



Improving Time Complexity and Accuracy of the Machine Learning Algorithms Through Selection of Highly Weighted Top k Features from Complex Datasets

Abdul Majeed¹

Received: 16 November 2018 / Revised: 12 April 2019 / Accepted: 16 April 2019 /
Published online: 2 May 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

Machine learning algorithms (MLAs) usually process large and complex datasets containing a substantial number of features to extract meaningful information about the target concept (a.k.a class). In most cases, MLAs suffer from the latency and computational complexity issues while processing such complex datasets due to the presence of lesser weight (i.e., irrelevant or redundant) features. The computing time of the MLAs increases explosively with increase in the number of features, feature dependence, number of records, types of the features, and nested features categories present in such datasets. Appropriate feature selection before applying MLA is a handy solution to effectively resolve the computing speed and accuracy trade-off while processing large and complex datasets. However, selection of the features that are sufficient, necessary, and are highly co-related with the target concept is very challenging. This paper presents an efficient feature selection algorithm based on random forest to improve the performance of the MLAs without sacrificing the guarantees on the accuracy while processing the large and complex datasets. The proposed feature selection algorithm yields unique features that are closely related with the target concept (i.e., class). The proposed algorithm significantly reduces the computing time of the MLAs without degrading the accuracy much while learning the target concept from the large and complex datasets. The simulation results fortify the efficacy and effectiveness of the proposed algorithm.

Keywords Machine learning algorithms · Features · Computational complexity · Weights · Class · Accuracy · Datasets

✉ Abdul Majeed
Abdulmajid09398@kau.kr

¹ School of Information and Electronics Engineering, Korea Aerospace University, Deogyang-gu, Goyang-si, Gyeonggi-do 412-791, South Korea

1 Introduction

Feature selection has become a de-facto standard in machine learning and data mining applications because of many advantages in terms of significant reduction in computation time and processing power [1]. Feature selection is the process of choosing a small subset of features from given N features that are necessary and adequate to describe the target concept (a.k.a class, the target-concept/class is the output category of the data, for example, income levels, disease categories, stock-market trends, and age-groups etc.). More concisely, it is the process of detecting/extracting relevant features and removing irrelevant, redundant, zero or negative weight features to support informative analysis. Feature selection is one of the key solutions for pre-processing complex datasets in order to overcome computational complexity issues. There are many potential benefits of selecting most relevant features for informative analysis such as big data analytics, recommendations, big data visualization, complex data understanding, reducing the measurement errors, storage capacity, real-time decision making, responding to user queries in real-time, and defying the curse of dimensionality to improve prediction performances [2]. Richard Bellman [3] coined the term the curse of dimensionality to describe the complex problems involving substantial number of features that do not come under the umbrella of low-dimensional settings such as the three-dimensional physical space of everyday experience. The author explained about the common experience that the cost of an algorithm grows exponentially with features dimension, making the cost prohibitive for moderate or large values of the dimension in complex datasets. Tomaso et al. [4] explained the core concepts to avoid curse of dimensionality in complex problems. Amin Belarbi et al. [5] explained the core concepts about the dimensionality reduction using principle component analysis (PCA). The proposed study can significantly reduce the computational cost of image features with high retrieval performance. The comprehensive study about the curse of dimensionality and related concepts can be found in recent studies [6–10]. Selection of the relevant features is very challenging because there exists a trade-off between accuracy and appropriate feature selection. How to select the minimum number of feature subset such that the resulting class distribution/conclusions given only by the values for the selected k features must be the same with the original class distribution/conclusions, given all N features. Additionally, the variables co-relation with each other also affect the feature selection process. In many cases, it is possible that a variable that is not useful at all by itself can yield a better performance when jointly taken with other variables. Similarly, two variables that are individually useless can become extremely useful together. Therefore, a well-defined criterion is needed to select the subset of appropriate features.

In many real-world problems related to knowledge extraction such as, appropriate gene selection from the microarray data [11], sentiment analysis for extracting user opinions about some topic [12], text categorization for frequency analysis [13, 14], information retrieval [15], pattern classification [16], determinants of individuals salary or prices of any industry products, sufficient attention is needed to perform informative analysis. Unfortunately, the most relevant features that explain the target concept well and with higher weights are mostly unknown and are difficult to identify because there exist plenty of useless features. The presence of irrelevant features in

the dataset not only degrade the performance (due to high dimensionality) but also the predictive accuracy (due to irrelevant information) of many machine learning models [17]. Apart from the performance and accuracy, the classifiers build on the large dataset which contains thousands of irrelevant features need additional computation, extensive storage, and take more time to describe the concept. Therefore, it is necessary to find the most appropriate features subsets which have high weight in explaining the target concept. By selecting the highly weighted features accurately, the large amount of the computation power and time can be saved. There are three well-known feature selection methods categories: wrappers [18], filters [18], embedded methods [19, 20] and their improved versions. Each of the three methods employ different evaluation functions to select the most relevant features from given features space. Wrapper methods utilize the learning machine of interest (LOI) as a pre-requisite to score subsets of variables according to their prediction ability. Filter methods choose important variables as a pre-processing step without relying on chosen predictor. Embedded methods contain properties of both filter and wrapper methods, they perform variable selection in the training process. PCA is also used for selecting fewer feature inputs than using all features initially considered relevant [21]. The evaluation criteria used by most of the existing methods are correlation, skewness, t test, ANOVA (Analysis of variance), entropy, information gain, Chi square test, fisher score, recursive feature elimination, sequential feature selection algorithms, genetic algorithms and regularization etc.

Mostly the existing feature selection schemes do not provide thorough insights into relevant feature selection, particularly with the help of the classifiers. Current feature selection schemes mainly focus on univariate or bivariate analysis. However, in many real-world cases, the appropriate selection of the features considering the variable interactions is necessary to select the appropriate features. Meanwhile, it's necessary to develop methods that leverage the capabilities of classifiers to select most relevant features for data mining applications. In real-world data mining applications, each variable affects the decision making with some degree. For example, variable v_i contains student IDs and C indicates whether a student learns computer science or not. v_i is helpful in explaining the students who have taken the computer science in the data. Meanwhile, it is useless to make some decision based on values of v_i since the student IDs will not be same next time, and decision can be highly unreliable. In contrast, the gender v_j variable has more predictive power compared to the ID variable since male students are more likely to study computer science than female students, and this trend has not changed for a past few years. Therefore, the gender variable has higher weight in terms of decision making than the IDs. In this example, whether a student takes computer science or not is the target concept (a.k.a class with two values, Yes or No), where both attributes gender and IDs have influence upon this target concept with varying degree, which we call weight in this study. In this work, we propose a new feature selection algorithm to reduce the computation time with comparable accuracy while building different classifiers on the selected top k relevant features. We first find each feature weight with the help of random forest [22] for the target concept and separate the highly relevant features from the low weight features. By doing so, we retain the highly informative features and discard further processing on the low weight features to reduce computational complexity. By employing the weights concept, we can reduce the processing power and computation

time significantly on large and complex dataset. For a discussion of relevance versus usefulness and various relevance measure, we refer interested readers to the review articles of Kohavi and John [18] and Blum and Langley [20].

