International Journal of Data Science and Analytics
https://doi.org/10.1007/s41060-024-00509-w

REVIEW

A review of random forest-based feature selection methods for data science education and applications

Reza Iranzad[1] · **Xiao Liu**[2]

Received: 7 November 2022 / Accepted: 6 January 2024

Abstract
Random forest (RF) is one of the most popular statistical learning methods in both data science education and applications. Feature selection, enabled by RF, is often among the very first tasks in a data science project, such as the college capstone project, industry consulting projects. The goal of this paper is to provide a comprehensive review of 12 RF-based feature selection methods for classification problems. The review provides necessary description of each method and the software packages. We show that different methods typically do not provide consistent feature selection results, and the model performance also varies when different RF-based feature selection approaches are employed. This observation suggests that caution must be taken when performing feature selection tasks using RF. Feature selection cannot be blindly done without a sound understanding of the methods adopted, which is not always the case in industry and many senior capstone projects that we have observed. The paper serves as a one-stop reference where students, data science consultants, engineers, and data scientists can access the basic ideas behind these methods, the advantages and limitations of different approaches, as well as the software packages to implement these methods.

Keywords Random forest · Feature selection · Feature importance · Classification · Data science education · Data science consulting projects · Capstone projects

1 Introduction

Random forest (RF) is one of the most popular statistical learning methods in both data science education and industry applications. One important topic under RF is to perform feature selection that identifies important or relevant features included in a statistical model. For example, when advising undergraduate Capstone projects, we have clearly seen an increasing number of projects that involve building statistical learning models and feature selection is often one of the very first tasks of these projects [4]. This is also the case in many multidisciplinary data analytics-related projects in industry [9, 35, 37]. However, our interactions with students, engineers, and data analytics consultants over the past couple of years clearly indicate that there is a lack of comprehensive review and comparison among various feature selection methods. Existing methods and software tools are scattered in different books, research papers, and conference proceedings from various academic communities. More importantly, features selected by different approaches often differ from each other. Hence, the goal of this paper is to provide a comprehensive review of 12 commonly used RF-based feature selection methods for classification problems. The paper serves as a one-stop reference where students, data science consultants, engineers, and data scientists can access the basic ideas behind these methods, the advantages and limitations of different approaches, as well as the software packages to implement these methods.

In many supervised statistical learning problems, there often exist a large number of candidate features that can potentially be used to establish the mapping between features and responses [19, 22, 26, 31–34, 38, 47]. However, having more features (i.e., data) does not necessarily mean one should include all features in a statistical learning model. On the contrary, many candidate features could be irrelevant or redundant and need to be excluded. Keeping only the right features in a model can greatly improve model interpretability, reduce model complexity, and enhance model predictive

✉ Xiao Liu
xiao.liu@isye.gatech.edu

[1] FedEx Express, Memphis, USA

[2] H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, USA

Published online: 03 February 2024

capabilities. Hence, feature selection is a fundamental task when constructing statistical learning models. Such a task becomes even more critical in the age of big data when it is easier than ever to access larger datasets but with more redundant information. After all, a dataset is a combination of useful information and noise. Irrelevant features, like the noise in data, should be removed. There is a wide range of applications in which feature selection techniques can be employed [36, 39].

Feature selection algorithms can be classified into three categories: "Filters," "Embedded," and "Wrappers" [17, 27]. "Filters" select features in a preprocessing step independent of the statistical learning process; for example, it first examines the correlation between features and responses and then identifies those important features to be included. Although the "Filters" approach is usually fast and relatively easier to be implemented, it may not be able to detect complex interactions among features nor take into account the model performance given selected features. Unlike "Filters," "Embedded" algorithms integrate feature selection into the model training process and identify important features by optimizing the model performance. Finally, "Wrappers" methods are also built around a specific statistical learning approach and utilize the statistical learning model to score/rank candidate features according to their predictive power. Random forest (RF)—one of the most widely used ensemble learning methods for both classification and regression—involves constructing a multitude of de-correlated decision trees [2, 19, 20]. In particular, RF successfully leverages the "Wrapper" approach to generate variable importance (VI) scores.

One challenge arising from practice is that, although many feature selection techniques have been proposed based on the framework of RF, it is not always clear to students which approaches should be used given the problem of interest. For example, some RF-based feature selection approaches are *performance-based*. These approaches typically leverage a forward selection and/or a backward elimination strategy to select or remove features for improving prediction accuracy. On the other hand, some RF-based feature selection approaches are *test-based*. These approaches utilize involve statistical tests to identify statistically significant features [40]. In addition, different feature selection methods employ different objectives when performing feature selection. For example, some approaches aim to find a *minimal optimal set* which includes a subset of features to perform prediction tasks, while others identify *all relevant* features (both strongly relevant and weakly relevant) such that removing these features has a negative impact on prediction accuracy [30].

Hence, it is timely for this paper to provide a comprehensive review of 12 commonly used RF-based feature selection methods for classification problems. In particular, we only include those methods where software packages are available. The review provides necessary descriptions of each method as well as the packages implementing the approach. Table 1 presents a high-level overview of the feature selection methods reviewed in this paper, including Boruta, RRF, GRRF, GRF, r2VIM, PIMP, NTA, varSelRF, VSURF, RF-SRC, AUC-RF, and RFE.

The rest of this paper is organized as follows: Sect. 2 firstly provides a short review of RF. A comprehensive review of the 12 feature selection methods is presented in Sect. 3. Numerical results are provided in Sect. 4 to illustrate the applications of these methods, and Sect. 5 concludes the paper.

2 Random forest

RF involves constructing a collection of weakly correlated classification or regression trees and then aggregating them (i.e., an ensemble approach). For regression problems, the average output of the individual trees is returned. For classification problems, the idea of majority voting is employed to determine the final prediction. RF inherits many advantages of decision trees (e.g., invariant under scaling and transformations of feature values, robust to the inclusion of irrelevant features, the ability to capture complex interactions among features, and handle noisy and missing data, etc.) and enhances the capabilities of decision trees such as the robustness against over-fitting and the capability of computing feature importance.

The general method of random decision forests can be traced back to Ho [20] and was later extended by Breiman [2] to incorporate the idea of "bagging" for reducing the variance of the estimator. Two important ideas are integrated into RF, i.e., bagging and random node split. Bagging is a general technique of bootstrap aggregating that repeatedly selects a random sample (with replacement) of the training data and fits trees to these samples. Random node split refers to the technique that only a random subset of features is considered at each tree node split. For example, for classification problems with p features, one may only consider $\sqrt{p}$ randomly selected candidate features at each tree node split. As a result, trees grown from different bootstrap samples become more different from each other and are less correlated. Even if there exist one or more features that are very strong predictors for the response, the random node split prevents these features from being selected by many of the trees, effectively de-correlating the trees.

