Journal of Big Data

RESEARCH

Open Access

Analysis of Bayesian optimization algorithms for big data classification based on Map Reduce framework

Chitrakant Banchhor* and N. Srinivasu

*Correspondence: banchhorchitrakant@gmail.com
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh 522502, India

Abstract

The process of big data handling refers to the efficient management of storage and processing of a very large volume of data. The data in a structured and unstructured format require a specific approach for overall handling. The classifiers analyzed in this paper are correlative naïve Bayes classifier (CNB), Cuckoo Grey wolf CNB (CGCNB), Fuzzy CNB (FCNB), and Holoentropy CNB (HCNB). These classifiers are based on the Bayesian principle and work accordingly. The CNB is developed by extending the standard naïve Bayes classifier with applied correlation among the attributes to become a dependent hypothesis. The cuckoo search and grey wolf optimization algorithms are integrated with the CNB classifier, and significant performance improvement is achieved. The resulting classifier is called a cuckoo grey wolf correlative naïve Bayes classifier (CGCNB). Also, the performance of the FCNB and HCNB classifiers are analyzed with CNB and CGCNB by considering accuracy, sensitivity, specificity, memory, and execution time.

Keywords: Map Reduce, Correlative naive Bayes classifier, Classification, Big data, Holoentropy

Introduction

The data size is increased because of the technological developments in the new data computing field. Big data refers to the large dataset unable to manage and handle the classical database systems [1]. Big data handling is a major challenge in every application due to rapid data gathering and storage, networking, and other related techniques [2, 3]. For example, big data classification is certainly needed because huge data is created, approximately 2.5 quintillion bytes of data every day [4]. Big data refers to the collection and processing of a huge amount of data [5]. The definition of big data is based upon the facts, including its size and the inability of processing by certain systems due to the high demand for primary storage and computing time [6, 7]. The extensive voluminous data processed in a year leads to the frequent use of the term big data in data analytics. This causes every organization to require data management options to handle big data [8, 9] effectively.

The characterization of big data is based on the factors such as volume, variety, and veracity associated with it. The enhanced difficulty level associated with big data

© The Author(s) 2021. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Springer Open

processing is because data arrives in a continuous pattern from the internet sources [10, 11]. Data analytics is considered one of the challenging research problems, and data mining and machine learning techniques are used to perform analytics. The inherent complexities associated with the data are its large size, and non-uniform makes the limitations on the current data mining software tools and techniques [12]. The standard data mining and machine learning methods are not directly dealing with extensive size data effectively [13, 14]. The analysis and knowledge extraction process related to big data can be further improved.

The two major categories, namely clustering, and classification are included in data mining schemes. The big data classification process is primarily [15, 16] influenced by the various classifiers, such as Naïve Bayes [17], Support Vector Machine [18], and Extreme Learning Machine [19]. The data classification approach provided by ELM [10] algorithm is based on multiclassification rather than binary classification [20]. The observed fact about big data processing is the increased computational complexity because of the high volume [21]. The analysis of big data in supervised classification is based on the learning algorithms, and after that, it finds the appropriate classes for the datasets[22].

The analytics include the extraction of useful insights from the extensive data, and one of the major tasks associated with it is classification [23, 24]. The deep study suggests the redesign of typical classification algorithms be adopted to classify very large amounts of data. The techniques are required to classify large amounts of data for different applications. The redesign of a classification technique for large amounts of data needs to be taken care of after that; the applications that use these algorithms can limit the growth of the extensive size of data[25].

The analysis and organization techniques applicable to traditional systems are insufficient in addressing big data-related issues and challenges [26, 27]. There are several algorithms developed for performing big data analysis tasks. The compatibility of MapReduce in handling big data processing makes it most suitable for analysis tasks [28].

The problem of processing large-size data is solved using the Map-Reduce scheme, which contains Mapper and Reducer tasks in parallel on datasets [29].

The Map-Reduce principle is derived from the divide and conquer strategy of problem-solving, in which sub-data samples are created and handled independently by splitting the data samples [30, 31]. The Map level divides the source data by producing dissimilar pairs of the key value. The efforts obtained from the key value in the map function are integrated by reducing functions at the Reduce level [32].

The number of smart data analytics approaches such as image processing, multitemporal processing, and automatic classification is increasingly in demand due to big data processing [33]. The data mining approaches integrated with the recent growing technologies are used for performing big data analytics with reduced limitations and drawbacks [34]. The MapReduce technique and distributed file system introduced by Google provides a suitable environment for processing large-scale datasets over a group of systems. The MapReduce framework is used to perform big data mining using several processors efficiently [12]. Generally, Hadoop, one of the systems, provides a parallel programming environment [35] used to execute the MapReduce framework.

The classification algorithms are employed to solve data mining issues in big data because of the fact that it is the gathering of data from different sources. The classification task is based on building a classifier model for resultant target class prediction of data items included in the dataset [36]. The emphasis on data classification motivates the adaptive use of techniques for classification in big data analysis. The classification techniques, including Bayes networks, genetic algorithms, genetic programming, and decision trees [37], are adopted to perform the classification of large data.

It is found in the literature that big data classification is performed using machine learning methods [38, 39], optimization algorithms [32, 40], Decision Tree [41], primarily along with some other approaches. The inclusion of fuzzy theory with correlative naïve Bayes classifier creates a model named the Fuzzy Correlative Naive Bayes Classifier (FCNB). The FCNB is developed for big data classification because it is implemented using the MapReduce framework. Further enhancement of the CNB classifier is done using the holoentropy function. The resultant model developed for big data classification using the MapReduce and named as Holoentropy based Correlative Naive Bayes Classifier (HCNB).

The main contribution of this paper is the analysis of big data classification techniques based on the Map-Reduce model using the classifiers, such as Correlative naïve Bayes classifier (CNB), Cuckoo Grey wolf CNB (CGCNB), Fuzzy CNB (FCNB), and Holoentropy CNB. (HCNB). The performance of the classifiers is evaluated based on accuracy, sensitivity, specificity, memory, and time. At first, CNB is compared with the existing naïve Bayes classifier. After that, further analysis shows the significant performance improvement of CGCNB in comparisons with NB and CNB. The other two developed models for big data classification named FCNB and HCNB are compared with Naïve Bayes [24], Correlative Naive Bayes (CNB) [20], Cuckoo Grey Wolf based CNB (CGCNB), and Fuzzy Naïve Bayes classifier (FNB) [24]. The classifiers are implemented in the JAVA programming language. The localization dataset and cover type dataset are taken from the UCI machine learning repository for experimentation.