The rest of the paper is organized as follows: Sect. 2 describes the related literature about the feature selection models and their applications in various domains. Section 3 presents the conceptual overview of proposed feature selection method and outlines its principal steps. Section 4 discusses the simulation and results on six small size and five large sized datasets and comparison with the existing algorithms. Finally, conclusions are offered in Sect. 5.

2 Background and Related Work

Feature selection techniques has been extensively studied in recent years because advances in information and communication technology have produced large amounts of data with thousands of features [23]. Recent advancement in technology and information surges have made the data generation, data processing, data collection, and data storage very efficient for production, services, communications, and research. Today, the generation of vast amount of data for pattern analysis, predictions and understanding need significant data processing prior to data mining [24]. Feature selection is a core technique for dimensionality reduction, and it is a promising achievement for benefiting from data mining techniques. Several studies have been proposed for this problem due to the data of increasingly dimensionality. Its direct benefits include building robust machine learning models, improving data mining techniques efficiency, and helping formulate, cleaning, and visualize data.

Feature selection or determining high weight features is the process of selecting only relevant features that have direct relation with the problem under investigation. Concisely, it is a procedure for removing irrelevant and redundant information as much as possible. It reduces the number of dimensions in the data that allows machine learning algorithms to quickly converge. Through feature selection the accuracy can also improve in some cases. Feature selection is a handy tool for learning the target concept at faster rate. If the original dataset contains N total number of features, accordingly, the entire number of competing candidates features to be generated is 2^N . This is an enormous number even for medium-sized N . There exist three approaches for solving this complicated problem of candidates features selection such as complete, heuristic, and random. In complete candidate selection approach, all variables are evaluated for candidate feature subset selection which makes this approach highly computationally expensive because a complete search is carried out for the optimal subset selection. An exhaustive search is comprehensive, and it makes some adjustments to select the optimal feature subsets without sacrificing the accuracy. Later it was found that the exhaustive search is not complete, and it uses the attribute dependence for optimal feature subset selection [25]. In heuristic generation procedure, all features to be selected or rejected are determined through many iterations using without replacement strategy. There are many variations to this straightforward process, but generation of features set is basically incremental (either increasing or decreasing). Random methods typi-

cally search lesser number of subsets compared to 2^N by setting a maximum number of iterations by considering the problem space.

An optimal subset of relevant features is always determined by using an evaluation function (i.e., each evaluation function usually select different number of features). Generally, evaluation functions try to measure the discriminating ability of a feature toward class labels. Estevez et al. [26] classified various feature selection methods into two broad categories (i.e., filter and wrapper) considering their close dependence on the selection algorithm. Ben-Bassat [27] grouped the evaluation functions into three categories: information or uncertainty, distance, and dependence, and it was added that the dependence measure can be part of the first two categories. Meanwhile, the authors have not considered the classification error rate as an evaluation function. Doak [28] classified the evaluation functions into three categories: data intrinsic, estimated or incremental error rate, and classification error. The data intrinsic category comprises of distance, entropy, and dependence measures. Considering the prior divisions and the latest developments, the authors [29] further categorized the evaluation functions into five categories: information (or uncertainty), distance, consistency, dependence, and classifier error rate. According to the authors a comprehensive overview of each evaluation functions is given in Table 1.

The ‘–’ in the last accuracy column means that no direct conclusions can be made about the accuracy of the given evaluation function. Surprisingly the classifier error rate has high time complexity, but the accuracy is better. However, the accuracy is very high as compared to all methods.

Many closely related methods have used the feature selection as the pre-processing tool for extracting the desired concepts from the datasets. The authors [30] used feature selection for opinion classification in web forums for different users. The entropy weighted genetic algorithm (EWGA) is a hybrid genetic algorithm that uses the information-gain heuristic criteria for the relevant feature selection. A closely related work in object detection using feature selection was proposed by the authors [31], they used PCA for feature extraction and support vector machines (SVMs) for classification. Power load forecasting using appropriate feature selection was given by the authors [32]. A core concept of the dimensionality reduction with the help of relevant features was also used by many studies [33–35]. Recently the text-mining for classification of documents and clustering has gained popularity. In this regard a comprehensive work is offered by the authors [36–39]. Feature selection has lot of utility in biomedical data classification, protein function prediction and DNA analysis etc. [40–43]. Feature selection has lots of utility in marketing applications for appropriate customers analysis, clustering and recommendations [44–48]. Some classifiers such as ID3 and PLSI [49, 50] select the appropriate features by themselves. However, features interactions are also important to be considered while selecting the relevant subset of features [51]. Therefore, for hard problems such as whether prediction and protein folding the feature interactions modelling is a pre-requisite to be considered.

Recently, some evolutionary methods focused on the optimal subset feature selection for various applications [52–56]. The proposed studies identify the relevant features for various scenarios such as classifying spam emails, stock market analysis, natural language processing, opinion mining, users profiling, query answering, and energy budget predictions etc. Felipe et al. [57] proposed a genetic program-

Table 1 A comparison of different feature selection evaluation functions

Sr. #	Evaluation function	Description about measures	Complexity	Accuracy
1	Distance measure	A feature X is preferred to another feature Y if X results in a greater difference between the two-classes conditional probabilities than Y	Low	–
2	Information measure	Feature X is superior than feature Y if the information gain value of feature X is higher than that from feature Y (e.g., entropy measure)	Low	–
3	Dependence measure	If the correlation of feature X with class C is greater than the correlation of feature Y with C, then feature X will be given preference over Y	Low	–
4	Consistency measure	These measures are heavily dependent on the training dataset. Min-features bias is used in selecting a subset of features	Moderate	–
5	Classifier error rate	The features are selected using the classifier employed for the feature selection	High	Very high

ming approach for feature selection in highly dimensional skewed data. The proposed approach has ability to deal with the data skewness issues, and it selects the most relevant features for informative analysis. The proposed approach is able to reduce the data space by 83% without sacrificing the guarantees on accuracy. Ismail Sayed et al. [58] proposed a novel meta-heuristic optimizer to optimize feature selection problem for maximizing the classification and minimizing the number of features. The evaluation of the proposed approach was carried out on the twenty datasets. The proposed approach finds smaller feature subsets with high accuracy. Zheng et al. [59] proposed a feature selection method for sentiment analysis of Chinese reviews. The proposed approach extracts the most relevant features from the text data that are sufficient to perform the sentiment analysis with high accuracy. Neshatpour et al. [60] proposed an adaptive Iterative Convolutional Neural Networks (ICNN) based algorithm for extracting relevant features from images. The proposed approach is superior in terms of accuracy and computing time compared to the existing approaches. The

proposed approach can be applied to the realtime deadline-driven applications due to the desired thresholding policies.

