Received <day> <Month>, <year>; | Revised <day> <Month>, <year>; | Accepted <day> <Month>, <year>

DOI: xxx/xxxx

ARTICLE TYPE

Standard errors and confidence intervals for variable importance in random forest regression, classification, and survival†

Hemant Ishwaran*[1] | Min Lu[1]

[1]Division of Biostatistics, Miller School of Medicine, University of Miami, Florida, USA

Correspondence
*Hemant Ishwaran, Division of Biostatistics, 1120 NW 14th Street, University of Miami, Miami, FL 33136. Email: hemant.ishwaran@gmail.com

Abstract

Random forests is a popular nonparametric tree ensemble procedure with broad applications to data analysis. While RF's widespread popularity stems from its prediction performance, an equally important feature is that it provides a fully nonparametric measure of variable importance (VIMP). A current limitation of VIMP however is that no systematic method exists for estimating its variance. As a solution, we propose a subsampling approach that can be used to estimate the variance of VIMP and for constructing confidence intervals. The method is general enough that it can be applied to many useful settings, including regression, classification, and survival problems. Using extensive simulations we demonstrate the effectiveness of the subsampling estimator and in particular find that the delete-*d* jackknife variance estimator, a close cousin, is especially effective under low subsampling rates due to its bias correction properties. These two estimators are highly competitive when compared to the .164 bootstrap estimator, a modified bootstrap procedure designed to deal with ties in out-of-sample data. Most importantly, subsampling is computationally fast, thus making it especially attractive for big data settings.

KEYWORDS:
bootstrap; delete-*d* jackknife; subsampling; prediction error; permutation importance; VIMP

1 | INTRODUCTION

Random forests (RF)[1] is a popular tree-based learning method with broad applications to machine learning and data mining. RF was originally designed for regression and classification problems, but over time the methodology has been extended to other important settings. For example, random survival forests (RSF)[2, 3] extends RF to right-censored survival and competing risk settings (see also Hothorn et al.[4] and Zhu and Kosorok[5] for other tree-ensemble approaches to survival analysis). Two guiding principles are at the core of RF's success. One is the use of deep trees. Another is injecting randomization into the tree growing process. First, trees are randomly grown by using a bootstrap sample of the data. Secondly, random feature selection is used when growing the tree. Thus, rather than splitting a node using all variables, the node is split using the best candidate from a randomly selected subset of variables. The purpose of this two-step randomization is to decorrelate trees, which encourages low variance for the ensemble due to bagging.[6] When combined with the strategy of using deep trees, which is a bias reduction technique, this reduces generalization error and results in superior performance for the ensemble.

While RF's popularity stems from its prediction performance, an equally important feature is that it provides a fully nonparametric measure of variable importance (VIMP).[1, 2, 7, 8, 9] VIMP allows users to identify which variables play a key role in prediction, thus providing insight into the underlying mechanism for what otherwise might be considered a black-box. We note

†This research was supported by NIH grant R01 GM125072.

that the concept of variable importance is not specific to RF and has a long history. One of the earliest examples was CART,[10] which calculated variable importance by summing the reduction in node impurity due to a variable over all tree nodes. Another approach calculated importance using surrogate splitting (see Chapter 5.3 of Breiman et al.[10]).

Early prototypes of RF software developed by Leo Breiman and his student Adele Cutler provided for various options for calculating VIMP.[11] One procedure used for classification forests was to estimate VIMP using the forest averaged decrease in Gini impurity (somewhat akin to the node impurity approaches of CART). However, while Gini importance[12] saw widespread initial use with RF, over time it has become less popular.[8] By far, the most frequently used measure of importance was another measure provided by the Breiman-Cutler software, called permutation importance (sometimes also referred to as Breiman-Cutler importance). Unlike Gini importance which estimates importance using in-sample impurity, permutation importance adopts a prediction based approach by using prediction error attributable to the variable. A clever feature is that rather than using cross-validation, which is computationally expensive for forests, permutation importance estimates prediction error by making use of out-of-bootstrap cases. Recall that each tree is calculated from a bootstrap sample of the original data. The approximately $1 - .632 = .368$ left from the bootstrap represents out-of-sample data which can be used for estimating prediction performance. This data is called out-of-bag (OOB) and prediction error obtained from it is called OOB error.[13] Permutation importance permutes a variable's OOB data and compares the resulting OOB prediction error to the original OOB prediction error—the motivation being that a large positive value indicates a variable with predictive importance.

