mathematics

MDPI

Article

An Intelligent Metaheuristic Binary Pigeon Optimization-Based Feature Selection and Big Data Classification in a MapReduce Environment

Felwa Abukhodair [1], Wafaa Alsaggaf [1], Amani Tariq Jamal [2], Sayed Abdel-Khalek [3,4] and Romany F. Mansour [5,*]

[1] Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; felwaabukhodair@gmail.com (F.A.); waalsaggaf@kau.edu.sa (W.A.)
[2] Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; Atjamal@kau.edu.sa
[3] Department of Mathematics and Statistics, College of Science, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; sayedquantum@yahoo.co.uk
[4] Department of Mathematics, Faculty of Science, Sohag University, Sohag 82524, Egypt
[5] Department of Mathematics, Faculty of Science, New Valley University, El-Kharga 72511, Egypt
* Correspondence: romanyf@sci.nvu.edu.eg

Citation: Abukhodair, F.; Alsaggaf, W.; Jamal, A.T.; Abdel-Khalek, S.; Mansour, R.F. An Intelligent Metaheuristic Binary Pigeon Optimization-Based Feature Selection and Big Data Classification in a MapReduce Environment. *Mathematics* **2021**, *9*, 2627. https://doi.org/10.3390/math9202627

Academic Editor: Bogdan Oancea

Received: 23 September 2021
Accepted: 15 October 2021
Published: 18 October 2021

Abstract: Big Data are highly effective for systematically extracting and analyzing massive data. It can be useful to manage data proficiently over the conventional data handling approaches. Recently, several schemes have been developed for handling big datasets with several features. At the same time, feature selection (FS) methodologies intend to eliminate repetitive, noisy, and unwanted features that degrade the classifier results. Since conventional methods have failed to attain scalability under massive data, the design of new Big Data classification models is essential. In this aspect, this study focuses on the design of metaheuristic optimization based on big data classification in a MapReduce (MOBDC-MR) environment. The MOBDC-MR technique aims to choose optimal features and effectively classify big data. In addition, the MOBDC-MR technique involves the design of a binary pigeon optimization algorithm (BPOA)-based FS technique to reduce the complexity and increase the accuracy. Beetle antenna search (BAS) with long short-term memory (LSTM) model is employed for big data classification. The presented MOBDC-MR technique has been realized on Hadoop with the MapReduce programming model. The effective performance of the MOBDC-MR technique was validated using a benchmark dataset and the results were investigated under several measures. The MOBDC-MR technique demonstrated promising performance over the other existing techniques under different dimensions.

Keywords: big data; metaheuristics; feature selection; Hadoop; MapReduce; data classification

1. Introduction

The term Big Data (BD) represents a huge amount of information [1], which can be unstructured and structured. With regards to data processing, the great significance is the organization that employs this information [2]. Nowadays, it can be extensively employed for outperforming peers and can be determined as variety, volume, and velocity. Variety refers to the data being structured or unstructured, volume refers to the amount of data produced, and velocity refers to the speed at which the data are being generated at. The significant advantages of BD are cost and time savings, large data processing, forecasting and analysis, and efficacy because of an innovative tool support [3]. The adoption of data mining (DM) tools to solve BD problems might require the remodeling of the algorithm and its inclusion in parallel environments. Amongst the distinct alternatives is the MapReduce (MR) model [4] and its distributed file systems, first presented by Google, which provides a robust and effective architecture for addressing the analysis of big datasets. Currently, this method is considered in DM models, instead of other parallelization systems such as Message Passing Interface (MPI), due to its simplicity and fault tolerance.

Several studies have concentrated on the parallelization of machine learning (ML) tools with the MR method [5]. In recent times, more flexible and advanced tasks have emerged to expand the typical MR approaches, such as Apache Spark, effectively employed on several ML and DM challenges. Clustering and classification are the primary types of techniques in DM methods. Nevertheless, the performance of classification and clustering methods is greatly influenced by an increase in the dataset dimension, as algorithms in both categories operate on the dataset dimension. Moreover, the drawbacks of higher-dimension datasets include redundant information, degraded quality, and high model build time, which make the data analysis highly complex [6].

For resolving these problems, the election of the feature is employed as a fundamental preprocessing step to select a subset of features from a huge dataset. This increases the accurate clustering and classification methods, which trigger foreign, ambiguous, and noisy data reduction. The method of feature selection (FS) depends on a search strategy and an efficient calculation of subsets. The initial phase employs tools for picking subsets of features in the search approach. Later, the values of the subset attained from the search approach are estimated by classification systems [7]. The choice of feature should achieve two objectives: Eliminate/reduce the numbers of selected features and maximize the output precision performance. The elected feature is determined via several exploration approaches, such as depth, breadth, random, or a combination thereof. Nonetheless, an exhaustive search is not adequate for large datasets. When the feature size is d, the suitable subsets of a feature could hardly be selected from 2d alternatives.

Recently, an interesting proposal to apply the FS model to BD sets was proposed in [8]. In this work, the author described a method capable of effectively handling ultrahigh dimension data sets and choosing smaller subsets of interesting features from them. However, the numbers of elected features are considered to be numerous orders of magnitude less than the overall features, and the algorithms are developed to be performed in a single machine. Thus, this method is not scalable to randomly huge datasets. A specific manner of addressing high-dimensional features is by employing an evolutionary algorithm [9]. Generally, the group of features is encoded as a binary vector, in which every place defines whether the features are elected or not. This allows the implementation of FS models using the exploration abilities of evolutionary techniques. However, these techniques are lacking the scalability requirement for addressing big datasets (from millions of instances onward). Because of the properties of BD, the present FS method faces difficult problems in many stages, for example, tracing concept drifts, speed of data processing, and handling incomplete or noisy data. Therefore, exploring relevant FS methods for BD is of considerable urgency. Despite that the access method is highly accurate, how to extract useful data from BD based on analyzing and tackling them is still an open issue.

This study focuses on the design of metaheuristic optimization based on big data classification in a MapReduce (MOBDC-MR) environment. Moreover, the MOBDC-MR technique involves the design of a binary pigeon optimization algorithm (BPOA)-based FS technique for reducing the complexity and improving the accuracy. To classify Big Data, the beetle antenna search (BAS) with long short-term memory (LSTM) technique was applied. The presented MOBDC-MR technique was realized on Hadoop with the MR programming model. The experimental validation of the MOBDC-MR technique took place on the benchmark dataset, and the results were examined under several measures.