Peng et al. [61] proposed an efficient feature selection framework named, minimal-redundancy-maximal-relevance (mRMR). The proposed framework is a major advance in feature selection methods, and it effectively resolves the limitations of the mutual information based feature selection (MIFS) algorithms [62]. Both mRMR and MISF algorithms select the feature β assuming that the features are independent of each other. Meanwhile these methods cannot effectively select the optimal number of features from the small datasets and are computationally expensive. A unique approach to circumvent the feature selection problems of these two methods was given by [63]. The used mutual information (MI) for discretization and feature selection (DSM). The proposed approach can select the appropriate features from the large and complex datasets by employing the MI. The proposed approach effectively resolves the limitations of the previous studies and work well with increasing number of features. The proposed approach is able to select the most informative features from the datasets. However, a key limitation of the DSM algorithm is that it doesn't consider the dependence of more than one features jointly that can lead to misleading results in large and complex datasets.

The contributions of this research in the field of feature selection for machine learning/data mining applications can be summarized as follows: (1) it proposes a new feature selection algorithm based on random forest that has potentials to obtain the minimum number of features that are necessary and sufficient to describe the target concept from complex datasets; (2) it computes and selects the highly relevant k features by employing the error rate as an evaluation criteria without discarding any relevant feature; (3) it determines the weights of each feature in relation with target concept to support informative analysis; (4) It reduces the computing speed significantly without sacrificing the guarantees on accuracy for the real-life applications utilizing complex datasets. Additionally, the proposed algorithm support generality which means that selected features give consistent results with different classifiers such as support vector machines (SVM) [64], random forest (RF) [65], classification and regression trees (CART) [66], naïve Bayesian (NB) [67] and K-Nearest Neighbors (K-NN) [68]. Our proposed method performs consistently better results with all five classifiers using six small and five large sized datasets. Furthermore, the proposed approach results can be applied to the realtime applications such as responding to user queries with relevant features for informative analysis, selecting the best features for recommendations, opinion mining from the raw text, predictions about the energy consumption, and forecasting about the future activities in enterprise environments.

3 Proposed Classifiers-Based Feature Selection Algorithm

The classifiers-based feature selection method is necessary to account for the performance issues stemming from highly dimensional data and redundant features. This method not only improves accuracy, it improves computation time significantly and support generality (i.e., to perform consistently better with most classifiers and datasets). This section presents the conceptual overview of the proposed method and

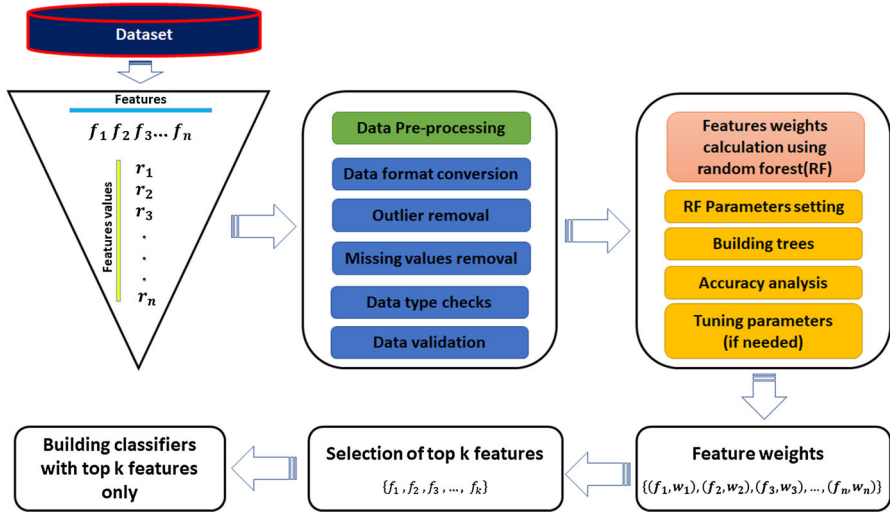


Fig. 1 Conceptual overview of the proposed feature selection algorithm

outlines its procedural steps. Figure 1 shows the conceptual overview of our proposed method.

To improve the performance of any classifiers such as RF, SVM, CART, NB, and KNN, the following four principal concepts are introduced: (1) pre-processing of the datasets; (2) determining the feature weights with the RF model by employing the classification error rate criteria; (3) Selection of the top k most relevant features; and (4) building classifiers from the k relevant features. This approach is chosen to reduce the computation time while processing the complex dataset without sacrificing the guarantees on accuracy, rejection of highly important variable, and yielding the small number of features that are highly co-related with the target concept. Concise details of the four major components with formulation and procedural steps are as follows.

3.1 Pre-processing of the Dataset

The datasets in the original form may contains many missing values and outliers. So, in order to ensure the correctness and preciseness of the results, we pre-process the dataset. At this stage, we remove the missing values and redundant records. The records are regarded as redundant if two or more rows contain exactly the same values. We standardize the values of features for the correct analysis by removing the outliers, feature f can be standardize using Eq. 1.

$$f_{new} = \frac{f - \mu}{\sigma} \tag{1}$$

Apart from the standardization, we transform the data into binary values (i.e., Yes or No, 0 or 1) for different features to extract the top relevant features accurately. With the help of pre-processing, we ensure that all feature values implicitly weight equally

in their representation at start, and data is error free. We remove outliers from the data and perform the data types checks prior to weights calculation. We identify the variable which needs modification in their data types (i.e., for date several formats exist.) to perform the informative analysis. Besides the data conversion, we validate the data types of each features for the correct use in later stages. With the help of data pre-processing which includes format conversion, outlier removal, missing values removal, data types information, and validation, we can obtain the correct data that can be used for further processing directly.

3.2 Determining the Features Weights Using Random Forest

This subsection presents the proposed mechanism through which the weights of the features are determined using random forest (RF). RF is an ensemble machine learning method, and we employed it in our work to quantify the weights of each feature to identify the most relevant features with high prediction power w.r.t target concept. Determining the weights of features is helpful in selecting the best features that are needed for the specific purpose or application. Without determining the feature weights, it is not clear whether the feature (e.g., full column of a given data) used by any classifier while building model is desirable to be used in many real-world cases or not. The detailed knowledge about the degree of information and weights of each feature to an acceptable level of granularity leads to processing power preservation significantly and fewer time performance degradation issue. Further details about the RF can be explored from [22]. The flowchart of the proposed method for weight calculation using RF is given in Fig. 2.

