

Opportunities and Challenges of Feature Selection Methods for High Dimensional Data: A Review



Siva Sankari Subbiah^{1*}, Jayakumar Chinnappan²

¹ Department of Information Technology, Kingston Engineering College, Anna University, Vellore 632059, India

² Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Pennalur 602117, India

Corresponding Author Email: sivasankari@kingston.ac.in

<https://doi.org/10.18280/isi.260107>

ABSTRACT

Received: 1 September 2020

Accepted: 22 December 2020

Keywords:

big data analysis, data mining, feature selection, machine learning, Hadoop, Spark, centralized processing, distributed parallel processing

Now a day, all the organizations collecting huge volume of data without knowing its usefulness. The fast development of Internet helps the organizations to capture data in many different formats through Internet of Things (IoT), social media and from other disparate sources. The dimension of the dataset increases day by day at an extraordinary rate resulting in large scale dataset with high dimensionality. The present paper reviews the opportunities and challenges of feature selection for processing the high dimensional data with reduced complexity and improved accuracy. In the modern big data world the feature selection has a significance in reducing the dimensionality and overfitting of the learning process. Many feature selection methods have been proposed by researchers for obtaining more relevant features especially from the big datasets that helps to provide accurate learning results without degradation in performance. This paper discusses the importance of feature selection, basic feature selection approaches, centralized and distributed big data processing using Hadoop and Spark, challenges of feature selection and provides the summary of the related research work done by various researchers. As a result, the big data analysis with the feature selection improves the accuracy of the learning.

1. INTRODUCTION

Big data is a large volume of complex dataset captured from many individual sources. Due to the fast technological development, all domains collecting the data rapidly. As a result, large scale data analytics has become essential for business enterprises to gain actionable insights from their increasing volumes of data. Internet giants like Facebook, Yahoo, Amazon and Google deal with large volume of user generated data in the form of photographs, audio files, video files, status messages and blog posts. These data should be analyzed for uncovering meaningful patterns about user behavior [1-3]. The big data has the significant characteristics like volume, velocity and variety. The volume represents size of the data that ranges from Zeta bytes to Peta bytes. It denotes enormous volume of data generated by machines, sensors, human interaction application programs on systems and sensors. For example, Walmart and Facebook handle billions of photos and millions of transactions respectively. In earlier days, the human genome decoding process taken 10 years of time but, now it takes only one week of time. The IBM generated 90% of its stored data in the last two years only [4]. The second characteristic represents a variety of big data which may range from unstructured to structured data. The structured data is stored in some formats like spread sheet, databases, etc., whereas the unstructured data is stored in the form of emails, photos, videos, PDFs, and audio [5]. The third important characteristic is velocity which represents the rate at which the data is collected and processed. The velocity of big data ranges from batch data to streaming data, i.e., the data is collected in the particular interval and processed as a batch and

the data can be collected as it comes and processed immediately as a stream before collecting the next data [6]. The continuous increment of the increase of dimension increases the complexity and the time of the analysis performed on that dataset. The available machine learning algorithms become insufficient for extracting meaningful information from big data due to the existence of incompleteness, noisy and redundancy. The processing of big data requires the new analysis and processing tools [1].

The size of the data needs to be reduced to reduce the complexity of the computation and also processing overhead [7]. The dimensionality reduction can be done by using two popular methods namely, feature selection and feature extraction [8, 9]. The feature extraction extracts new features from old features by transforming available features into the reduced number of new features. The feature selection method identifies relevant features and removes an irrelevant features from the given list of features [10]. In case of big datasets, the feature selection serves as an important pre-processing technique for reducing the dimensionality. The implementation of feature selection uses two important components, namely, search strategy and an objective function. The search strategy selects the candidate subsets from the original training dataset which has the complete feature set and an objective function for evaluating these candidates. As a result, it produces the measurement of goodness which will be used by the search strategy for selecting new candidates [11]. The feature selection is an important pre-processing technique for big data which becomes an indispensable process for the machine learning to speed up and improve the accuracy of the prediction process [12].

The comparison of the number of datasets with more than 100 features in UCI (University of California, Irvine) repository is shown in Figure 1. The analysis of datasets from the UCI Repository for the period 2008 to 2017 as a sample shows that the number of datasets coming with large number of features increase year by year. So, in present and future, there is the need for the feature selection methodologies for reducing the complexities of machine learning with huge number of features.

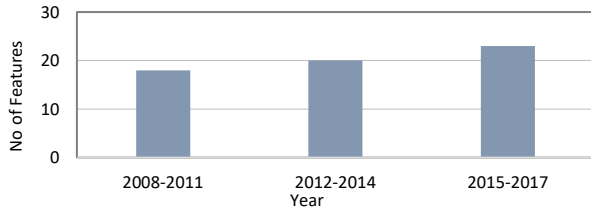


Figure 1. Comparison of number of datasets with more than 100 features-UCI repository

The rest of the paper is organized as follows: Section 2 discusses the significance of the feature selection and the types of feature selection methods. Section 3 discusses the centralized, distributed and distributed parallel big data analysis using Hadoop and Spark environments. It also describes the possibilities of doing feature selection by keeping the data in centralized and distributed databases. Section 4 explores the results of the review. Finally, the paper concludes in section 5.

2. FEATURE SELECTION METHODOLOGIES

The feature selection is broadly categorized into three types namely filter feature selection, wrapper feature selection and hybrid feature selection based on the ability of combining feature selection with model building.

2.1 Filter feature selection

The filter feature selection selects the relevant features without considering any model. It considers only the statistical relationship between each feature and target feature [13]. Figure 2 shows the working of the filter feature selection.

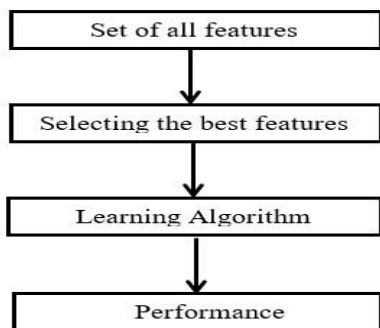


Figure 2. Filter feature selection

The filter feature selection selects the highly significant features from the list of original features by considering only the statistical characteristics of the features. After that, the dataset with the selected features are utilized as the input for the learning model and produces the output. It helps to

improve the accuracy of the classification and prediction process. Hence, it also help to reduce the computational time and overfitting. As it considers only the statistical relationship between the feature and target, it is faster compared to wrapper methods. It is suitable for reducing dimensionality of the high dimensional data [14, 10].

