# Regression-Enhanced Random Forests

Haozhe Zhang*, Dan Nettleton, and Zhengyuan Zhu
Department of Statistics, Iowa State University, Ames, IA 50011
*Email: haozhe@iastate.edu

## Abstract

Random forest (RF) methodology is one of the most popular machine learning techniques for prediction problems. In this article, we discuss some cases where random forests may suffer, and propose a new boosted method of Lasso and RFs to address the challenges of RFs, namely regression-enhanced random forests (RERFs). The algorithm for constructing RERFs and the tuning parameter selection procedure are described in the paper. Both simulation study and real data examples show that RERFs have a better predictive performance than the standard RFs in specific cases. Moreover, RERFs may incorporate known relationships between the response and the predictors, and may give reliable predictions in extrapolation problems where predictions are required at points out of the domain of the training dataset. It is possible that the idea of combing penalized parametric regression and machine learning methodology can be generalized in other areas.

## 1 Introduction

Random forest (RF) methodology, proposed by L. Breiman [3], is one of the most popular machine learning techniques for regression and classification problems. In the last few years, there have been many methodological and theoretical advances in the random forests approach. Some methodological developments and extensions include case-specific random forests [14], multivariate random forests [11], quantile regression forests [8], random survival forests [6], and predictor augmentation in random forests [13] among others. For theoretical developments, the statistical and asymptotic properties of random forests have been intensively investigated. Advances have been made in the areas such as consistency [1] [10], variable selection [5] and the construction of confidence intervals [12].

Although random forests have proven themselves to be a reliable predictive approach in many application areas [2], there are some cases where random forests may suffer. First, as a fully nonparametric predictive algorithm, random forests may not incorporate known relationships between the response and the predictors. Second, random forests may fail in extrapolation problems where predictions are required at points out of the domain of the training dataset. For regression problems, a random forest prediction is an average of the predictions produced by the trees in the forest. Because each tree prediction corresponds to some weighted average of the responses $Y_1, \ldots, Y_n$ observed in the original training data, we can view the final random forest prediction at some given value of predictors $\boldsymbol{X}_0$ as a convex combination of the training responses

$$\widehat{Y}(\boldsymbol{X}_0) = \sum_{i=1}^{n} w_i(\boldsymbol{X}_0) Y_i, \tag{1}$$

involving nonnegative weights $w_i(\boldsymbol{X}_0)$ with the constraint $\sum_{i=1}^{n} w_i(\boldsymbol{X}_0) = 1$. It follows that

$$\min_{1\le i\le n} Y_i \le \widehat{Y}(\boldsymbol{X}_0) \le \max_{1\le i\le n} Y_i. \tag{2}$$

As a consequence, the predictions given by random forests are always within the range of response values in the training dataset, which is problematic if the response values in the target dataset tends to fall outside this range.

We illustrate the above issues by considering the problem of forecasting Iowa corn yield. The dataset, that will be further introduced in Section 4.2, contains county-level corn yield data and predictor variables that provide information about soil quality and environmental conditions during 28 growing seasons. We used random forests to forecast the corn yield in the coming year by using the yield and predictor data in previous years as a training dataset. The performance of random forests was not as good as we expected and failed to outperform multivariate linear regression in this problem. For example, the root mean square error (RMSE) of random forests for predicting 2015 corn yield was slightly more than 10% higher than the RMSE of multivariate linear regression. There are at least two reasons why multivariate linear regression outperforms random forests for predicting Iowa corn yield. First, in some years, the weather was so hot and dry that the values of temperature and precipitation were beyond the ranges of those in the training dataset, which creates an extrapolation problem. Second, corn yield has been increasing generally over time, due to consistent genetic improvement of maize and agricultural technology developments. When forecasting corn yield for a future year using random forests, as shown by Equation 2, each

forecast is bounded above by the largest corn yield in the training dataset, even if the past trend suggests a record-setting crop for that future year.