Apart from the weight calculation procedure flowchart shown in Fig. 2, the complete pseudocode used to calculate the feature weights is given in algorithm 1. In Algorithm 1, a high dimensional dataset with N number of rows, and a collection of features (M) which are the columns, a number of trees (T), and a small subset of features (n) used to split the node of classification or regression tree are provided as an input. The feature weights set (w) is obtained as an output from the algorithm. RF constructs an ensemble of classification or regression trees and determine the misclassification error commonly known as out-of-bag (OOB) error (Lines 1–5). We partition the original dataset into two parts while conducting the experiments (i.e., training data and testing data). two-thirds ($2/3$) in Line 4 represents the training data (e.g., the partitioned data on which the algorithm was trained) and the remaining one-third ($1/3$) of the data in Line 5 was used for validation and testing purposes. T_{error} in line 6 represents the error threshold used to compare the error produced by the RF while building random trees. However, this threshold value can be adjusted according to the objectives and nature of the problem. In our experiments we set it to 10% for decision making about highly relevant features. The acceptable values of accuracy should be above 90% to correctly determine the weight values of features. Therefore, we rigorously compare OOB error with defined threshold (T_{error}) to maintain higher values of accuracy. However, accuracy values can be adjusted according to the objectives. If the OOB is high, then a parameter setting of T and n is performed to achieve the appropriate accuracy (Line 7). By tuning, we mean choosing the optimal combinations of T and n

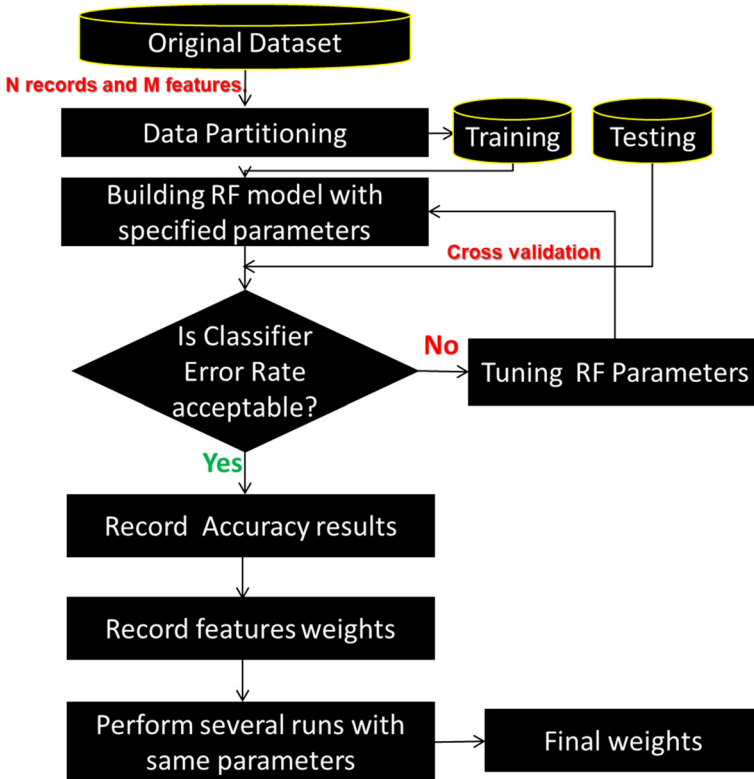


Fig. 2 Feature's weights calculation procedure flowchart

to get the desired results. In contrast, if the OOB falls within the acceptable range (i.e., $er_b \leq 10\%$), then the values of each feature are shuffled in a column, and its impact on the OOB is observed (Lines 9–12). The variable i in Line 1 and line 9 refers to the same features that are part of sample originally drawn from dataset D . The same process is repeated for all features in the dataset. Variable j in Line 15 is a reference variable to denote an error difference of a specific tree before and after permutations. Once the OOB values of all features are calculated, the mean (\bar{x}), variance (s^2) and standard deviation (s) are computed using Eqs. (2)–(4). Subsequently, the feature's weights (W) of each feature are calculated using Eq. (5) by taking joint importance of the variable from all trees. We add each feature weight in set W and at the end a features weight set is obtained (Line 23). All the features in set W have different weights values. Higher weight value of some feature indicates that variable is important. For example, five features, $[f_1, f_2, f_3, f_4, f_5]$ can be processed through the above-mentioned process, and their weights $W = [wf_1, wf_2, wf_3, wf_4, wf_5]$ can be obtained as an output for further analysis (i.e., building classifiers).

3.3 Selection of the Top k Most Relevant Features from the N Features

RF gives the weights of all N features in the form of matrix. We select the top k most influential/relevant features out of N features by comparing the weights of each features with defined threshold. The threshold value can be adjusted dynamically considering the application scenarios, type of the data analysis, and field of application. Setting the optimal threshold value is an NP-hard optimization problem. The threshold value can affect the number of features and the classifiers performance accordingly. The low value of threshold can yield substantial number of features, thereby increasing computing time. In contrast, the high value of threshold can discard many informative features, thereby causing loss in accuracy. This results in a trade-off between accuracy and computing time which can be exploited by designing an adaptive decision making mechanism where the application scenario and nature of the problem are jointly taken into account to reduce the computing time issues while improving the accuracy. In this work, we setup this threshold from the obtained weights averages as a general solution. Given a set F of all features we find another subset k features which have the weight higher than the desired feature selection (FS) threshold.

$$FS_{Thr} = \frac{\sum_{i=1}^N w_i}{N} \quad (6)$$

where w denotes the weights of each feature and N denotes the total number of features in the dataset. The selection of the features is made according to the Eq. 7.

$$f(SS_{Thr}) = \begin{cases} k & \text{if } w_i \geq FS_{Thr} \\ \text{ignore} & \text{otherwise} \end{cases} \quad (7)$$

where w_i denotes the weights of a specific feature i and FS_{Thr} is the feature selection threshold. After appropriate selection we ignore further processing on the rejected features and use only relevant features for further tasks (i.e., building the classifiers).

3.4 Building Classifiers from the Selected k Features Only

After selecting the k -relevant features, we build five classifiers named, RF, SVM, CART, NB, and KNN to determine the classifiers performance with reduced data. We mainly evaluate the accuracy and computation time for the evaluation of classifiers build from the top k selected features. The data partitioning into training and testing and model parameters are explained in the simulation section.

