Manuscript for *Expert Systems with Applications*

A Comparison of Random Forest Variable Selection Methods for Regression Modeling of Continuous Outcomes

**Authors**: Nathaniel S O’Connell a, Byron C Jaeger b, Garrett S Bullock c, Jaime Lynn Speiser d

a Department of Biostatistics and Data Science, Wake Forest University School of Medicine, Winston-Salem, NC 27157, USA (n.oconnell@wakehealth.edu)

b Department of Biostatistics and Data Science, Wake Forest University School of Medicine, Winston-Salem, NC 27157, USA (bjaeger@wakehealth.edu)

c Departments of Orthopedic Surgery and Biostatistics and Data Science, Wake Forest University School of Medicine, Winston-Salem, NC 27157, USA (gbullock@wakehealth.edu)

d Department of Biostatistics and Data Science, Wake Forest University School of Medicine, Winston-Salem, NC 27157, USA (jspeiser@wakehealth.edu)

**Author contact**:

Jaime Lynn Speiser

Wake Forest University School of Medicine

Division of Public Health Sciences

Medical Center Boulevard

Winston-Salem, NC 27157

[jspeiser@wakehealth.edu](mailto:jspeiser@wakehealth.edu)

Phone: (336)-713-3469

**Funding:** Research reported in this publication was supported by the National Center for Advancing Translational Sciences of the National Institutes of Health under Award Number UL1TR001420. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

**Declarations of interest**: none.

**Author contributions:**

NSO: Analysis, funding acquisition, methodology, writing original draft

BCJ: Analysis, funding acquisition, methodology, editing draft

GSB: Funding acquisition, methodology, editing draft

JLS: Conceptualization, analysis, funding acquisition, methodology, writing original draft

**Abstract**

Random forest (RF) regression is frequently used to develop prediction models for continuous outcomes. Variable selection (VS), i.e., selecting a subset of predictor variables to use in prediction models, is also frequently used to improve the prediction accuracy and computational efficiency of RFs and also reduce the burden of data collection. Several VS methods leveraging RFs have been proposed, but there is limited evidence to guide decisions on which VS methods may be preferable for different types of datasets. Using 59 publicly available datasets, we evaluated 13 VS techniques that leverage RFs. Performance of VS was measured via out-of-sample R2 of a RF that used the variables selected by the technique. Simplicity of VS was measured via the percent reduction in the number of variables selected versus number of variables available. Efficiency was measured via computational time required to complete the VS technique. We compared VS techniques by summarizing these three metrics overall and in subgroups based on the type of random forest used for prediction (axis-based or oblique), the number of predictors and observations in training data, and among test based or performance based VS techniques. Based on our benchmarking study, the best variable selection methods for most datasets were the method in the R package Boruta and Menze’s method for axis-based random forest models and Menze’s method and permute-oblique for oblique random forest models. A significant contribution of this study is the ability to assess different variable selection techniques in the setting of random forest regression for continuous outcomes to identify preferable methods based on applications in expert and intelligent systems using an open science approach.

**Keywords**: random forest, variable selection, feature reduction, regression, continuous outcome

**1 Introduction**

Random forest (RF) regression is a popular machine learning algorithm used to develop models for continuous outcomes. First introduced in 2001 (Breiman, 2001), RFs are a collection of randomized decision trees (Breiman, Friedman, Olshen, & Stone, 1984). In a regression setting, the RF computes predictions by taking the mean of predictions from each tree. Prior studies have shown RFs can provide better prediction accuracy compared to a single decision tree, regression model, and other machine learning algorithms (refs). RFs also allow for interpretation of relationships between predictors and outcome (Speiser, Durkalski, & Lee, 2015).

In addition to developing prediction models, the RF can also be used to conduct variable selection (VS), i.e., selecting a subset of predictors to include in a prediction model. VS can improve prediction accuracy and decrease the burden of applying prediction models in practice. For example, rather than using all variables available in a registry dataset, one may prefer to use only a subset of the most important variables when developing a model (17, 18). For datasets with many possible predictor variables, VS is a critical step as it can remove superfluous variables and improve the prediction accuracy of downstream models, i.e., models that are trained using the selected subset of variables.