Some of the advantages and limitations of filter methods observed from the literature are as follows. The FCBF (Fast Correlation Based Filter) algorithm reduces the dimensionality of dataset dramatically for Microarray data [15]. The correlation-based feature selection handles irrelevant and redundant features effectively and works well only for smaller datasets [16, 17]. The interact algorithm is used for improving the classification accuracy. But it decreases the mining performance while the dimensionality of the dataset increases [18]. Wang [14] utilized the conditional dynamic mutual information technique to perform feature selection and achieved better performance. But it is much sensitive to noise. Quinbao Song et al. [15] have designed a system that uses Fast clustering-based feature selection algorithm to identify the most relevant features from the huge volume of high dimensional dataset. It is a graph theoretic clustering method that clusters the big dataset first and then chooses the best features from the clusters. It works well for the high dimensional microarray data and effectively reduces the dimensionality [15]. Zhu and Yang [19] developed a framework in which the significant features are identified by using cluster based sequential feature selection algorithm. The clusters are formed by using an affinity propagation clustering and the sequential feature selection was applied to each cluster for obtaining best subsets of features.

The Relief algorithm, not as interact, is scalable for high dimensional dataset. But it cannot identify the redundant features for elimination. The ReliefF algorithm is the advanced version of Relief algorithm which can effectively handle noisy, incomplete and multiclass datasets. It works well for domains with strong interdependencies between features but, it is myopic and cannot detect conditional dependencies among the features in regression problems. Another algorithm in the Relief family of algorithms is the RReliefF algorithm. It is the advanced algorithm of ReliefF algorithm which overcomes the drawbacks of ReliefF algorithm. It supports non-myopic discretization of numeric features and also best to discretize features in regression problems. The RReliefF algorithm requires high computational cost when compared to distance functions [20-22]. Kira and Rendel [23] developed a Relief algorithm with relevance evaluation for finding the best set of features and concluded that the technique is scalable for high dimensional dataset.

2.2 Wrapper feature selection

The wrapper feature selection selects the best subset of features by eliminating the irrelevant and redundant features using the learning algorithm as the feature evaluator. The working of wrapper feature selection is shown in Figure 3.

First, it takes all features as input and then, it generates the best subset of features by considering the learning algorithm. It improves the performance by reducing the over fitting [24]. Hence, when the number of features is large, it takes much computation time. Some of the advantages of wrapper method as seen from literature are as follows. An evolutionary based wrapper method uses K-means clustering for forming clusters that covers a large number of feature combinations, but it

decreases the cluster quality with the increasing number of features [25]. Even though, an affinity propagation sequential feature selection algorithm produces low accuracy compared to sequential feature selection, it is faster than Sequential Feature Selection [1, 2, 26]. The recursive feature elimination (RFE) and the Boruta feature selection are utilized by many researchers as an important wrapper methodology for selecting the best subset of features.

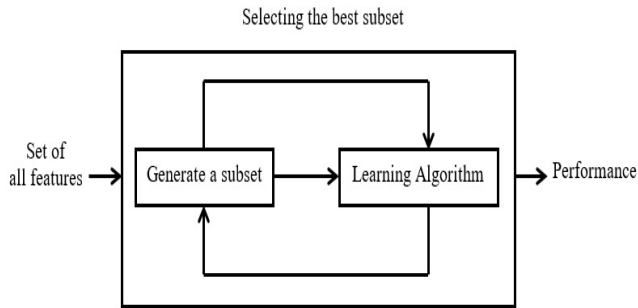


Figure 3. Wrapper feature selection

2.3 Hybrid feature selection

The hybrid feature selection combines the advantages of both filter and wrapper feature selection. It considers the learning model for selecting the best subset of features. But, the criteria for good selection should be known in advance [27]. It handles redundant features, irrelevant features and improves the accuracy by reducing the overfitting [28]. Filter methods pick up the internal characteristics of features by using statistical tests as an evolution criteria instead of using cross validation. In contrast, the wrapper methods measure performance of features based on the learning algorithm and the usefulness of features. Hence, the wrapper methods produce best subset of features than filter methods by considering biases. But, they are computationally more expensive due to the practice of repeated learning steps and the involvement of cross-validation measures. A hybrid method is quite similar to wrapper method and used for optimizing the objective function [2].

Wang [28] designed a hybrid system that selects the best features from the dataset consists of thousands of features. It uses the combination of mutual information & wrapper based feature subset selection method and improves the prediction accuracy. Tsai et al. [29] developed a system for performing parallel and distributed computing on big data and compared the MapReduce based methodologies with distributed methodologies. The result shows that the MapReduce based procedure is very stable and takes less computational cost. Moran-Fernandez et al. [30] designed a centralized and distributed feature selection in which the horizontal and vertical partitioning are applied first on big datasets. Then, the datasets are processed in a centralized and a distributed manner. For finding the best feature subsets five feature selections such as correlation based feature selection (CFS), interact (INT), consistency-based filter (Cons), information gain (IG) and ReliefF are utilized. Finally, the centralized and distributed approaches are compared and the result shows that the distributed approach achieves improved accuracy and less runtime than centralized approach. Wang et al. [31] introduced an efficient feature selection for hybrid data in which the given dataset is decomposed on the basis of variance and sample size. The Naive-Bayes and decision tree classifiers were applied on

the decomposed dataset. The results of each decomposed dataset were combined for achieving an increased accuracy and reducing the computation time [32].

Hodge et al. [33] performed a feature selection in parallel and distributed manner using mutual information, CFS, gain ratio, chi-square and odds ratio. It reduces redundancies, removes the noise and improves the accuracy of machine learning algorithm. Bolon Canado et al. [1] discussed challenges available in the feature selection and analyzed the various types of feature selection. To reduce the size of big dataset ReliefF, chi-squared test, IG, INT and Cons feature selection are used and the performance of these feature selection are compared. As a result, ReliefF works well for noisy and incomplete big datasets. Rahman et al. [34] discussed machine learning for mining big data. An efficient forecasting system for electricity generation was developed using MapReduce based approach under Hadoop environment for enhancing the scalability and achieving high performance [35]. The back propagation neural network (BPNN) was utilized for forecasting the electricity generation.

Gandomi and Haider [36] discussed the big data concepts and methods for processing big data. Hashem et al. [37] suggested that the big dataset can be efficiently handled by using cloud and analyzed in a cloud environment which gives an efficient data management, reduction in computation time and achieve scalable framework. Deisy et al. [38] highlighted a feature selection as an enhancement of traditional information theoretic algorithm called Information Theoretic Interact Algorithm (IT-IN) and compared against CFS, Relief and fast correlation based feature selection. It showed that IT-IN requires minimum computation time and improves accuracy.

Ma [39] developed the short term load forecasting model in which the Support Vector Machine (SVM) was developed and its parameters were tuned by utilizing Particle Swarm Optimization (PSO). It produces accurate result. But, it is difficult to define parameters. The aim of feature selection technique is to reduce the dimensionality of dataset by finding and selecting only the relevant features which should in turn help for improving the accuracy of machine learning algorithm [9, 40, 41]. Now a days, lot of feature selection algorithms available to achieve this goal, but there is not a single feature selection algorithm available for all types of datasets and in all situations. The best machine learning algorithm depends on the given dataset. The input dataset should be in good quality to achieve good accuracy of machine learning algorithm. Hence, not all the feature selection algorithms are suitable for all machine learning algorithms [42]. All datasets do not have same category of instances. So, first identify the dataset belongs to which category and then find which feature selection algorithm is suitable for that type of dataset. The feature selection techniques identified best are related to the popular classification algorithms like Relief, information gain, gain-ratio, voting and Linear Forward Selection (LFS) [43].