Next we use a simulated example to illustrate this point. Let the data-generating model be $Y = f(\boldsymbol{X}) + 10Z + \epsilon$, where $Y$ is the response variable, $\boldsymbol{X} = (X_1, \ldots, X_{10})$ and $Z$ are the predictors, and $\epsilon$ is a mean-zero error term. Suppose $f(\boldsymbol{X})$ is a partially nonlinear additive function of equation (56) in J. H. Friedman's "MARS" paper [4]. We want to predict $Y$ by using the predictors $\boldsymbol{X}$ and $Z$. The distributions of predictors $\boldsymbol{X}$ in both the training and the target datasets are identical, and they are independently simulated from the uniform distribution unif$(0, 1)$. In the training dataset, the predictor $Z$ is sampled from unif$(0, 0.8)$, while $Z$ is sampled from unif$(0, 1)$ in the target dataset. The sample sizes for the training and the target datasets are 1500 and 300.

Figure 1 presents prediction errors when analyzing the simulated data with a random forest and with a regression-enhanced random forest (RERF), the method we introduce in this paper. The red points and the red smoothed curve in the Figure 1 illustrate the relationship between the predictor $Z$ and the pointwise prediction errors $Y - \widehat{Y}$ given by the standard RFs. When the predictor $Z > 0.8$, the predicted errors are relatively large. This example indicates that the random forest approach suffers in linear extrapolation.

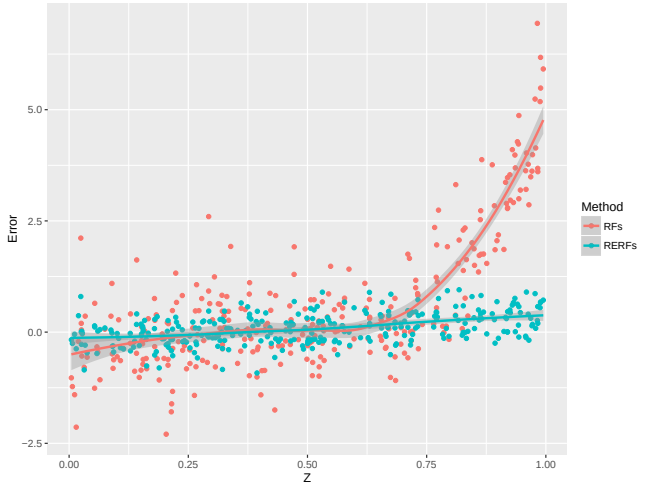


**Figure 1:** The pointwise prediction errors $Y - \widehat{Y}$ given by a random forest (red) and a regression-enhanced random forest (blue) against the predictor $Z$ and the corresponding Loess smooth curves of $Y - \widehat{Y}$ against $Z$.

To address the challenge, we propose a method, "Regression-Enhanced Random Forests" (RERFs), which is a hybrid method of penalized parametric regression and random forests. The purpose of this paper is to introduce the RERFs and investigate the prediction performance of the RERFs in comparison with the standard RFs. The parametric methods, such as linear regression and Lasso, can account for main effects, reflect the scientific mechanism and extrapolate out of the domain. The nonparametric machine learning algorithms, such as random forests and neural network, can account for complex factor interactions and incorporate a large number of predictors. We expect that the RERFs can borrow the strength of the two types of methods and overcome the corresponding disadvantages. The simulation study and the two real data examples in this paper reach the same conclusion: the prediction performance of the RERFs is better than that of the standard RFs in both the interpolation and extrapolation problems. Furthermore, in the extrapolation cases, the RERFs far outperforms the standard RFs.

The rest of the paper is organized as follows. Section 2 introduces the algorithm for building RERFs and also discusses tuning parameter selection. In Section 3, we conduct a simulation study to examine the prediction performance of RERFs in comparison to RFs and Lasso. To illustrate the proposed methodology and demonstrate its relevance in practice, Section 4 and Section 5 provides two real data examples involving high-performance concrete strength prediction and Iowa corn yield forecasting. In Section 6, we discuss and explain some special features and the limitations of RERFs.