To elaborate, if we consider a set of B random variables, each with variance σ^2, the variance of the mean of these random variables is given by

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \tag{1}$$

Table 1 A summary of wrapper methods based on RF for classification problems

Method	References	R Package	Approach	Strategy
Boruta	Kursa and Rudnicki [29]	*Boruta*	Test-based	Permutation based algorithm
RRF	Deng and Runger [7]	*RRF*	Performance-based	Forward feature selection using regularized RF
GRRF	Deng and Runger [8]	*RRF*	Performance-based	Guide RRF using ordinary RF
GRF	Deng [6]	*RRF*	Performance-based	A subset of GRRF that allows parallel computation
r2VIM	Szymczak et al. [41]	*Pomona*	Test-based	Permutation based algorithm
PIMP	Altmann et al.[1]	*vita*	Test-based	Permutation test
NTA	Janitza et al. [25]	*vita*	Test-based	Hold-out VIM test
varSelRF	Díaz-Uriarte and Alvarez de Andrés [11]	*varSelRF*	Performance-based	Backward feature elimination using OOB-error rate
VSURF	Genuer et al. [14]	*VSURF*	Performance-based	Two-step procedure, first applies backward elimination then forward selection
RF-SRC	Ishwaran et al. [23]	*randomForestSRC*	Performance-based	Forward stepwise regularized feature selection
AUCRF	Calle et al. [3]	*AUCRF*	Performance-based	Backward feature elimination using OOB-AUC
RFE	Guyon et al. [18]	*caret*	Performance-based	Backward feature elimination using OOB-error rate

where ρ is the positive pairwise correlation. Hence, it is seen that the variance of the mean becomes smaller as B increases and ρ decreases. This explains why RF works well by (i) leveraging the idea of bagging to grow a large number of trees (i.e., to make B larger) and (ii) adopting the idea of random node split to make the tree less similar to each other (i.e., to make the correlation ρ smaller).

The algorithm of RF is summarized as follows [19]:

Algorithm 1: Random Forest for Regression and Classification

Set the number of trees $ntree = B$
Set the value of $mtry = m$
for b=1,...,B **do**
  Grow tree t by repeating the following steps:
  (i) Draw a bootstrap sample of size N from the training set.
  (ii) Grow a binary decision tree T_b to the bootstrapped data as follow, until the stoppage criterion is met:
  - Randomly select m variable from the feature set
  - Find the best-split variable and split value that minimize impurity
  - Split the node into two daughter nodes

  (iii) Output the ensemble of trees $\{T_b\}_1^B$
end
To predict a new data point: use majority voting for classification problems and the average for regression problems

One important built-in capability of RF is the computation of feature importance, such as the Gini index and mean decrease accuracy. For example, the Gini index at each tree node ν is defined as:

$$\text{Gini}(\nu) = \sum_{c=1}^{C} p_c^{\nu}(1 - p_c^{\nu}) \tag{2}$$

where p_c^{ν} is a proportion of class-c observations at node ν. The Gini impurity of the feature X_i for the two daughter nodes of ν is then given by

$$\begin{aligned}\text{Gain}(X_i, \nu) &= \text{Gini}(X_i, \nu) - \omega^L \text{Gini}(X_i, \nu^L) \\ &\quad -\omega^R \text{Gini}(X_i, \nu^R)\end{aligned} \tag{3}$$

where ν^L and ν^R denote the two daughter nodes of ν, and ω^L and ω^R are the proportions of observations in each daughter node. Hence, at each tree node split, the improvement in the split criterion (i.e., the Gini index) is computed for the splitting feature and is accumulated over all trees for each feature.

RF also uses another approach to compute the feature importance in terms of the prediction strength of a feature. After a tree has been added for the forest, the prediction accuracy is evaluated using the OOB samples (i.e., construct the RF predictor for a data point by averaging only those trees

corresponding to bootstrap samples in which this data point did not appear). Then, the values of a particular feature in the OOB samples are randomly shuffled, and the prediction accuracy is again evaluated using the new OOB data (as if we were feeding the incorrect input to the forest). By doing this, we expect the OOB prediction accuracy to decrease, and the decrease in accuracy is averaged over all trees. The amount of decrease (i.e., the mean decrease in accuracy) can be used as an important measure for the feature whose values have been permuted. If the feature greatly affects the prediction performance, the decrease is expected to be large. If permuting the values of a feature merely affects the prediction accuracy, one may naturally expect this feature to be less influential. The basic built-in capability of RF described above enables one to generate the feature importance ranking.

3 Feature selection methods

In this section, we provide a comprehensive review of the 12 RF-based feature selection methods for classification problems as summarized in Table 1. For each method, we describe its main idea as well as the R packages for implementation.

• **Boruta**

Boruta, proposed by Kursa and Rudnicki [29], is a wrapper feature selection method built around the RF classification algorithm. Boruta is a test-based heuristic approximation algorithm that attempts to find a threshold for feature selection rather than ranking features with some VIMs. In other words, it solves the challenging all-relevant feature selection problem rather than finding a minimal set of features. Boruta has been implemented in the R package, `Boruta` [29].

The core idea behind the Boruta is that a feature is not important if its calculated importance is less than the importance of a randomly permuted feature. Because removing irrelevant features is expected to increase the accuracy in computing the variable importance, Boruta hinges upon sequentially removing features that are found significantly irrelevant.

At each iteration, the Boruta algorithm duplicates each feature that exists in the feature set with a random permutation of their observations. The duplicated features are referred to as the *shadow features*. The RF is performed upon all features, including the shadow features, and the feature importance is computed. Then, the features that have higher importance compared to the maximum importance among all the shadow features are deemed relevant, while the remaining features are considered irrelevant and removed from the feature set. Since the RF classifier produces different importance measures due to its randomness and the presence of shadow features, Boruta usually repeats the process above multiple times until no more irrelevant features can be removed. Finally, Boruta categorizes each feature as relevant (irrelevant) if it has an importance significantly higher (lower) than the maximum Z-score among all random shadow features (MZSF). For those features with close importance measure to the best shadow feature (known as tentative features), the Boruta algorithm does not make any decisions and lets the user decide if those tentative features should be included given the context of the application.

• **RRF, GRRF, and GRF**

Regularized random forest (RRF) is a wrapper feature selection technique built over the RF binary classification problems. Unlike Boruta, RRF attempts to find a minimal optimal set of relevant features and remove non-relevant features. RRF is available in the R package `RRF`, Deng and Runger [7].

Note that although it is often possible to select the first K features with the highest importance scores using the RF algorithm, selecting non-relevant features among the K features is likely in the presence of correlated features. Hence, the RRF is an ensemble and greedy feature selection technique that employs a regularized framework together with an upper bound of the Gini information gain value when computing the feature importance. The regularized feature selection by RRF helps identify a compact feature subset possible to perform the prediction. Let F be the empty set of indices, $\lambda \in (0, 1]$ be the penalty coefficient, and $Gain^* = 0$ be the initial upper bound of Gini information gain. At each tree node, for any feature X_i that is not in the set of indices $i \notin F$, RRF penalizes it by multiplying λ with the Gini information gain. The regularized gain for variable X_i at a non-leaf node ν is then calculated as follows:

$$\text{Gain}_R(X_i, \nu) = \begin{cases} \lambda \times \text{Gain}(X_i, \nu) & \text{if } \ i \notin F \\ \text{Gain}(X_i, \nu) & \text{if } \ i \in F \end{cases} \tag{4}$$

Hence, the split on X_i only occurs if the $\text{Gain}_R(X_i, \nu)$ exceeds the upper bound $Gain^*$ obtained from previous splits, and the $Gain^*$ is updated after the node splitting. A comprehensive description of the algorithm is provided in Deng and Runger [7].

Note that although RRF can be used as an ensemble classifier, RRF is often recommended to use only for feature selection purposes. It is also noted that as the depth of a tree increases, fewer observations may drop in non-leaf nodes. The lack of enough observations may affect calculating the Gini information gain, known as the node sparsity issue. Because of the node sparsity issue, selecting a subset of features that includes weakly relevant features is probable.