Permutation (Breiman-Cutler) Importance

In the OOB cases for a tree, randomly permute all values of the jth variable. Put these new covariate values down the tree and compute a new internal error rate. The amount by which this new error exceeds the original OOB error is defined as the importance of the jth variable for the tree. Averaging over the forest yields VIMP.

— Measure 1 (Manual On Setting Up, Using, And Understanding Random Forests V3.1)

We focus on Breiman-Cutler permutation importance in this manuscript (for simplicity, hereafter simply refered to as VIMP). One of the tremendous advantages of VIMP is that it removes the arbitrariness of having to select a cutoff value when determining the effectiveness of a variable. Regardless of the problem, a VIMP of zero always represents an appropriate cutoff, as it reflects the point at which a variable no longer contributes predictive power to the model. However, in practice one may observe values close to zero, and the meaning of what constitutes being zero becomes unclear. One way to resolve this is to calculate the variance of VIMP, but this is challenging due to the complex nature of RF. Unfortunately, while the empirical properties of VIMP are well documented,[14, 15, 16] much less is known about VIMP's theoretical properties outside of a few studies.[7, 17]

Given the difficulties of theoretical analysis, an alternative approach is to approximate the distribution of VIMP through some form of resampling. This has been the favored approach used for RF regression for assessing variability of RF predicted values. Methods that have been used include bootstrapping[18] for estimating the variance, and the infinitesimal jackknife[19] and infinite order U-statistics[20] for confidence intervals. These methods however only apply to RF predicted values and not to VIMP which involves prediction error. This greatly complicates matters and requires a more general approach.

For this reason, we base our approach on subsampling,[21] a general methodology for approximating the distribution of a complex statistic. Section 3 provides a description of of our subsampling procedure for estimating the variance. Notational framework and a formal definition of VIMP are provided in Section 2. Section 3 begins by introducing a bootstrap solution to be used as a comparison procedure. Interestingly, we find the bootstrap cannot be applied directly due to ties that occur in the OOB data. This is precisely due to the fact that VIMP is prediction error based. We propose a solution to this problem called the .164 bootstrap estimator. The subsampling variance estimator and the delete-d jackknife variance estimator,[22] a close cousin, are described later in Section 3. Sections 4, 5, and 6 consider regression, classification, and survival settings and extensively evaluate performance of the two subsampling methods and the .164 bootstrap estimator. We also show how to construct confidence intervals for VIMP using the estimated variance. The results are very promising for the subsampling methods. Section 7 summarizes our findings and provides practical guidelines for use of the methodology. Some theoretical results for VIMP are provided in the Appendix.

2 | NOTATIONAL FRAMEWORK AND DEFINITION OF VIMP

2.1 | Notation

We assume $Y \in \mathcal{Y}$ is the response and $\mathbf{X} \in \mathcal{X}$ is the p-dimensional feature where Y can be continuous, binary, categorical, or survival, and $\mathbf{X}$ can be continuous or discrete. We assume the underlying problem involves a nonparametric regression framework where the goal is to estimate a functional $h(\mathbf{x})$ of the response given $\mathbf{X} = \mathbf{x}$. Estimation is based on the learning data $\mathcal{L} = \{(\mathbf{X}_1, Y_1), \ldots, (X_n, Y_n)\}$, where $(\mathbf{X}_i, Y_i)$ are independently distributed with the same distribution $\mathbb{P}$ as $(\mathbf{X}, Y)$.

Examples of $h(\mathbf{x})$ are:

1. The conditional mean $h(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$ in regression.
2. The conditional class probabilities $h(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_K(\mathbf{x}))$ in a K-multiclass problem, where $p_k(\mathbf{x}) = \mathbb{P}\{Y = k|\mathbf{X} = \mathbf{x}\}$.
3. The survival function $h(\mathbf{x}) = \mathbb{P}\{T^o > t|\mathbf{X} = \mathbf{x}\}$ in survival analysis. Here $Y = (T, \delta)$ represents the bivariate response comprised of the observed survival time $T = \min(T^o, C^o)$ and censoring indicator $\delta = 1\{T^o \leq C^o\}$, where (T^o, C^o) are the unobserved event and censoring times.