## 2 Method

Regression-enhanced random forests (RERFs) is a boosted method of penalized parametric regression and random forests. RERFs can improve random forests in prediction accuracy and also incorporate known relationships between the response variable and the predictors. Penalized parametric regression is a class of parametric regression with a penalty function applied on the regression coefficients, which amounts to solving a minimization problem of the form

$$\min_{\boldsymbol{\beta}} \{L(Y_i, \widehat{Y}_i) + \lambda P(\boldsymbol{\beta})\}, \tag{3}$$

where $L(\cdot, \cdot)$ is a loss function and $P(\cdot)$ is the penalty function. $\lambda$ is called the penalty parameter. Lasso with a $\ell^1$ penalty function and Ridge regression with a $\ell^2$ penalty function are two examples of penalized parametric regression. Penalized parametric regression can be used to capture the global trend and incorporate scientific knowledge about linear structure, but may not be flexible enough to accommodate nonlinearity. Random forests offer a flexible nonparametric approach for prediction, which leads to small fitting errors compared with parametric methods. However, small fitting errors do not necessarily imply small prediction errors, especially in extrapolation problems. As shown by Figure 1 and the simulated example

in Section 1, random forests may suffer in extrapolation problems.

Let $Y$ be a continuous response variable and $\boldsymbol{X}$ a $p$-dimensional vector of predictor variables. We assume a standard data-generating model given by

$$Y = f(\boldsymbol{X}) + \epsilon \tag{4}$$

for both the training dataset and the test dataset. We assume a training dataset $\boldsymbol{C} = \{C_i = (\boldsymbol{X}_i, Y_i) : i = 1, \ldots, N\}$ with a sample size $N$ is available to fit the model for prediction. The random forests algorithm has two tuning parameters [3], mtry and nodesize, denoted as $m$ and $s$. The RERF algorithm is described as follows:

**Regression-Enhanced Random Forest Algorithm**

- Step 1: Extend the $p$-dimensional predictor $\boldsymbol{X}$ to a $(p+q)$-dimensional predictor $\boldsymbol{X}^*$ by adding higher-order , interaction or other known parametric functions of $\boldsymbol{X}$.
- Step 2: Run Lasso of $Y$ on $\boldsymbol{X}^*$ with a pre-specified penalty parameter $\lambda$. Let $\widehat{\boldsymbol{\beta}}_\lambda$ be the estimated coefficient, and $Y^\lambda = Y - \boldsymbol{X}^*\widehat{\boldsymbol{\beta}}_\lambda$ be the residuals from the Lasso. Create a new training dataset $\boldsymbol{C}^\lambda = \{C_i^\lambda = (\boldsymbol{X}_i, Y_i^\lambda) : i = 1, \cdots, N\}$.
- Step 3: Build random forests $\boldsymbol{T}_{m,s}$ using $\boldsymbol{C}^\lambda$ with pre-specified $m$ and $s$. A prediction for the response at a given predictor value $\boldsymbol{X}_0$ is $\widehat{Y}(\boldsymbol{X}_0) = \boldsymbol{X}_0\widehat{\boldsymbol{\beta}}_\lambda + \boldsymbol{T}_{m,s}(\boldsymbol{X}_0)$.
- Step 4: Select the tuning parameters $(\lambda, m, s)$ through k-fold cross validation by repeating step 2 and step 3. The selected tuning parameters are denoted by $\lambda^*, m^*$ and $s^*$.
- Step 5: The RERF prediction for the response at $\boldsymbol{X}_0$ is given by $\widehat{Y}(\boldsymbol{X}_0) = \boldsymbol{X}_0\widehat{\boldsymbol{\beta}}_{\lambda^*} + \boldsymbol{T}_{m^*,s^*}(\boldsymbol{X}_0)$.