To alleviate the node sparsity issue, Deng and Runger [8] proposed an enhanced version of the RRF called the Guided Regularized Random Forest (GRRF). GRRF is also available in the R package `RRF`. GRRF aims to select a compact feature set by building multiple ensembles, and features are evaluated on the entire training set. GRRF incorporates the

importance scores calculated by an ordinary RF to guide the feature selection procedure in RRF, while the penalty coefficient λ is no longer fixed for the entire feature set. In other words, GRRF dynamically penalizes features out of the set of indices as follows:

$$\text{Gain}_R(X_i, \nu) = \begin{cases} \lambda_i \times \text{Gain}(X_i, \nu) & \text{if} \quad i \notin F \\ \text{Gain}(X_i, \nu) & \text{if} \quad i \in F \end{cases} \tag{5}$$

where λ_i is given by

$$\lambda_i = 1 - \gamma(1 - \text{Imp}_i) \tag{6}$$

with γ and Imp_i, respectively, being the weighting control parameter and the normalized variable importance score of the variable X_i.

Based on experimental results on 10 gene datasets, it has been shown that GRRF may be more robust than RRF against parameter changes, while the overall accuracy of RRF is higher. Hence, RRF is recommended when accuracy is a major concern, while one may choose GRRF if shrinking the feature set is the priority.

Finally, Deng [6] proposed the Guided Random Forest (GRF) as a special case of GRRF. GRF also shares the same idea of using a regular RF VIM to guide the feature selection process. However, despite the GRRF that uses sequentially grown trees, GRF intends to build independent trees in which parallel computation is applicable. Moreover, GRF removes the regularized part in GRRF, making each split in a tree node highly dependent on previous splits. In a numerical investigation based on 10 gene datasets, GRF, in general, selects more features than GRRF does. However, it ends up building a more accurate classification RF model than building the RF using the entire features. GRF is available in the R package `RRF` as well.

• **r2VIM**

Recurrent Relative Variable Importance Measure (r2VIM) was proposed by Szymczak et al. [41] and is implemented in the R package `Pomona` [12]. r2VIM is a test-based feature selection technique built around the RF. The method is applicable to both regression and classification problems. Like Boruta, the main goal is to find all-relevant features. In contrast, r2VIM establishes a criterion to determine the number of features that need to be selected.

The main idea of r2VIM is that relevant features have relatively high variable importance scores no matter how many times and with what seeds we run the RF. On the other hand, irrelevant features only occasionally have high importance scores.

The r2VIM algorithm involves three main components. First, it builds several RFs and calculates VIMs associated with the ensembles. Second, since observing negative VIM for a noise feature is probable, each variable importance score is divided by the observed absolute value of the minimum importance score, called the relative importance score. Finally, all variables with relative importance score larger than a threshold in all runs are selected as all-relevant features. The main advantage of r2VIM is that it is able to limit the number of false positives under the null hypothesis.

• **PIMP**

Permutation Importance (PIMP) was proposed by Altmann et al. [1] to distinguish relevant predictors from less important ones. The key idea of using permutation in PIMP is to destroy any sort of correlations between features and the response variable. PIMP is a heuristic approach that attempts to find unbiased importance scores by fitting RFs to different permutations of the response vector.

The PIMP algorithm starts by obtaining variable importance scores using the intact response vector. Then, RFs are fitted on N different permutations of the response vector, and the importance scores for all variables are calculated. Finally, based on the N sets of important scores, the p-value is computed for each variable. Here, the p-values can be obtained by computing the fraction of N importance scores that exceed the original importance score. Very often, to reduce the number of permutations, a prior distribution such as Gaussian, log-normal, or gamma can be assumed. The PIMP algorithm fits the distribution by computing the maximum likelihood estimates. After that, the p-values for each variable are defined as the probability of observing an importance score greater than the original importance score. Once the p-values for all variables have been computed, variables with p-values less than a predetermined threshold (e.g., 0.05) are statistically significant and thus selected. The R code for this method is available in the R package `vita` [5].

• **NTA**

One limitation that almost all test-based feature selection techniques is that RF does not provide a threshold for selecting features. Hereby, Janitza et al. [25] proposed a Naive and New Testing Approach (NTA), a computationally fast heuristic variable importance test, that aims to find a cutoff point in the VIMs generated by RF so as to find all-relevant features for classification problems. NTA is available in the R package `vita`. Celik [5]

NTA uses the hold-out variable importance, also known as the twofold cross-validation method. It first splits the entire dataset into two equal-sized subsets. Then, RF is applied to one of the subsets, and the variable importance scores are calculated for the other set. This process is repeated for the other subset as well. The hold-out variable importance is defined as the mean variable importance scores. The main idea of NTA is that irrelevant features do not create positive variable importance. Based on the non-positive variable importance scores, NTA constructs an approximate null hypothesis distribution and, subsequently, computes the p-values corresponding to the features of interest. Finally, the

features that are statistically significant are included as all-relevant features.

• **varSelRF and RFE**

Díaz-Uriarte and Alvarez de Andrés [11] proposed a performance-based variable selection method using RF (varSelRF) and is available in the R package `varSelRF` [10]. Using a backward elimination strategy under the aggressive variable selection framework, the goal of this method is to find the smallest possible set of variables that has relatively good predictive accuracy for either two-class or multi-class classification problems. The varSelRf algorithm calculates variable importance only once by fitting RF on all features. Then, it fits several RFs successively, and for each RF model, a predefined fraction of features with the lowest importance scores (e.g., 20%) is removed. Finally, the algorithm identifies the smallest set of features that has the smallest OOB error rate with an error rate within a chosen standard error of the minimum error rate among the fitted RFs.

It is noted that the varSelRF algorithm above ranks feature importance only once, which may not be suitable in the existence of highly correlated predictors. Hence, Guyon et al. [18] proposed a modified version of varSelRf called Recursive Feature Elimination (RFE) to handle the issue above. RFE is a performance-based feature selection algorithm that works under a backward elimination strategy to find a minimal set of features with the minimum OOB-error rate. RFE is an iterative algorithm where at each step:

◇ A RF model is fitted.

◇ Variable importance is calculated.

◇ OOB-error rate is estimated using samples that were not used in the model.

◇ A predefined ratio of the least important features is removed from the feature set.

One needs to stress that, unlike varSelRF, RFE repeatedly calculates variable importance. The algorithm stops when only a single feature remains. In the end, RFE identifies a set of features that has the minimum OOB-error rate or has an error within a small range of the minimum error. RFE is available in the R package `caret` [28].

• **VSURF**

Variable Selection Using Random Forest (VSURF) is another performance-based feature selection technique applicable to both regression and classification problems. VSURF outputs two subsets of features: one subset for interpreting purposes, including all features that are highly correlated with the response variable, and the other subset (known as the interpretation subset) for predicting purposes that exclude redundant features so that the model involves only a smaller number of features [14]. The algorithm is available in the R package `VSURF`. [15]

The VSURF algorithm consists of two steps: (*i*) preliminary elimination and ranking and (*ii*) variable selection. In the first step, a number of RFs are constructed, and the variable importance scores for all variables are obtained. Then, taking the average overall importance scores from all RF runs, variables are sorted in descending manner in terms of their importance, and those with the least importance scores are removed from the feature set.

In the second step, using features retained from step one with the same order, VSURF fits an RF to the most important variable and iteratively adds another variable until the last RF is fitted to all variables. The interpretation subset thus becomes the smallest feature set that yields the lowest OOB error rate with an error rate within its standard deviation. In particular, VSURF uses a forward selection strategy to further shrink the feature set to achieve a better prediction performance. A threshold is derived as follows:

$$\frac{1}{m-m'}\sum_{j=m'}^{m-1}|\text{errOOB}(j+1)-\text{errOOB}(j)| \tag{7}$$

where m is the number of variables selected in the first step, m' is the number of variables that exist in the interpretation subset, and $\text{errOOB}(j)$ is the OOB error of the j-th RF constructed. Hence, a variable is selected if the decrease of OOB error exceeds the threshold.