This paper is organized into different sections: "Literature review" section discusses the Literature review covering the number of techniques developed using the MapReduce framework to perform classification on big data. "Descriptions of Bayesian classification methods" section describes the developed Bayesian classifiers, namely Correlative Naïve Bayes, Cuckoo Search Grey Wolves Correlative naïve Bayes, Fuzzy Correlative Naïve Bayes, and Holoentropy based correlative naïve Bayes classifiers. The developed classifiers' analysis is presented with comparisons based on the obtained results of accuracy, sensitivity, specificity, memory, and time is shown in "Results and discussion" section. Finally, the conclusion and future scope for improvements of the classifiers discussed in "Conclusion" section

Literature review

In this section, various algorithms for performing big data classification by different researchers and their importance in terms of advantages and disadvantages are presented.

Shen and Kai Gao [42] created a technique to be adopted for big data classification, and it works in the internet environment. The performance of this approach was stable, and also it can reduce the computation complexity. The approach has lower complexity than the traditional segmentation approaches, but one of the demerits of this approach is its strongest feature independence.

Simone Scardapane et al. [43] designed an algorithm for big data classification known as Echo State Networks based on utilizing the multipliers optimization procedure. The method performed the local exchanges among the nearby elements by utilizing the multiplier optimization procedure without connecting nodes. In addition to this, the communicating nodes did not require to use of training patterns. The synthetic datasets were used in experimentation and showed improvements in computation time and generalization accuracy metrics. In this method, pilot symbols are not required during communication. In contrast, the weights depend on error values is one of the drawbacks of this approach.

Jemal H. Abawajy et al. [44] implemented large, iterative multitier ensemble (LIME) classifiers and analyzed their suitability for big data classification. This classifier integrated varieties of meta-classifiers at lower levels and an iterative system developed at the higher level. The experimentation results showed that the developed classifier has improved performance compared with existing classifiers for big data. The computational cost was a major drawback of the developed ensemble classifier.

Xin et al. [45] integrated Map-Reduce framework and extreme learning machine to develop a classifier to work in a distributed environment. In the initial stage of this classifier, learning weights are computed through the multiplication of data matrices. Further stage proposed elastic ELM with Map Reduce technique. This approach's positive side was the rapid training speed and negligible human intervention because of its capability of efficient learning using dynamic processing massive training dataset.

Victoria López et al. [13] proposed an algorithm based on fuzzy rule-based classification named Chi-FRBCS-BigDataCS to deal with big imbalanced data. This algorithm dealt with the uncertainty in big data, with the strength of learning in the minimal class. This algorithm performs the fuzzy logic processes and MapReduce concept for design processes using the cost-sensitive learning approach. The performance of this algorithm is significantly improved in terms of classification accuracy and computation time.

Reshma C. Bhagat and Sachin S. Patil [46] performed a classification of imbalanced data and created a multi-classifier. At its first step, they utilized the techniques for conversion of the original datasets in the small group containing binary classes. In the next step, balanced classes were formed from the imbalanced binary class using the SMOTE algorithm. The classification was performed in the final step by utilization of the Random Forest classifier. The oversampling issues in this algorithm were handled using the MapReduce concept so that the developed method acquired the capability to handle large datasets. The fixing of dataset dependent oversampling rates was one of the problems with this algorithm.

Triguero et al. [47] used the evolutionary under-sampling method and developed a parallel model for handling big data and its related issues. Processes involved in the classification task were distributed to various cluster computing nodes due to the MapReduce technique for accomplishing classification work effectively. In addition, they also introduced a windowing approach for handling imbalanced class data, and it also increases the under-sampling speed. The advantage of this model was the scalability achievement during experimental studies. The abrupt change in computation time based on the imbalance ratio was the considerable drawback of this method.

Alessio Bechini et al. [48] developed a classifier by combining Map-Reduce concept with association rules and then named it MapReduce-based Associative Classifier (MRAC). This classifier utilized distributed classification method with the Map-Reduce model of programming and association rules. The method uses the FP-Growth algorithm and mined Classification Association Rules (CARs) and handled a distributed rule pruning classifying unlabeled patterns. The advantage of this method was related to speed up and scalability, but it failed when performed experiments on large datasets.

Shichao Zhang et al. [49] utilized the combination of clustering and classification approaches of K-Nearest Neighbors (KNN) and k-means clustering. The k-means clustering was used for dividing the entire dataset into various parts, and kNN classification was performed in each part of the cluster. The results of the experiments performed on medical images and big data showed between efficiency and accuracy. This approach had a drawback of selecting the value of k because of the degradation of classification accuracy when the value of k was increased.

The nature-inspired meta-heuristic algorithm developed by Seyedali Mirjalili et al. [50] which was based on the social and hunting behavior of grey wolves. The algorithm was adopted in computer science for the optimization process and named Grey Wolves Optimization algorithm (GWO). The grey wolves' leadership hierarchy and hunting technique were observed, and accordingly, the GWO algorithm was framed. The GWO algorithm has superior performance improvements when compared with other algorithms. The algorithms compared with GWO were Evolution Strategy (ES), Evolutionary Programming (EP), Differential Evolution (DE), Gravitational Search Algorithm (GSA), and Particle Swarm Optimization (PSO).

Cuckoo search is another meta-heuristic algorithm developed by Xin-She Yang and Suash Deb [51] to provide an approach towards the solution of optimization problems. The algorithm combines the cuckoo species' brood parasitic behavior [52] with the Levy flight behavior of some birds and fruit flies. The experimental results of this algorithm showed improved performance as compared with the genetic algorithm and particle swarm optimization techniques.

Simon Fong et al. [53] developed a mining method from streaming data while avoiding the computational challenges. The developed algorithm was based on lightweight feature selection approaches and used to avoid high dimensionality of the accepted data mining. This method was suitable for real-time applications and showed effectiveness in terms of high accuracy, low latency, and robustness for solving big data problems. S.Md. Mujeeb et al. [54] developed Big data classification based on MapReduce Framework for effective data management using the E-Bat algorithm. The performance is analyzed based on accuracy and Total Positive rate (TPR). The data management is better but, the accuracy obtained is not high. Shichao Zhang et al. [55] developed big data classification using kNN (k nearest neighbors). It is well known easy learning algorithm for real time applications and efficient algorithm. However, the selection of k value must be very small to get better accuracy. Such selection is not possible for all data sets. William et al. [56] developed multi-class imbalanced big data classification on Spark. In this, the clustering-based data partitioning is done to eliminate the big data classification problems.