To explain the mechanics of RERFs, we will discuss each step in the algorithm in detail. In Step 1, expanding the design matrix is optional. Whether to add higher-order, interaction or other parametric terms should be decided by exploratory analysis or knowledge of the relationship between $Y$ and $X$. The main aim of the Lasso regression in Step 2 is to select variables in order to find a parametric structure that incorporates the global trend and known relationships between the response and predictors. The penalty parameter $\lambda$ controls the strength of variable selection. When $\lambda = 0$, the Lasso regression in Step 2 is equivalent to multivariate regression without regularization. When $\lambda \to \infty$, the Lasso regression in Step 2 is equivalent to regressing on a constant value, i.e., intercept. Thus, RERFs will be reduced to RFs for sufficiently large $\lambda$, and $RFs$ can be viewed as a special case of RERFs.

Tuning parameter selection in Step 4 is critical to the performance of RERFs. As a hybrid method of the Lasso and RFs, regression-enhanced random forests have three tuning parameters, the Lasso penalty parameter ($\lambda$), nodesize ($s$) and mtry ($m$). The value of $\lambda$ plays an important role in the prediction performance of the RERFs. We should note that, unlike the standard Lasso, the optimal value of $\lambda$ for the RERFs is not determined based on the residuals from the Lasso regression. Instead, the optimal value of $\lambda$ is determined based on the residuals from random forests in the step 3, which takes the residuals from the Lasso in the step 2 as response values. The plausible values of $\lambda$ are positive and unbounded. In our numerical example, we choose $\lambda$ from among the values in the set $\{\exp((\log(0.001) + h \times \frac{\log(100)-\log(0.001)}{99}) : h = 0, \ldots, 99\}$, which is a set of 100 points from 0 to 100 equally spaced on the logarithm scale. Following the advice of Breiman as recounted by Liaw and Wiener (2002) [7], we consider mtry from the default value of $\max\{1, \lfloor p/3 \rfloor\}$ as well as half and twice the default value. For the nodesize, we consider the default value of 5 as well as 1 (the value recommended by Breiman for classification problems).

One approach for simultaneous selection of these three tuning parameters is an exhaustive search on 3-dimensional tuning parameter space. Because such a search is time consuming, parallel computing can be applied to lessen the computing intensity. The other approach is to apply one-step iteration as follows. First, fixing the nodesize and mtry to be the default values, we select a value of penalty parameter by cross validation. Second, using the selected value of penalty parameter $\lambda$, choose values of nodesize and mtry. Lastly, using the selected values of the nodesize and mtry obtained in the previous step, we update the value of the penalty parameter by cross validation. The above procedure reduces the computing intensity and yields values of the tuning parameter with cross-validation performance similar to parameters obtained by an exhaustive search. Throughout the paper, all the results from RERFs, Lasso and RFs were obtained by selecting tuning parameters by cross validation. Particularly, we applied one-step iteration to the tuning parameters selection in RERFs.

# 3 Simulation study

In this section, we conduct a simulation study to examine the prediction performances of RERFs compared with the RFs for both the interpolation cases and the extrapolation cases. We simulated data from a data-generating model given by

$$Y = f(\boldsymbol{X}) + \epsilon. \tag{5}$$

The independent random errors $\epsilon$ follow $N(0, 0.5^2)$. We considered three different structures for $f(\cdot)$, labeled as $L$, $P$ and $N$, described as follows,

- $L$: a linear model with additive structure

$$f(\boldsymbol{X}) = x_1 + x_2 + 2x_3 + 2x_4 + 0\sum_{i=5}^{10} x_i,$$

- $P$: a partially linear model with additive structure

$$f(\boldsymbol{X}) = sin(\pi x_1) + \frac{4}{1+e^{-20x_2+10}} + 2x_3 + 2x_4 + 0\sum_{i=5}^{10} x_i,$$

- $N$: a non-additive partially linear model

$$f(\boldsymbol{X}) = sin(\pi x_1) + \frac{4}{1+e^{-20x_2+10}} + 2x_3 + 2x_4 + 3x_3x_4 + 0\sum_{i=5}^{10} x_i.$$

We also considered two different sampling distributions for $\boldsymbol{X}$, denoted as $I$ and $E$, described as follows.