As many methods to fit RFs and conduct VS with them have been developed since their introduction in 2001, general comparisons between existing methods using real data can offer empirical edVS RFsfew studies have evaluated VS methods for regression RFs.However, RF VS have differential performance in classification versus regression due to differing outcome definitions, performance statistics, and criteria for growing decision treesTo provide data-driven recommendations for regression, we edusing publicly available data RF VS

The remainder of this paper is structured in the following manner. Section 2 summarizes methods and implementation for random forest variable selection for regression in the current literature. The design of the current study is presented in Section 3, including the datasets used and evaluation metrics for the variable selection procedures. Section 4 provides a summary of results comparing performance, parsimony and computation time for the variable selection procedures. Discussion and conclusions are presented in Section 5.

**2 Methods for Random Forest Variable Selection for Regression**

We included a total of 13 RF VS methods in the current analysis (Cadenas, et al., 2013; Cano, et al., 2017; Degenhardt, et al., 2017; Hapfelmeier & Ulm, 2013; Sanchez-Pinto, et al., 2018). Each method is described in detail by its corresponding paper, which is presented alongside a brief description of the method in Table 1. In brief, Svetnik, Jiang, Caret, Menze, Negation, and Permutation – Oblique each use a specific form of recursive feature elimination, a stepwise procedure where a single predictor is dropped at each step until a stopping criterion is met. Altmann, Boruta, and Hapfelmeier use permutation tests to identify significant predictors. RRF uses forward selection based on regularization, SRC uses minimal depth of maximal subtrees, and VSURF applies a three step (threshold, interpret, and predict) selection procedure,

Methods that use a backward elimination approach with conditional inference forest include Svetnik’s method (2004) and Jiang’s method (2004). Hapfelmeier employs a permutation test based approach based on conditional inference forests (2013, 2023) . Menze (2011) and Jaeger (2023) use backward selections procedures for oblique random forests. Some methods use a backward elimination procedure with standard implementation of random forest, including *caret* (Kuhn, 2008) and *randomForestSRC* (Ishwaran & Kogalur, 2014). A stepwise selection procedure is implemented in *VSURF* (Genuer, et al., 2015), whereas *RRF* (Deng & Runger, 2013) uses a regularized random forest procedure and a forward selection approach. Altmann’s method (2010) and *Boruta* (Kursa & Rudnicki, 2010) use random forest importance measures to perform variable selection. We additionally use permutation variable importance values greater than zero to select variables for axis-based and oblique random forest methods.

**Table 1**: Summary of variable selection methods for random forest regression

Similar to Hapfelmeier’s categorization (2013) and our previous study (ref), we define variable selection methods as being test based or performance based. Performance based approaches select variables based on changes in the prediction accuracy of the continuous outcome when variables are added or deleted from models, and include methods by Svetnik, Jiang and Menze, and the R packages *caret, RRF, randomForestSRC,* and *VSURF*. Test based approaches select variables based on statistical or permutation tests on the variables themselves, and include methods by Altmann and Hapfelmeier, and the R packages *Boruta*, *aorsf* (permutation method)*, ranger* (permutation method). Our goal was to be as inclusive as possible in terms of using all available variable selection methods for random forest regression with available R code to thoroughly evaluate and compare methods.

**3 Design of Study**

All datasets analyzed were pulled from and are freely available with the *OpenML* (website: https://www.openml.org/home) and *modeldata*  R packages (Casalicchio, et al., 2017). We included datasets from these repositories designated as supervised regression tasks with less than 50% missing observations. We restricted this set to include datasets with 10 -1000 predictors, and 100 - 10,000 observations. Simulated datasets were excluded and redundant datasets (i.e. data sets in which there exist multiple versions) were limited to the most recent version. Last, we required the continuous outcome to have at least 10 unique values.