• **RF-SRC**

Random Forest for Survival, Regression, and Classification problems (RF-SRC) was proposed and developed in the R package `randomForestSRC` [24]. In the confrontation with high-dimensional data, the number of features p may exceed the sample size n. As a result, the trees in an RF model cannot reach the sufficient depth where the predictive variables can be identified. Motivated by this limitation, Ishwaran et al. [23] proposed the Variable Hunting (RSF-VH) approach, which performs as a forward stepwise regularized feature selection method.

The key idea behind RSF-VH is that features that are being split in nodes closer to the root node are likely to be more important. Hence, a new concept of order statistic called the minimal depth of maximal subtrees, is employed to calculate variable importance scores. The algorithm of RSF-VH works as follows:

◇ Fit an RF and select features using minimal depth thresholding.

◇ The set of features obtained in step (1) is used as an initial model. After that, other features are added to the initial model based on minimal depth ranking until the variable importance becomes stabilized in the nested models.

◇ Repeat steps 1 and 2 several times. Finally, a feature that appears most is selected if its size is greater than the average.

It is reported that, for very high-dimensional microarray datasets, RSF-VH is able to select a small set of features, and genes in these particular datasets, along with low predictive error compared to other state-of-the-art methods [44].

• **AUC-RF**

AUC-RF was proposed by Calle et al. [3] and is available in the R package `AUCRF` [43]. AUC-RF is a performance-based variable selection method using RFs for problems with a labeled response vector. Using a backward elimination strategy under the aggressive variable selection framework, the goal of this method is to find the set of features that has the highest area under the ROC curve (AUC).

Like varSelRf, AUC-RF fits several RF models successively. For each RF constructed, feature importance scores are obtained and features with the lowest importance scores are eliminated by a predetermined ratio between 0 and 1 (e.g., 0.2). Finally, the algorithm selects a set of features that has the highest OOB-AUC value among all RFs constructed.

Table 2 provides a summary of the advantages as well as some practical considerations of the methods reviewed above.

4 Examples and discussions

In this section, we apply the feature selection methods reviewed in Sect. 3 to three datasets and compare the results generated by different approaches. The data used in this section are obtained from the UCI Machine Learning Database Repository [13].

The first dataset is the Sonar, Mines vs Rocks (SMR) dataset [16]. This dataset contains the sonar (sound navigation and ranging) data that can be used to distinguish metal sea mines and rocks on seafloor. This dataset contains 60 features and 208 instances where 111 of them are metal samples and the rest are rock samples. The features are derived from the receiving sonar signals under various conditions and from different angles, spanning 90 degrees for the cylinder and 180 degrees for the rocks.

The second dataset is the Wisconsin Breast Cancer (WBC) dataset from patients diagnosed with cancer [45, 46]. This dataset can be used to predict whether a cancerous tumor is malignant or benign, utilizing the information obtained from biopsy procedures. The dataset has 9 features and involves 699 patients.

The third dataset is the Spam Emails (SE) dataset [21]. The data are collected from personal and work-related emails and include 4601 sample emails categorized into two groups: spam and non-spam. A total number of 57 features are available, including the ratio of 48 different words that repeat in an email over the total number of words, the frequency of 6 different characters occurring in the email, and 3 other features relating to uninterrupted sequences of capital letters.

For all three datasets, we apply RF (using all features) as well as the 12 feature selection approaches (using selected features). Figures 1, 2, and 3, respectively, show the number of features selected by each method for the three datasets.

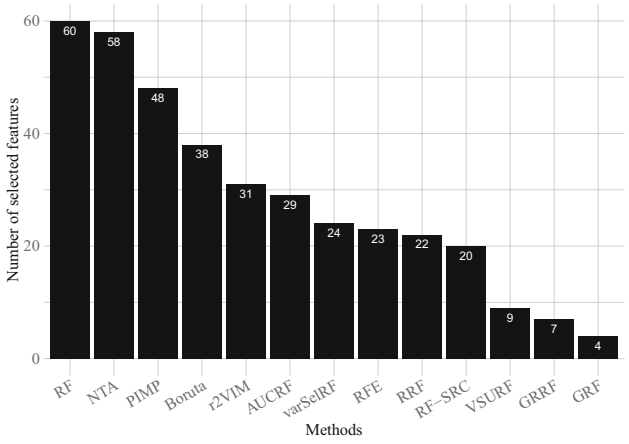

Fig. 1 Number of features selected by each method for the SMR dataset

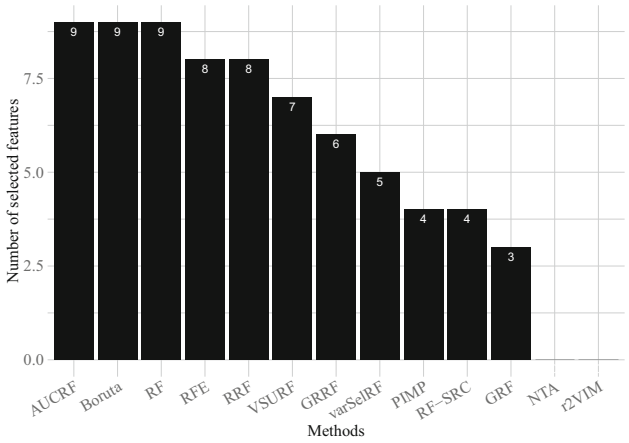

Fig. 2 Number of features selected by each method for the WBC dataset

An initial examination immediately suggests that *the number of features selected by different feature selection methods varies dramatically*. This is a natural consequence as these feature selection methods have different definitions of feature importance and use different algorithms to identify those important features. For all three datasets, GRF always selects the smallest number of features based on all three datasets, while Boruta tends to include most of the features. Such a lack of consistency strongly suggests that caution is needed when choosing feature selection methods given the problem of interest.

Furthermore, Tables 3, 4, and 5, respectively, show exactly what features are selected by different approaches for the three datasets. In these tables, "1" and "0," respectively, denote that a feature is included or excluded, and the last column shows the number of times that a feature is selected by the 12 feature selection methods. It is not surprising to see that different feature selection methods select different sets of features, as these methods are built upon different ideas.

Table 2 A summary of advantages and practical considerations for RF-based feature selection methods

Method	Advantages	Some practical considerations
Boruta	High interpretability, robustness to noise, handles mixed data types	Computationally intensive, sensitivity to hyperparameters. Requires sufficient samples, limited to small to medium-sized feature sets
RRF, GRF, and GRRF	Reduces overfitting, handles high-dimensional data, and provides feature importance	Requires parameter tuning and may be sensitive to the number of reference features. Assumes linear relationships, requires sufficient samples
r2VIM	Provides feature importance, handles high-dimensional data	Computationally expensive, may be sensitive to algorithm choice. Assumes linear relationships, requires sufficient samples
PIMP	Provides feature importance, handles high-dimensional data	Computationally expensive, may be sensitive to algorithm choice. Assumes linear relationships, requires sufficient samples
NTA	Considers variable interactions, handles high-dimensional data, provides variable importance	Computationally intensive, requires parameter tuning. Assumes linear relationships, may struggle with highly correlated features
VarSelRF	Handles both continuous and categorical variables, and handles high-dimensional datasets	Computationally expensive for large datasets, requires parameter tuning, may not capture subtle interactions or nonlinear relationships. Assumes correlation between important features and target variable, assumes linear relationships, may struggle with highly correlated features
VSURF	Considers variable interactions, handles high-dimensional data, provides variable importance	Computationally expensive, requires parameter tuning. Assumes linear relationships, may struggle with highly correlated features
RF-SRC	Considers variable interactions, handles high-dimensional data, provides feature importance	Computationally expensive, requires parameter tuning. Assumes linear relationships, may struggle with highly correlated features
AUCRF	Considers variable interactions, handles high-dimensional data, provides feature importance	Computationally expensive, requires parameter tuning. Assumes linear relationships, may struggle with highly correlated features
RFE	Reduces feature space, provides feature ranking, can handle any machine learning algorithm	Computationally intensive, sensitive to algorithm choice, may not capture complex interactions. Requires sufficient samples, may struggle with high-dimensional data