3. BIG DATA ANALYSIS

The organizations generate big data continuously in different forms from different data sources. But, it faces challenges during the analysis of big data. In the traditional processing, the data is stored in data warehouse, where it resides in structural form. But, now a day, huge volume of semi-structured and unstructured data are generated indirectly

by websites and many other sources. So, the data collected in different forms like unstructured and semi-structured should be preprocessed before performing the analysis [44]. The following section discusses the centralized and distributed analysis of big data.

3.1 Centralized approach

The analysis of big data in centralized manner is a complex task. There are two ways of achieving centralized processing. First, the powerful high-end machines can be introduced to process high volume of big data. Second, instead of a single powerful server, more number of servers can be utilized for processing big data. Whenever the number of servers increases, the access to the shared database also increases, as per Amdahl's Law. The law says that the number of data access by number of server reduce the efficiency of an individual server. All these scales out and scale in approaches for the relational database becomes an inexpensive, time consuming and does not satisfy the real time demands [11]. While the dataset size is small, the centralized data processing becomes an effective approach.

3.2 Distributed approach

In the distributed approach big datasets are divided into number of smaller datasets. The analytical processing is done separately on each dataset and then combines the analytical results obtained from these smaller datasets. Based on the samples and features, there are two types of partitioning methods, namely, horizontal partitioning and vertical partitioning. In the horizontal partitioning, the given dataset is partitioned into several smaller datasets in the horizontal manner. The partitioned datasets have the same set of features but they vary in the number of instances. So, the horizontal partition is applied to the datasets that have huge number of instances and less numbers of attributes [45]. Hence, the horizontal partitioning is more preferable in applications where the reduction of runtime and storage requirements are more important than accuracy. In the vertical partitioning, the given data set is partitioned into several smaller datasets in the vertical manner. The partitioned datasets have the same set of instances but they vary in the number of features. So, the vertical partition is applied to the datasets that have high dimensionality and less numbers of samples. Hence, in vertical partitioning, the features were distributed across the smaller datasets and the smaller datasets may have redundant features. The vertical partitioning is more preferable for applications where the accuracy is more important than the reduction of runtime and storage requirements [30].

3.2.1 Hadoop

In the distributed parallel processing, the data is distributed at many nodes and operated in parallel. The number of nodes should be increased in linear scale with the size of dataset. In order to achieve fault tolerance, the data is replicated at more than one node and if any updation to any data then the data should be propagated to each replica to maintain consistency. The distributed parallel processing is implemented easily using the powerful parallel processing tool named Hadoop. It is an open source environment that provides the MapReduce paradigm for big data processing. The Hadoop environment includes Hadoop Distributed File System (HDFS), YARN (Yet Another Resource Negotiator) and MapReduce for

achieving high throughput during data access, for scheduling and clustering resource management and for processing the big datasets respectively. The HDFS follows the master-slave principle that uses Namenode as master and datanode as slave. The namenode partitions the dataset into number of smaller datasets and then, assigns them to each slave nodes. The HDFS processing model is shown in Figure 4. The two functions map and reduce articulates the MapReduce model. It performs the map function first by taking data from the file systems and transforming it into the intermediate key/value pairs. Then the reduce function reduces these intermediate key/value pairs and produces the final output. The Hadoop MapReduce follows the divide and conquer technique called data partitioning for scaling well the large volume of data.

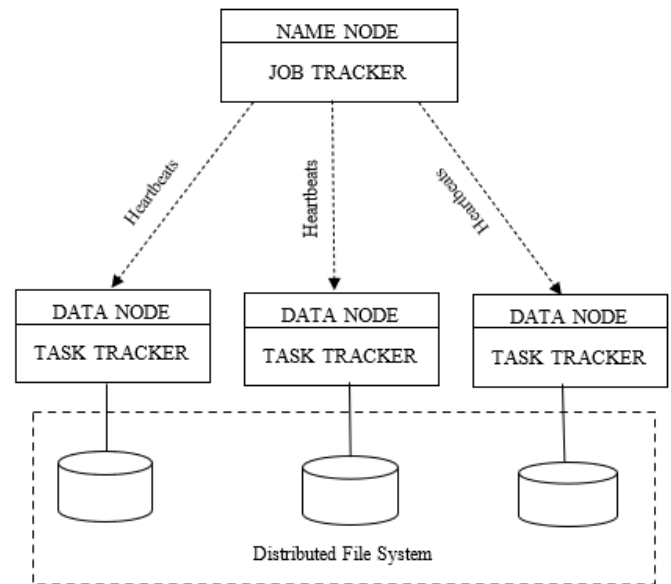


Figure 4. Hadoop distributed file system processing model

The working of Hadoop MapReduce is illustrated in Figure 5. The HDFS distributes the data among the cluster of nodes. In the cluster one of the node is Namenode which acts as a master that monitors the datanode and the maintains the metadata. The remaining nodes are slave nodes that only holds the block of data. Similarly, the computing resources follow the master slave model. The master is called the JobTracker that receives the job, partitions the job and assigns the job to the TaskTrackers. The TaskTrackers are the slave computing nodes in the cluster of computing nodes.

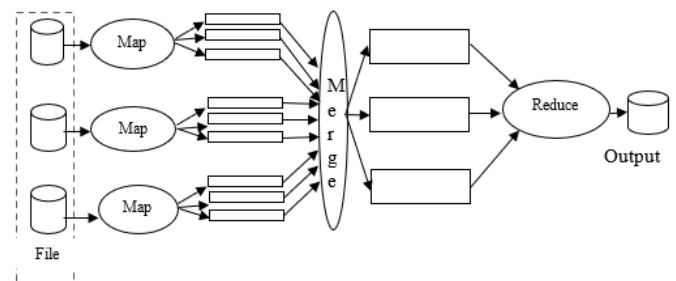


Figure 5. MapReduce workflow

The advent of MapReduce influenced the design of libraries like Spark MLlib, Hadoop Mahout, etc., for supporting the machine learning process on big data [46]. The libraries consist of several machine learning techniques including

supervised and unsupervised algorithms such as random forest, Naive Bayes, principle component analysis, etc. The fault tolerance can be achieved effectively with a MapReduce model. Even though, MapReduce model has major benefits, it is not suitable for all types of problems like imbalanced dataset, but it saves lots of computational time [47].

3.2.2 Spark

Spark is an efficient alternate for Hadoop MapReduce framework which is 100 times more efficient than Hadoop. The comparison of Hadoop and Spark in terms of elapsed time and rate is shown in Figure 6.