Based on these criteria we analyzed a total of 59 data sets. We provide a summary of dataset characteristics in Figure 1, which demonstrates the distributions for the number of predictors, number of observations, and coefficient of variation (CV). In general, each of these characteristics was right-skewed. The number of predictors ranged from 10 to 614, with a median of 22. The number of observations ranged from 120 to 10,000 with a median of 534. Most datasets (46; 78%) had a CV <1 indicating low variability relative to the mean, while 4 data had high variability with CV > 10. The CVs ranged near 0 to 86, with a median of 0.48. The ratio of predictors to sample size (P:N) ranged from 0.001 to 0.65, with a median of 0.04. In total, 13 datasets had fewer than 10 observations per predictor (i.e. P:N ratio > 0.1). In total 7 datasets contained missing data (11%), with missing proportions of 1%, 2%, 3%, 11%, 13%, 42%, and 50%. Details about the characteristics of datasets used in this study are included within the Supplementary Dataset Listing File. Datasets are from a variety of domains including medicine, manufacturing, agriculture, economics, education, and more.

**Figure 1:** Characteristics of datasets used for the study

We used repeated 5 replications of split sample validation (i.e., Monte-Carlo cross validation) for each dataset to evaluate RF VS methods. First, a dataset was split into training (75%) and testing (25%) sets. Second, each VS technique was applied to the training data, and the variables selected by each technique were saved. Third, each set of variables were used to fit an axis-based and oblique RF. Fourth, predictions for the testing data were computed with each RF. If any missing values were present in the training or testing data, they were imputed prior to running VS methods using the mean and mode for continuous and categorical predictors, respectively, computed in the training data.

For methods from available R packages, we assumed default parameters for each method, unless noted otherwise in Table 1. Computation time and the number of variables were recorded. We then fit a standard axis-based random forest model using the R package *ranger* and a random oblique forest using the package *aorsf* on the training data set using variables selected from each method, and recorded R-squared for each method from the test data. For comparison, we fit a standard random forest model using no form of variable selection. Some of the simulation runs for certain datasets and data splits resulted in some methods not selecting any variables. We recorded cases for which no variables were selected and forced the model to use the most important variable using the permutation importance metric as the one predictor variable included. We also provide a sensitivity analysis in which any simulation run containing a method that selected no variables was deleted for that dataset and random split for all methods.

We summarize results evaluating computation time (and log-computation time), percent variable reduction, and R-squared aggregated across all dataset replications for each method. Additionally, we compare variable selection methods based on standardized metrics, where for each dataset replication, we calculate z-scores for log-computation time, percent variable reduction and r-squared by variable selection method by dividing the mean of the metric by the standard deviation of the metric. These standardized outcomes allow us to compare how each method, in general, performs relative to other methods for a given replicate (i.e. random split) of a given dataset.

We used R version 4.3.0 on a computer with an AMD Ryzen 9 3900x 12-core/24-thread 3.8 GHz CPU with 2 x 16GB 3200 MHz DDR4 RAM. Code used to implement and evaluate the variable selection methods is provided on Github at: https://github.com/NateOConnellPhD/rfvs-regression.

**4 Results**

4.1 Results for all datasets

Performance statistics were aggregated across replications and datasets by method of variable selection. We present descriptive statistics for R-squared across the 13 variable selection procedures (plus RF with no variable selection) for both axis- and oblique forests, along with computation time (in seconds) and percent variable reduction in Table 2. Figures 2 – 5 demonstrate both original and standardized distributions by replicated sample for log-time, percent variable reduction. RF with no variable selection is included in Figures 4 and 5 for accuracy comparison but excluded from the computation time and percent reduction in Figures 2 and 3.