2. Existing Big Data Classification Approaches

This section performs a detailed review of existing FS-based Big Data classification models available in the literature. In Alweshah et al.'s study [10], a new optimization method, the monarch butterfly optimization (MBO) model, was executed by a wrapper FS technique, which employs K-nearest neighbor (KNN) classifiers. Research has been conducted on eighteen standard datasets. El-Hasnony et al. [11] presented a novel binary variant of the wrapper FS–grey wolf optimization (GWO) and particle swarm optimization (PSO). The KNN classifiers using Euclidean separation matric were employed for finding an optimum solution. A chaotic tent map helped in evading the approach from locked to optimal local problems. The sigmoid function was used to convert the search space from continual vectors to a binary one to be appropriate for the problems of the FS model, and cross-validation K-fold was employed for overcoming the over-fitting problem. BenSaid and Alimi [12] presented an online FS model that resolves these problems. The presented optimal feature selection (OFS) method, named MOANOFS, examines the current developments of online ML methods and a conflict resolution method (automatic negotiation) to improve the classification performance of the ultrahigh-dimension database.

In Al-Sarem et al.'s study [13], the ensemble method, CatBoost, RF, and XGBoost were employed for finding the primary feature for Parkinson's Disease (PD) prediction. The effects of this feature using distinct thresholds were examined for obtaining optimal performance for the PD prediction. The result showed that CatBoost methods attained an optimal result. Wang et al. [14] proposed a Big Data analytics (BDA) manner for the FS model for obtaining each explanatory factor of computation time (CT) to shed light on the fluctuation of CT. First, relative analysis was executed among two candidate factors using mutual data metrics for constructing an observed network. Next, the network deconvolutions were examined for inferring the direct dependencies among the candidate factors and the CT by eliminating the effect of transitive connections from the network.

Shehab et al. [15] presented a new hybrid FS cloud-based model for imbalanced data based on the KNN method. The presented method integrates the Euclidean and firefly distance metrics employed in the KNN. The experimental results showed a better insight into feature weights and time usage in comparison to the weighted nearest neighbor. Li et al. [16] integrated multimodal FS and grouped feature extraction and proposed a new faster hybrid reduction dimension technique, integrating their benefits of removing redundant and irrelevant data. First, the intrinsic dimension of the dataset was evaluated using the maximal probability model. The information Gain and Fisher score-based FS was employed as a multimodal method for removing inappropriate features. With the redundancy amongst the elected features as clustering conditions, they were grouped into a specific number of clusters.

Spencer et al. [17] assessed the efficiency of the model acquired with the ML technique through a related FS model. The four generally employed heart disease datasets were estimated by a Chi-squared test, PCA model, symmetrical uncertainty, and ReliefF to make distinct feature sets. Next, several classification methods were employed for creating methods that were later related to search an optimum features combination, to enhance the accurate predictions of the heart condition. Abdel-Basset et al. [18] introduced hybrid versions of the hybrid Harris Hawks optimization algorithm-based simulated annealing (HHOBSA) method to resolve the FS problems for classification purposes with the wrapper method. The two bitwise operations (OR and bitwise operations) could arbitrarily transmit the most informative feature from an optimal solution to others in the population to increase their quality. Simulated annealing (SA) boosts the performance of the HHOBSA model and helped to flee from the local optimal. A typical wrapper model KNN using Euclidean distance metric works as an evaluator for the novel solution.

In Mohammed et al.'s study [19], a set of hybrid and effective genetic algorithm (GA) models were presented for solving FS problems, while the processed data had a huge feature size. The presented algorithm employs a novel gene-weighted method, which can adoptively categorize the feature into weak or redundant features, unstable features, and strong relative features in the evolution method. According to this classification, the presented method provides a weak featureless priority and a strong feature high priority, while producing a novel candidate solution. Alarifi et al. [20] presented new BD and ML techniques to evaluate the SA process for overcoming these problems. The data are gathered from a large number of datasets, useful in efficient analyses of systems. The noise in the information is removed by pre-processing DM concepts. From the cleaned sentimental data, an efficient feature is elected by a greedy method, which selects an optimum feature handled by an optimum classifier named CSO-LSTMNN.

3. The Proposed Model

In this study, a new BD classification model using the MOBDC-MR technique was derived, which intends to choose optimal features and effectively classify BD. The MOBDC-MR technique follows a two-stage process, namely, BPOA-based FS and BAS-LSTM based classification. Furthermore, the presented MOBDC-MR technique was realized on Hadoop with the MR programming model. Figure 1 illustrates the overall working process of the proposed MOBDC-MR model. The detailed working of these modules is given in the following sub-sections.

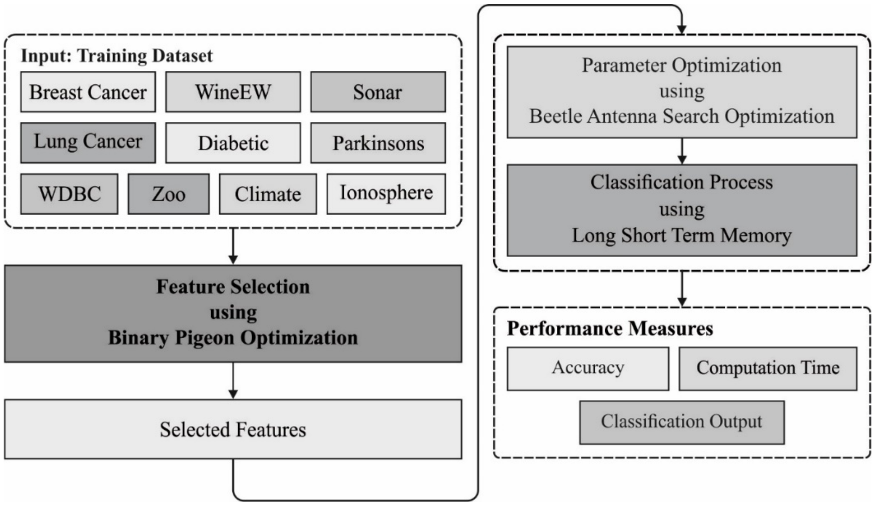

Figure 1. Overall process of the MOBDC-MR model.