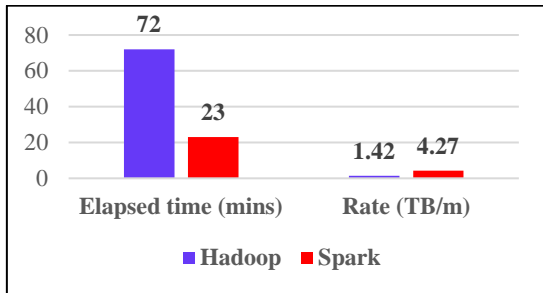


Figure 6. Comparison of Hadoop and Spark in terms of rate and elapsed time

The Spark sorts the data three times faster using ten times fewer nodes. For an example, the Hadoop can process 102.5TB of data at 72mins with the rate of 1.42TB/m. But, the Spark can process 100TB of data at 23mins with the rate of 4.27TB/m [47]. Not like the Hadoop, the Spark performs all the sortings at the HDFS instead of the in-memory cache. The Hadoop is most suitable for batch processing and has limitations in stream processing [48] whereas, the Spark is suitable for both batch and stream processing. It is more powerful and capable complementary tool for Hadoop. It can query, process and transform several terabytes of big data at a time. It supports other bigdata frameworks and runs on different platforms. So, the programs have been written on any of the languages like Scala, Python, R, Structured Query Language (SQL), Java, etc., can be executed easily in Spark. Spark has the limitation in providing security. It uses shared secret password only for providing security whereas, Hadoop provides security through Kerberos and access control list. Hadoop provides strong security than Spark. The Apache Spark architecture consists of two major modules namely, master daemon and worker daemon. The Apache Spark general architecture is given in Figure 7.

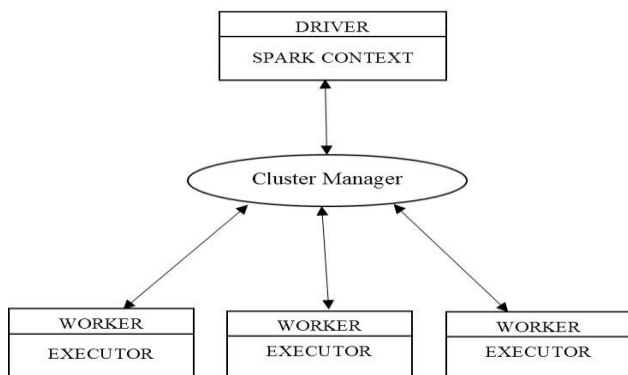


Figure 7. Architecture of apache spark

In the Apache Spark, the user codes are translated into jobs by the master daemon called driver and then, it is assigned to the number of worker daemons. The main task of worker daemons are the execution of the task assigned to it, store all the data needed for the task at in-memory cache, reading and writing the data to and from an external source. For analyzing the bigdata, thousands of processing tools are available at present [49]. These tools are grouped under four major categories, data extraction tools such as Octoparse, Content Grabber, Import.io, Parsehub, Mozenda and Scraper, data visualization tools such as Datawrapper, Solver, Qlik, Tableau Public, Google Fusion Tables and Infogram, open source data tools such as Knime, OpenRefine, R-Programming, Orange, RapidMiner, Pentaho, Talend, Weka, NodeXL & Gephi and sentiment analysis tools such as Opentext, Semantria, Trackur, SAS Sentiment Analysis and Opinion Crawl.

Some of the bigdata processing tools, scalability and fault tolerance capability are compared in Table 1. Hadoop Mahout supports the algorithms like logistic regression, Naive Bayes, random forest, hidden Markov model and multilayer perceptron. But, it does not support the machine learning algorithms like Kernel SVM, conjugate gradient descent, multivariate logistic regression, etc. On the other hand the Spark MLlib does not support K-means clustering, multivariate logistic regression in general form. Some of the algorithms supported by Spark MLlib and the problems where these algorithms can be better utilized are given in Table 2.

Table 1. Big data processing tools

SNO	TOOLS	FAULT TOLERANCE	SCALABILITY
1	SAS	No	Vertical
2	R	No	Vertical
3	Weka	No	Vertical
4	Mahout	Yes	Horizontal over Hadoop
5	Pentaho	Yes	Horizontal over Hadoop
6	Cascading	Yes	Horizontal over Hadoop
7	Spark	Yes	Horizontal beyond Hadoop
8	Halooop	Yes	Horizontal beyond Hadoop
9	GraphLab	No	Horizontal beyond Hadoop
10	Pregel	No	Horizontal beyond Hadoop
11	Giraph	No	Horizontal beyond Hadoop
12	ML over	No	Horizontal beyond Hadoop
13	Storm	No	Horizontal beyond Hadoop

Table 2. Algorithms supported by Spark

SNO	ALGORITHM	TYPE OF PROBLEM
1	Linear SVM	Binary classification
2	Logistic Regression	Binary classification
3	Decision Tree	Multiclass classification, Regression
4	Random Forest	Binary classification, Multiclass classification, Regression
5	Gradient-Boosted trees	Binary classification, Regression
6	Naïve Bayes	Binary classification, Multiclass classification
7	Linear least Squares	Regression
8	Lasso	Regression
9	Ridge Regression	Regression
10	Isotonic Regression	Regression

3.3 Feature selection in big data analysis

Traditionally, feature selection methods are developed for reducing the dimensionality of the dataset in a centralized computing environment. With the advent of big data, we cannot process the large volume of data on a single machine. So, the more powerful environment should be used for analysis of big data [41]. In recent years, the big data application uses distributed approach instead of centralized approach. Hence, the parallel processing concept also introduced in distributed environment for achieving high performance. The parallel processing can be done by keeping multiple nodes at the same geographical location or by keeping nodes in multiple geographical locations [29].

In the distributed feature selection, the original dataset is decomposed into a number of small datasets and the feature selection is applied on each smaller dataset separately. The feature selection methodologies may be a homogeneous or heterogeneous. Then, it executes same or different machine learning (ML) algorithms on the selected features. Finally, the outputs of the machine learning algorithms are merged using the combination methods like voting, weighted voting, etc. [29]. In the distributed processing, a number of processors are required for a parallel run. Hence, the partitioning of datasets into number of subsets should be done manually. Each node should be managed manually where each node consumes same computing resource over the number of subsets. When the dataset size increases, the storage requirement, time complexity, over fitting and noise also increases. The traditional algorithms do not work well with the high dimensional dataset. So, it introduces scaling up problem. So, the dataset is partitioned into several packets by samples or features. The dataset partitioning is normally done for manageability, performance, availability and load balancing. In distributed approach the partition should be done manually but in MapReduce approach the partitioning is achieved automatically by the powerful Hadoop environment [29].