**Table 2**: Distribution of R-Squared, Computation Time, and Percent Variable Reduction for variable selection procedures

First, we present computational efficiency of the methods as time in seconds (Table 2, Figure 2). In terms of computation time, the fastest variable selection methods were Permute-Axis, RRF, Menze, Permutation Oblique, and Negation, each faster than average (i.e. z-scores <0 on the standardized plot), with median computation times less than 5 seconds. The least efficient methods were Hapfelmeier, CARET, and Svetnik, with median computation times of 144 seconds, 171 seconds, and 245 seconds respectively. These three methods had some simulation runs where variable selection took more than 2 hours, while permutation-axis had the fastest instance, running in only 2.65 seconds. Generally, each approach demonstrated similar variation in computation time across dataset replications.

**Figure 2: Distributions of Log computation time and Standardized Log-computation Time**

We assess model simplicity in terms of the percent reduction in the variables selected by the methods (Table 2, Figure 3). Percent variable reduction ranged from a median of more than 90% for Hapfelmeier, to near 0% for RRF. The most restrictive methods achieving the largest median percent reduction were Hapfelmeier (93%), VSURF (84%), Altman (82%), and Svetnik (76%). The selection methods CARET, Boruta, and SRC, had the greatest variability in percent variable reduction, demonstrating the widest range of IQRs for both absolute percent reduction as well as standardized percent reduction. In contrast, Hapfelmeier had the smallest variability in percent reduction.

**Figure 3: Distributions of Percent Reduction (Absolute and Standardized)**

We present model performance using R-squared (Table 2, Figures 4-6). There was not a large difference in the distribution of R-squared across datasets by method of selection for axis-based (Figure 4) or oblique (Figure 5) random forest models, with the exception of Hapfelmeier which achieved lower median R-squared across datasets. The standardized distribution comparisons corroborate this, showing all methods are within +/-1 SD of average performance by dataset replication. Comparing ranges in performance, VSURF, Svetnik, RRF, and Altman were the most variable in performance relative to other methods by dataset for both axis-based and oblique forests based on standardized comparisons (Figures 4 and 5). Oblique forests generally provide higher R squared compared to axis-based forests for all variable selection methods (Figure 6). Overall, the best performers in terms of mean and median R-squared were oblique forests based on the two oblique forest methods of variable selection: Menze and Permutation-Oblique.

**Figure 4: R-Squared for Axis Forests Variable Selection Method**

**Figure 5:** **R-Squared for Oblique Forests Variable Selection Method**

**Figure 6: Comparison of Mean and Median R-Squared by Forest type and Method of Variable Selection**

Finally, we compare performance based on all three metrics: median accuracy measured by R-squared, median computation time, and median percent reduction in variables (Table 2, Figure 7). Top performers are displayed in Panel B of Figure 7 for axis-based and oblique forests. Hapfelmeier’s method, Svetnik’s method, and CARET had higher computation times compared to the other methods (Panel A in Figure 7). As such, these methods are not considered “top performers” for the following analysis of the performance metrics. The fastest methods were permutation approaches (axis and oblique), RRF, Negation, Boruta, and Menze, all with median computation times <10 seconds. The most accurate methods were Boruta, Menze, and Jiang for axis forests and Permutation-Oblique and Menze for oblique forests. The most parsimonious models with the highest median percent reduction in variables were VSURF and Altman, with median percent reduction of greater than 80%, with Menze and Jiang as the next most parsimonious with median percent reduction greater than 65%. Using a holistic approach across all three metrics for axis-based forests, Boruta and Menze demonstrate high R-squared, low computation time, and high percent reduction in variables relative to the other methods. For oblique fitted forests, Permutation-Oblique and Menze have high R-squared, low computation time, and high percent reduction in variables relative to the other methods.

**Figure 7: Comparison of Accuracy by Time and Percent Reduction**

4.2 Sensitivity Analysis: Excluding runs for which any method had no variables selected

Conducing this benchmarking study, we found that some methods did not select any variables for certain datasets and splits of the data (i.e., simulation runs). We summarize the number of times this occurred, across all replications and the number of times it occurred in at least one replication by dataset (Table 3). Notably, in 94 out of 290 replications (58 datasets by 5 replications each), and occurring in 42 of the 58 datasets, no variables were selected based on Hapfelemeier’s method. This issue also occurred, albeit more scarcely with Altman’s method and VSURF in 5 and 4 datasets respectively. The other methods not included in Table 3 selected variables for every simulation run.