2.2 | RF predictor

As in Breiman,[1] we define a RF as a collection of randomized tree predictors $\{h(\cdot, \Theta_m, \mathcal{L}), m = 1, \ldots, M\}$. Here $h(\mathbf{x}, \Theta_m, \mathcal{L})$ denotes the mth random tree predictor of $h(\mathbf{x})$ and $\{\Theta_m\}$ are independent identically distributed random quantities encoding the randomization needed for constructing a tree. Note that Θ_m is selected prior to growing the tree and is independent of the learning data, $\mathcal{L}$.

The tree predictors are combined to form the finite forest estimator of $h(\mathbf{x})$,

$$h(\mathbf{x}, \Theta_1, \ldots, \Theta_M, \mathcal{L}) = \frac{1}{M} \sum_{m=1}^{M} h(\mathbf{x}, \Theta_m, \mathcal{L}). \tag{1}$$

The infinite forest estimator is obtained by taking the limit as $M \to \infty$ and equals

$$h(\mathbf{x}, \mathcal{L}) = \mathbb{E}_\Theta \left[h(\mathbf{x}, \Theta, \mathcal{L})\right]. \tag{2}$$

2.3 | Loss function

Calculating VIMP assumes some well defined notion of prediction error. Therefore, we assume there is an appropriately prechosen loss function $\ell(Y, \hat{h}) \geq 0$ used to measure performance of a predictor $\hat{h}$ in predicting h. Examples include:

1. Squared error loss $\ell(Y, \hat{h}) = (Y - \hat{h})^2$ in regression problems.
2. For classification problems, widely used measures of performance are the misclassification error or the Brier score. For the latter, $\ell(Y, \hat{h}) = (1/K) \sum_{k=1}^{K} \left(1\{Y = k\} - \hat{p}_k\right)^2$, where $\hat{p}_k$ is the estimator for the conditional probability p_k.
3. For survival, the weighted Brier score[23, 24] can be used. Section 6 provides further details.

The choice of ℓ can be very general and we do not impose any specific conditions on how it must be selected. As described later in Section 3, the conditions needed for our methodology to hold require only the existence of a limiting distribution for VIMP. Although such a limit may be satisfied by imposing specific conditions on ℓ, such as requiring the true function h to yield the minimum value of $\mathbb{E}[\ell(Y, h)]$, we do not impose such assumptions so as to retain as general an approach as possible.

2.4 | Tree VIMP

Let $\mathcal{L}^*(\Theta_m)$ be the mth bootstrap sample and let $\mathcal{L}^{**}(\Theta_m) = \mathcal{L} \setminus \mathcal{L}^*(\Theta_m)$ be the corresponding OOB data. Write $\mathbf{X} = (X^{(1)}, \ldots, X^{(p)})$ where $X^{(j)}$ denotes the jth feature coordinate. The permuted value of the jth coordinate of X is denoted by $\tilde{X}^{(j)}$. Substituting this into the jth coordinate of $\mathbf{X}$ yields $\tilde{\mathbf{X}}^{(j)}$:

$$\tilde{\mathbf{X}}^{(j)} = (X^{(1)}, \ldots, X^{(j-1)}, \tilde{X}^{(j)}, X^{(j+1)}, \ldots, X^{(p)}).$$

VIMP is calculated by taking the difference in prediction error under the original $\mathbf{X}$ to prediction error under the perturbed $\tilde{\mathbf{X}}^{(j)}$ over OOB data. More formally, let $I(X^{(j)}, \Theta_m, \mathcal{L})$ denote the VIMP for $X^{(j)}$ for the mth tree. It follows that

$$I(X^{(j)}, \Theta_m, \mathcal{L}) = \frac{\sum_{i \in \mathcal{L}^{**}(\Theta_m)} \ell(Y_i, h(\tilde{\mathbf{X}}_i^{(j)}, \Theta_m, \mathcal{L}))}{\sum_{i \in \mathcal{L}^{**}(\Theta_m)} 1} - \frac{\sum_{i \in \mathcal{L}^{**}(\Theta_m)} \ell(Y_i, h(\mathbf{X}_i, \Theta_m, \mathcal{L}))}{\sum_{i \in \mathcal{L}^{**}(\Theta_m)} 1}. \tag{3}$$