The Hadoop MapReduce also helps for converting the unstructured data into structured data using the statistical column normalization technique on big dataset. The performance of machine learning process on big dataset can be improved by using MapReduce methodology instead of distributed methodology. The MapReduce methodology helps for achieving good accuracy with small memory consumption for the datasets of any size, but it cannot work well with imbalanced datasets [29]. Rahman et al. [34] performed the forecasting of electricity generation using United States electricity generation dataset of past 15 years from 2016. The dataset in text and comma delimited formats were processed and normalized using MapReduce and converted to structured dataset. Then, the dataset was analyzed using BPNN. The result shows that the mean absolute percentage error (MAPE) of all US states electricity load is only 4.13% and the MAPE for individual states is 4 to 9%.

The ranking methods such as ReliefF and Gain ratio are best suitable for centralized feature selection whereas, the feature selection such as CFS and FCBF are the best suited for the distributed feature selection. The ranking filter has much stability than subset filters. The centralized feature selection is the best approach still the size of the dataset is small. When the dataset size is high, either distributed or MapReduce methodology is the best approach [50]. But, when the dataset has imbalanced instances, the MapReduce methodology is not a good choice. The feature selection on big data can be easily

performed using Spark framework. As illustrated in literature, the algorithms like MapReduce based evolutionary feature selection, evolutionary feature weighting, greedy information theoretic feature selection, fast-mRMR, prototype reduction algorithm, random oversampling and random forest are utilized for efficient feature selection in Spark environment [13].

Reggiani et al. [46] designed a new system for processing both tall-narrow and wide-short big datasets. The most popular feature selection algorithm, minimum redundancy maximum relevancy (mRMR) was utilized for selecting best feature subset by considering the mutual information of the feature with the class and also the process is done in distributed parallel environment using Hadoop MapReduce. The proposed system achieved good accuracy and scalability. Yamada et al. [51] designed a new system for an accurate prediction in biological and healthcare applications. Where the novel algorithm named, Hilbert-Schmidt Independence Criterion Lasso (HSIC Lasso), was utilized. The HSIC Lasso algorithm was derived from two feature selections namely, Least Angles Regression (LARS) and Hilbert-Schmidt independence criterion. The proposed HSIC Lasso algorithm is also called Least Angle Nonlinear Distributed feature selection (LAND). The result shows that the LAND works well for handling non-linear big datasets and for the prediction of biological and medical big datasets. It achieves high scalability with high prediction power compared to an existing feature selection like mRMR. The LAND was evaluated using both synthetic and benchmark datasets.

Zhao et al. [52] introduced a new technique for an efficient analysis of the economic bigdata. First, the technique preprocesses the data and then it finds the best feature subsets as the economic indicators. It uses two layer feature selection in both horizontal and vertical manner using the clustering technique. Finally, the economic model was constructed using the correlative and also the collaborative analysis. As a result the technique achieved high accuracy. Ramirez-Gallego et al. [53] presented a fast-mRMR feature selection as an enhanced version of traditional mRMR. The traditional mRMR can handle efficiently the small and medium size datasets and provides good accuracy. But, it has the scalability problem. The big dataset quadratically scales the complexity with the number of attributes and also linearly scales with the number of instances. Hence, it can provide only locally optimal feature subset but not the globally optimal feature subset. So, the existing mRMR algorithm does not suitable for growing big datasets. The fast-mRMR method overcomes these all drawbacks of existing mRMR feature selection. The fast-mRMR is evaluated with sequential execution using central processing unit, parallel execution using Graphics processing units (GPUs) and distributed execution using Apache Spark and proved that the fast-mRMR provides high accuracy than existing methods.

4. CHALLENGES OF FEATURE SELECTION

The process of decision making is a tedious task with the datasets. The feature selection plays a major role in providing the solution to these types of complicated task by reducing the size in terms of features. But, the strong dependencies between the features of the large dataset create difficulty in identifying the most relevant features. This problem is known as multi-collinearity problem. Let X, Y and Z are three features, then the dependencies between these features is represented as

$Z=f(X,Y)$. The multi-collinearity problem represents a strong relationship between two or more features. The available methods such as heuristics, greedy searches and regularization solve the dependency problem but they fail to guarantee the optimality of the feature subset. The quadratic programming approach handles this problem effectively by taking feature presence as binary vector and defining the feature subset quality criterion in quadratic form. The efficient feature selection algorithm identifies the subset of features that should minimize the number of similar features and maximize the number of relevant features [54]. Other challenges available for feature selection include structured features, linked data, multi-source data and multi-view data, streaming data and features, scalability and stability. Big data analysis is an

important research area in recent years. But, the general problem associated with is the curse of dimensionality [55]. All big datasets have enormous number of features which may create challenges in machine learning process. The performance of the machine learning process increases with increasing number of features for certain levels only, after that it turns to decrease. So, the number of features should be an optimal. An alternate way for the conventional feature selection is the use of coefficient variance (CV) such as signal to noise ratio (SNR), relative standard deviation (RSD), variance to mean ratio (VTM) and Efficiency (E). The summary of different feature selection algorithms utilized in literature for big data analysis are shown in Table 3.

Table 3. Summary of the related works

Sl. No	Author & References	Feature Selection Types	Method	Application	Dataset	Advantages	Result	Remarks
1	Rana et al [56]	Hybrid	RRRelieff and Mutual Information (MI)	Electricity Load Forecasting	New South Wales in Australia Electric Load Data at 2006	Considers both Linear and non-Linear dependency between predictor variable and class variable. RRRelieff, MI with Back Propagation Neural Network (BPNN) provides good accuracy than RRRelieff, MI with Linear Regression	MAPE: RRRelieff+MI+BPNN = 0.28% RRRelieff+MI+LR = 0.29%	Interaction or synergy of features not considered
2	Sarhani et al. [57]	Filter	CFS,Support Vector Regression (SVR), PSO	Electricity Load Forecasting	Historical electricity dataset used in EUNITE competition from January 1997 to December 1998 and hourly data from NEPOOL region from 2004 to 2007 collected from Mathwork website	CFS with SVR and PSO improves performance of Machine Learning and eliminate an irrelevant features effectively.	MAPE: CFS+SVR+PSO (16 Features) = 0.055% CFS+SVR+PSO (8 Features) = 0.03%	Need to optimize SVR parameters
3	Rana et al. [58]	Filter	Mutual Information, CFS and partial auto-correlation	Interval Forecasting of Electricity Demand	Electricity demand data of Australia and United Kingdom	MI and CFS techniques effectively find out forecasting interval. Effectively balances demand supply. Provides efficient risk management in demand-supply	Accuracy: MI+ Ensemble NN = 97.14% CFS+ Ensemble NN = 96.68% Autocorrelation+ Ensemble NN = 69.69%	Over fitting problem may occur when dataset size increases
4	Huang et al. [59]	Filter	Mutual Information	Electricity Load Forecasting	North China data from 2005-2012	G-mRMR feature selection technique with Random Forest provides good accuracy. It reduces the over fitting problem and provides optimal subset.	MAPE: Gm-RMR+RandomForest=2.530% Gm-RMR+SVR = 3.320% Gm-RMR+BPNN = 2.718%	Cannot handle redundant features. The performance reduces as noise and redundancy increases
5	Wang et al. [60]	Hybrid	Relieff and Random forest(RF), Kernel Principle Component Analysis(KPCA),Differential Evolution-SVM(DE-SVM)	Electricity Price Forecasting	ISO New England Energy offer data from 2010 to 2015	Relieff with RF technique handles high dimensional data effectively. Hybrid feature selection method (HFS) and feature extraction with DE-SVM improves the performance of the learning process compared to HFS with Naive Bayes and HFS with Decision tree methods.	Relieff+Random Forest + DE-SVM : 4.6% error at 1.1 threshold	Computation time increases as number of features increases
6	Oreski et al. [43]	Filter	Information Gain, Gain Ratio, Relief, Linear Forward selection and Voting Feature Selection	-	Publicly available datasets from UCI repository, Staelib-Carnegie Mellon University, Sociology Dataset Server-Saint Joseph's University, Arizona State University	Relief with the Neural Network provides high accuracy compared to other feature selection techniques	Accuracy: Relieff+Artificial neural network = 87.50% Relieff+Decision Tree = 83% Relieff+Discreminant Analysis = 75%	Fail to consider the interaction and relevancy of the features. Over fitting problem may occur