**Table 3: Instances when no variables were selected**

As a sensitivity analysis, we present the comparison of accuracy by time and percent reduction for a complete case analysis, in which we compare methods only on dataset replications in which all methods selected at least one variable (Figure 8). This sensitivity analysis consisted of 2,464 replications (compared to 4,260 in the primary analysis) across 53 datasets. The median performance in terms of R-squared decreased across all datasets with the elimination of the datasets completely excluded, however the general conclusions remain the same. For axis-based forests, Boruta and Permutation-Oblique had high R-squared, low computation time, and high percent reduction in variables selected. For oblique forests, Menze and Permutation-Oblique had had high R-squared, low computation time, and high percent reduction in variables selected.

**Figure 8: Comparison of Accuracy by Time by Percent Reduction (Complete Case)**

4.3 Results for datasets with high versus low P:N Ratio

We conducted a subgroup analysis by datasets that had differing ratios of the number of predictors to the sample size (Figure 9). There were 13 datasets with a P:N >0.10, defined as high, and the remaining datasets had P:N <0.10, defined as low. A dataset with a high P:N ratio corresponds to a dataset with more than 10 observations per predictor. In the scenario of high P:N ratio, oblique forests provide a better fit than axis-based forests for all variable selection methods, with the oblique-forest based selection methods (Permutation-Oblique and Menze) providing the best results. However, for datasets with low P:N ratio, the results are much closer between forests types, with the difference in median R-squared between forests among the top performing variable selection types being less than 1%. In this scenario, methods that had the highest R squared were Menze, Jiang, Boruta and Permute-Oblique.

**Figure 9: Median R-Squared by Variable Selection type based on High vs Low P:N**

4.4 Comparing methods grouped by characteristics of the methods

4.4.1 Results for comparing axis-based, conditional and oblique random forests in the variable selection method

The type of random forest implemented in each of the variable selection methods differed (Table 1). Most methods used standard random forest, with some using conditional random forest (Svetnik’s method, Jiang’s method, and Hapfelmeier’s method) and some using oblique random forest (Menze’s method, Negation, and permutation-oblique). In this section, we analyze Figure 7 by random forest implementation type. Methods employing conditional inference forest were computationally slow. The methods using oblique random forest performed fairly well in terms of low computation time and accuracy, although negation had lower accuracy when oblique random forest was also used in the modeling. Of the methods using standard random forest, Boruta had the highest accuracy and lowest computation time, while VSURF had fairly high accuracy it had longer median computation times.

4.4.2 Results for comparing test based and performance based methods

In addition to the type of random forest method used in the variable selection, we categorize methods as test based and performance based (Table 1). Test based methods use tests on the variables, either statistical or permutation based, to determine which variables should be in the model, and included Altman, Boruta, Permutation-Oblique, Permutation-Axis, and Hapfelmeier. The other methods were all considered performance based, which uses changes in model performance to determine variables to include. Comparing performance via computation time and accuracy in Figure 7, there were no discernable patterns when comparing test based and performance based methods. For axis based random forests, Boruta and Menze had the highest accuracy and lowest computation time, and these were test based and performance based, respectively. Similarly for oblique random forests, Permute-Oblique and Menze had the highest accuracy and lowest computation time, and these were test based and performance based, respectively.

**5 Discussion**

We presented a comparison of variable selection methods for random forest modeling of continuous outcomes using 59 datasets freely available in R packages. Variable selection methods assessed include methods based on standard random forest, conditional inference forests, and oblique forests. As such, we compare final fitted model results using random forests and oblique forests. The best methods are considered to be those that provide the best accuracy in terms of R-Squared first and foremost, with low computation time and parsimony favored among competing methods with comparable accuracy.