- $I$: all 10 predictor observations are i.i.d. sampled from unif$(0,1)$ in both training and validation datasets.

- $E$: $x_3$ observations are i.i.d. sampled from Beta$(4,8)$ in the training dataset and i.i.d. sampled from Beta$(5,1)$ in the validation dataset, for which predictions of $y$ are desired. The other 9 predictor observations are i.i.d. sampled from unif$(0,1)$ in both training and validation datasets.

Most of the data generated from Beta$(4,8)$ are less than 0.6, while most of the data generated from Beta$(5,1)$ are larger than 0.6. Thus, we can treat the first case ($I$) as interpolation and the second case ($E$) as extrapolation. Then, there are 6 simulation scenarios formed by all the combinations of choices of $f(\cdot)$ and choices of sampling distributions for $\boldsymbol{X}$. These can be labeled as $I \times L$, $I \times P$, $I \times N$, $E \times L$, $E \times P$ and $E \times N$. For each scenario, 1000 simulation runs were conducted. In each run, 1000 training observations and 100 validation observations were randomly and independently generated from the joint distribution of $(\boldsymbol{X}, Y)$. We fitted models using training data, which then were used for prediction on the validation data. The root mean square errors (RMSEs) were then calculated over the validation dataset.

The RMSEs from the simulation are shown in Figure 2. We can conclude that the prediction performance of the RERFs is better than that of the standard RFs in both the interpolation and extrapolation problems, no matter if the Lasso is better than the RFs or not. Particularly, in extrapolation cases, the RERFs far outperforms the standard RFs. We should also note that the performance of RERFs is better than that of Lasso for all the models except $I \times L$ and $E \times L$. Since $I \times L$ and $E \times L$ are both linear models, Lasso is correct under assumption, and we do expect Lasso to do well. Nevertheless, the RMSEs of Lasso and RERFs are very close for $I \times L$ and $E \times L$.

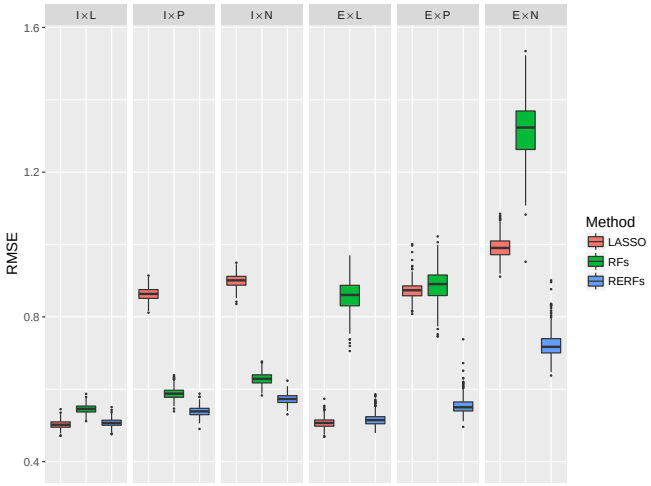


**Figure 2:** The boxplot of the RMSEs against data-generating models from simulation study using Lasso, RFs and RERFs.

As stated in Section 2, the value of penalty parameter $\lambda$ has a significant effect on the prediction performance of the RERFs. In general, the selected penalty parameters of the RERFs are larger than those of the Lasso. Because random forests can utilize some predictor variables to predict response, less variables need to be selected in Step 2 of the RERFs, in comparison with the standard Lasso. We believe this is the reason that the selected penalty parameter of the RERFs is larger than those of the Lasso.