3.1. MapReduce Programming Framework

Hadoop is a technique that meets the requirements of BD. It can be horizontally scalable and designed as a software model to process huge amounts of data. Hadoop is a public domain execution for Google MR; also, it depends on the programming MR framework. Hadoop frameworks handle the processing of details and permits the developer to focus easily on the application logic. Additionally, Hadoop is a familiar technique, i.e., employed to realize BD. Huge corporations such as Oracle, IBM, and Intel are also based on an extension of Hadoop support for its BD solution. MR and the Hadoop Distributed File System (HDFS) are the fundamental models of Hadoop.

YARN is the resource manager of Hadoop and is accountable for distributing the requested resources (memory and CPU) of a Hadoop cluster to several workloads. In this method, specific jobs could be allocated more or less resources that can be configured based on the user and application. HDFS is the primary element of Hadoop clusters. It is a Java-based DFS, which permits reliable and persistent storage and faster access to a huge amount of data. It splits the file to block and save them redundantly on the cluster, i.e., only perceived and slightly influenced by the users. When the files are saved in it, it is usually not noted, even when a single file is saved on many computers. It has been presented by Google as a way of resolving a class of terabyte or petabyte magnitude problems with huge clusters of inexpensive machinery [21]. Frameworks, alternative algorithms, and database management systems have been designed for resolving the dramatically increasing data and their model.

The MR model is employed for the parallel and distributed processing of huge amounts of unstructured and structured data, where Hadoop is usually stored in HDFS, clustered by a huge computer [21]. MR is a programming framework for expressing a distributed computation on a huge scale. MR is a method of splitting all requests into small requests, which are transmitted to several smaller servers for making a scalable usage of the CPU promising. Hence, scaling in small steps is potential (scale-out). The framework depends on two distinct phases for an application:

(i) Map—a primary transformation, as well as a recording phase, where single-input records can be treated in parallel.

(ii) Reduce—a consolidation/aggregation phase, where each interrelated record is treated in an individual entity.

The two major benefits are related to the consolidation phase: Logical block and map task. The main idea is that the input data could be divided into logical blocks. All of these blocks can be independently treated initially through map tasks. The result from this individually functioning block could be physically separated into distinct sets and later sorted. All of the sorted blocks are later transferred to the reduce task (RT). The RT: A map task can run-in some computed nodes on the cluster, and many map tasks can run in parallel, which could be accountable for transforming the input record to a value or key pair. The output from each map is separated and then sorted. However, there is single division for all RTs. The keys of all sorted divisions and the value related to the keys are treated in the RT. Many RTs can later run in parallel.

3.2. Feature Selection Using the BPOA-FS Technique

During the feature selection process, the BPOA-FS technique is executed on the input data and derives an optimal subset of features. Mainly, the POA consists of three operators: The landmark, map, and compass operators. In the map and compass operators, pigeons sense the geomagnetic fields to form a map for homing. Assume that the searching space is N dimensions, and the i-th pigeon of swarms can be denoted as N-dimensional vectors $X_i = (X_{i,1}, X_{i,2}, \dots, X_{i,N})$. The velocity of a pigeon, representing the changing location of the pigeon, can be denoted as other N-dimensional vectors $V_i = (V_{i,1}, V_{i,2}, \dots, V_{i,N})$. The formerly visited location of i-th pigeons are represented by $P_i = (P_{i,1}, P_{i,2}, \dots, P_{i,N})$. The global optimal location of the swarms are $= (g_1, g_2, \dots, g_N)$. All pigeons fly based on Equations (1) and (2):

$$V_i(t+1) = V_i(t) \times e^{-Rt} + rand \times \left(X_g - X_i(t)\right) \tag{1}$$

$$X_i(t+1) = X_i(t) + V_i(t+1), \tag{2}$$

where R indicates map and compass factors, in which $rand$ denotes an arbitrary value between 0 and 1, X_g denotes global optimal solutions, $X_i(t)$ denotes the present location of a pigeon at time t, and $V_i(t)$ denotes the present velocity of a pigeon at iteration t.

In the landmark operator, each pigeon is ranked based on their fitness values [22]. In all generations, the number of pigeons is upgraded by Equation (3), where only half of the pigeons are deliberated to evaluate the preferred location of the center pigeon, where every other pigeon alters their terminus, as follows:

$$N_p(t+1) = \frac{N_p(t)}{2}, \tag{3}$$

where N_p represents the number of pigeons in the present iteration t. The location of the preferred terminus is evaluated by Equation (4), where every other pigeon updates their location toward Equation (5).

$$X_c(t+1) = \frac{\sum X_i\,(t+1) \times Fitness(X_i(t+1))}{N_p \sum F\ itness(X_i(t+1))} \tag{4}$$

$$X_i(t+1) = X_i(t) + rand \times (X_c(t+1) \times X_i(t))) \tag{5}$$

where X_c represents the location of the center pigeon (preferred destinations). In order to implement the parameter election, a BPOA was presented. Unlike the typical POA, where the solution is upgraded in the searching space toward continuous valued position, in the BPOA, the search space is modeled as an n-dimensional Boolean lattice. Moreover, the solution is upgraded over the corner of a hypercube. Additionally, to solve the problem of whether to elect or not, a provided parameter and a binary solution vector are applied, in which 1 corresponds to a parameter being elected to comprise the novel datasets, and 0 corresponds to something else.

In binary algorithms, one uses the step vectors to evaluate the likelihood of changing the position, and the transfer function considerably impacts the balance between exploitation and exploration. In the FS method, when the size of feature vectors is N, the amount of distinct feature combination tends to be 2^N, i.e., a massive space for exhaustive searching. The proposed hybrid algorithms are employed for this purpose for searching the feature space vigorously, also generating the right combinations of features. The FS falls within multi-objective problems, since it needs to satisfy several objectives for getting an optimal solution, which minimizes the subsets of FS while simultaneously maximizing the accuracy of output to provided classifiers.