Algorithm 1: Finding the weights of the features using Random Forest

Input: Data sets $D_{N \times M}$ where N are number of rows and M are number of columns in the dataset, Number of trees (T), Number of features for splitting tree node (n).

Output: Set W of features weights

Procedure:

1. for each feature $i \in M$ do
2. for $t= 1$ to T
3. Draw a bootstrap sample S^* of size N from dataset D .
4. Build classification or regression tress T_b using $2/3$ of data (e.g., training data).
5. Predict classification-error of the leftover $1/3$ data (e.g., training data), and calculate OOB, represented as er_b .
6. **if** ($er_b > T_{error}$) **then**
7. Tune RF model parameters(n and T) and go back to (step 3 and 4);
8. else
9. **for** each feature i , permute the values of feature vertically.
10. Repeat (step 3 & 4), record OOb value with (Er_b) after permutation.
11. **Subtract** both OOB error values ($c=Er_b - er_b$)
12. **End for**
13. end *if-else*
14. **End for**
15. Find commutative-OOB error (c_j) from all trees(T).
16. Find average-OOB (\bar{x}), Variance(s^2) and standard deviation(s) using (Eqs. 2,3 and 4) respectively.

$$17. \quad \bar{x} = \frac{1}{T} \sum_{j=1}^T c_j \quad (2)$$

$$18. \quad s^2 = \frac{1}{T-1} \sum_{j=1}^T (c_j - \bar{x})^2 \quad (3)$$

$$19. \quad s = \sqrt{\frac{1}{T-1} \sum_{j=1}^T (c_j - \bar{x})^2} \quad (4)$$

20. Calculate feature weights of the feature i using following equation

$$w_i = \frac{s}{\bar{x}} \quad (5)$$

21. Add feature weight in set W , $W = W \cup \{w_i\}$

22. **End for**

23. return W

Table 2 Detailed description about the dataset used in experiments

Sr.#	Dataset name	Number of features	Number of records	Number of classes
1	Sonar dataset	60	208	2
2	Libras dataset	91	360	15
3	Arrhythmia dataset	279	452	16
4	German credit dataset	20	1000	2
5	Lung cancer dataset	56	32	3
6	Toyota corolla dataset	38	1436	1

4 Simulation Results and Discussions

This section presents the results obtained from various experiments. The improvements of the proposed method are compared using two criteria—the improvements in accuracy, and the computational time—with benchmark feature selection algorithms. To validate the proposed method, we compared the proposed method results with parallel large scale feature selection (PLSFS) [51], DSM [63] and mRMR [61] methods. All these methods are viable candidates for comparing the performance of our proposed algorithm in terms of computing time and accuracy values. The proposed algorithm efficacy has been verified and tested on six datasets including: Toyota Corolla dataset, Lungs cancer dataset, German credit data, Arrhythmia dataset, Libras dataset and Sonar dataset were used in conducting proof of concept experiments. All datasets were obtained from the UCI machine learning repository [69]. All the results are produced and compared on PC computer running Windows 10 having CPU of 2.6 GHz with 8.00 GB RAM. A comprehensive overview of the datasets used in experimented is provided in Table 2. The proposed method works well for both regression and classification problems. The RF has ability to build either classification or regression tree depending upon the type of the target class. The RF build regression trees when the target class is numeric or continuous variable (i.e., houses prices, income, and tax etc.). In contrast, RF builds classification tree for the target class consists of discrete or categorical values. Both classification and regression trees come under the umbrella of supervised learning. The proposed algorithm has ability to deal with both types of the classes (i.e., numeric and categorical). Therefore, both types of the trees were used in proposed algorithm evaluation. Meanwhile, the variable selection criteria are different for classification and regression problems. For the classification trees, the Gini index is applied, whereas variance reduction is used for the regression trees.

To validate the proposed algorithm performance, we build the five classifiers (i.e., machine learning algorithms), named RF, SVM, CART, NB, and KNN to record the

Table 3 Computation time of the five different classifiers on original datasets

Different classifiers\different datasets	Sonar dataset Time (ms)	Libras dataset Time (ms)	Arrhythmia dataset Time (ms)	German credit dataset Time (ms)	Lung cancer dataset Time (ms)	Toyota corolla dataset Time (ms)
Random forest (RF)	1.38	2.38	3.33	3.48	1.08	4.38
Support vector machines (SVM)	1.23	1.31	1.56	2.05	1.11	2.21
Classification and regression tree (CART)	0.10	0.17	0.2	0.24	0.014	0.28
Naïve Bayes (NB)	0.11	0.19	0.19	0.22	0.016	0.25
K-nearest neighbours (K-NN)	4.54	6.78	7.52	7.98	2.53	8.58

computing time and accuracy values. At this stage we use all six datasets without any modifications. The computing time results obtained from the simulations are shown in Table 3.

From the simulation results it can be seen that computing time heavily depends on the number of features and number of records in datasets. Meanwhile, in some cases, the selected features are too less in number compared to the original dataset. Therefore, the computing time for considerably large size dataset is smaller compared to the larger datasets. The accuracy results obtained from the unaltered data by employing all five classifiers (i.e., machine learning algorithms), named RF, SVM, CART, NB, and KNN are depicted in Table 4.

These results are obtained with the standard algorithms with slight modification in the classifiers parameters. The appropriate number of trees used for the RF were 230–270 and the variables used for the tree split were set to the maximum numbers. The tree types were chosen according to the target variable whether it is classification or regression.

4.1 Improvement in the Time Complexity

To prove the algorithm efficiency, we did extensive simulations to verify the computing efficiency of the proposed method while selecting the highly weighted features from the total number of features present in the dataset. We performed the performance

Table 4 Accuracy of the five different classifiers on original datasets

Different classifiers\different datasets	Sonar dataset Accuracy (%)	Libras dataset Accuracy (%)	Arrhythmia dataset Accuracy (%)	German credit dataset Accuracy (%)	Lung cancer dataset Accuracy (%)	Toyota corolla dataset Accuracy (%)
Random forest (RF)	80	73	79	81	73	90
Support vector machines (SVM)	71	72	69	76	53	87
Classification and regression tree (CART)	74	58	73	71	63	85
Naïve Bayes (NB)	63	52	72	72	60	72
K-nearest neighbours (K-NN)	70	67	69	71	63	81
Average accuracy (%)	71	64	72	45	62	83

comparisons with the existing methods on two-fold-the complexity of determining the relevant features, and classifiers performance on the selected features of the both methods. Table 5 shows the performance comparison of the proposed method with the existing methods while determining the feature weights. Proposed algorithm saves almost 30% computation time as compared to the existing method.