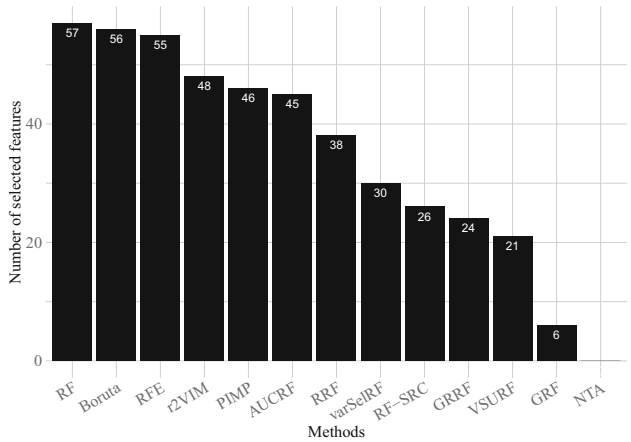

Fig. 3 Number of features selected by each method for the SE dataset

Of the all feature selection methods reviewed in this paper, only Boruta, r2VIM, varSelRF, and VSURF require one or more thresholds to be setup which we used the default values used in the packages. For example, when using Boruta, we used $p\text{-}value = 0.1$ as a significance level or a threshold for deciding whether a feature is important or not.

One interesting question to answer is to see which of these feature selection methods tend to agree with other methods. To shed some light on this question, we perform an association analysis on the features to uncover the hidden relationship among the outputs from different feature selection methods [42]. Tables 6, 7, and 8, respectively, show the results from our association analysis for the SMR, WBC, and SE datasets. For a pair of feature selection methods (given in the first two columns of these tables), association analysis returns the "support" and "confidence" of the two methods. Here, "support" refers to the proportion of times that a feature is selected by both methods, while "confidence" refers to the proportion of times that a feature is selected by the method in the second column given that this feature is selected by the method in the first column. For SMR data in Table 6, we see that the results generated by Boruta, PIMP, and NTA seem to agree with each other. For example, the empirical probability that a feature is selected by both PIMP and NTA methods reaches 80%, and a feature is also selected by NTA if this feature is selected by PIMP. However, by examining the results from Tables 6, 7, and 8, we do not see any consistent association rules among different feature selec-

Table 3 Sonar, mines vs rocks selected features

Features	Boruta	varSelRF	r2VIM	RFE	RRF	GRRF	GRF	NTA	PIMP	VSURF	AUCRF	RF-SRC	Sum
V1	1	0	1	1	0	0	0	1	1	0	0	0	5
V2	1	0	0	0	1	0	0	1	0	0	0	0	3
V3	0	0	0	0	0	0	0	1	0	0	0	0	1
V4	1	1	1	1	0	0	0	1	1	0	1	1	8
V5	1	0	1	1	0	0	0	1	1	0	1	1	7
V6	0	0	0	0	0	0	0	1	0	0	1	0	2
V7	0	0	0	0	0	0	0	1	0	0	0	0	1
V8	1	0	0	1	1	0	0	1	1	0	0	0	5
V9	1	1	1	1	0	1	1	1	1	0	1	1	10
V10	1	1	1	0	0	1	1	1	1	0	1	1	9
V11	1	1	1	0	1	1	1	1	1	1	1	1	11
V12	1	1	1	1	1	1	1	1	1	1	1	1	12
V13	1	1	1	0	1	0	0	1	1	0	1	1	8
V14	0	0	0	1	0	0	0	1	0	0	0	0	2
V15	1	1	1	0	0	0	0	1	1	0	0	0	5
V16	1	1	1	0	1	0	0	1	1	1	1	1	9
V17	1	1	1	1	1	0	0	1	0	1	1	1	9
V18	1	1	1	0	0	1	0	1	0	0	1	0	6
V19	1	0	0	0	0	0	0	1	1	0	0	0	3
V20	1	1	1	0	1	0	0	1	1	0	1	0	7
V21	1	1	1	1	0	0	0	1	1	1	1	1	9
V22	1	0	1	0	1	0	0	1	1	0	0	0	5
V23	1	1	1	1	1	0	0	1	1	0	1	0	8
V24	0	0	0	0	0	0	0	1	1	0	0	0	2
V25	0	0	0	1	0	0	0	1	1	0	0	0	3
V26	1	0	0	0	0	0	0	1	1	0	0	0	3
V27	1	1	1	0	0	0	0	1	1	1	1	0	7
V28	1	1	1	1	0	0	0	1	1	0	1	1	8
V29	1	0	0	0	0	0	0	1	1	0	0	0	3
V30	0	0	0	1	0	0	0	1	1	0	0	0	3
V31	1	0	1	1	1	0	0	1	1	0	1	0	7
V32	0	0	0	0	1	0	0	1	1	0	0	0	3
V33	0	0	0	0	0	0	0	1	0	0	0	0	1
V34	1	0	0	1	0	0	0	1	1	0	0	0	4
V35	1	1	0	0	0	0	0	1	1	0	1	0	5
V36	1	1	1	1	1	1	0	1	1	1	1	1	11
V37	1	1	1	0	0	0	0	1	1	1	1	1	8
V38	0	0	0	0	0	0	0	1	1	0	0	0	2
V39	0	0	1	0	0	0	0	1	1	0	1	0	4
V40	0	0	0	0	0	0	0	1	0	0	0	0	1
V41	0	0	0	0	0	0	0	0	0	0	0	0	0
V42	0	0	0	1	0	0	0	1	1	0	0	0	3
V43	1	0	1	1	1	0	0	1	1	0	0	0	6
V44	1	1	1	1	1	0	0	1	1	0	1	0	8
V45	1	1	1	0	1	0	0	1	1	0	1	1	8
V46	1	1	1	0	1	0	0	1	1	0	1	1	8

Table 3 continued

Features	Boruta	varSelRF	r2VIM	RFE	RRF	GRRF	GRF	NTA	PIMP	VSURF	AUCRF	RF-SRC	Sum
V47	1	1	1	1	1	0	0	1	1	0	1	1	9
V48	1	1	1	0	1	0	0	1	1	1	1	1	9
V49	1	1	1	1	0	0	0	1	1	0	1	1	8
V50	0	0	0	0	0	0	0	1	1	0	0	0	2
V51	1	0	1	1	0	0	0	1	1	0	1	1	7
V52	1	0	1	1	1	1	0	1	1	0	1	1	9
V53	0	0	0	0	1	0	0	1	1	0	0	0	3
V54	0	0	0	0	0	0	0	1	1	0	0	0	2
V55	0	0	0	0	0	0	0	1	1	0	0	0	2
V56	0	0	0	0	1	0	0	0	0	0	0	0	1
V57	0	0	0	0	0	0	0	1	0	0	0	0	1
V58	0	0	0	0	0	0	0	1	1	0	0	0	2
V59	1	0	0	0	0	0	0	1	1	0	0	0	3
V60	0	0	0	0	0	0	0	1	1	0	0	0	2

Table 4 Breast cancer selected features

Features	Boruta	varSelRF	r2VIM	RFE	RRF	GRRF	GRF	NTA	PIMP	VSURF	AUCRF	RF-SRC	Sum
Cl.thickness	1	1	0	1	1	0	0	0	1	1	1	0	7
Cell.size	1	1	0	0	1	1	1	0	1	1	1	1	9
Cell.shape	1	1	0	1	1	1	1	0	0	1	1	1	9
Marg.adhesion	1	0	0	1	1	0	0	0	0	0	1	0	4
Epith.c.size	1	0	0	1	1	1	0	0	0	1	1	0	6
Bare.nuclei	1	1	0	1	1	1	1	0	1	1	1	1	10
Bl.cromatin	1	0	0	1	1	1	0	0	0	1	1	1	7
Normal.nucleoli	1	1	0	1	1	1	0	0	1	1	1	0	8
Mitoses	1	0	0	1	0	0	0	0	0	0	1	0	3

tion methods. The results in these three tables suggest that the association among different feature selection methods depends on the problem itself (i.e., data). This observation once again demonstrates the challenges associated with feature selection. It shows the necessity of applying different feature selection methods on the same dataset. It also suggests that users' experiences, domain knowledge, and theoretical guidance are all needed in performing feature selection tasks.