Based on the abovementioned, the fitness function (FF) to determine a solution under this condition was made to attain a balance among the two objectives as:

$$fitness\ = \alpha\Delta_R(D) + \beta\frac{|Y|}{|T|} \tag{6}$$

where $\Delta_R(D)$ is the classifier's error rate, $|Y|$ denotes the size of the subset that the technique selects, $|T|$ is the overall number of features contained in the present datasets, α denotes parameter $\in [0,1]$, related to the weight of error rate of classifications, and correspondingly, $\beta = 1 - \alpha$ represents the importance of reduction features. The classification performance permits an important weight instead of the number of selected features. When the estimation function only considers the classification accuracy, the effects would be neglect of the solution, which might contain similar accuracy; however, there would be less selected features, which serves as the main aspect in reducing the dimensionality problem.

3.3. Data Classification Using the BAS-LSTM Technique

Once the features have been chosen, they are fed into the LSTM model to allot proper class labels. LSTM extends the recurrent neural network (RNN) with memory cells, rather than recurrent units, for storing and outputting data, ease the learning of temporal relationships on a longer time scale. LSTM utilizes the idea of gating: The procedure that depends on elementwise multiplication of the input that determines the performance of all separate memory cells. LSTM updates its cell states, based on the activation of the gate. The input provided to LSTM is fed into distinct gates, which control the process, and implemented on the memory of cells: Write (input gate) and reset (forget gate)/read (output gate). The computations of the hidden values h_t of an LSTM cell are upgraded in each time step t. The vectorial depiction (vector representing each unit in a layer) of LSTM layers is given in the following equations:

$$i_t = \sigma_i(W_{ai}a_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{7}$$

$$f_t = \sigma\left(W_{af}a_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \tag{8}$$

$$c_t = f_t c_{t-1} + i_t\sigma_c(W_{ac}a_t + W_{hc}h_{t-1} + b_c) \tag{9}$$

$$o_t = \sigma_0(W_{ao}a_t + W_{ho}h_{t-1} + W_{co}c_t + b_0) \tag{10}$$

$$h_t = o_t\sigma_h(c_t) \tag{11}$$

where i, f, o, and c represent the input gate, forget gate, output gate, and cell activation vector, respectively, with similar sizes to vector h determining the hidden values; σ represents a nonlinear function; a_t represents the input to the memory cell layer at time t;

W_{ai}, W_{hi}, W_{ci}, W_{af}, W_{hf}, W_{cf} W_{ac}, W_{hc}, W_{ao}, W_{ho}, and W_{co} indicate the weight matrices, with the subscripts demonstrating the from–to relationship (W_{ai} indicates the input–input gate matrix, W_{hi} denotes the hidden-input gate matrix, etc.); b_j, b_f, b_c, and b_0 signify bias vectors. Layer representation was neglected for clarity network with LSTM cells, have achieved an outstanding performance when compared to regular recurrent units in speech detection, in which they provide an advanced result in phoneme detection [23]. Figure 2 depicts the architecture of the LSTM model.

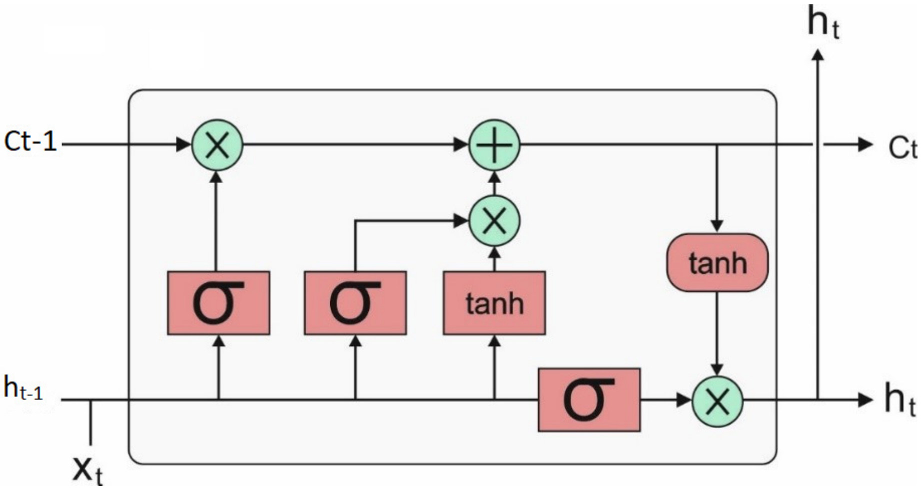

Figure 2. LSTM structure.

For improving the efficacy of the LSTM model, the hyperparameter tuning process is carried out by the use of BAS technique. This approach is one of the smart optimization methods that simulate beetles' foraging behaviors. Once a beetle starts foraging, it employs its right and left antennas for sensing the odor intensity of food. When the odor intensity obtained through the left antennas are larger, it flies to the left through the stronger odor intensity; otherwise, it flies to the right. The modeling procedure of BAS algorithms is given in the following:

$$\vec{b} = \frac{rands(Dim, 1)}{\|rands(Dim, 1)\|} \tag{12}$$

where Dim represents the spatial dimension. The space coordinates of a beetle's right and left sides and its antennae are made by:

$$\begin{cases} x_{rt} = x^t + d_0 * \vec{b}/2 \\ x_{lt} = x^t - d_0 * \vec{b}/2 \end{cases} \tag{13}$$

Amongst others, x^t represents the location of a beetle's antennae in the *t*-th iteration, x_{rt} represents the location of a beetle's right antennae in the *t*-th iteration, x_{lt} represents the location of a beetle's left antennae in the *t*-th iteration, and d_0 represents a beetle's two locations. Based on the elected FF, the corresponding fitness value of the right and left antennae are estimated, and beetles move toward the antennae using a smaller fitness number [24,25]. The positions of beetles are iteratively upgraded by:

$$x^{t+1} = x^t + \delta^t * \vec{b} * sign\big(f(x_{rt}) - f(x_{lt})\big) \tag{14}$$

Amongst others, δ^t denotes the step factor, sign indicates a sign function, and eta represents the variable step factor, which is generally 0.95.

4. Performance Validation

This section investigates the performance of the MOBDC-MR technique under different dimensions. The results were examined against ten benchmark datasets, and the results are given in Table 1. The results were inspected under several aspects and a detailed comparative results analysis was conducted in terms of different measures.