The proposed algorithm results are promising with respect to achieving better computing efficiency as compared to the closely related feature selection algorithms. The number of features selected by each algorithm are different in numbers and labels. The cumulative performance of the proposed algorithm and its comparison with the exiting algorithm is summarized in Table 6. The proposed algorithm yields less computing time in both feature weights computation and building the classifiers from k highly weighted features. We take the average of five runs of both the feature weights computation and building the classifiers time. The significant reduction in the proposed algorithm result is due to RF which is superior than the other classifiers such as, SVM and CARTs etc.

Apart from the cumulative computing efficiency performance shown in Table 6, we conducted the dataset specific experiments to support the conclusions of the proposed approach. The performance of the computing time on all six datasets listed in Table 1

Table 5 Feature weights computation time: proposed method versus existing method

Methods	Run 1 time in (ms)	Run 2 time in (ms)	Run 3 time in (ms)	Run 4 time in (ms)	Run 5 time in (ms)
Proposed method	2.27	2.33	2.32	2.28	2.34
PLSFS algorithm [51]	3.38	3.28	3.38	3.4	3.29
mRMR algorithm [61]	3.1	3.02	3.11	3.15	3.17
DSM algorithm [63]	2.98	2.88	2.78	2.82	2.91

Table 6 Cumulative execution time performance of the proposed and existing algorithms

Different classifiers/runs	Proposed algorithm		Existing algorithms	
	Feature weight calculation Average time (ms)	Classifiers building Average time (ms)	Feature weight calculation Average time (ms)	Classifiers building Average time (ms)
Random forest (RF)	2.308	1.918	3.346	1.976
Support vector machines (SVM)	2.308	1.778	3.346	1.796
CART (RT in this case)	2.308	0.196	3.346	0.236
Naïve Bayes (NB)	2.308	0.206	3.346	0.216
K-nearest neighbours (K-NN)	2.308	6.982	3.346	7.99
Total execution time (ms)	2.6776		3.112	
Percentage improvements (%)	16.2%			

is shown in Table 7. Through simulation and comparison, on average, the proposed algorithm reduces computing time by 44.6% compared to all three existing methods. Furthermore, the proposed algorithm reduces computing time by 59.45% compared to the original datasets. Furthermore, the proposed algorithm significantly reduces the computing time while selecting the top k features from all N features. The proposed

Table 7 Dataset specific results comparison with the existing algorithms

Sr. #	Dataset name	Proposed algorithm Avg. computing time (ms)	PLSFS algorithm Avg. computing time (ms)	DSM algorithm Avg. computing time (ms)	mRMR algorithm Avg. computing time (ms)
1	Sonar dataset	1.16	1.37	1.77	2.03
2	Libras dataset	2.63	3.4	3.99	4.58
3	Arrhythmia dataset	1.20	1.37	1.83	2.10
4	German credit dataset	1.05	1.34	1.59	1.83
5	Lung cancer dataset	0.84	0.94	1.01	1.19
6	Toyota corolla dataset	8.94	11.73	11.94	13.71

algorithm yields the most informative features that are closely related with the target class. The proposed algorithm shows remarkable improvements in computation time in all datasets. Additionally, the proposed algorithm has less overheads of computing feature weights compared to the existing algorithms as shown in Table 5.

4.2 Improvement in the Classification/Regression Accuracy

Apart from the computing time, we compared the accuracy results of the proposed algorithm with the three existing algorithms on six datasets. The accuracies results are summarized in Table 8 along with the accuracies of other three algorithms. From the results, it can be observed that proposed algorithm gives better accuracy results compared to existing algorithms. Furthermore, the proposed algorithm shows only marginal degradation compared to the original dataset (i.e., baseline). We used the original dataset accuracy values as a baseline for evaluating our method. We compared the techniques based on the number of features selected by each technique from the datasets. Each technique selects different number of features from each dataset. Each technique employs different evaluation criteria for selecting k features from datasets. For example, in the German credit dataset, the number of features selected by proposed, PLSFS, DSM, and mRMR are 12, 14, 13, and 16, respectively. Furthermore, the accuracy values depend not only on the number of features but also the features itself. Existing techniques choose substantial number of features compared to the proposed study, but their accuracy values are lower due to the inclusion of lesser weight features. All the classifiers were designed considering the data size. The parameters of the all classifiers were kept same for the experiments except the k features. All the parameter, such as number of trees, variables used for tree split, iterations, cross validations, tree depths, sampling strategy, seed values etc. were the same for each method evaluation. In this work, we ignored the F-statistics and other related measures for algorithm

Table 8 classification/regression accuracy results comparison with the existing algorithms

Dataset name	mRMR accuracy (%)	DSM accuracy (%)	PLSFS accuracy (%)	Proposed method accuracy (%)	Original dataset accuracy (%)
Sonar dataset	70	77	69	79	80
Libras dataset	41	60	62	69	73
Arrhythmia dataset	67	72	69	73	79
German credit dataset	73	70	70	75	81
Lung cancer dataset	63	66	68	72	73
Toyota corolla dataset	73	74	71	85	90
Average accuracy (%)	64	69	68	75.5	78.3

evaluation. We considered only the accuracy values for evaluating proposed algorithm effectiveness.

For the sake of the simplicity, we compare the algorithm performance using the RF which is superior than other methods in accuracy. The main reason to use RF is the parameter adjustment in flexible way. The proposed algorithm on average shows 9.8% improvements as compared to the existing algorithms and only marginal degradation as compared to the original dataset. These results emphasize the validity of the proposed method in terms of achieving better accuracy in most cases.

To further validate the proposed algorithm efficacy and effectiveness, we compared the proposed algorithm performance on five large sized datasets. Table 9 presents the brief details of the datasets utilized in the experiments. We utilized full information about the dataset while finding the accuracy and computing time shown in Table 9. The computing time depends on both the number of records and features present in the datasets.

The performance comparisons of the proposed approach with the recent study [60] are shown in Table 10. From the results it can be observed that proposed algorithm yields superior performance in terms of computing time and accuracy. The marginal loss in accuracy in some cases is due to the iterative nature of ICNN method. From the results it can be observed that proposed algorithm reduces the computing time by 29.% and 22.1% compared to the original dataset and ICNN method respectively. From the accuracy point of view, the proposed algorithm improves the accuracy by 2% compared to the ICNN method. Meanwhile, the proposed algorithm shows marginal degradation only in accuracy compared to the original dataset.