Note that in the first sum, we implicitly assume Θ_m embeds the additional randomization for permuting OOB data to define $\tilde{\mathbf{X}}_i^{(j)}$. Because this additional randomization only requires knowledge of OOB membership, and therefore can be parameterized in terms of Θ, we assume without loss of generality that Θ encodes both the randomization for growing a tree and for permuting OOB data.

Expression (3) can be written more compactly by noting that the denominator in each sum equals the OOB sample size. Let $N(\Theta_m)$ be this value. Then

$$I(X^{(j)}, \Theta_m, \mathscr{L}) = \frac{1}{N(\Theta_m)} \sum_{i \in \mathscr{L}^{**}(\Theta_m)} \left[\ell(Y_i, h(\tilde{\mathbf{X}}_i^{(j)}, \Theta_m, \mathscr{L})) - \ell(Y_i, h(\mathbf{X}_i, \Theta_m, \mathscr{L})) \right].$$

2.5 | VIMP

Averaging tree VIMP over the forest yields VIMP:

$$I(X^{(j)}, \Theta_1, \ldots, \Theta_M, \mathscr{L}) = \frac{1}{M} \sum_{m=1}^{M} I(X^{(j)}, \Theta_m, \mathscr{L}). \tag{4}$$

An infinite forest estimator for VIMP can be defined analogously by taking the limit as $M \to \infty$,

$$I(X^{(j)}, \mathscr{L}) = \mathbb{E}_\Theta \left[I(X^{(j)}, \Theta, \mathscr{L}) \right]. \tag{5}$$

It is worth noting that (4) and (5) do not explicitly make use of the forest predictors (1) or (2). This is a unique feature of permutation VIMP because it is a tree-based estimator of importance.

3 | SAMPLING APPROACHES FOR ESTIMATING VIMP VARIANCE

3.1 | The .164 bootstrap estimator

The bootstrap is a popular method that can be used for estimating the variance of an estimator. So why not use the bootstrap to estimate the standard error for VIMP? One problem is that running a bootstrap on a forest is computationally expensive. Another more serious problem, however, is that a direct application of the bootstrap will not work for VIMP. This is because RF trees already use bootstrap data and applying the bootstrap creates double-bootstrap data that affects the coherence of being OOB.

To explain what goes wrong, let's simplify our previous notation by writing $I_{n,M}^{(j)}$ for the finite forest estimator (4). Let $\mathbb{P}_n$ denote the empirical measure for $\mathscr{L}$. The bootstrap estimator of $\text{Var}(I_{n,M}^{(j)})$ is

$$\text{Var}^*(I_{n,M}^{(j)}) = \text{Var}_{\mathbb{P}_n}(I_{n,M}^{*(j)}). \tag{6}$$

To calculate (6), we must draw a sample from $\mathbb{P}_n$. Call this bootstrap sample $\mathscr{L}^*$. Because $\mathscr{L}^*$ represents the learning data, we must draw a bootstrap sample from $\mathscr{L}^*$ to construct a RF tree. Let $\mathscr{L}^*(\Theta^*)$ denote this bootstrap sample where Θ^* represents the tree growing instructions. This is a double-bootstrap draw. The problem is that if a specific case in $\mathscr{L}^*$ is duplicated $l > 1$ times there is no guarantee that all l cases appear in the bootstrap draw, $\mathscr{L}^*(\Theta^*)$. These remaining duplicated values are assigned to the OOB data but these values are not truly OOB which compromises the coherence of the OOB data.