The Random Forest and Naive Bayes classification method is used. They obtained better predictive power for the distributed environment. The major drawback of this method is its time complexity. Selvi et al. [57] developed Optimal Feature Selection for big data classification. The Firefly with Lion-Assisted Model is employed for the feature selection. The model obtains the effective classification, but it is not applicable for the noisy data. Mujeeb et al. [58] developed deep learning for big data classification. The Adaptive Exponential Bat algorithm is devised for training the classifier. They obtained better accuracy, TPR, and TNR. The security constraints are the major drawback of this method.

The major challenges in the existing techniques are computational complexity, time, cost, oversampling, and speed. These drawbacks were overcome by using the Bayesian classifiers. The CNB classifier is well suited for imbalanced datasets. By using the optimization algorithms, better convergence is obtained with improved accuracy with low computational complexity.

Descriptions of Bayesian classification methods

Naïve Bayes classifier (NB)

The different classifiers based on the Bayesian theorem are known as Bayesian classifiers; for example, NB is one of the bayesian classifiers. The Bayesian classification is based on the posterior probability calculation by assumed prior probabilities and the probability of different data object under given assumptions.

The NB classifier is based on the fact that each attribute of the object to be classified is independent of each other. The NB classifier is based on the approach where the probability of each category is calculated, and the object belongs to the category with the largest probability associated.

Correlative Naive Bayes classifier (CNB)

One of the highly utilized classifiers is NB classifier, and the typical classifier is adopted with the Map-Reduce framework and used for big data classification [59]. At the initial phase of the training process, the input data are arranged in different groups based on the number of classes.

$$\mathrm{CNB}._{\mathrm{QXm}} = \{\mu_{\mathrm{Q}\times\mathrm{m}}, \sigma_{\mathrm{Q}\times\mathrm{m}}, \mathrm{R}_{\mathrm{Q}\times 1}\} \tag{1}$$

where, $\mu_{Q\times m}$ is the mean to be calculated, $\sigma_{Q\times m}$ is specified as variance, $R_{Q\times 1}$ denotes correlation function, and it is illustrated in vector form. Testing data result is represented using the following equation:

$$\mathrm{C} = \mathrm{arg}^{\mathrm{Max}}_{\mathrm{q}=\mathrm{i}\ldots\mathrm{Q}}\left[\mathrm{P}\left(\mathrm{C}_{\mathrm{q}}\right) \times \mathrm{P}(\mathrm{X}\,|\mathrm{C}_{\mathrm{q}})\right] \times \mathrm{R}^{\mathrm{q}} \tag{2}$$

The Eq. (2) indicates that the highest posterior value is only selected as a consequential class.

Cuckoo Grey Wolf optimization with correlative Naïve Bayes classifier (CGCNB)

The integrated CGWO and CNB classifier with MapReduce framework is named as CGCNB-MRP [60]. CGWO is the integration of the cuckoo search algorithm and grey wolf Optimization (GWO). The block diagram of the developed model for big data classification is depicted in Fig. 1.

Fuzzy correlative Naive Bayes classifier (FCNB)

One more extension of CNB classifier is the inclusion of fuzzy theory and it is named as FCNB [61]. FCNB classifier is shown in the following equation:

$$\mu_q^s = \frac{\left|m_q^s\right|}{d} \tag{3}$$

The term μ_q^s represents membership degree of symbol sth in the qth element of the training model. The whole incidence of sth symbol in qth element is represented by the term $\left|m_q^s\right|$ and d is utilized for representing the data sample in the attribute. Each data sample is classified into the number of classes let it be called K. It is represented as follows:

$$\mu_c^k = \frac{\left|m^k\right|}{d}. \tag{4}$$

Here the whole incidence of Kth class in ground truth information is represented by $|m^k|$. The outcome of FCNB classifier is represented as below:

$$\mathrm{FCNB} = \{\mu_{\mathrm{q}}^{\mathrm{k}}, \mu_{\mathrm{c}}^{\mathrm{k}}, \mathrm{C}\}. \tag{5}$$

In the developed FCNB classifier's testing stage, the testing sample is classified into the appropriate classes by using the posterior probability of naïve Bayes, fuzzy membership degree, and a correlation factor among attributes. The output of FCNB is expressed as below equation:

$$G = \arg_{k=1toK}^{Max} P\left(\mathrm{g}_k|X\right) * C^k. \tag{6}$$

The term here is represented as $\mathrm{P(g_k \mid X)}$ denoted as a posterior probability by using test data **X** for given class $\mathbf{g_k}$. The expression $\mathbf{C^k}$signifies correlation for class **K**.

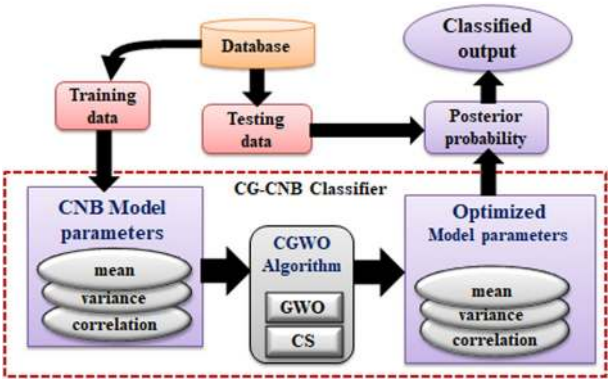

Fig. 1 Control and Data Flow of CGCNB-MRP classifier

Holoentropy using correlative naïve bayes classifier for a big data classification (HCNB)

The new classification technique called HCNB [62] is introduced by combining the existing CNB classifier with the holoentropy function. The handling is based on the holoentropy estimation for each attribute using the following formula:

$$H_v^b = \mathrm{F} \times \mathrm{T(ib)}. \tag{7}$$

Here, F represents the weight function, and $\mathrm{T(i_b)}$ is the entropy. The formulae for the weight function and entropy is described in the following equations.

$$\mathrm{F} = 2\left[1 - \frac{1}{1 + \exp(-\mathrm{T(i_b)})}\right]. \tag{8}$$

$$\mathrm{T(i_b)} = -\sum_{\mathrm{b=1}}^{\mathrm{M(i_b)}} \mathrm{P_b} \times \log \mathrm{P_b}\,. \tag{9}$$

Here, $\mathrm{M(i_b)}$ is the unique value of the attribute vector $\mathrm{i_b}$. The training phase of the HCNB based on the training data samples produces the result in the following vector form:

$$\mathrm{HCNB_{a\times s}} = \left\{\mathrm{\mu_{a\times s}, \sigma^2_{a\times s}, C_{a\times 1}, H_{a\times s}}\right\}. \tag{10}$$

Here, $\mathrm{\mu_{a\times s}}$ and $\mathrm{\sigma^2{}_{a\times s}}$ represent the computed mean value and computed variance value between the attributes ***a*** and ***s***, respectively. Also, the correlation is represented by $\mathrm{C_{ax1,}}$ and the holoentropy function is represented using $\mathrm{H_{a\times s}}$. The individual class is selected by estimating posterior probability independently during a testing phase, which the below expression can represent:

$$P(Y|k_v) = \prod_{b=1}^{s} P\left(Y = y_b|k_v\right) \tag{11}$$

where, y_b illustrates *y*th data of *b*th element, and k_v indicates *v*th class number. The block diagram of the HCNB classifier is depicted in Fig. 2.