Overall, the best variable selection method Menze’s method, which uses an oblique random forest approach. This method provided among the best performance in terms of R-squared, with a median computation time less than 5 seconds and a median of 67% variable reduction across the random dataset split replications. Menze’s method provides good results whether the final model is fit by standard random forest or oblique forest and performs well for data that with both low and high P:N ratios. Boruta performed close to Menze for axis-based random forests, but had a lower median R-squared than Menze for oblique forests. Similarly, Permutation-Oblique performs similarly to Menze for oblique forests, but not as well in axis-based forests. Jiang’s method and VSURF perform similarly to Boruta, but with slower median computation times. RRF and SRC perform similarly in computation time to Menze, Boruta, and Permutation-Oblique, but have slightly worse performance in terms of r-squared and variable percent reduction. CARET and Svetnik’s method provide comparable accuracy in terms of r-squared but have high computation time, with some replications on the more complex data sets taking over 2 hours to run. Hapfelmeier’s method had the highest computation time and largest percent reduction in variables, failing to select a single variable in at least one replication of 42 of the 59 datasets.

A novel contribution of our study is consideration of axis-based and oblique random forest implementations. We found that oblique fitted forests generally led to better overall accuracy in terms of r-square than axis-based random forests, regardless of the methodology used for variable selection. In the overall results, the magnitude of this difference was largely driven by datasets with a large P:N ratio. In our secondary analyses, we found that in datasets with less than 10 observations per included predictor, oblique forests provided better accuracy than axis-based forests, which is consistent with previous investigations of oblique forests (Menze 2011). For datasets with a smaller P:N ratio, the differences between axis-based and oblique forests for a given variable selection method were much smaller (within 1% in R-squared) for the top 5 performing variable selection methods. In fact, for datasets with P:N <0.1, there was a small difference in performance between Menze, Jiang, Boruta, and Permutation-Oblique with respect to R-squared.

Our results should be considered in the greater context of previous literature comparing random forest variable selection methods for continuous outcomes. Most previous work has focused on classification and prediction of categorical outcomes (Cadenas, et al., 2013; Sanchez-Pinto, et al., 2018,). These studies suggested that VSURF and Boruta may be preferable. In our previous work for categorical outcomes (Speiser ref), VSURF and Jiang’s method were identified as optimal for datasets with binary outcomes, whereas varSelRF and Boruta were preferable for large datasets. Degenhardt (ref) used simulated and real data for continuous outcomes and found that Boruta and Altman are suitable for low dimensional data. This is consistent with our finding that Boruta is a top performer; however, the study by Degenhardt and colleagues (ref) did not include oblique random forest methods or implementations as in our study. It was interesting that VSURF did not perform as well as other methods in our study for continuous outcomes, since it was consistently identified as a top performer for classification models. Another major difference between our work and previous studies is that methods for variable selection using conditional inference forest were not optimal performers. This is largely because conditional inference forests have high computation times, whereas axis-based and oblique random forest implementations have alternative implementations that are quite fast.

There are some limitations of our study. We were constrained to 59 datasets that were freely available in R packages. Regression datasets with continuous outcomes are less available than classification datasets as analyzed in our previous work (ref). However, this is a sizeable number of datasets from different domains, and we believe it provides a suitable basis for benchmarking random forest variable selection methods. A few of the datasets had missing values, but we decided to include them to increase the number of datasets included in the study. Given that only 7 of the 59 datasets had missing values, the imputation of missing values likely did not bias the study results much. A future study could investigate the impact of missing data on random forest variable selection. Despite these limitations, our study provides a thorough benchmarking experiment to compare random forest variable selection methods in terms of accuracy, computational efficiency, and parsimony.

A primary contribution of our study is the ability to assess random forest variable selection methods for continuous outcomes using axis-based and oblique random forest models. Based on our benchmarking study of 59 datasets and optimizing accuracy, computational efficiency, and parsimony, Boruta and Menze’s method may be preferable for axis-based random forest models, whereas permute-oblique and Menze’s method may be preferable for oblique random forest models.

**References**

Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. *Bioinformatics, 26*, 1340-1347.

Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5-32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees.* Monterrey, CA, USA: Wadsworth and Brooks.

Cadenas, J. M., Garrido, M. C., & MartíNez, R. (2013). Feature subset selection filter–wrapper based on low quality data. *Expert Systems with Applications, 40*, 6241-6252.

Cano, G., Garcia-Rodriguez, J., Garcia-Garcia, A., Perez-Sanchez, H., Benediktsson, J. A., Thapa, A., & Barr, A. (2017). Automatic selection of molecular descriptors using random forest: Application to drug discovery. *Expert Systems with Applications, 72*, 151-159.

Casalicchio, G., Bossek, J., Lang, M., Kirchhoff, D., Kerschke, P., Hofner, B., Seibold, H., Vanschoren, J., & Bischl, B. (2017). OpenML: An R package to connect to the machine learning platform OpenML. *Computational Statistics*, 1-15.

Conn, D., Ngun, T., Li, G., & Ramirez, C. (2015). Fuzzy Forests: Extending Random Forests for Correlated, High-Dimensional Data.

Degenhardt, F., Seifert, S., & Szymczak, S. (2017). Evaluation of variable selection methods for random forests and omics data sets. *Briefings in bioinformatics*.

Deng, H., & Runger, G. (2013). Gene selection with guided regularized random forest. *Pattern Recognition, 46*, 3483-3489.

Díaz-Uriarte, R., & De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC bioinformatics, 7*, 3.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems. *J. Mach. Learn. Res, 15*, 3133-3181.

Genuer, R., Poggi, J.-M., & Tuleau-Malot, C. (2015). VSURF: an R package for variable selection using random forests. *The R Journal, 7*, 19-33.

Hapfelmeier, A., & Ulm, K. (2013). A new variable selection approach using random forests. *Computational Statistics & Data Analysis, 60*, 50-69.

Hothorn, T., Hornik, K., Strobl, C., & Zeileis, A. (2010). Party: A laboratory for recursive partytioning. In.

Ishwaran, H., & Kogalur, U. (2014). Random Forests for Survival, Regression and Classification (RF-SRC), R package version 1.6. *URL* [*http://CRAN*](http://CRAN)*. R-project. org/package= randomForestSRC*.

Janitza, S., Celik, E., & Boulesteix, A.-L. (2015). A computationally fast variable importance test for random forests for high-dimensional data. *Advances in Data Analysis and Classification*, 1-31.

Jiang, H., Deng, Y., Chen, H.-S., Tao, L., Sha, Q., Chen, J., Tsai, C.-J., & Zhang, S. (2004). Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC bioinformatics, 5*, 81.

Kuhn, M. (2008). Caret package. *Journal of statistical software, 28*, 1-26.

Kursa, M. B., & Rudnicki, W. R. (2010). Feature selection with the Boruta package. *J Stat Softw, 36*, 1-13.

Liaw, A., & Weiner, M. (2002). Classification and Regression by randomForest. *R News, 2*, 18-22.

Sanchez-Pinto, L. N., Venable, L. R., Fahrenbach, J., & Churpek, M. M. (2018). Comparison of variable selection methods for clinical predictive modeling. *International journal of medical informatics, 116*, 10-17.

Speiser, J. L., Durkalski, V. L., & Lee, W. M. (2015). Random forest classification of etiologies for an orphan disease. *Statistics in medicine, 34*, 887-899.

Svetnik, V., Liaw, A., Tong, C., & Wang, T. (2004). Application of Breiman’s random forest to modeling structure-activity relationships of pharmaceutical molecules. In *International Workshop on Multiple Classifier Systems* (pp. 334-343): Springer.

Urrea, V., & Calle, M. (2012). AUCRF: variable selection with random forest and the area under the curve. *R package version 1.1*.

Urrea, V., & Calle, M. L. (2012). AUCRF: Variable Selection with Random Forest and the Area Under the Curve. *R package version 1.1*.

Wei, R., Wang, J., & Jia, W. (2019). R package: multiROC.