Double bootstrap data lowers the probability of being truly OOB to a value much smaller than .368, which is the value expected for a true OOB sample. We can work out exactly how much smaller this probability is. Let n_i be the number of occurrences of case i in $\mathscr{L}^*$. Then,

$$\Pr\{i \text{ is truly OOB in } \mathscr{L}^*(\Theta^*)\} = \sum_{l=1}^{n} \Pr\{i \text{ is truly OOB in } \mathscr{L}^*(\Theta^*) | n_i = l\} \Pr\{n_i = l\}. \tag{7}$$

We have

$$\begin{aligned} (n_1, \ldots, n_n) &\sim \text{Multinomial}(n, (1/n, \ldots, 1/n)) \\ n_i &\sim \text{Binomial}(n, 1/n) \asymp \text{Poisson}(1). \end{aligned}$$

Hence, (7) can be seen to equal

$$\sum_{l=1}^{n} \left(\frac{n-l}{n} \right)^n \Pr\{n_i = l\} \asymp \sum_{l=1}^{n} \left(\frac{n-l}{n} \right)^n \left(\frac{\mathrm{e}^{-1} 1^l}{l!} \right) = \mathrm{e}^{-1} \sum_{l=1}^{n} \left(1 - \frac{l}{n} \right)^n \frac{1}{l!} \asymp \mathrm{e}^{-1} \sum_{l=1}^{n} \frac{\mathrm{e}^{-l}}{l!} \asymp .1635.$$

Therefore, double bootstrap data has a OOB size of .164n.

The above discussion points to a simple solution to the problem which we call the .164 bootstrap estimator. The .164 estimator is a bootstrap variance estimator but is careful to use only truly OOB data. Let $\mathscr{L}^* = \{\mathbf{Z}_1 = (\mathbf{X}_{i_1}, Y_{i_1}), \ldots, \mathbf{Z}_n = (\mathbf{X}_{i_n}, Y_{i_n})\}$ denote the bootstrap sample used for learning and let $\mathscr{L}^*(\Theta^*) = \{\mathbf{Z}_i : i \in \Theta^*\}$ be the bootstrap sample used to grow the tree. The OOB data for the double-bootstrap data is defined as $\{(\mathbf{X}_{i_l}, Y_{i_l}) \notin \mathscr{L}^*(\Theta^*)\}$. However, there is another subtle issue at play regarding duplicates in the OOB data. Even though $\{(\mathbf{X}_{i_l}, Y_{i_l}) \notin \mathscr{L}^*(\Theta^*)\}$ are data points from $\mathscr{L}^*$ truly excluded from the double-bootstrap sample, and therefore technically meet the criteria of being OOB, there is no guarantee they are all unique. This is because these values orginated from $\mathscr{L}^*$, a bootstrap draw, and therefore could very well be duplicated. To ensure this does not happen we further process the OOB data to retain only the unique values.

The steps for implementing the .164 estimator can be summarized as follows.

.164 bootstrap estimator for $\mathrm{Var}(I^{(j)}_{n,M})$

1. Draw a bootstrap sample $\mathscr{L}^* = \{\mathbf{Z}_1 = (\mathbf{X}_{i_1}, Y_{i_1}), \ldots, \mathbf{Z}_n = (\mathbf{X}_{i_n}, Y_{i_n})\}$.
2. Let $\mathscr{L}^*(\Theta^*) = \{\mathbf{Z}_i : i \in \Theta^*\}$ be a bootstrap draw from $\mathscr{L}^*$. Use $\mathscr{L}^*(\Theta^*)$ to grow a tree predictor.
3. Define OOB data to be the unique values in $\{(\mathbf{X}_{i_l}, Y_{i_l}) \notin \mathscr{L}^*(\Theta^*)\}$.
4. Calculate the tree VIMP, $I(X^{(j)}, \Theta^*, \mathscr{L}^*)$, using OOB data of step 3.
5. Repeat steps 2–4 independently M times. Average the VIMP values to obtain $\hat{\theta}^{*(j)}_n$.
6. Repeat the entire procedure $K > 1$ times obtaining $\hat{\theta}^{*(j)}_{n,1}, \ldots, \hat{\theta}^{*(j)}_{n,K}$. Estimate $\mathrm{Var}(I^{(j)}_{n,M})$ by the bootstrap sample variance, $(1/K)\sum_{k=1}^{K}(\hat{\theta}^{*(j)}_{n,k} - \frac{1}{K}\sum_{k'=1}^{K}\hat{\theta}^{*(j)}_{n,k'})^2$.