Results and discussion

This section presents the classifiers' evaluation results, and a comparative detail analysis is provided with the existing methods. The system requirements and implementation details are provided in the experiment setup.

Experimental setup

The system configuration used for performing the experiment contains Windows 10 OS running on Intel processor CPU 2.16 GHz with 2 GB RAM capacity. The methods included in the developed classifiers are implemented in JAVA programming language. The parameters used for the experimentation are maximum iteration-5, population size-6, and mapper size-5.

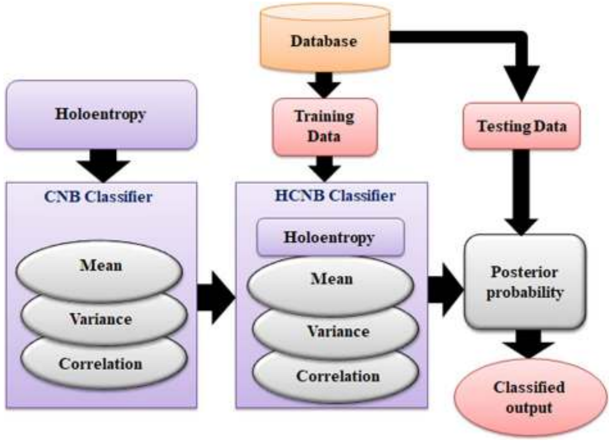

Fig. 2 Block diagram of HCNB classifier

Dataset description

The dataset utilized for the experimentation purpose is localization dataset and cover type dataset.

The localization dataset is taken from the UCI machine learning repository for experimentation [63]. The recorded activities of five people wearing tags, ankle left, ankle right, belt, and chest are collected in terms of the localization dataset. The dataset contains a total of 164,860 instances that include 8 attributes. Each instance in the dataset forms localization data for tags, and attributes are used to recognize them.

The cover type dataset is taken from the UCI machine learning repository for experimentation [64]. The dataset contains total of 581,012 instances with 54 attributes.

Metrics used for performance evaluation

Five metrics, such as accuracy, sensitivity, specificity, memory, and time, are used to evaluate the performance of the classifiers. The degree of veracity is measured using an accuracy metric defined as the proportion of true results. The sensitivity and specificity are referred to as the proportion of correctly identified true positives and true negatives.

$$Acc = \frac{T_N + T_P}{T_N + T_P + F_N + F_P} \tag{A}$$

$$Sens = \frac{T_P}{T_P + F_N} \tag{B}$$

$$Spec = \frac{T_N}{T_N + F_P} \tag{C}$$

where, T_N is True Negative, T_P is True Positive, F_N is False Negative, and F_P is False Positive.

Comparative analysis of Bayesian models

A comparative analysis is done to evaluate the developed classifiers with the existing models based on sensitivity, specificity, and accuracy.

In this paper, the first model C.N.B. is compared with the existing naïve Bayes classifier. After that, further analysis shows the significant performance improvement of CGCNB in comparisons with NB and CNB. The other two developed models for big data classification named FCNB and HCNB are compared with Naïve Bayes [24], Correlative Naive Bayes (CNB) [20], Cuckoo Grey Wolf based CNB (CGCNB), and Fuzzy Naïve Bayes classifier (FNB) [24].

The naïve Bayes classifier incorporating the fuzzy theory is named the Fuzzy Naïve Bayes classifier (FNB); the comparison of FCNB classifier shows enhanced performance achievement and adopted to perform the big data classification.

The holoentropy extension of the CNB classifier (HCNB) is created and compared with the existing models with similar conditions and parameters and observed improvements when its performance is assessed on the localization dataset and the cover type dataset. The comparative study of the models is done for training sample data taken from 75 to 90% for the number of number mappers as M = 5. The number of mappers represents the number of desktops used for simulation.

Analysis using localization dataset

Performance evaluation of different Bayesian classifiers

The developed classifiers CNB and CGCNB are evaluated based on accuracy, sensitivity, specificity, memory, and time analysis on the localization dataset. The performance evaluation is presented in this section. The performance evaluation process is carried out with a mapper size of 5 and on training data. Table 1 shows the analysis of NB, CNB, GWO + CNB, CGCNB, FCNB, and HCNB classifiers based on localization data.

Analysis based on training percentage

Consider Table 1, the accuracy of the NB classifier with 75% of training data is 76.3% while increasing the training percentage the accuracy of the classifier goes on increasing, similarly for all the matrices like sensitivity, specificity, memory and time the improved performance is achieved with an increase in training percentage. This improved performance with the increase in training percentage is achieved in all the classifiers.

Analysis based on mappers:

In Table 1, when considering the HCNB classifier for mapper size 2 with a training percentage of 75%, the performance matrices like accuracy, sensitivity, specificity, memory, and the execution time are 85, 94, 89.2, 166, and 1.99, respectively. Similarly, for mapper size 5 with a training percentage of 75%, the performance matrices like accuracy, sensitivity, specificity, memory, and the execution time are 85.3, 94.2, 89.3,

Table 1 Analysis of NB, CNB, GWO + CNB, CGCNB, FCNB, and HCNB based on localization data