# 4 Examples

## 4.1 High-performance concrete strength example

We first use the high-performance concrete strength dataset [15] as a real example to demonstrate our methodology. The concrete strength dataset is available on the UC Irvine Machine Learning Repository website, and has been widely used for evaluating machine learning algorithms. It contains 1030 observations, with eight quantitative predictors (cement, water, fly ash, blast furnace slag, superplasticizer, coarse aggregate, fine aggregate and age of testing), and a response variable (concrete compressive strength). The Abrams rule [9] implies the approximate proportionality between the cement-to-water ratio (C/W) and the strength of concrete, so the

cement-to-water ratio was computed as a predictor and added into the training dataset for prediction.

In our study, the prediction performance of RERFs and RFs are compared under six scenarios, as shown in Table 1. Scenario C1-1 and C1-2 are the case of interpolation, while the rest of the scenarios are the case of extrapolation. In scenario C1-1 and C1-2, the complete dataset is randomly splitted into the training and validation datasets. In scenario C2-1 and C2-2, the complete dataset is splitted based on the value of concrete compressive strength (CCS) so that the domains of the CCS in the training and validation datasets are disjoint. In scenario C3-1 and C3-2, the cement-to-water ratios in the training and validation set have disjoint domains. We run 1000 simulations for each scenario.

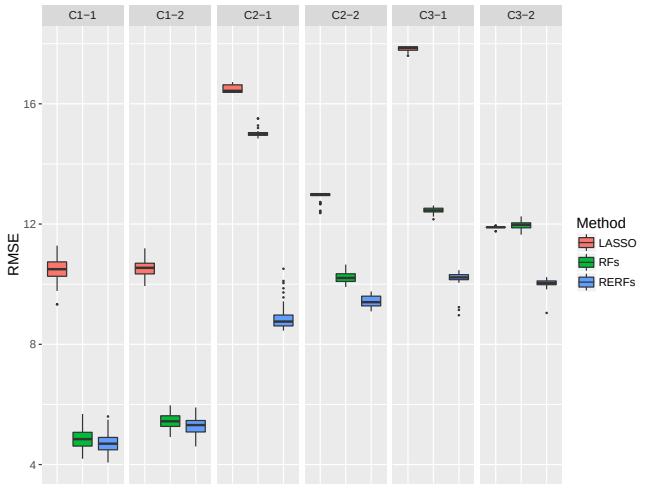


**Figure 3:** The boxplot of the RMSEs against scenarios for concrete dataset using the Lasso, RFs and RERFs

In the implementation of RERFs for concrete strength dataset, we did not include high-order or interaction terms of predictors in Step 1 of the RERFs algorithm. However, the exploratory analysis shows that the relationships between concrete strength and ingredients are nonlinear and there are interactions between predictors. To attain the minimal error, less predictors should be selected in Step 2 of the RERFs algorithm. The computing results show that the averaged penalty parameter for Lasso is 0.08, while the averaged penalty parameter for RERFs is 1.0.

Figure 3 implies the same conclusion as that in simulation study. For the problem of predicting concrete strength, RERFs has better prediction performance than RFs in both the interpolation and extrapolation cases, no matter whether the Lasso is better than the RFs or not. Particularly, in the extrapolation cases, RERFs approach far outperforms the RFs.

## 4.2 Iowa corn yield example

Forecasting corn yield is an age-old and important problem in agriculture and economics. The United States produced roughly 14.2 billion bushels of corn in the 2014-2015 crop marketing year, and the productions have been exported to more than 100 different countries. Iowa produces, by far, the most corn in the United States, supplying nearly 20 percent of the country's annual corn. Providing a valid corn yield prediction of Iowa before and within the harvesting season is of importance to land planning, livestock husbandry and option markets.

In this subsection, we compare the performance of the RERFs, RFs and Lasso in forecasting current year's corn yield in Iowa by using previous years' data and current year's meteorological and soil data. For instance, we used the complete data during 1988-2013 and the meteorological and soil data during January - September 2014 to forecast the corn yield in 2014.