3.2 | Subsampling and the delete-d jackknife

A problem with the .164 bootstrap estimator is that its OOB data set is smaller than a typical OOB estimator. Truly OOB data from a double bootstrap can be be less than half the size of OOB data used in a standard VIMP calculation (16.4% versus 36.8%). Thus in a forest of 1000 trees, the .164 estimator uses about 164 trees on average to calculate VIMP for a case compared with 368 trees used in a standard calculation. This can reduce efficiency of the .164 estimator. Another problem is computational expense. The .164 estimator requires repeatedly fitting RF to bootstrap data which becomes expensive as n increases.

To avoid these problems, we propose a more efficient procedure based on subsampling theory.[21] The idea rests on calculating VIMP over small i.i.d. subsets of the data. Because sampling is without replacement, this avoid ties in the OOB data that creates problems for the bootstrap. Also, because each calculation is fast, the procedure is computationally efficient, especially in big n settings.

3.2.1 | Subsampling theory

We begin by first reviewing some basic theory of subsampling. Let $X_1, \ldots, X_n$ be i.i.d. random values with common distribution $\mathbf{P}$. Let $\hat{\theta}_n = \hat{\theta}(X_1, \ldots, X_n)$ be some estimator for $\theta(\mathbf{P})$, an unknown real-valued parameter we wish to estimate. The bootstrap estimator for the variance of $\hat{\theta}_n$ is based on the following simple idea. Let $\mathbf{P}_n = (1/n)\sum_{i=1}^{n}\delta_{X_i}$ be the empirical measure for the data. Let $X^*_1, \ldots, X^*_n$ be a bootstrap sample obtained by independently sampling n points from $\mathbf{P}_n$. Because $\mathbf{P}_n$ converges to $\mathbf{P}$, we should expect the moments of the bootstrap estimator $\hat{\theta}^*_n = \hat{\theta}(X^*_1, \ldots, X^*_n)$ to closely approximate those of $\hat{\theta}$. In particular, we should expect the bootstrap variance $\mathrm{Var}_{\mathbf{P}_n}(\hat{\theta}^*_n)$ to closely approximate $\mathrm{Var}(\hat{\theta}_n)$. This is the rationale for the variance estimator (6) described earlier.

Subsampling[21] employs the same strategy as the bootstrap but is based on sampling without replacement. For $b := b(n)$ such that $b/n \to 0$, let S_b be the entire collection of subsets of $\{1, \ldots, n\}$ of size b. For each $s = \{i_1, \ldots, i_b\} \in S_b$, let $\hat{\theta}_{n,b,s} = \hat{\theta}(X_{i_1}, \ldots, X_{i_b})$ be the estimator evaluated using s. The goal is to estimate the sampling distribution of $n^{1/2}(\hat{\theta}_n - \theta(\mathbf{P}))$. It turns out that subsampling provides a consistent estimate of this distribution under fairly mild conditions. Let $\mathbb{Q}_n$ denote the distribution of $n^{1/2}(\hat{\theta}_n - \theta(P))$. Assume $\mathbb{Q}_n$ converges weakly to a proper limiting distribution $\mathbb{Q}$:

$$\mathbb{Q}_n \overset{\mathrm{d}}{\rightsquigarrow} \mathbb{Q}. \tag{8}$$

Then it follows[21] that the distribution function for the statistic $n^{1/2}(\hat{\theta}_n - \theta(P))$ can be approximated by the subsampling estimator

$$\widetilde{U}_{n,b}(x) = \frac{1}{C_b}\sum_{s \in S_b} 1\{b^{1/2}\left(\hat{\theta}_{n,b,s} - \hat{\theta}_n\right) \le x\}, \tag{9}$$

where $C_b = \binom{n}{b}$ is the cardinality of S_b. More formally, assuming (8) and $b/n \to 0$ for $b \to \infty$, then $\widetilde{U}_{n,b}(x) \xrightarrow{\text{p}} F(x) = \mathbb{Q}[-\infty, x]$ for each x that is a continuity point of the limiting cumulative distribution function F. The key to this argument is to recognize that due to (8) and $b/n \to 0$, $\widetilde{U}_{n,b}$ closely approximates

$$U_{n,b}(x) = \frac{1}{C_b} \sum_{s \in S_b} 1\{b^{1/2} \left(\hat{\theta}_{n,b,s} - \theta(\mathbf{P})\right) \leq x\},$$

which is a U-statistic[25] of order b. See Politis and Romano[21] for details.