We used RF as a learner for the statistical analysis. The data flow is to input data, get the subset of data with feature selection methods, train the RF classifier on the training set, and measure the performance metrics such as AUC, F1-score, and OOB-error rate on the out-of-sample set. Finally, a 10-fold cross-validation is performed for all approaches. The mean and standard deviation of the classification AUC (area under the curve), F1-score, and OOB-error rate are reported. Tables 9, 10, and 11, respectively, present the results obtained from the three datasets. Each table shows the 10-fold cross-validation AUC, F1-score, OOB-error rate, and the number of features selected by each method.

For the SMR dataset, we see from Table 9 that the number of features selected by different feature selection methods varies dramatically. While the NTA method identifies 58 features, the GRF method only includes 4 features. Remarkably, it is noted that the VSURF approach achieves the *highest* F-1 score and the *lowest* OOB-error rate by including the *smallest* number of features (i.e., only 9 out of the 60 features are included to achieve the best performance). VSURF also produces a reasonably high AUC, which is only slightly lower than that of PIMP, varSelRF, and NTA, while the latter three approaches involve way more features.

For the WBC dataset, we obtain from Table 10 a similar observation that the number of features selected by different feature selection methods can vary dramatically. While the Boruta and AUCRF approaches retain all 9 features, GRF method only selects 3 features. It is also noted that the r2VIM and NTA approaches are not able to produce valid results

Table 5 Spam emails selected features

Features	Boruta	varSelRF	r2VIM	RFE	RRF	GRRF	GRF	NTA	PIMP	VSURF	AUCRF	RF-SRC	Sum
make	1	0	0	1	1	0	0	0	0	0	1	0	4
address	1	0	1	1	0	0	0	0	1	0	1	0	5
all	1	1	1	1	1	0	0	0	1	0	1	0	7
num3d	1	0	0	1	0	0	0	0	0	0	0	0	2
our	1	1	1	1	1	1	0	0	1	1	1	1	10
over	1	1	1	1	1	1	0	0	1	0	1	0	8
remove	1	1	1	1	1	1	1	0	1	1	1	1	11
internet	1	1	1	1	1	0	0	0	1	0	1	1	8
order	1	0	1	1	1	1	0	0	1	0	1	0	7
mail	1	0	1	1	1	1	0	0	1	0	1	1	8
receive	1	1	1	1	0	0	0	0	1	0	1	1	7
will	1	1	1	1	1	1	0	0	1	1	1	1	10
people	1	0	1	1	1	0	0	0	1	0	1	0	6
report	1	0	0	1	0	0	0	0	1	0	1	0	4
addresses	1	0	1	1	0	0	0	0	1	0	0	0	4
free	1	1	1	1	1	1	1	0	1	1	1	1	11
business	1	1	1	1	1	0	0	0	1	1	1	1	9
email	1	1	1	1	1	1	0	0	1	0	1	1	9
you	1	1	1	1	1	1	0	0	1	1	1	1	10
credit	1	1	1	1	1	1	0	0	1	0	1	0	8
your	1	1	1	1	1	1	1	0	1	1	1	1	11
font	1	0	1	1	0	0	0	0	1	0	1	0	5
num000	1	1	1	1	1	0	0	0	1	1	1	1	9
money	1	1	1	1	0	0	0	0	1	0	1	1	7
hp	1	1	1	1	1	1	0	0	1	1	1	1	10
hpl	1	1	1	1	0	0	0	0	1	0	1	1	7
george	1	1	1	1	1	1	0	0	1	1	1	1	10
num650	1	1	1	1	1	0	0	0	1	1	1	0	8
lab	1	0	1	1	1	0	0	0	0	1	0	0	5
labs	1	1	1	1	0	0	0	0	1	0	1	0	6
telnet	1	0	1	1	0	0	0	0	1	0	0	0	4
num857	1	0	0	0	0	0	0	0	0	0	0	0	1
data	1	0	1	1	1	1	0	0	1	0	1	0	7
num415	1	0	0	0	0	0	0	0	0	0	0	0	1
num85	1	0	1	1	1	0	0	0	1	0	1	0	6
technology	1	0	1	1	1	0	0	0	1	0	1	0	6
num1999	1	1	1	1	1	0	0	0	1	1	1	1	9
parts	1	0	0	1	0	0	0	0	0	0	0	0	2
pm	1	0	1	1	0	0	0	0	0	0	1	0	4
direct	1	0	0	1	0	0	0	0	1	0	0	0	3
cs	1	0	0	1	1	0	0	0	0	0	0	0	3
meeting	1	1	1	1	1	1	0	0	1	1	1	1	10
original	1	0	1	1	0	0	0	0	0	0	0	0	3
project	1	0	1	1	1	0	0	0	1	0	1	0	6
re	1	1	1	1	1	1	0	0	1	1	1	1	10
edu	1	1	1	1	1	1	0	0	1	1	1	1	10

Table 5 continued

Features	Boruta	varSelRF	r2VIM	RFE	RRF	GRRF	GRF	NTA	PIMP	VSURF	AUCRF	RF-SRC	Sum
table	0	0	0	1	0	0	0	0	0	0	0	0	1
conference	1	0	1	1	0	0	0	0	1	0	0	0	4
charSemicolon	1	0	1	1	1	1	0	0	1	0	1	0	7
charRoundbracket	1	1	1	1	1	0	0	0	1	0	1	1	8
charSquarebracket	1	0	1	1	0	1	0	0	0	0	1	0	5
charExclamation	1	1	1	1	1	1	1	0	1	1	1	1	11
charDollar	1	1	1	1	1	1	1	0	1	1	1	1	11
charHash	1	0	1	1	1	0	0	0	1	0	1	0	6
capitalAve	1	1	1	1	1	1	1	0	1	1	1	1	11
capitalLong	1	1	1	1	1	1	0	0	1	1	1	1	10
capitalTotal	1	1	1	1	1	1	0	0	1	1	1	1	10

Table 6 Association analysis sorted by support on SMR dataset

Method 1	Method 2	Support	Confidence
PIMP	NTA	0.8	1
NTA	PIMP	0.8	0.82
Boruta	NTA	0.63	1
NTA	Boruta	0.63	0.65
Boruta	PIMP	0.58	0.92
PIMP	Boruta	0.58	0.72
Boruta, PIMP	NTA	0.58	1
Boruta, NTA	PIMP	0.58	0.92
NTA, PIMP	Boruta	0.58	0.72
r2VIM	NTA	0.51	1
NTA	r2VIM	0.51	0.53
r2VIM	Boruta	0.5	0.96
Boruta	r2VIM	0.5	0.78
Boruta, r2VIM	NTA	0.5	1
r2VIM, NTA	Boruta	0.5	0.96