The simulation results obtained from eleven dataset verify the proposed algorithm efficacy and effectiveness for various applications. The proposed algorithm effectively resolves the trade-off between accuracy and computing time by considering the weights of features. In either case, preserving more features gives promising accuracy

Table 9 Description about large sized dataset used in experiments

Sr. #	Dataset name	Number of features	Number of records	Original dataset	
				Avg. computing time (ms)	Avg. accuracy (%)
1	Clean1 dataset	166	476	5.2	88.3
2	Semeion dataset	265	1593	14.56	97.5
3	PenglungEW dataset	325	73	8.92	86.1
4	Colon dataset	2000	62	13.98	74.5
5	Leukemia dataset	7129	72	21.89	92.8

Table 10 Proposed algorithm results comparison with the existing method and original datasets

Sr. #	Dataset Name	Original dataset		Proposed method		ICNN method	
		Avg. computing time (ms)	Avg. accuracy (%)	Avg. computing time (ms)	Avg. accuracy (%)	Avg. computing time (ms)	Avg. accuracy (%)
1	Clean1 dataset	5.2	88.3	3.12	87.2	4.92	86.4
2	Semeion dataset	14.56	97.5	11.51	95.5	13.12	92.9
3	PenglungEW dataset	8.92	86.1	5.82	84.1	7.07	83.4
4	Colon dataset	13.98	74.5	10.21	73.2	12.11	74.2
5	Leukemia dataset	21.89	92.8	14.81	90.08	18.31	89.98

but computation time will be high. There exists a strong trade-off between accuracy and computation time which can be exploited by designing an adaptive feature selection mechanism where the utility of each feature and detailed analysis of each attribute in term of the computation are integrated to reduce the computation issues while improving the features accuracy. These experiments results emphasize the validity of the proposed method with respect to achieving better accuracy and low computational overheads. This study provides additional support for highly imbalanced and large dataset handling as compared to current related methods. Furthermore, the proposed approach can be applied to realtime application where the minimum response time is preferable and relevant features are necessary to perform required analysis. Therefore, the proposed approach is an ideal candidate for predictions, informative analysis, future activities forecasting, natural language processing, sentiment analy-

sis, opinion mining, activity analysis of blockchain driven smart home environments residents, personalized recommendations, and contingency planning in financial institutes. Moreover, the proposed approach can be applied as a prototype with an enterprise applications/framework for the intended purpose (i.e., appropriate feature selection) only.

5 Conclusions

In this paper, we proposed a random forest (RF) based top k highly weighted feature selection algorithm to reduce the time overheads of the various machine learning algorithms (MLAs) without sacrificing the accuracy. The main goals of the proposed method are to select top k most relevant feature out of N feature to reduce the computing time of MLAs with considerable accuracy even faced with datasets having substantial number of features. We propose a mechanism for quantifying the weights of each feature of the dataset using random forest to select most influential features by employing the classification error rate as the feature selection criteria. We adapt the optimal RF parameters considering the data distribution and number of features to yield unique and high predictive power features. The proposed method resolves the computation problem with only marginal loss in accuracy as compared to the original dataset and existing methods. The proposed methods results are promising with respect to significant reduction in computation time without sacrificing the guarantees on classifiers accuracy. Through simulations and comparative analysis, on average, our algorithm reduces the classifiers computing time by 24% compared to existing algorithms. From the accuracy point of view, it only marginally ($\sim 1.13\%$) degrades as compared to the original dataset's accuracy.

References

1. Bolón-Canedo V, Sánchez-Marroño N, Alonso-Betanzos A, Benítez JM, Herrera F (2014) A review of microarray datasets and applied feature selection methods. *Inf Sci (Ny)* 282:111–135
2. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. *J Mach Learn Res* 3(Mar):1157–1182
3. Kuo FY, Sloan IH (2005) Lifting the curse of dimensionality. *Not AMS* 52(11):1320–1328
4. Poggio T, Mhaskar H, Rosasco L, Miranda B, Liao Q (2017) Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *Int J Autom Comput* 14(5):503–519
5. Belarbi MA, Mahmoudi S, Belalem G (2017) PCA as dimensionality reduction for large-scale image retrieval systems. *Int J Ambient Comput Intell* 8(4):45–58
6. Wang S-H, Zhang Y-D, Dong Z, Phillips P (2018) Dimensionality reduction of brain image features. Springer, Singapore, pp 105–118
7. Heer B, Maußner A (2018) Projection methods and the curse of dimensionality. *J Math Finance* 08(02):317–334
8. Chow YT, Darbon J, Osher S, Yin W (2019) Algorithm for overcoming the curse of dimensionality for state-dependent Hamilton–Jacobi equations. *J Comput Phys* 387:376–409
9. Christiansen B, Christiansen B (2018) Ensemble averaging and the curse of dimensionality. *J Clim* 31(4):1587–1596
10. Agarwal S, Ranjan P (2018) High dimensionality characteristics and new fuzzy versatile particle swarm optimization. Springer, Singapore, pp 267–275