The ability to approximate the distribution of $\hat{\theta}_n$ suggests, similar to the bootstrap, that we can approximate moments of $\hat{\theta}_n$ with those from the subsampled estimator; in particular, we should be able to approximate the variance. Unlike the bootstrap, however, subsampled statistics are calculated using a sample size b and not n. Therefore to estimate the variance of $\hat{\theta}_n$ we must apply a scaling factor to correct for sample size. The subsampled estimator for the variance is (see Radulović[26] and Section 3.3.1 from Politis and Romano[21])

$$\hat{v}_b = \frac{b/n}{C_b} \sum_{s \in S_b} \left(\hat{\theta}_{n,b,s} - \frac{1}{C_b} \sum_{s' \in S_b} \hat{\theta}_{n,b,s'} \right)^2. \tag{10}$$

The estimator (10) is closely related to the delete-d jackknife.[22] The delete-d estimator works on subsets of size $r = n - d$ and is defined as

$$\hat{v}_{J(d)} = \frac{r/d}{C_r} \sum_{s \in S_r} \left(\hat{\theta}_{n,r,s} - \hat{\theta}_n\right)^2.$$

With a little bit or rearrangement, this can be rewritten as

$$\hat{v}_{J(d)} = \frac{r/d}{C_r} \sum_{s \in S_r} \left(\hat{\theta}_{n,r,s} - \frac{1}{C_r} \sum_{s' \in S_r} \hat{\theta}_{n,r,s'} \right)^2 + \frac{r}{d} \left(\frac{1}{C_r} \sum_{s \in S_r} \hat{\theta}_{n,r,s} - \hat{\theta}_n \right)^2.$$

Setting $d = n - b$, we obtain

$$\hat{v}_{J(d)} = \frac{b/(n-b)}{C_b} \sum_{s \in S_b} \left(\hat{\theta}_{n,b,s} - \frac{1}{C_b} \sum_{s' \in S_b} \hat{\theta}_{n,b,s'} \right)^2 + \frac{b}{n-b} \underbrace{\left(\frac{1}{C_b} \sum_{s \in S_b} \hat{\theta}_{n,b,s} - \hat{\theta}_n \right)^2}_{\text{bias}}. \tag{11}$$

The first term closely approximates (10) since $b/n \to 0$, while the second term is a bias estimate of the subsampled estimator. Thus, the delete-d estimator (11) can be seen to be a bias corrected version of (10). Furthermore this correction is always upwards because the bias term is squared and always positive.

3.2.2 | Subsampling and delete-d jackknife algorithms

We can now describe our subsampling estimator for the variance of VIMP. In the following we assume b is some integer much smaller than n such that $b/n \to 0$.

b-subsampling estimator for $\text{Var}(I^{(j)}_{n,M})$

1. Draw a subsampling set $s \in S_b$. Let $\mathscr{L}_s$ be $\mathscr{L}$ restricted to s.
2. Calculate $I^{(j)}_{n,M}(\mathscr{L}_s)$, the finite forest estimator for VIMP using $\mathscr{L}_s$. Let $\hat{\theta}^{(j)}_{n,b,s}$ denote this value.
3. Repeat $K > 1$ times obtaining $\hat{\theta}^{(j)}_{n,b,s_1}, \ldots, \hat{\theta}^{(j)}_{n,b,s_K}$. Estimate $\text{Var}(I^{(j)}_{n,M})$ by $[b/(nK)] \sum_{k=1}^{K} (\hat{\theta}^{(j)}_{n,b,s_k} - \frac{1}{K} \sum_{k'=1}^{K} \hat{\theta}^{(j)}_{n,b,s_{k'}})^2$.