Table 7 Association analysis sorted by support on WBC dataset

Method 1	Method 2	Support	Confidence
AUCRF	Boruta	1	1
Boruta	AUCRF	1	1
RFE	AUCRF	0.89	1
AUCRF	RFE	0.89	0.89
RFE	Boruta	0.89	1
Boruta	RFE	0.89	0.89
RRF	AUCRF	0.89	1
AUCRF	RRF	0.89	0.89
RRF	Boruta	0.89	1
Boruta	RRF	0.89	0.89
RFE, AUCRF	Boruta	0.89	1
Boruta, RFE	AUCRF	0.89	1
Boruta, AUCRF	RFE	0.89	0.89
RRF, AUCRF	Boruta	0.89	1
Boruta, RRF	AUCRF	0.89	1

Table 8 Association analysis sorted by support on SE dataset

Method 1	Method 2	Support	Confidence
RFE	Boruta	0.95	0.98
Boruta	RFE	0.95	0.96
r2VIM	RFE	0.84	1
RFE	r2VIM	0.84	0.87
r2VIM	Boruta	0.84	1
Boruta	r2VIM	0.84	0.86
r2VIM, RFE	Boruta	0.84	1
Boruta, r2VIM	RFE	0.84	1
Boruta, RFE	r2VIM	0.84	0.89
PIMP	RFE	0.81	1
RFE	PIMP	0.81	0.84
PIMP	Boruta	0.81	1
Boruta	PIMP	0.81	0.82
RFE, PIMP	Boruta	0.81	1
Boruta, PIMP	RFE	0.81	1

Table 9 Summary of RF 10-fold cross-validation using Sonar, Mines vs Rocks dataset

Method	AUC	F1-score	OOB(%)	Selected Features
RF	0.946 ± 0.036	0.824 ± 0.108	15.97 ± 1.17	60
Boruta	0.936 ± 0.040	0.802 ± 0.148	15.01 ± 1.65	38
varSelRF	0.938 ± 0.045	0.809 ± 0.096	14.79 ± 1.44	24
r2VIM	0.936 ± 0.043	0.838 ± 0.096	14.63 ± 1.54	31
RFE	0.929 ± 0.048	0.781 ± 0.137	16.93 ± 1.15	23
RRF	0.923 ± 0.049	0.798 ± 0.106	16.56 ± 1.44	22
GRRF	0.872 ± 0.073	0.777 ± 0.112	20.62 ± 1.97	7
GRF	0.800 ± 0.087	0.703 ± 0.085	26.87 ± 1.47	4
NTA	0.939 ± 0.039	0.796 ± 0.170	15.49 ± 1.65	58
PIMP	0.949 ± 0.037	0.828 ± 0.132	15.11 ± 0.88	48
VSURF	0.937 ± 0.049	0.842 ± 0.063	14.37 ± 1.03	9
AUCRF	0.936 ± 0.045	0.812 ± 0.112	15.06 ± 0.91	29
RF-SRC	0.922 ± 0.043	0.802 ± 0.140	17.25 ± 1.26	20

Table 10 Summary of RF 10-fold cross-validation using Wisconsin Breast Cancer dataset

Method	AUC	F1-score	OOB(%)	Selected Features
RF	0.993 ± 0.009	0.963 ± 0.029	2.86 ± 0.33	9
Boruta	0.993 ± 0.009	0.963 ± 0.029	2.86 ± 0.33	9
varSelRF	0.989 ± 0.011	0.949 ± 0.043	3.28 ± 0.53	5
r2VIM	NA	NA	NA	NA
RFE	0.992 ± 0.009	0.961 ± 0.035	2.50 ± 0.34	8
RRF	0.993 ± 0.008	0.963 ± 0.032	2.74 ± 0.35	8
GRRF	0.990 ± 0.011	0.948 ± 0.047	3.28 ± 0.53	6
GRF	0.990 ± 0.010	0.935 ± 0.043	4.52 ± 0.57	3
NTA	NA	NA	NA	NA
PIMP	0.989 ± 0.010	0.942 ± 0.052	3.52 ± 0.43	4
VSURF	0.992 ± 0.009	0.961 ± 0.029	2.89 ± 0.44	7
AUCRF	0.993 ± 0.009	0.961 ± 0.033	2.63 ± 0.25	9
RF-SRC	0.990 ± 0.010	0.927 ± 0.034	4.63 ± 0.61	4

Table 11 Summary of RF 10-fold cross-validation using the Spam Emails dataset

Method	AUC	F1-score	OOB(%)	Selected Features
RF	0.986 ± 0.006	0.936 ± 0.010	4.93 ± 0.14	57
Boruta	0.986 ± 0.006	0.935 ± 0.012	4.91 ± 0.10	56
varSelRF	0.986 ± 0.005	0.937 ± 0.010	4.91 ± 0.10	30
r2VIM	0.986 ± 0.006	0.938 ± 0.011	4.91 ± 0.05	48
RFE	0.986 ± 0.006	0.937 ± 0.012	4.95 ± 0.13	55
RRF	0.986 ± 0.005	0.941 ± 0.009	4.71 ± 0.11	38
GRRF	0.985 ± 0.005	0.935 ± 0.010	5.04 ± 0.07	24
GRF	0.957 ± 0.006	0.895 ± 0.010	7.82 ± 0.23	6
NTA	NA	NA	NA	NA
PIMP	0.986 ± 0.005	0.932 ± 0.010	4.82 ± 0.17	46
VSURF	0.986 ± 0.005	0.936 ± 0.010	4.79 ± 0.16	21
AUCRF	0.986 ± 0.005	0.938 ± 0.012	4.76 ± 0.11	45
RF-SRC	0.986 ± 0.005	0.936 ± 0.007	5.02 ± 0.12	26

(indicated by "NAs") in this table. As already mentioned in Sect. 3, this is because the r2VIM and NTA algorithms lean on negative variable importance created by non-relevant features, and error messages are returned when running the code.

For the SE dataset, all methods but NTA are able to produce valid results, as shown in Table 11. The number of features selected by different feature selection methods again varies dramatically. GRF approach identifies a much smaller number of features. However, it also has the worst overall performance among the 13 approaches. Because all methods (except for GRF) yield similar predictive performance, one naturally prefers those methods that include a smaller feature set features for a better balance of model complexity and model performance, such as VSURF (21 features), GRRF (24 features), and RF-SRC (26 features).

5 Conclusion

This paper provided a comprehensive review and discussion of 12 commonly used RF-based feature selection methods for classification problems. The fundamental ideas behind each method were described, the R packages were introduced, and numerical studies were presented that illustrate the implementation of these methods and compare the results generated from different methods. The numerical examples show that different methods often identify different important features as these feature selection methods are based on different ideas and approaches. This observation indicates that caution is often required when performing feature selection tasks. It is a good practice to try more than one feature selection method and identify important features by examining the outcomes from different approaches integrated with users' experiences, domain knowledge, and theoretical analysis. The paper serves as a one-stop reference where students, data science consultants, engineers, and data scientists can access the basic ideas behind these methods, the advantages and limitations of different approaches, as well as the software packages to implement these methods.

Author Contributions The first author, RI, was the Ph.D. student and graduate research assistant of the corresponding author. RI performed the literature review, completed the numerical examples, and prepared the draft of the paper. The corresponding author, XL, provided his advice during the process, led research meetings and discussions, and revised the manuscript.

Funding This material was partially supported by the National Science Foundation under Award No. OIA-1946391.