Training data (%)	Mappers (M)	Accuracy (%)	Sensitivity (%)	Specificity (%)	Memory (MB)	Execution time (s)
NB						
75	2	76.4	80.4	72.4	39.7	2.87
	3	76.1	80.5	72	38.4	2.98
	4	76.1	80.2	72.2	37.5	2.91
	5	76.3	80.1	72.1	36.3	2.75
80	2	76.7	80.6	72.5	38.8	2.75
	3	76.5	80.5	72.1	37.3	2.85
	4	76.2	80.5	72.5	36.6	2.65
	5	76.5	80.3	72.1	36.6	2.68
85	2	76.8	80.7	72.7	36.5	2.64
	3	76.7	80.6	72.5	36.9	2.73
	4	76.2	80.7	72.5	34.7	2.41
	5	76.9	80.4	72.7	34.1	2.49
90	2	76.9	80.9	72.9	34.6	2.42
	3	76.8	80.8	72.8	35.5	2.58
	4	76.5	80.7	72.9	32.5	2.35
	5	76.9	80.9	72.7	31.1	2.37
CNB						
75	2	77.1	81	73	39.7	2.27
	3	77	81.3	73.2	35.6	2.32
	4	77.5	81.2	73.4	40.2	2.55
	5	77.1	81	73.4	37.6	2.47
80	2	77.5	81.3	73.2	39.4	2.12
	3	77.3	81.5	73.6	34.1	2.21
	4	77.6	81.5	73.6	38.5	2.32
	5	77.3	81	73.6	36.7	2.56
85	2	77.7	81.4	73.5	38.8	2.01
	3	77.4	81.7	73.8	33.5	2.16
	4	77.9	81.7	73.7	37.4	2.16
	5	77.6	81.4	73.7	35.5	2.26
90	2	77.8	81.7	73.9	35.8	1.99
	3	77.9	81.9	73.8	32.9	2.08
	4	77.9	81.9	73.9	35.3	1.95
	5	77.8	81.5	73.7	34.8	2.22
GWO + CNB						
75	2	79.1	82.5	75.1	62.7	2.73
	3	79.4	82.9	75.3	61.2	2.64
	4	79.2	82.9	75.2	60.6	2.88
	5	79.2	82.9	75.1	59.8	2.61
80	2	79.2	82.7	75.3	61.2	2.66
	3	79.5	82.9	75.5	59.6	2.56
	4	79.5	83	75.3	59.2	2.33
	5	79.5	82.9	75.1	58.8	2.49
85	2	79.4	82.9	75.4	59.6	2.48
	3	79.6	83	75.7	58.4	2.36
	4	79.7	83.2	75.7	57.9	2.21
	5	79.6	83	75.2	57.3	2.16

Table 1 (continued)

Training data (%)	Mappers (M)	Accuracy (%)	Sensitivity (%)	Specificity (%)	Memory (MB)	Execution time (s)
90	2	79.9	82.9	75.7	58.1	2.32
	3	79.7	83.3	75.7	57.7	2.11
	4	79.9	83.2	75.9	54.5	2.09
	5	79.8	83.4	75.9	55.9	2.02
CGCNB						
75	2	80	83.7	76.2	12.9	2.98
	3	80	83.5	76.3	13.3	2.87
	4	80.5	83.9	76.5	14.7	2.55
	5	80.1	83.6	76	13.7	2.63
80	2	80.4	83.9	76.3	12.2	2.81
	3	80.3	83.6	76.3	12.7	2.82
	4	80.5	84	76.6	13.9	2.45
	5	80.2	83.9	76.2	13	2.36
85	2	80.8	84	76.5	11.4	2.73
	3	80.3	84	76.7	11.8	2.73
	4	80.7	84.2	76.9	12.4	2.26
	5	80.4	84	76.7	12.3	2.27
90	2	80.9	84	76.6	10.2	2.58
	3	80.4	84.1	76.7	10.5	2.53
	4	80.9	84.4	76.9	10.9	2.17
	5	80.7	84.5	76.9	11.1	2.09
FCNB						
75	2	93.1	96.2	91.5	823	3.33
	3	93.4	96.3	91	812	3.45
	4	93.3	96.5	91.2	826	3.19
	5	93.5	96.4	91.3	815	3.06
80	2	93.2	96.1	91.3	836	3.08
	3	93.6	96.4	91.4	831	3.02
	4	93.4	96.6	91.2	819	2.98
	5	93.3	96.3	91.5	813	2.91
85	2	93.2	96.1	91.2	845	2.55
	3	93.2	96.4	91.3	833	2.41
	4	93.4	96.3	91.5	824	2.22
	5	93.5	96.7	91.5	809	2.05
90	2	93.1	96.7	91.7	812	1.53
	3	93.7	96.6	91.9	826	1.23
	4	93.8	97	91.7	814	1.11
	5	93.9	97.1	91.8	803	1.05
HCNB						
75	2	85	94	89.2	166	1.99
	3	85.1	94.3	89.3	158	1.90
	4	85.2	94.2	89.1	137	1.88
	5	85.3	94.2	89.3	122	1.81
80	2	85.4	94.1	89.4	133	1.87
	3	85.3	94.2	89	122	1.77
	4	85.3	94.3	89.3	119	1.71
	5	85.4	94.5	89.6	118	1.67

Table 1 (continued)

Training data (%)	Mappers (M)	Accuracy (%)	Sensitivity (%)	Specificity (%)	Memory (MB)	Execution time (s)
85	2	85.3	94.3	89.7	122	1.77
	3	85.5	94.6	89.5	118	1.63
	4	85.4	94.7	89.4	115	1.57
	5	85.6	94.9	89.6	113	1.52
90	2	85.6	95	89.8	136	1.36
	3	85.7	95.2	89.5	131	1.44
	4	85.7	94.7	89.6	113	1.24
	5	85.9	94.8	90	105	1.11

122, and 1.81. From the above analysis, we can interpret that the execution time and memory decrease with the increase in the mapper size. In this proposed work, the mapper size depicts the number of desktops used.

Analysis using cover type Dataset

Performance evaluation of different Bayesian classifiers

The performance evaluation using the cover type dataset is presented in this section. The developed classifiers are evaluated based on accuracy, sensitivity, specificity, memory, and time. The performance evaluation process is carried out by varying the number of mappers and the training data. The number of mappers used is 5, representing the number of desktops used for simulation of big data analysis, and the data sample for training varies from 75 to 90%. Table 2 shows the analysis of NB, CNB, GWO + CNB, CGCNB, FCNB, and HCNB classifiers based on cover type data.

Analysis based on training percentage

Consider Table 1, the sensitivity of the CNB classifier with 75% of training data is 75.8% while increasing the training percentage the sensitivity of the classifier goes on increasing, similarly for all the matrices like accuracy, specificity, memory and time the improved performance is achieved with an increase in training percentage. This improved performance with the increase in training percentage is achieved in all the classifiers.

Analysis based on mappers

In Table 1, when considering the FCNB classifier for mapper size 3 with a training percentage of 90%, the performance matrices like accuracy, sensitivity, specificity, memory, and the execution time are 79.8, 82.9, 73.7, 26.6, and 18.4, respectively. Similarly, for mapper size 5 with the training percentage of 90%, the performance matrices like accuracy, sensitivity, specificity, memory, and the execution time are 80.1, 83.4, 73.9, 25.5, and 16.8. From the above analysis, we can interpret that the execution time and memory decrease with the increase in the mapper size. In this proposed work, the mapper size depicts the number of desktops used.