11. Duan K-B, Rajapakse JC, Wang H, Azuaje F (2005) Multiple SVM-RFE for gene selection in cancer classification with expression data. *IEEE Trans Nanobiosci* 4(3):228–234
12. Pang B, Lee L (2008) Opinion mining and sentiment analysis. *Found Trends Inf Retr* 2(1–2):1–135
13. Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. Springer, Berlin, Heidelberg, pp 137–142
14. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. *Inf Process Manag* 24(5):513–523
15. Croft WB, Metzler D, Strohman T (2010) Search engines information retrieval in practice. Addison-Wesley, Reading
16. Liu Huan, Lei Yu (2005) Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans Knowl Data Eng* 17(4):491–502
17. Sleeman D, Edwards P (eds) (2014) International conference on machine learning 9th : 1992 : Aberdeen, Machine learning : proceedings of the ninth international workshop (ML92). Morgan Kaufman/Elsevier Science
18. Kohavi R, John GH (1997) Wrappers for feature subset selection. *Artif Intell* 97(1–2):273–324
19. Langley P (1994) Selection of relevant features in machine learning. In: Proceedings of the AAAI Fall symposium on relevance, pp 1–5
20. Blum AL, Langley P (1997) Selection of relevant features and examples in machine learning. *Artif Intell* 97(1–2):245–271
21. Malhi A, Gao RX (2004) PCA-based feature selection scheme for machine defect classification. *IEEE Trans Instrum Meas* 53(6):1517–1525
22. Breiman L (2001) Random forests. *Mach Learn* 45:5–32
23. Liu H, Motoda H (1998) Feature selection for knowledge discovery and data mining. Springer, New York
24. Liu H, Motoda H, Setiono R, Zhao Z (2010) Feature selection: an ever evolving frontier in data mining. In: Fourth Workshop on Feature Selection in Data Mining, pp 4–13
25. Rokach L, Maimon O (2008) Data mining with decision trees: theory and applications. World Scientific, Singapore
26. Estevez PA, Tesmer M, Perez CA, Zurada JM (2009) Normalized mutual information feature selection. *IEEE Trans Neural Netw* 20(2):189–201
27. Su B, Ding X, Wang H, Wu Y (2018) Discriminative dimensionality reduction for multi-dimensional sequences. *IEEE Trans Pattern Anal Mach Intell* 40(1):77–91
28. Doak J (1992) Cse-92-18- an evaluation of feature selection methods and their application to computer security
29. Dash M, Liu H (1997) Feature selection for classification. *Intell Data Anal* 1(1–4):131–156
30. Abbasi A, Chen H, Salem A (2008) Sentiment analysis in multiple languages. *ACM Trans Inf Syst* 26(3):1–34
31. Sun Z, Bebis G, Miller R (2004) Object detection using feature subset selection. *Pattern Recognit* 37(11):2165–2176
32. Niu D, Wang Y, Wu DD (2010) Power load forecasting using support vector machine and ant colony optimization. *Expert Syst Appl* 37(3):2531–2539
33. Mitra P, Murthy CA, Pal SK (2002) Unsupervised feature selection using feature similarity. *IEEE Trans Pattern Anal Mach Intell* 24(3):301–312
34. Raymer ML, Punch WF, Goodman ED, Kuhn LA, Jain AK (2000) Dimensionality reduction using genetic algorithms. *IEEE Trans Evol Comput* 4(2):164–171
35. Saeys Y, Inza I, Larranaga P (2007) A review of feature selection techniques in bioinformatics. *Bioinformatics* 23(19):2507–2517
36. Aggarwal CC, Zhai C (2012) Mining text data. Springer, Berlin
37. Li T, Zhang C, Ogihara M (2004) A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics* 20:2429–2437
38. Liu H, Motoda H (2008) Computational methods of feature selection. Chapman & Hall/CRC, Boca Raton
39. Oliveira LS, Sabourin R, Bortolozzi F, Suen CY (2003) A methodology for feature selection using multiobjective genetic algorithms for handwritten digit string recognition. *Int J Pattern Recognit Artif Intell* 17(06):903–929
40. Peng Y, Wu Z, Jiang J (2010) A novel feature selection approach for biomedical data classification. *J Biomed Inform* 43(1):15–23

41. Parkka J, Ermes M, Korpipaa P, Mantyjärvi J, Peltola J, Korhonen I (2006) Activity classification using realistic data from wearable sensors. *IEEE Trans Inf Technol Biomed* 10(1):119–128
42. Ding C, Peng H (2005) Minimum redundancy feature selection from microarray gene expression data. *J Bioinform Comput Biol* 03(02):185–205
43. Somorjai RL, Dolenko B, Baumgartner R (2003) Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. *Bioinformatics* 19(12):1484–1491
44. Lessmann S, Voß S (2009) Feature selection in marketing applications. Springer, Berlin, Heidelberg, pp 200–208
45. Meiri R, Zahavi J (2006) Using simulated annealing to optimize the feature selection problem in marketing applications. *Eur J Oper Res* 171(3):842–858
46. Liu H, Motoda H (1998) Feature extraction, construction and selection: a data mining perspective. Kluwer Academic, Dordrecht
47. Teranol T, Ishino Y (1998) Interactive genetic algorithm based feature selection and its application to marketing data analysis. In: Feature extraction, construction and selection. Springer, Boston, MA, pp 393–406
48. Gheyas IA, Smith LS (2010) Feature subset selection in large dimensionality domains. *Pattern Recognit* 43(1):5–13
49. Quinlan JR (1986) Induction of decision trees. *Mach Learn* 1(1):81–106
50. Bolc L (1987) Computational models of learning. Springer, Berlin Heidelberg
51. Singh S, Kubica J, Larsen S, Sorokina D (2009) Parallel large scale feature selection for logistic regression. In: Proceedings of the 2009 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, pp 1172–1183
52. Faris H, Al-Zoubi AM, Heidari AA, Aljarah I, Mafarja M, Hassonah MA, Fujita H (2019) An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. *Inf Fusion* 48:67–83
53. Liu J, Abbass HA, Tan KC (2019) Evolutionary computation. In: Evolutionary computation and complex networks. Springer, Cham, pp 3–22
54. Gu S, Cheng R, Jin Y (2018) Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Comput* 22(3):811–822
55. Hancer E, Xue B, Zhang M (2018) Differential evolution for filter feature selection based on information theory and feature ranking. *Knowl Based Syst* 140:103–119
56. Mafarja M, Aljarah I, Heidari AA, Hammouri AI, Faris H, Al-Zoubi AM, Mirjalili S (2018) Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowl Based Syst* 145:25–45
57. Viegas F, Rocha L, Gonçalves M, Mourão F, Sá G, Salles T, Andrade G, Sandin I (2018) A genetic programming approach for feature selection in highly dimensional skewed data. *Neurocomputing* 273:554–569
58. Sayed GI, Hassanien AE, Azar AT (2019) Feature selection via a novel chaotic crow search algorithm. *Neural Comput Appl* 31(1):171–188
59. Zheng L, Wang H, Gao S (2018) Sentimental feature selection for sentiment analysis of Chinese online reviews. *Int J Mach Learn Cybern* 9(1):75–84
60. Neshatpour K, Behnia F, Homayoun H, Sasan A (2018) ICNN: An iterative implementation of convolutional neural networks to enable energy and computational complexity aware dynamic approximation. In: 2018 design, automation and test in Europe conference and exhibition (DATE), pp 551–556
61. Hanchuan Peng H, Fuhui Long F, Ding C (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans Pattern Anal Mach Intell* 27(8):1226–1238
62. Hu Q, Zhang L, Zhang D, Pan W, An S, Pedrycz W (2011) Measuring relevance between discrete and continuous features based on neighborhood mutual information. *Expert Syst Appl* 38(9):10737–10750
63. Sharmin S, Ali AA, Khan MAH, Shoyab M (2017) Feature selection and discretization based on mutual information. In: 2017 IEEE international conference on imaging, vision and pattern recognition (icIVPR), pp 1–6
64. Wang L (2005) Support vector machines: theory and applications. Springer, Berlin
65. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
66. Kamiński B, Jakubczyk M, Szufel P (2018) A framework for sensitivity analysis of decision trees. *Cent Eur J Oper Res* 26(1):135–159

67. Bayes T (1991) An essay towards solving a problem in the doctrine of chances. 1763. *MD Comput* 8(3):157–171
68. Fix E, Hodges Jr JL (1952) Discriminatory analysis-nonparametric discrimination: small sample performance (No. UCB-11). California Univ Berkeley
69. Blake C, Merz C (1998) UCI repository of machine learning databases. University of California, Dept. Information and Computer Science, Irvine, CA, USA

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