Agricultural knowledge implies that extremely high and extremely low temperature may both cause low corn yield. The flood (high precipitation) and drought (low precipitation) may cause low yield as well. As a consequence, we choose to add the second-order terms of variables related with temperature and precipitation into the feature matrix $\boldsymbol{X}^*$ in the step 1 of RERFs. For all other predictors, only the first-order terms are included in $\boldsymbol{X}^*$. We should note that, the meteorological recordings are time series data. We regard the meteorological variable in each time point as an individual predictor. For instance, we have the data of the mean soil moisture for each month from January to September, so there are nine mean soil moisture predictors in the feature matrix.


**Figure 4:** The RMSEs against the year for Iowa corn yield dataset using the Lasso, RFs and RERFs.

The RMSEs of Lasso, RFs and RERFs are shown in Figure 4. The conclusion from Iowa corn yield dataset is

Table 1: Description of the training sets and validation sets of concrete strength dataset

| Scenario | Training set | Validation set | Sample size of training set | Sample size of validation set |
|---|---|---|---|---|
| C1-1 | Random 3/4 | The rest 1/4 | 772 | 258 |
| C1-2 | Random 1/2 | The rest 1/2 | 515 | 515 |
| C2-1 | CCS $>$ 25 | CCS $\leq$ 25 | 735 | 295 |
| C2-2 | CCS$<$ 16 or CCS$>$ 56 | 16 $\leq$ CCS $\leq$ 56 | 761 | 269 |
| C3-1 | C/W$<$ 2 | C/W$\geq$ 2 | 793 | 237 |
| C3-2 | C/W$<$ 1 or C/W$>$ 3 | 1 $\leq$ CCS $\leq$ 3 | 804 | 226 |

consistent with the conclusion from the simulation study and the analysis of the concrete strength dataset. The prediction performance of the RFs can be improved by using the RERFs.

# 5 Discussion

In this article, we introduce two cases where random forests may not perform well, and propose a new boosted method of Lasso and RFs named as regression-enhanced random forests. We focus on the comparison between RFs and RERFs in prediction performance for regression problems. The methodology for classification problems will be developed in the future work. The two approaches for simultaneous selection of these three tuning parameters are discussed in this paper in detail. We also present the simulation study, high-performance concrete strength example and Iowa corn yield example. They all show that RERFs have a better predictive performance than the standard RFs in specific cases. Moreover, it is possible that the idea of combing penalized parametric regression and machine learning methodology can be generalized in other areas.

# References


[1] Gérard Biau, Luc Devroye, and Gábor Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9(Sep):2015–2033, 2008.

[2] Gérard Biau and Erwan Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Jerome H Friedman. Multivariate adaptive regression splines. *The annals of statistics*, pages 1–67, 1991.

[5] Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236, 2010.

[6] Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. Random survival forests. *The annals of applied statistics*, pages 841–860, 2008.

[7] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[8] Nicolai Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7(Jun):983–999, 2006.

[9] Snador Popovics. Analysis of the concrete strength versus water cement ratio relationship. *ACI Materials journal*, 87(Title No. 87-M56), 1990.

[10] Erwan Scornet, Gérard Biau, Jean-Philippe Vert, et al. Consistency of random forests. *The Annals of Statistics*, 43(4):1716–1741, 2015.

[11] Mark Segal and Yuanyuan Xiao. Multivariate random forests. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):80–87, 2011.

[12] Stefan Wager, Trevor Hastie, and Bradley Efron. Confidence intervals for random forests: the jackknife and the infinitesimal jackknife. *Journal of Machine Learning Research*, 15(1):1625–1651, 2014.

[13] Ruo Xu, Dan Nettleton, and Daniel J Nordman. Predictor augmentation in random forests. *Statistics and its interface*, 7:177–186, 2014.

[14] Ruo Xu, Dan Nettleton, and Daniel J Nordman. Case-specific random forests. *Journal of Computational and Graphical Statistics*, 25(1):49–65, 2016.

[15] I-C Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808, 1998.