Table 1. Dataset Descriptions.

Dataset	Name	Features	Labels
Dataset-1	Breast cancer	9	2
Dataset-2	WineEW	13	3
Dataset-3	Zoo	16	7
Dataset-4	Sonar	60	2
Dataset-5	Lung cancer	23	2
Dataset-6	Diabetic	19	2
Dataset-7	Parkinson’s	22	2
Dataset-8	WDBC	30	2
Dataset-9	Climate	20	2
Dataset-10	Ionosphere	34	2

The results of the classification analysis of the MOBDC-MR technique on all the applied datasets are given in Table 2 and Figures 3 and 4. The results depict that the MOBDC-MR technique attained maximum accuracy after the FS process and chose only a minimum number of features on the applied datasets. The accuracy analysis was carried out using the MOBDC-MR technique with and without the FS process. The results demonstrate that the MOBDC-MR technique accomplished proficient results with higher accuracy after the FS process. For instance, the MOBDC-MR technique resulted effective in classification after the FS process by obtaining an accuracy of 97.98%, 92.88%, 92.98%, 86.25%, 89.15%, 66.11%, 92.86%, 98.05%, 92.75%, and 91.18% on the applied datasets 1–10 respectively. Similarly, the MOBDC-MR technique chose a minimal number of 3, 4, 6, 14, 6, 5, 4, 5, 4, and 5 features on the applied datasets 1–10, respectively.

Table 2. Results analysis of the proposed MOBDC-MR model.

Dataset	Accuracy		Time (s)	Total Features	Selected Features
	Before FS	After FS			
Dataset-1	96.83	97.98	6.25	9	3
Dataset-2	92.72	92.88	5.57	13	4
Dataset-3	89.27	92.98	4.96	16	6
Dataset-4	65.37	86.25	5.78	60	14
Dataset-5	83.20	89.15	4.14	23	6
Dataset-6	60.77	66.11	7.21	19	5
Dataset-7	78.01	92.86	5.59	22	4
Dataset-8	92.38	98.05	6.48	30	5
Dataset-9	88.84	92.75	6.56	20	4
Dataset-10	84.84	91.18	5.92	34	5

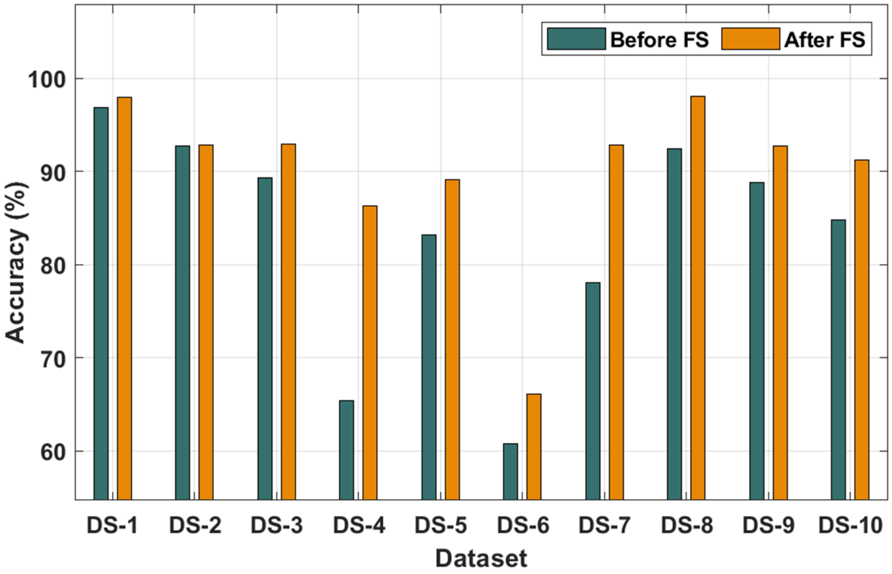

Figure 3. Result analysis of the MOBDC-MR model.

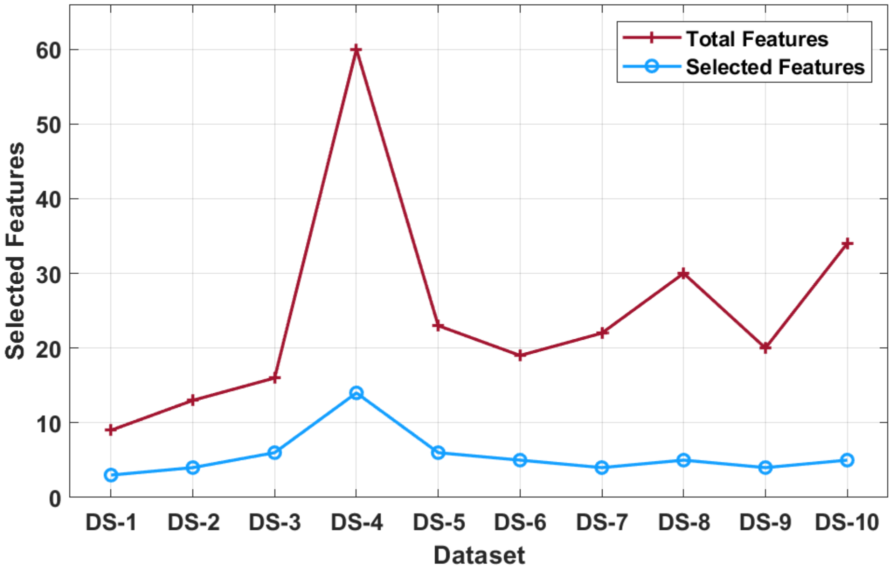

Figure 4. Selected feature analysis of the MOBDC-MR model.

Table 3 reports the comparative analysis of different FS techniques such as the social spider optimization (SSO) and pigeon-inspired optimization (PIO) algorithms in terms of the number of selected features. The results show that the BPOA-FS technique demonstrated an effectual outcome with the least number of selected features on all of the applied datasets.

Table 3. Comparison of the various feature selection methods.

