalatha Saleti, and R. B. V. Subramanyam, "A novel Bit Vector Product algorithm for mining frequent itemsets from large datasets using MapReduce framework," *Cluster Computing*, vol. 21, no. 2, pp. 1365-1380, June 2018. Article (CrossRef Link).
[31] Yaling Xun, Jifu Zhang, and Xiao Qin, "FiDoop: Parallel Mining of Frequent Itemsets Using MapReduce," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 3, pp. 313-325, March 2016. Article (CrossRef Link).

Hemn Barzan Abdalla Holds a Ph.D. degree in the field of Communication and Information Engineering. He possesses one decade of experience in teaching and worked as a project assistant in various higher education places. Also, currently, he is working as a lecturer at Wenzhou-Kean University with a member of the Institute of training and development in Sulaimani (KRG). He is an Editorial board member/reviewer of International/ National Journals and Conferences. He has more than 100 project systems for several places. His research interests include big data and data security, NoSQL, application.

Awder Mohammed Ahmed is a Ph.D. student at Duhok Polytechnic University (DPU). He obtained his MSc degree in Network Technology and Management from the University of Huddersfield, United Kingdom, in 2012. Currently, he is working as a lecturer at Sulaimani Polytechnic University (SPU) - Technical College of Engineering. His research interest includes Network Technology, E-Learning , Information Security, Fog and Cloud Computing.

Mustafa A. Al Sibahee is an Researcher of Huazhong University of Science and Technology Shenzhen Institute, Shenzhen, China. Lecturer at Department of Communication Engineering at Iraq University College, Basrah, Iraq. He is received his Ph.D. (2018) from Huazhong University of Science and Technology, Wuhan-China. His research interests include Computer Networks and Information Security, Computer Network Measurements, Machine Learning Algorithms Applications, Wireless Sensor Network (WSN), Software Defined Networking (SDN), Embedded Systems and Cyber Physical Systems (CPS).