Table 2 Analysis of CNB and CGCNB

Training data (%)	Mappers (M)	Accuracy (%)	Sensitivity (%)	Specificity (%)	Memory (MB)	Time (s)
NB						
75	2	68	69.4	61	39.8	38.4
	3	68.1	69.6	61.1	39	36.5
	4	68.2	69.8	61.3	38.2	32
	5	68.3	70.2	61.5	37	30.5
80	2	69.5	70.5	62.2	39.5	29.6
	3	69.7	70.7	62.3	38.2	29.2
	4	69.9	71.2	62.4	37.7	28.7
	5	70.1	71.4	62.5	36.8	28.4
85	2	71.6	71.6	62.6	33.3	27.8
	3	71.7	71.9	62.8	33.1	27
	4	71.8	72	62.9	32.6	26.2
	5	72.1	72.1	63.1	32.4	25.5
90	2	74.6	72.6	65.2	34.5	24.4
	3	74.8	72.8	65.7	33.1	23.8
	4	75	73.1	65.9	31.8	22.1
	5	75.2	73.3	66.1	30.1	21.2
CNB						
75	2	73.6	75.8	64.7	39.7	31.8
	3	73.7	75.9	64.8	38.8	30.8
	4	73.9	76.1	65	37.1	29.9
	5	74.1	76.2	65.2	36.2	29
80	2	74.8	77.1	65.3	36.8	29.9
	3	74.9	77.2	65.4	36.4	29.4
	4	75	77.4	65.5	36	29
	5	75.1	77.5	65.6	35.8	28.8
85	2	76.2	78.2	65.7	35.5	28.6
	3	76.3	78.3	65.8	34.4	27.9
	4	76.4	78.4	66	33.5	27.1
	5	76.6	78.5	66.1	32.2	26.6
90	2	77.5	79.6	68.2	32.1	26.8
	3	77.7	79.7	68.5	31.4	25.9
	4	77.8	79.8	68.7	30.9	25
	5	77.9	79.9	68.8	30.5	24.1
GWO + CNB						
75	2	74	76.6	66.6	37.4	29.4
	3	74.1	76.8	66.8	36.6	28.9
	4	74.2	76.9	67	36	28.6
	5	74.3	77.1	67.1	35.8	28.2
80	2	74.7	77.7	66.8	36.3	29.6
	3	74.9	77.9	66.9	35.7	28.4
	4	75	78.2	67.1	35.1	27.9
	5	75.1	78.5	67.2	34.4	27.7
85	2	75.7	77.8	67.9	35.5	28.8
	3	75.8	77.9	68.1	34.1	28.1
	4	75.9	78	68.2	33.2	27.4
	5	76.1	78.1	68.4	32.3	26.5

Table 2 (continued)

Training data (%)	Mappers (M)	Accuracy (%)	Sensitivity (%)	Specificity (%)	Memory (MB)	Time (s)
90	2	76.9	79.6	69.5	33.8	27.7
	3	77.1	79.7	69.6	33.7	26.5
	4	77.2	79.8	69.8	32.4	25.9
	5	77.4	79.9	69.9	31.2	25.5
CGCNB						
75	2	75.1	81.1	64.6	36.4	28.8
	3	75.2	81.3	64.8	35.8	28.2
	4	75.4	81.4	65	34.1	26.5
	5	75.5	81.5	65.1	32.7	25.5
80	2	75.7	81.5	64.8	34.4	27.2
	3	75.8	81.6	64.9	33.6	26.4
	4	76.1	81.7	65.1	32.1	25.4
	5	76.2	81.9	65.2	31.2	24.1
85	2	76.7	81.9	66.2	34.5	24.4
	3	76.8	82.1	66.4	33.2	23.8
	4	76.9	82.3	66.5	31.2	23
	5	77.1	82.4	66.7	30.6	22.2
90	2	78.7	82.1	66.8	31.2	23.8
	3	78.9	82.2	66.9	30.7	22.1
	4	79	82.3	67	30.7	21.7
	5	79.1	82.5	67.1	29.8	20.5
FCNB						
75	2	77.7	79.8	70.7	36.6	24.4
	3	77.8	79.9	70.9	35.4	23.4
	4	78	80.1	71.1	32.4	22.9
	5	78.2	80.2	71.2	31.7	22
80	2	78.5	80.5	71.8	33.3	24.3
	3	78.6	80.7	71.9	32.1	23.7
	4	78.7	80.9	72.1	31.7	22.3
	5	78.8	81.1	72.3	30.6	21.2
85	2	78.8	81.5	73.1	31.2	22.2
	3	78.9	81.7	73.3	30.5	21.3
	4	79	82	73.4	29.9	20.6
	5	79.1	82.1	73.5	29.6	19.8
90	2	79.6	82.7	73.6	27.8	19.9
	3	79.8	82.9	73.7	26.6	18.4
	4	79.9	83.2	73.8	26	17.1
	5	80.1	83.4	73.9	25.5	16.8
HCNB						
75	2	78.8	80.9	71.8	31.2	14.5
	3	78.9	81	71.9	30.7	13.8
	4	79.1	81.2	72	30	11.2
	5	79.2	81.4	72.2	29.5	10.5
80	2	79.7	81.7	72.5	31.4	13.3
	3	79.9	81.9	72.6	30.9	12.1
	4	80.1	82.1	72.7	29.8	11.5
	5	80.2	82.2	72.8	29.1	10.3

Table 2 (continued)

Training data (%)	Mappers (M)	Accuracy (%)	Sensitivity (%)	Specificity (%)	Memory (MB)	Time (s)
85	2	80.9	83	73.8	28.6	9.92
	3	81.2	83.2	73.9	27.9	9.83
	4	81.3	83.4	74	27.1	9.79
	5	81.4	83.5	74.1	26.8	9.76
90	2	81.8	83.7	74.6	26.7	9.55
	3	82	83.9	74.7	25.9	9.42
	4	82.1	84	74.9	25	9.26
	5	82.2	84.1	75.1	24.1	9.12

Comparative discussion

Tables 1 and 2 for all the classifiers, the increase in training percentage increases the system's overall performance in terms of accuracy, sensitivity, specificity, memory, and execution time. Likewise, while increasing the mapper size, the memory requirement and the execution time decrease. For the localization dataset, the FCNB classifier has improved performance accuracy, sensitivity, and specificity compared to other methods. Similarly, for the cover type dataset, the HCNB classifier has enhanced performance accuracy, sensitivity, and specificity compared to other techniques.