Sl. No	Author & References	Feature Selection Types	Method	Application	Dataset	Advantages	Result	Remarks
7	Kumar et al. [61]	Filter	Linear correlation, Rank correlation, Regression Relief, Random forest	Stock Market Forecasting	Stock datasets from different countries. Dataset is collected from Yahoo Finance (January 2008-December 2013)	Random Forest technique used with proximal support vector machine (PSVM) provides good performance and reduces the computational complexity	Accuracy: Random Forest+BPNN = 60.2% Random Forest+PSVM=62.72%	Little difficult to tune regularization parameter for prediction model. Less effective to handling non-linear dataset
8	Yamada et al [51]	Hybrid	Hilbert-Schmidt Independence Criterion Lasso	Healthcare	Synthetic dataset and Small scale benchmark datasets	Hilbert-Schmidt Independence Criterion Lasso efficiently handles nonlinear datasets in distributed environment.	Accuracy = 99.99%	-
9	Abedinia et al. [62]	Hybrid	MRMRMS (Mutual Information, Mutual Redundancy, Information Gain)	Electricity Load Forecasting and Electricity Price Forecasting	Pennsylvania, New Jersey, and Maryland (PJM) and Spanish electricity market data, New York electric utility data and weather data obtained from Central Park	Filter method provides best subset by considering relevance, redundancy and interaction. Wrapper method fine tunes the setting of the filter. Effectively handles non-linear datasets. MRMRMS technique with Wavelet Neural Network improves accuracy	MAPE: MRMRMS+Wavelett neural network = 4.09%	Considers two-way MRMRMS interactions. Computational time increases as the dataset size, relevancy and interaction increases

Let 'X' be the training dataset that consists of 'n' instances and 'm' attributes of 'C' classes. ' v_a ' be the sum of all coefficients of variance for each class, \bar{x} be the sample mean, ' μ ' be the mean and 's' be the standard deviation. The RSD, SNR, VTM and E are calculated as follows,

$$RSD: v_a = \frac{\bar{x} - \mu}{s/\sqrt{n}} \quad (1)$$

$$SNR: v_a = \frac{\mu}{\sigma} \quad (2)$$

$$VTM: v_a = \frac{\sigma^4}{\sigma} \quad (3)$$

$$E: v_a = \frac{\sigma^4}{\mu^2} \quad (4)$$

where

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} \quad (5)$$

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (6)$$

These coefficients of variance techniques are applied on each feature and the relationship with the target feature is identified. Finally, the outcome of each techniques are compared and the best coefficient variance is selected for further processing [2, 63].

5. CONCLUSION

The size of the dataset plays an important role in performing the classification and prediction process. When the dimension of the dataset is large and also the dataset contains the irrelevant features, the big data cannot be analyzed effectively. The machine learning methods achieve a good performance with the big datasets, only after reducing the dimensionality of the dataset by removing the irrelevant and redundant features.

The dimension can be reduced effectively by the feature selection. In this paper, several feature selection methods and its pros and cons were discussed. The feature selection is an important preprocessing task to produce the good quality of prediction and classification in case of big datasets. So, the best feature selection should be identified for the given dataset. After that the suitable prediction or classification algorithms are applied on that dataset to obtain good results.

Now a days, the datasets collected from many of the applications have multiple dependencies. These datasets can be processed by using quadratic programming approach to select feature subsets of good quality. The feature selection can also be done in centralized, distributed and distributed parallel processing environments using the central processing unit, graphics processing unit and the powerful high dimensional processing and analytical engines like Hadoop and Spark. The study shows that ReliefF, mRMR, random forest, mutual information, fast-mRMR, RFE and boruta feature selection are an important feature selection in improving the performance of machine learning process. The decomposition techniques, optimization techniques, clustering concepts, hybrid system and deep learning are presented as the future research ideas to enhance the performance of the big data analysis. The decomposition and clustering can be utilized for removing the noise and outliers. The optimization can be introduced for tuning the parameters of the machine learning in big data analysis. The hybrid system can be utilized to greatly reduce the dimension at high speed and can solve the curse of dimensionality issues for the big datasets and to improve the accuracy of learning. The deep learning can be utilized to handle the complicated big data of large size and to improve the accuracy of the learning.

REFERENCES

- [1] Bolón-Canedo, V., Sánchez-Marroño, N., Alonso-Betanzos, A. (2015). Recent advances and emerging challenges of feature selection in the context of big data. Knowledge-Based Systems, 86: 33-45. <http://dx.doi.org/10.1016/j.knsys.2015.05.014>
- [2] Rani, D.S., Rani, T.S., Bhavani, S.D. (2015). Feature subset selection using consensus clustering. In 2015