Dataset	Total Features	SSO-FS	PIO-FS	BPOA-FS
Dataset-1	9	9	6	3
Dataset-2	13	12	8	4
Dataset-3	16	11	9	6
Dataset-4	60	25	18	14
Dataset-5	23	15	10	6

Dataset-6	19	14	8	5
Dataset-7	22	16	10	4
Dataset-8	30	18	12	5
Dataset-9	20	10	7	4
Dataset-10	34	18	12	5

Table 4 and Figure 5 portray the accuracy analysis of different FS techniques with LSTM as the classification model. The results demonstrate that the MOBDC-MR technique resulted in a superior classification performance over the other techniques. For instance, with dataset-1, the MOBDC-MR technique offered a maximum accuracy of 97.98%, whereas the SSO-FS and PIO-FS techniques obtained minimal accuracies of 96.78% and 96.97%, respectively. At the same time, with dataset-3, the MOBDC-MR approach obtained a maximal accuracy of 92.98%, whereas the SSO-FS and PIO-FS methods reached lesser accuracies of 90.66% and 91.42%, respectively. In line with dataset-6, the MOBDC-MR algorithm reached a higher accuracy of 66.11%, whereas the SSO-FS and PIO-FS techniques obtained lower accuracies of 65.28% and 65.69%, respectively. At last, with dataset-10, the MOBDC-MR system offered a maximum accuracy of 91.18%, whereas the SSO-FS and PIO-FS techniques gained reduced accuracies of 90.50% and 90.83%, respectively.

Table 4. Results of the various feature selection methods in terms of accuracy.

Dataset	Before FS	SSO-FS	PIO-FS	MOBDC-MR
Dataset-1	96.83	96.78	96.97	97.98
Dataset-2	92.72	92.36	92.38	92.88
Dataset-3	89.27	90.66	91.42	92.98
Dataset-4	65.37	87.33	89.10	86.25
Dataset-5	83.20	88.38	89.75	89.15
Dataset-6	60.77	65.28	65.69	66.11
Dataset-7	78.01	88.25	90.76	92.86
Dataset-8	92.38	97.18	97.69	98.05
Dataset-9	88.84	92.31	92.56	92.75
Dataset-10	84.84	90.50	90.83	91.18

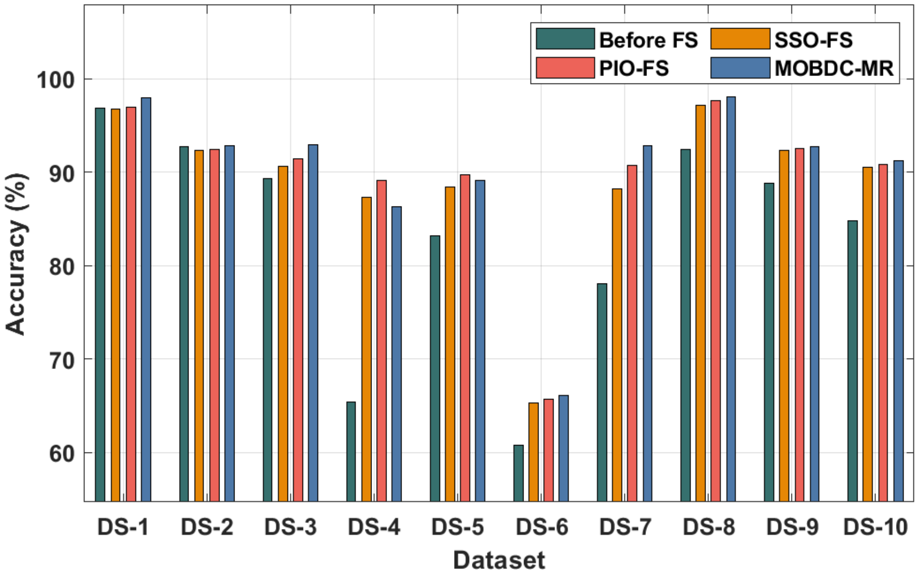

Figure 5. Accuracy analysis of the MOBDC-MR model.

The results of a brief computation time (CT) analysis of the MOBDC-MR technique with existing methods are displayed in Table 5 and Figure 6. The resultant values portray that the MOBDC-MR technique attained minimal CT on all of the tested datasets. For instance, with dataset-1, a lower CT of 5.25 s was required by the MOBDC-MR technique, whereas the SSO-FS, PIO-FS, and IFS-BA techniques needed higher CTs of 8.55 s, 8.42 s, and 7.93 s, respectively. Following this, with dataset-3, a lower CT of 4.96 s was essential for the MOBDC-MR methodology, whereas the SSO-FS, PIO-FS, and IFS-BA techniques required higher CTs of 7.05 s, 6.88 s, and 6.47 s, respectively.

Table 5. Results of the various methods in terms of computation time (s).

Dataset	SSO-FS	PIO-FS	IFS-BA	MOBDC-MR
Dataset-1	8.55	8.42	7.93	6.25
Dataset-2	8.17	7.95	7.35	5.57
Dataset-3	7.05	6.88	6.47	4.96
Dataset-4	8.56	8.32	7.82	5.78
Dataset-5	6.87	6.69	6.25	4.14
Dataset-6	9.36	9.26	8.66	7.21
Dataset-7	8.21	8.11	7.63	5.59
Dataset-8	8.44	8.20	7.70	6.48
Dataset-9	8.64	8.47	7.91	6.56
Dataset-10	8.79	8.61	8.04	5.92

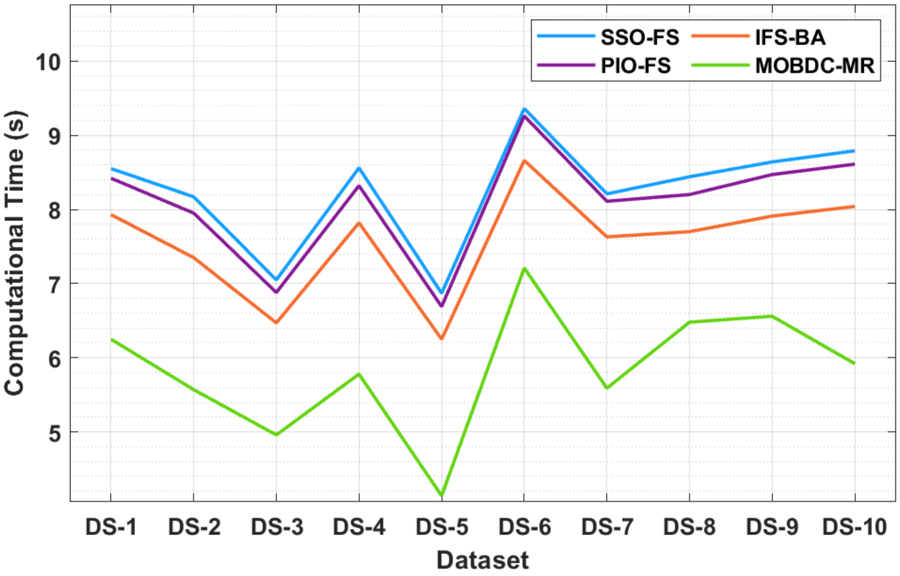

Figure 6. Computational time analysis of the MOBDC-MR model.