Data availability All datasets are publicly available and the sources of data are stated in the paper.

Declarations

Conflict of interest The authors are not aware of any potential conflict of interest.

Human and animals participants The paper does not involve human participants and/or animals.

Informed consent The paper does not involve any informed consent.

References

1. Altmann, A., Toloşi, L., Sander, O., Lengauer, T.: Permutation importance: a corrected feature importance measure. Bioinformatics **26**, 1340–1347 (2010)
2. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
3. Calle, M.L., Urrea, V., Boulesteix, A.-L., Malats, N.: AUC-RF: a new strategy for genomic profiling with random forest. Hum. Hered. **72**, 121–132 (2011)
4. Capstone: 6th Annual Industrial Engineering Capstone Symposium, Industrial Engineering, University of Arkansas

(2022). https://industrial-engineering.uark.edu/academics/undergraduate-program/capstone-2021-2022.php
5. Celik, E.: vita: variable importance testing approaches, r package version 1.0.0 (2015)
6. Deng, H.: Guided random forest in the RRF package, arXiv preprint arXiv:1306.0237 (2013)
7. Deng, H., Runger, G.: Feature selection via regularized trees. In: The 2012 International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 1–8 (2012)
8. Deng, H., Runger, G.: Gene selection with guided regularized random forest. Pattern Recogn. **46**, 3483–3489 (2013)
9. Detzner, A., Eigner, M.: Feature selection methods for root-cause analysis among top-level product attributes. Qual. Reliab. Eng. Int. (2020). https://doi.org/10.1002/qre.2738
10. Diaz-Uriarte, R.: GeneSrF and varSelRF: a web-based tool and R package for gene selection and classification using random forest. BMC Bioinform. **8**, 1–7 (2007)
11. Díaz-Uriarte, R., Alvarez de Andrés, S.: Gene selection and classification of microarray data using random forest. BMC Bioinform. **7**, 1–13 (2006)
12. Fouodo, C.: Pomona: identification of relevant variables in omics data sets using Random Forests, r package version 1.0.2 (2022)
13. Frank, A.: UCI machine learning repository (2010). http://archive.ics.uci.edu/ml
14. Genuer, R., Poggi, J.-M., Tuleau-Malot, C.: VSURF: an R package for variable selection using random forests. R J. **7**, 19–33 (2015)
15. Genuer, R., Poggi, J.-M., Tuleau-Malot, C.: VSURF: Variable Selection Using Random Forests, r package version 1.1.0 (2019)
16. Gorman, R.P., Sejnowski, T.J.: Analysis of hidden units in a layered network trained to classify sonar targets. Neural Netw. **1**, 75–89 (1988)
17. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
18. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. **46**, 389–422 (2002)
19. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer, New York (2009)
20. Ho, T.K.: Random decision forests. In: The 3rd International Conference on Document Analysis and Recognition, pp. 278–282 (1995)
21. Hopkins, M., Reeber, E., Forman, G., Suermondt, J.: Spambase data set, Hewlett-Packard Labs, 1 (1999)
22. Hua, J., Xiong, Z., Lowey, J., Suh, E., Dougherty, E.R.: Optimal number of features as a function of sample size for various classification rules. Bioinformatics **21**, 1509–1515 (2005)
23. Ishwaran, H., Kogalur, U.B., Gorodeski, E.Z., Minn, A.J., Lauer, M.S.: High-dimensional variable selection for survival data. J. Am. Stat. Assoc. **105**, 205–217 (2010)
24. Ishwaran, H., Kogalur, U.B., Kogalur, M.U.B.: Package randomForestSRC. Breast **6**, 1 (2022)
25. Janitza, S., Celik, E., Boulesteix, A.-L.: A computationally fast variable importance test for random forests for high-dimensional data. Adv. Data Anal. Classif. **12**, 885–915 (2018)
26. Jirapech-Umpai, T., Aitken, S.: Feature selection and classification for microarray data analysis: evolutionary methods for identifying predictive genes. BMC Bioinform. **6**, 1–11 (2005)
27. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artif. Intell. **97**, 273–324 (1997)
28. Kuhn, M.: caret: Classification and Regression Training, r package version 6.0-86 (2020)
29. Kursa, M.B., Rudnicki, W.R.: Feature selection with the Boruta package. J. Stat. Softw. **36**, 1–13 (2010)
30. Kursa, M.B., Rudnicki, W.R.: The all relevant feature selection using random forest, arXiv preprint arXiv:1106.5112 (2011)
31. Lee, J.W., Lee, J.B., Park, M., Song, S.H.: An extensive comparison of recent classification tools applied to microarray data. Comput. Stat. Data Anal. **48**, 869–885 (2005)
32. Liu, H., Liu, L., Zhang, H.: Ensemble gene selection for cancer classification. Pattern Recogn. **43**, 2763–2772 (2010)
33. Liu, X., Pan, R.: Analysis of large heterogeneous repairable system reliability data with static system attributes and dynamic sensor measurement in big data environment. Technometrics **62**, 206–222 (2020)
34. Liu, X., Pan, R.: Boost-R: gradient boosting for recurrent event data. J. Qual. Technol. **53**, 545–565 (2021)
35. Mahajan, S., Pandit, A.K.: Hybrid method to supervise feature selection using signal processing and complex algebra techniques. Multimed. Tools Appl. (2021). https://doi.org/10.1007/s11042-021-11474-y
36. Mansoor, M., Ur Rehman, Z., Shaheen, M., Khan, M.A., Habib, M.: Deep learning based semantic similarity detection using text data. Inf. Technol. Control **49**, 495–510 (2020)
37. Pudjihartono, N., Fadason, T., Kempa-Liehr, A.W., O'Sullivan, J.M.: A review of feature selection methods for machine learning-based disease risk prediction. Front. Bioinform. (2022). https://doi.org/10.3389/fbinf.2022.927312
38. Ruiz, R., Riquelme, J.C., Aguilar-Ruiz, J.S.: Incremental wrapper-based gene selection from microarray data for cancer classification. Pattern Recogn. **39**, 2383–2392 (2006)
39. Shaheen, M., Shahbaz, M.: An algorithm of association rule mining for microbial energy prospection. Sci. Rep. **7**, 46108 (2017)
40. Speiser, J.L., Miller, M.E., Tooze, J., Ip, E.: A comparison of random forest variable selection methods for classification prediction modeling. Expert Syst. Appl. **134**, 93–101 (2019)
41. Szymczak, S., Holzinger, E., Dasgupta, A., Malley, J.D., Molloy, A.M., Mills, J.L., Brody, L.C., Stambolian, D., Bailey-Wilson, J.E.: r2VIM: a new variable selection method for random forests in genome-wide association studies. BioData Min. **9**, 1–15 (2016)
42. Tan, P.N., Steinbach, M., Karpatne, A., Kumar, V.: Introduction to Data Mining, 2nd edn. Pearson, London (2019)
43. Urrea, V., Calle, M.: AUCRF: Variable Selection with Random Forest and the Area Under the Curve, r package version 1.1 (2012)
44. Wang, H., Li, G.: A selective review on random survival forests for high dimensional data. Quant. Bio Sci. **36**, 85 (2017)
45. Wolberg, W.H., Mangasarian, O.L.: Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proc. Natl. Acad. Sci. **87**, 9193–9196 (1990)
46. Zhang, J.: Selecting typical instances in instance-based learning. In: Machine Learning Proceedings 1992. Elsevier, pp. 470–479 (1992)
47. Zhu, Z., Ong, Y.-S., Dash, M.: Markov blanket-embedded genetic algorithm for gene selection. Pattern Recogn. **40**, 3236–3248 (2007)