For both the datasets, CNB has improved performance compared to NB, because the highest posterior value is only selected as a consequential class. GWO + CNB is better than both NB and CNB because the GWO algorithm is used to train the CNB classifier. Similarly, CGCNB has improved performance compared to NB, CNB, and GWO + CNB. In CGCNB, the Cuckoo search algorithm is incorporated with GWO; hence the better result is obtained. Finally, both FCNB and HCNB have better performance. HCNB is well suited for big data classification.

Conclusion

This paper focused on big data classification based on different functions incorporated with Map-Reduce framework. The basic model is CNB classifier and later it is integrated with optimization algorithms, like cuckoo search and grey wolf optimization. The adoption of fuzzy theory with CNB classifier with membership degree of attributes included in the dataset provides performance achievements comparatively with CNB and CGCNB classifiers. The models, such as FCNB, HCNB, and CGCNB classifier, demonstrate the enhanced performance in localization and covertype databases from simulation outcomes. In the future, the performance of the classifiers will be analyzed using log loss and training loss.

Acknowledgements

I would like to express my very great appreciation to Dr. N. Srinivasu for his valuable and constructive suggestions during the planning and development of this research work. His willingness to give his time so generously has been very much appreciated. Also, I wish to thank my parents for their support and encouragement throughout my study.

Authors' contributions

All authors contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript. Both authors read and approved the final manuscript.

Funding

None.

Availability of data and materials
The data used for the analysis is taken from the localization dataset, available at, https://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity and cover type dataset link is available at, "dataset link: https://archive.ics.uci.edu/ml/datasets/Covertype".

Declarations

Ethics approval and consent to participate
This paper does not use any of the animals or humans for the authors' studies.

Consent for publication
None.

Competing interests
The authors declare that they have no competing interests.

Received: 17 December 2020 Accepted: 10 May 2021
Published online: 05 June 2021

References

1. Benabderrahmane S, Mellouli N, Lamolle M, Paroubek P. Smart4Job: a big data framework for intelligent job offers broadcasting using time series forecasting and semantic classification. Big Data Research. 2017;7:16–30.
2. Thanekar SA, Subrahmanyam K, Bagwan AB. Big data and MapReduce challenges, opportunities and trends. Int J Electr Comput Eng. 2016; 6(6): 2911–2919. https://doi.org/10.11591/ijece.v6i6.10555.
3. Raghav RS, Amudhavel J, Dhavachelvan P. A survey on tools used in big data platform. Adv Appl Math Sci. 2017;17(1):213–29.
4. Wu X, Zhu X, Wu GQ, Ding W. Data mining with big data. IEEE Trans Knowl Data Eng. 2014;26(1):97–107.
5. Marx V. The big challenges of big data. Nature. 2013;498(7453):255–60.
6. Minelli M, Chambers M, Dhiraj A. Big Data, big analytics: emerging business intelligence and analytic trends for today's businesses. 1st ed. New York: Wiley Publishing; 2013.
7. Pole G, Gera P. A recent study of emerging tools and technologies boosting big data analytics. 2016. https://doi.org/10.1007/978-981-10-0419-3_4.
8. Lin W, Wu Z, Lin L, Wen A, Li J. An Ensemble Random Forest Algorithm for Insurance Big Data Analysis. IEEE Access. 2017;5:16568–75.
9. Patil SS, Sonavane SP. Enriched Over_Sampling Techniques for Improving Classification of Imbalanced Big Data. In: Proceedings of IEEE Third International Conference on Big Data Computing Service and Applications (Big Data Service), San Francisco, CA, pp. 1–10, 2017.
10. Chen J, Chen H, Wan X, Zheng G. MR-ELM: a MapReduce-based framework for large-scale ELM training in big data era. Neural Comput Appl. 2016;27(1):101–10.
11. Radha K, Thirumala Rao B. Research issues and challenges of big data. Int J Contr Theory Appl. 2016;9(17):8437–44.
12. Tsai C-F, Lin W-C, Ke S-W. Big data mining with parallel computing: a comparison of distributed and MapReduce methodologies. J Syst Softw. 2016;122:83–92.
13. López V, del Río S. José Manuel Benítez, Francisco Herrera, "Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data." Fuzzy Sets Syst. 2015;258:5–38.
14. Sucharita V, Jyothi S, Rao PV. Comparison of machine learning algorithms for classification of penaeid prawn species. In: Paper presented at the Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom 2016, 1610–1613.
15. Haque A, Parker B, Khan L, Thuraisingham B. Evolving Big Data Stream Classification with MapReduce. In: Proceedings of IEEE 7th International Conference on Cloud Computing, Anchorage, AK, pp. 570–577, 2014.
16. Hegazy O, Safwat S, Bakry ME. A mapreduce fuzzy techniques of big data classification. In: Proceedings of the SAI Computing Conference (SAI), London, pp. 118–128, 2016.
17. Santafe G, Lozano JA, Larranaga P. Bayesian Model averaging of naive bayes for clustering. IEEE Trans Syst Man Cybern. 2006;36(5):1149–61.
18. Huang X, Shi L, Suykens JK. Support vector machine classifier with pinball loss. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 5, pp. 984–997, 2014.
19. Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and applications. Neurocomputing. 2006;70(1):489–501.
20. Duan M, Li K, Liao X, Li K. A parallel multiclassification algorithm for big data using an extreme learning machine. IEEE Trans Neural Netw Learn Syst. 2017;29(6):2337–51.
21. Arnaiz-González Á, González-Rogel A, Díez-Pastor JF, López-Nozal C. MR-DIS: democratic instance selection for big data by MapReduce. Progr Artif Intell. 2017;6(3):211–9.
22. Potharaju SP, Sreedevi M. Distributed feature selection (DFS) strategy for microarray gene expression data to improve the classification performance. Clin Epidemiol Global Health. 2019;7(2):171–6.
23. Segatori A, Marcelloni F, Pedrycz W. On distributed fuzzy decision trees for big data. IEEE Trans Fuzzy Syst. 2018;26(1):174–92.
24. Bechini A, Marcelloni F, Segatori A. A MapReduce solution for associative classification of big data. Inf Sci. 2016;332:33–55.