In addition, with dataset-5, a lower CT of 4.14 s was required by the MOBDC-MR technique, whereas the SSO-FS, PIO-FS, and IFS-BA approaches needed increased CTs of 6.87 s, 6.69 s, and 6.25 s, respectively. Eventually, with dataset-10, a lower CT of 5.92 s was required by the MOBDC-MR algorithm, whereas the SSO-FS, PIO-FS, and IFS-BA techniques required superior CTs of 8.79 s, 8.61 s, and 8.04 s, respectively.

In order to ensure better performance of the MOBDC-MR technique, a comprehensive comparative results analysis with the binary crow search algorithm (BCA), tent map-based grey wolf optimization (TMGWO), and improved feature selection for Big Data analytics (IFS-BA) was performed, as shown in Table 6 and Figure 7. The experimental results highlight that the MOBDC-MR technique accomplished effectual outcomes with maximum overall accuracy compared to the other techniques. For instance, with dataset-

1, the MOBDC-MR technique obtained an increased accuracy of 0.980, whereas the BCA, TMGWO, and IFS-BA techniques attained decreased accuracies of 0.958, 0.960, and 0.967, respectively. Similarly, with dataset-3, the MOBDC-MR approach gained a superior accuracy of 0.930, whereas the BCA, TMGWO, and IFS-BA algorithms attained decreased accuracies of 0.890, 0.910, and 0.895, respectively. Moreover, with dataset-5, the MOBDC-MR manner obtained an increased accuracy of 0.892, whereas the BCA, TMGWO, and IFS-BA approaches attained reduced accuracies of 0.870, 0.880, and 0.882. Lastly, with dataset-10, the MOBDC-MR technique attained an improved accuracy of 0.912, whereas the BCA, TMGWO, and IFS-BA methodologies gained lower accuracies of 0.900, 0.930, and 0.901, respectively.

Table 6. Overall accuracy analysis of MOBDC-MR with existing techniques.

Dataset	BCSA	TMGWO	IFS-BA	MOBDC-MR
Dataset-1	0.958	0.960	0.967	0.980
Dataset-2	0.935	0.947	0.924	0.929
Dataset-3	0.890	0.910	0.895	0.930
Dataset-4	0.670	0.740	0.855	0.863
Dataset-5	0.870	0.880	0.882	0.892
Dataset-6	0.690	0.710	0.651	0.661
Dataset-7	0.860	0.860	0.920	0.929
Dataset-8	0.940	0.940	0.970	0.981
Dataset-9	0.910	0.920	0.923	0.928
Dataset-10	0.900	0.930	0.901	0.912

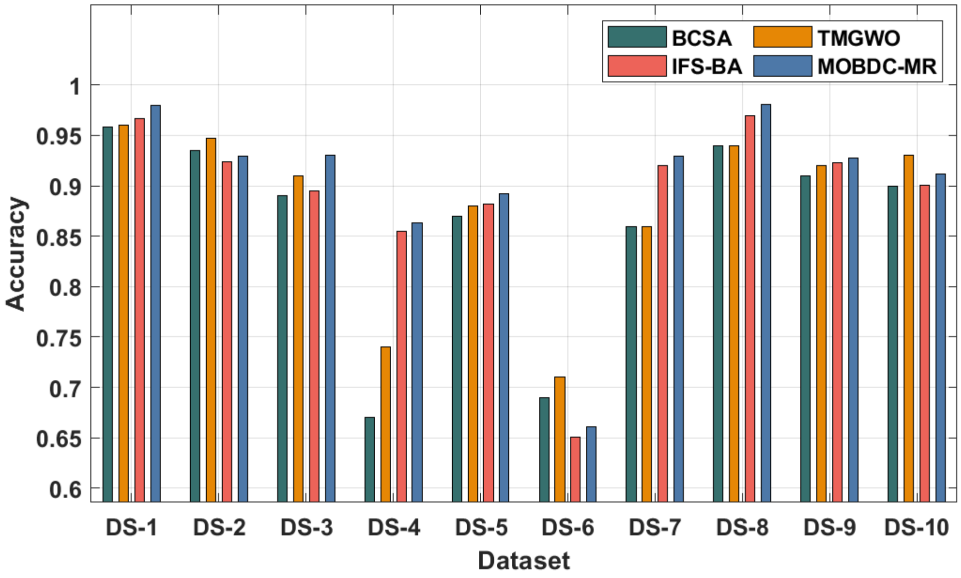

Figure 7. Overall accuracy analysis of the MOBDC-MR model.

By observing the detailed results analysis, it is apparent that the MOBDC-MR technique achieved accomplished maximum performance on the Big Data classification process compared to recent techniques.

5. Conclusions

In this study, the MOBDC-MR technique was designed to effectively classify the BD on MR environment. The MOBDC-MR technique follows a two-stage process, namely, BPOA-based FS and BAS-LSTM-based classification. The design of the BPOA-FS technique helped to considerably reduce the computation time and significantly increase the classification accuracy. Following this, hyperparameter tuning of the LSTM model using the BSA technique paved the way to accomplishing enhanced classifier results. The presented MOBDC-MR technique was realized on Hadoop with the MR programming model. The experimental validation of the MOBDC-MR technique took place on the benchmark dataset, and the results were examined under several measures. The simulation results demonstrated the promising performance of the MOBDC-MR technique over the other existing techniques under different dimensions. As part of future scope, BD clustering techniques with optimization algorithms can be designed to boost the overall classification accuracy further.