- Eighth International Conference on Advances in Pattern Recognition (ICAPR), pp. 1-6. <http://doi.org/10.1109/ICAPR.2015.7050659>.
- [3] Sutha, K., Tamilselvi, J.J. (2015). A review of feature selection algorithms for data mining techniques. *International Journal on Computer Science and Engineering*, 7(6): 63.
- [4] Ilango, S.S., Vimal, S., Kaliappan, M., Subbulakshmi, P. (2019). Optimization using artificial bee colony based clustering approach for big data. *Cluster Computing*, 22(5): 12169-12177. <http://dx.doi.org/10.1007/s10586-017-1571-3>
- [5] Sivasankari, S., Baggiya Lakshmi, T. (2016). Operational analysis of various text mining tools in bigdata. *International Journal of Pharmacy & Technology (IJPT)*, 8(2): 4087-4091.
- [6] Du, W., Cao, Z., Song, T., Li, Y., Liang, Y. (2017). A feature selection method based on multiple kernel learning with expression profiles of different types. *BioData Mining*, 10(1): 4. <http://dx.doi.org/10.1186/s13040-017-0124-x>
- [7] Subbiah, S.S., Chinnappan, J. (2020). A review of short term load forecasting using deep learning. *International Journal on Emerging Technologies*, 11(2): 378-384.
- [8] Kumar, V., Minz, S. (2014). Feature selection: A literature review. *SmartCR*, 4(3): 211-229. <http://dx.doi.org/10.6029/smartcr.2014.03.007>
- [9] Melinda, S. (2016). A survey of feature selection approaches for scalable machine learning. Doctoral Dissertation, Technische Universität Berlin.
- [10] Swaroop, G., Kumar, S. (2014). An efficient model for share market prediction using data mining techniques. *International Journal of Applied Engineering Research*, 9(17): 3807-3812.
- [11] Senthil Kumar, P., Lopez, D. (2016). A review on feature selection methods for high dimensional data. *International Journal of Engineering and Technology*, 8(2): 669-672.
- [12] Kalousis, A., Prados, J., Hilario, M. (2007). Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and Information Systems*, 12(1): 95-116. <http://dx.doi.org/10.1007/s10115-006-0040-8>
- [13] Subbiah, S.S., Chinnappan, J. (2020). An improved short term load forecasting with ranker based feature selection technique. *Journal of Intelligent & Fuzzy Systems*, 39(5): 6783-6800. <http://dx.doi.org/10.3233/JIFS-191568>
- [14] Wang, L.P. (2015). Feature selection algorithm based on conditional dynamic mutual information. *International Journal on Smart Sensing & Intelligent Systems*, 8(1): 316. <http://dx.doi.org/10.21307/ijssis-2017-761>
- [15] Song, Q., Ni, J., Wang, G. (2011). A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering*, 25(1): 1-14. <http://dx.doi.org/10.1109/TKDE.2011.181>
- [16] Hall, M.A., Smith, L.A. (1999). Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In *FLAIRS Conference*, pp. 235-239.
- [17] Senthil, K.P. (2019). Improved prediction of wind speed using machine learning. *EAI Endorsed Transactions on Energy Web*, 6(23).
- [18] Zhao, Z., Liu, H. (2009). Searching for interacting features in subset selection. *Intelligent Data Analysis*, 13(2): 207-228. <http://doi.org/10.3233/IDA-2009-0364>
- [19] Zhu, K., Yang, J. (2013). A cluster-based sequential feature selection algorithm. In *2013 Ninth International Conference on Natural Computation (ICNC)*, pp. 848-852. <http://dx.doi.org/10.1109/ICNC.2013.6818094>
- [20] Senthil Kumar, P., Lopez, D. (2015). Feature selection used for wind speed forecasting with data driven approaches. *Journal of Engineering Science and Technology Review*, 8(5): 124-127. <http://dx.doi.org/10.25103/jestr.085.17>
- [21] Paramasivan, S.K., Lopez, D. (2016). Forecasting of wind speed using feature selection and neural networks. *International Journal of Renewable Energy Research (IJRER)*, 6(3): 833-837.
- [22] Senthil, K.P. (2017). A review of soft computing techniques in short-term load forecasting. *International Journal of Applied Engineering Research*, 12(18): 7202-7206.
- [23] Kira, K., Rendell, L.A. (1992). The feature selection problem: Traditional methods and a new algorithm. In *Aaai*, 2: 129-134.
- [24] Hwang, Y.S. (2014). Wrapper-based feature selection using support vector machine. *Life Sci. J*, 11(7): 632-636.
- [25] Kim, Y., Street, W.N., Menczer, F. (2000). Feature selection in unsupervised learning via evolutionary search. In *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 365-369. <http://dx.doi.org/10.1145/347090.347169>
- [26] Gandhi, S.S., Prabhune, S.S. (2017). Overview of feature subset selection algorithm for high dimensional data. In *2017 International Conference on Inventive Systems and Control (ICISC)*, pp. 1-6. <http://dx.doi.org/10.1109/ICISC.2017.8068599>
- [27] Bostani, H., Sheikhan, M. (2017). Hybrid of binary gravitational search algorithm and mutual information for feature selection in intrusion detection systems. *Soft Computing*, 21(9): 2307-2324. <http://dx.doi.org/10.1007/s00500-015-1942-8>.
- [28] Liu, J., Wang, G. (2010). A hybrid feature selection method for data sets of thousands of variables. In *2010 2nd International Conference on Advanced Computer Control*, 2: 288-291. [10.1109/ICACC.2010.5486671](http://dx.doi.org/10.1109/ICACC.2010.5486671)
- [29] Tsai, C.F., Lin, W.C., Ke, S.W. (2016). Big data mining with parallel computing: A comparison of distributed and MapReduce methodologies. *Journal of Systems and Software*, 122: 83-92. <http://dx.doi.org/10.1016/j.jss.2016.09.007>
- [30] Morán-Fernández, L., Bolón-Canedo, V., Alonso-Betanzos, A. (2017). Centralized vs. distributed feature selection methods based on data complexity measures. *Knowledge-Based Systems*, 117: 27-45. <http://dx.doi.org/10.1016/j.knosys.2016.09.022>
- [31] Wang, F., Liang, J. (2016). An efficient feature selection algorithm for hybrid data. *Neurocomputing*, 193: 33-41. <http://dx.doi.org/10.1016/j.neucom.2016.01.056>
- [32] Xue, B., Zhang, M., Browne, W.N., Yao, X. (2015). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4): 606-626. <http://dx.doi.org/10.1109/TEVC.2015.2504420>
- [33] Hodge, V.J., O'Keefe, S., Austin, J. (2016). Hadoop neural network for parallel and distributed feature selection. *Neural Networks*, 78: 24-35.