25. Manekar AK, Pradeepini G. Cloud based big data analytics a review. In: Paper presented at the Proceedings - 2015 International Conference on Computational Intelligence and Communication Networks, CICN 2015, 2016; 785–788. https://doi.org/10.1109/CICN.2015.160.
26. Hu H, Wen Y, Chua TS, Li X. Toward scalable systems for big data analytics: a technology tutorial. IEEE Access. 2014;2:652–87.
27. Bechini A, Marcelloni F, Segatori A. A MapReduce solution for associative classification of big data. Inform Sci. 2016;332:33–55.
28. Priyadarshini A, Agarwal S. A Map-Reduce based support vector machine for big data classification. Int J Database Theory Appl. 2015;8(5):77–98.
29. Bhukya R, Gyani J. Fuzzy associative classification algorithm based on MapReduce framework. In: Proceedings of the international conference on applied and theoretical computing and communication technology (iCATccT), Davangere, pp. 357–360, 2015.
30. Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. ACM Commun. 2008;51(1):107–13.
31. Elkano M, Galar M, Sanz J, Bustince H. CHI-BD: A Fuzzy Rule-Based Classification System for Big Data classification problems. Fuzzy Sets Syst. 2018;348:75–101.
32. Polepally V, Chatrapati KS. Dragonfly optimization and constraint measure-based load balancing in cloud computing. Cluster Comput. 2019;22(1):1099–111.
33. Cavallaro G, Riedel M, Richerzhagen M, Benediktsson JA, Plaza A. On understanding big data impacts in remotely sensed image classification using support vector machine methods. IEEE J Select Top Appl Earth Observ Remote Sens. 2015;8(10):4634–46.
34. Triguero I, Peralta D, Bacardit J, García S, Herrera F. MRPR: A MapReduce solution for prototype reduction in big data classification. Neurocomputing. 2015;150:331–45.
35. Dean J, Ghemawat S. Map reduce: a flexible data processing tool. Commun ACM. 2010;53(1):72–7.
36. Kamal MS, Parvin S, Ashour AS, Shi F, Dey N. De-Bruijn graph with MapReduce framework towards metagenomic data classification. Int J Inform Technol. 2017;9(1):59–75.
37. Arnaiz-González Á, González-Rogel A, Díez-Pastor JF, López-Nozal C. MR-DIS: democratic instance selection for big data by MapReduce. Progr Artif Intell. 2017;6(3):211–9.
38. Ratre A. Taylor series based compressive approach and Firefly support vector neural network for tracking and anomaly detection in crowded videos. J Eng Res. 2019;20(7):4.
39. Arul VH. An approach for speech enhancement using deep convolutional neural network. Multimedia Res. 2019;2(1):37–44.
40. More NS, Ingle RB. Energy-aware VM migration using dragonfly–crow optimization and support vector regression model in Cloud. Int J Model Simul Sci Comput. 2018;9(06):1850050.
41. Daga BS, Bhute AN. Predicting recurrence pattern in breast cancer Using Decision Tree. 2009.
42. Gao S, Gao K. Modelling on Classification and Retrieval Strategy in Map-Reduce Based IR System. In: proceedings of 2014 International Conference on Modelling, Identification and Control, Melbourne, Australia, December 3–5, 2014.
43. Scardapane S, Wang D, Panella M. A decentralized training algorithm for echo state networks in distributed big data applications. Neural Networks. 2016;1(78):65–74.
44. Abawajy JH, Kelarev A, Chowdhury M. Large iterative multitier ensemble classifiers for security of bigdata. IEEE Trans Emerg Top Comput. 2014;2(3):352–63.
45. Xin J, Wang Z, Luxuan Qu, Wang G. Elastic extreme learning machine for big data classification. Neurocomputing. 2015;149:464–71.
46. Bhagat RC, Patil SS. Enhanced SMOTE Algorithm for Classification of Imbalanced Big-Data usingRandom Forest. In: Proceedings of IEEE International on Advance Computing Conference (IACC), pp. 403–408, 2015.
47. Triguero I, Galar M, Vluymans S, Cornelis C, Bustince H, Herrera F, Saeys Y. Evolutionary undersampling for imbalanced big data classification. Evol Comput. 2009;17(3):275–306.
48. AlessioBechini FM. A MapReduce solution for associative classification of big data". Inf Sci. 2016;332:33–55.
49. Deng Z, Zhu X, Cheng D, Zong M, Zhang S. Efficient kNN classification algorithm for big data. Neurocomputing. 2016;195:143–8.
50. SeyedaliMirjalili SM, Mirjalili AL. Grey Wolf Optimizer. Adv Eng Softw. 2014;69:46–61.
51. Yang XS, Deb S. Cuckoo search via Levy flights. In: Proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), Coimbatore, IEEE Publications, USA, pp. 210–214. 2009.
52. Thirugnanasambandam K, Prakash S, Subramanian V, et al. Reinforced cuckoo search algorithm-based multimodal optimization. Appl Intell. 2019;49:2059–83. https://doi.org/10.1007/s10489-018-1355-3.
53. Fong S, Wong R, Vasilakos AV. Accelerated PSO swarm search feature selection for data stream mining big data. IEEE Trans Serv Comput. 2016;9(1):33–45.
54. Mujeeb SM, Sam RP, Madhavi K. Adaptive hybrid optimization enabled stack autoencoder-based MapReduce framework for big data classification. In: 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE) 2020, pp. 1–5.
55. Deng Z, Zhu X, Cheng D, Zong M, Zhang S. Efficient kNN classification algorithm for big data. Neurocomputing. 2016;26(195):143–8.
56. William C, Sleeman IV, and Bartosz Krawczyk. Multi-class imbalanced big data classification on Spark. Knowledge-Based Systems, 2020.
57. Selvi RS, Valarmathi ML. Optimal feature selection for big data classification: firefly with lion-assisted model. Big data. 2020;8(2):125–46.
58. Mujeeb SM, Sam RP, Madhavi K. Adaptive Exponential Bat algorithm and deep learning for big data classification. Sādhanā. 2021;46(1):1–5.
59. Chitrakant B, Srinivasu N. CNB-MRF: Adapting correlative naive bayes classifier and mapreduce framework for big data classification. Int Rev Comput Softw. 2016. https://doi.org/https://doi.org/10.15866/irecos.v11i11.10116

60. Chitrakant Banchhor N, Srinivasu, . Integrating Cuckoo search-Grey wolf optimization and Correlative Naive Bayes classifier with Map Reduce model for big data classification. Data Knowl Eng. 2020. https://doi.org/10.1016/j.datak.2019.101788.
61. ChitrakantBanchhor NS. FCNB: Fuzzy Correlative naive bayes classifier with mapreduce framework for big data classification. J Intell Syst. 2018. https://doi.org/10.1515/jisys-2018-0020.
62. ChitrakantBanchhor NS. Holoentropy based Correlative Naive Bayes classifier and MapReduce model for classifying the big data. Evol Intel. 2019. https://doi.org/10.1007/s12065-019-00276-9.
63. Localization dataset, https://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity.
64. 'CovertypeDataset', UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/Covertype. 2020.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen journal and benefit from:

► Convenient online submission
► Rigorous peer review
► Open access: articles freely available online
► High visibility within the field
► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com