Author Contributions: F.A.: Conceptualization, methodology, software, and writing—original draft. W.A.: Conceptualization, methodology, software, and writing—original draft. A.T.J.: Conceptualization, writing—review and editing and project administration. S.A.-K.: Conceptualization, writing—review and editing, and project administration. R.F.M.: Investigation, validation, writing—reviewing and editing, and supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was funded by Institutional Fund Projects under grant no. (IFPHI-047-611-2020). Therefore, authors gratefully acknowledge technical and financial support from the Ministry of Education and King Abdulaziz University, Jeddah, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable to this article, as no datasets were generated during the current study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dubey, A.K.; Kumar, A.; Agrawal, R. An efficient ACO-PSO-based framework for data classification and preprocessing in big data. *Evol. Intell.* **2021**, *14*, 909–922.
2. Lozada, N.; Arias-Pérez, J.; Perdomo-Charry, G. Big data analytics capability and co-innovation: An empirical study. *Heliyon* **2019**, *5*, e02541.
3. Hashmi, M.R.; Tehrim, S.T.; Riaz, M.; Pamucar, D.; Cirovic, G. Spherical Linear Diophantine Fuzzy Soft Rough Sets with Multi-Criteria Decision Making. *Axioms* **2021**, *10*, 185.
4. Yaqoob, N.; Gulistan, M.; Kadry, S.; Wahab, H.A. Complex Intuitionistic Fuzzy Graphs with Application in Cellular Network Provider Companies. *Mathematics* **2019**, *7*, 35. https://doi.org/10.3390/math7010035.
5. Garg, H.; Riaz, M.; Khokhar, M.A.; Saba, M. Correlation Measures for Cubic m-Polar Fuzzy Sets with Applications. *Math. Probl. Eng.* **2021**, *2021*, 19. https://doi.org/10.1155/2021/9112586.
6. Dean, J.; Ghemawat, S. Map reduce: A flexible data processing tool. *Commun. ACM* **2010**, *53*, 72–77.
7. Minelli, M.; Chambers, M.; Dhiraj, A. *Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses*, 1st ed.; (Wiley CIO); Wiley: Hoboken, NJ, USA, 2013.
8. Marx, V. The big challenges of big data. *Nature* **2013**, *498*, 255–260.
9. Tan, M.; Tsang, I.W.; Wang, L. Towards ultrahigh dimensional feature selection for big data. *J. Mach. Learn. Res.* **2014**, *15*, 1371–1429.
10. de la Iglesia, B. Evolutionary computation for feature selection in classification problems. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2013**, *3*, 381–407.
11. Alweshah, M.; Al Khalaileh, S.; Gupta, B.B.; Almomani, A.; Hammouri, A.I.; Al-Betar, M.A. The monarch butterfly optimization algorithm for solving feature selection problems. *Neural Comput. Appl.* **2020**, 1–15. https://doi.org/10.1007/s00521-020-05210-0.
12. El-Hasnony, I.M.; Barakat, S.I.; Elhoseny, M.; Mostafa, R.R. Improved feature selection model for big data analytics. *IEEE Access* **2020**, *8*, 66989–67004.
13. BenSaid, F.; Alimi, A.M. Online feature selection system for big data classification based on multi-objective automated negotiation. *Pattern Recognit.* **2021**, *110*, 107629.
14. Al-Sarem, M.; Saeed, F.; Boulila, W.; Emara, A.H.; Al-Mohaimeed, M.; Errais, M. Feature Selection and Classification Using CatBoost Method for Improving the Performance of Predicting Parkinson's Disease. In *Advances on Smart and Soft Computing*; Springer: Singapore, Singapore, 2021; pp. 189–199.
15. Wang, J.; Zheng, P.; Zhang, J. Big data analytics for cycle time related feature selection in the semiconductor wafer fabrication system. *Comput. Ind. Eng.* **2020**, *143*, 106362.
16. Shehab, N.; Badawy, M.; Ali, H.A. Toward feature selection in big data preprocessing based on hybrid cloud-based model. *J. Supercomput.* **2021**, 1–40. https://doi.org/10.1007/s11227-021-03970-7.

17. Li, M.; Wang, H.; Yang, L.; Liang, Y.; Shang, Z.; Wan, H. Fast hybrid dimensionality reduction method for classification based on feature selection and grouped feature extraction. *Expert Syst. Appl.* **2020**, *150*, 113277.
18. Spencer, R.; Thabtah, F.; Abdelhamid, N.; Thompson, M. Exploring feature selection and classification methods for predicting heart disease. *Digit. Health* **2020**, *6*, 2055207620914777.
19. Abdel-Basset, M.; Ding, W.; El-Shahat, D. A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection. *Artif. Intell. Rev.* **2021**, *54*, 593–637.
20. Mohammed, T.A.; Bayat, O.; Uçan, O.N.; Alhayali, S. Hybrid efficient genetic algorithm for big data feature selection problems. *Found. Sci.* **2020**, *25*, 1009–1025.
21. Alarifi, A.; Tolba, A.; Al-Makhadmeh, Z.; Said, W. A big data approach to sentiment analysis using greedy feature selection with cat swarm optimization-based long short-term memory neural networks. *J. Supercomput.* **2020**, *76*, 4414–4429.
22. Mathiya, B.J.; Desai, V.L. Apache Hadoop Yarn MapReduce job classification based on cpu utilization and performance evaluation on multi-cluster heterogeneous environment. In *Proceedings of International Conference on ICT for Sustainable Development, New York, NY, USA, 21–22 September 2016*; Springer: Singapore, 2016; pp. 35–44.
23. Algamal, Z.Y.; Qasim, M.K.; Lee, M.H.; Ali, H.T.M. High-dimensional QSAR/QSPR classification modeling based on improving pigeon optimization algorithm. *Chemom. Intell. Lab. Syst.* **2020**, *206*, 104170.
24. Ordóñez, F.J.; Roggen, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* **2016**, *16*, 115.
25. Zhou, L.; Chen, K.; Dong, H.; Chi, S.; Chen, Z. An Improved Beetle Swarm Optimization Algorithm for the Intelligent Navigation Control of Autonomous Sailing Robots. *IEEE Access* **2020**, *9*, 5296–5311.