- <http://dx.doi.org/10.1016/j.neunet.2015.08.011>
- [34] Rahman, M.N., Esmailpour, A., Zhao, J. (2016). Machine learning with big data an efficient electricity generation forecasting system. *Big Data Research*, 5: 9-15. <http://dx.doi.org/10.1016/j.bdr.2016.02.002>
- [35] Peralta, D., Del Río, S., Ramírez-Gallego, S., Triguero, I., Benitez, J.M., Herrera, F. (2015). Evolutionary feature selection for big data classification: A mapreduce approach. *Mathematical Problems in Engineering*. <http://dx.doi.org/10.1155/2015/246139>
- [36] Gandomi, A., Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2): 137-144. <http://dx.doi.org/10.1016/j.ijinfomgt.2014.10.007>
- [37] Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A., Khan, S.U. (2015). The rise of “big data” on cloud computing: Review and open research issues. *Information Systems*, 47: 98-115. <http://dx.doi.org/10.1016/j.is.2014.07.006>
- [38] Deisy, C., Baskar, S., Ramraj, N., Koori, J.S., Jeevanandam, P. (2010). A novel information theoretic-interact algorithm (IT-IN) for feature selection using three machine learning algorithms. *Expert Systems with Applications*, 37(12): 7589-7597. <http://dx.doi.org/10.1016/j.eswa.2010.04.084>
- [39] Ma, W. (2008). Power system short-term load forecasting based on improved support vector machines. In *2008 International Symposium on Knowledge Acquisition and Modeling*, pp. 658-662. <http://dx.doi.org/10.1109/KAM.2008.68>
- [40] Poostchi, M., Bunyak, F., Palaniappan, K., Seetharaman, G. (2013). Feature selection for appearance-based vehicle tracking in geospatial video. In *Geospatial InfoFusion III (Vol. 8747, p. 87470G)*. International Society for Optics and Photonics. <http://dx.doi.org/10.1117/12.2015672>
- [41] Kiruthika, V.G., Arutchudar, V., Paramasivan, S. (2014). Highest humidity prediction using data mining techniques. 9: 3259-3264. https://www.researchgate.net/publication/283625830_Highest_humidity_prediction_using_data_mining_techniques.
- [42] Dowlatshahi, M.B., Derhami, V., Nezamabadi-pour, H. (2017). Ensemble of filter-based rankers to guide an epsilon-greedy swarm optimizer for high-dimensional feature subset selection. *Information*, 8(4): 152. <http://dx.doi.org/10.3390/info8040152>
- [43] Oreski, D., Oreski, S., Klicek, B. (2017). Effects of dataset characteristics on the performance of feature selection techniques. *Applied Soft Computing*, 52: 109-119. <http://dx.doi.org/10.1016/j.asoc.2016.12.023>
- [44] Del Río, S., López, V., Benítez, J.M., Herrera, F. (2014). On the use of MapReduce for imbalanced big data using Random Forest. *Information Sciences*, 285: 112-137. [10.1016/j.ins.2014.03.043](http://dx.doi.org/10.1016/j.ins.2014.03.043)
- [45] Gangurde, H.D. (2014). Feature selection using clustering approach for big data. *Int. J. Comput. Appl*, 975: 1-3.
- [46] Reggiani, C., Le Borgne, Y.A., Bontempi, G. (2017). Feature selection in high-dimensional dataset using MapReduce. In *Benelux Conference on Artificial Intelligence*, pp. 101-115. http://dx.doi.org/10.1007/978-3-319-76892-2_8
- [47] Ramírez-Gallego, S., Fernández, A., García, S., Chen, M., Herrera, F. (2018). Big Data: Tutorial and guidelines on information and process fusion for analytics algorithms with MapReduce. *Information Fusion*, 42: 51-61. <http://dx.doi.org/10.1016/j.inffus.2017.10.001>
- [48] Maillo, J., Ramírez, S., Triguero, I., Herrera, F. (2017). kNN-IS: An iterative spark-based design of the k-Nearest Neighbors classifier for big data. *Knowledge-Based Systems*, 117: 3-15. <http://dx.doi.org/10.1016/j.knosys.2016.06.012>
- [49] Karmel, A., Adhithiyar, M., Senthil, K.P. (2018). Machine learning based approach for pothole detection. *International Journal of Civil Engineering and Technology (IJCIET)*, 9(5): 882-888.
- [50] Aldehim, G., Wang, W. (2017). Determining appropriate approaches for using data in feature selection. *International Journal of Machine Learning and Cybernetics*, 8(3): 915-928. <http://dx.doi.org/10.1007/s13042-015-0469-8>
- [51] Yamada, M., Tang, J., Lugo-Martinez, J. et al. (2018). Ultra high-dimensional nonlinear feature selection for big biological data. *IEEE Transactions on Knowledge and Data Engineering*, 30(7): 1352-1365. <http://dx.doi.org/10.1109/TKDE.2018.2789451>
- [52] Zhao, L., Chen, Z., Hu, Y., Min, G., Jiang, Z. (2016). Distributed feature selection for efficient economic big data analysis. *IEEE Transactions on Big Data*, 4(2): 164-176. <http://dx.doi.org/10.1109/TBDATA.2016.2601934>
- [53] Ramírez-Gallego, S., Lastra, I., Martínez-Rego, D., Bolón-Canedo, V., Benítez, J.M., Herrera, F., Alonso-Betanzos, A. (2017). Fast-mRMR: Fast minimum redundancy maximum relevance algorithm for high-dimensional big data. *International Journal of Intelligent Systems*, 32(2): 134-152. <http://dx.doi.org/10.1002/int.21833>
- [54] Katrutsa, A., Strijov, V. (2017). Comprehensive study of feature selection methods to solve multicollinearity problem according to evaluation criteria. *Expert Systems with Applications*, 76: 1-11. <http://dx.doi.org/10.1016/j.eswa.2017.01.048>
- [55] Agila, N., Senthil Kumar, P. (2020). An efficient crop identification using deep learning. *International Journal of Scientific & Technology Research*, 9(1): 2805-2808.
- [56] Rana, M., Koprinska, I., Agelidis, V.G. (2012). Feature selection for electricity load prediction. In *International Conference on Neural Information Processing*, pp. 526-534. http://dx.doi.org/10.1007/978-3-642-34481-7_64
- [57] Sarhani, M., El Afia, A. (2015). Electric Load Forecasting Using Hybrid Machine Learning Approach Incorporating Feature Selection. In *BDCA*, 1-7.
- [58] Rana, M., Koprinska, I., Khosravi, A. (2015). Feature selection for interval forecasting of electricity demand time series data. In *Artificial Neural Networks*, pp. 445-462. http://dx.doi.org/10.1007/978-3-319-09903-3_22
- [59] Huang, N., Hu, Z., Cai, G., Yang, D. (2016). Short term electrical load forecasting using mutual information based feature selection with generalized minimum-redundancy and maximum-relevance criteria. *Entropy*, 18(9): 330. <http://dx.doi.org/10.3390/e18090330>
- [60] Wang, K., Xu, C., Zhang, Y., Guo, S., Zomaya, A.Y. (2017). Robust big data analytics for electricity price forecasting in the smart grid. *IEEE Transactions on Big Data*, 5(1): 34-45. <http://dx.doi.org/10.1109/TBDATA.2017.2723563>
- [61] Kumar, D., Meghwhani, S.S., Thakur, M. (2016).

- Proximal support vector machine based hybrid prediction models for trend forecasting in financial markets. *Journal of Computational Science*, 17: 1-13. <http://dx.doi.org/10.1016/j.jocs.2016.07.006>.
- [62] Abedinia, O., Amjady, N., Zareipour, H. (2016). A new feature selection technique for load and price forecast of electrical power systems. *IEEE Transactions on Power Systems*, 32(1): 62-74.
- <http://dx.doi.org/10.1109/TPWRS.2016.2556620>
- [63] Fong, S., Biuk-Aghai, R.P., Si, Y.W. (2016). Lightweight feature selection methods based on standardized measure of dispersion for mining big data. In 2016 IEEE International Conference on Computer and Information Technology (CIT), pp. 553-559. <http://dx.doi.org/10.1109/CIT.2016.120>