The delete-d jackknife estimator is obtained by a slight modification to the above algorithm:

delete-d jackknife estimator ($d = n - b$) for $\text{Var}(I^{(j)}_{n,M})$

1. Using the entire learning set $\mathscr{L}$, calculate the forest VIMP estimator $I^{(j)}_{n,M}(\mathscr{L})$. Let $\hat{\theta}^{(j)}_n$ denote this value.
2. Run the b-subsampling estimator, but replace the estimator in step 3 with $\left\{b/[(n-b)K]\right\} \sum_{k=1}^{K} (\hat{\theta}^{(j)}_{n,b,s_k} - \hat{\theta}^{(j)}_n)^2$.

4 | RANDOM FOREST REGRESSION, RF-R

4.1 | Simulations

In the following sections (Sections 4, 5, and 6) we evaluate the performance of the .164 bootstrap estimator, the b-subsampling estimator, and the delete-d jackknife variance estimator. We begin by looking at the regression setting. We used the following simulations to assess performance.

1. $y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon$; $\{x_j\} \sim \text{U}(0,1)$; $\varepsilon \sim \text{N}(0,1)$.
2. $y = (x_1^2 + [x_2 x_3 - (x_2 x_4)^{-1}]^2)^{1/2} + \varepsilon$; $x_1 \sim \text{U}(0,100)$, $x_2 \sim \text{U}(40\pi, 560\pi)$, $x_3 \sim \text{U}(0,1)$, $x_4 \sim \text{U}(1,11)$; $\varepsilon \sim \text{N}(0, 125^2)$.
3. $y = \tan^{-1}([x_2 x_3 - (x_2 x_4)^{-1}]/x_1) + \varepsilon$; $x_1 \sim \text{U}(0,100)$, $x_2 \sim \text{U}(40\pi, 560\pi)$, $x_3 \sim \text{U}(0,1)$, $x_4 \sim \text{U}(1,11)$; $\varepsilon \sim \text{N}(0, .1^2)$.
4. $y = x_1 x_2 + x_3^2 - x_4 x_7 + x_8 x_{10} - x_6^2 + \varepsilon$; $\{x_j\} \sim \text{U}(-1,1)$; $\varepsilon \sim \text{N}(0, .1^2)$.
5. $y = 1\{x_1 > 0\} + x_2^3 + 1\{x_4 + x_6 - x_8 - x_9 > 1 + x_{10}\} + \exp(-x_2^2) + \varepsilon$; $\{x_j\} \sim \text{U}(-1,1)$; $\varepsilon \sim \text{N}(0, .1^2)$.
6. $y = x_1^2 + 3x_2^2 x_3 \exp(-|x_4|) + x_6 - x_8 + \varepsilon$, $\{x_j\} \sim \text{U}(-1,1)$; $\varepsilon \sim \text{N}(0, .1^2)$.
7. $y = 1\{x_1 + x_4^3 + x_9 + \sin(x_2 x_8) + \varepsilon > 0.38\}$; $\{x_j\} \sim \text{U}(-1,1)$; $\varepsilon \sim \text{N}(0, .1^2)$.
8. $y = \log(x_1 + x_2 x_3) - \exp(x_4 x_5^{-1} - x_6) + \varepsilon$; $\{x_j\} \sim \text{U}(0.5,1)$; $\varepsilon \sim \text{N}(0, .1^2)$.
9. $y = x_1 x_2^2 |x_3|^{1/2} + \lfloor x_4 - x_5 x_6 \rfloor + \varepsilon$; $\{x_j\} \sim \text{U}(-1,1)$; $\varepsilon \sim \text{N}(0, .1^2)$.
10. $y = x_3(x_1 + 1)^{|x_2|} - (x_5^2[|x_4| + |x_5| + |x_6|]^{-1})^{1/2} + \varepsilon$; $\{x_j\} \sim \text{U}(-1,1)$; $\varepsilon \sim \text{N}(0, .1^2)$.
11. $y = \cos(x_1 - x_2) + \sin^{-1}(x_1 x_3) - \tan^{-1}(x_2 - x_3^2) + \varepsilon$; $\{x_j\} \sim \text{U}(-1,1)$; $\varepsilon \sim \text{N}(0, .1^2)$.
12. $y = \varepsilon$; $\varepsilon \sim \text{N}(0,1)$.

In all 12 simulations, the dimension of the feature space was set to $p = 20$. This was done by adding variables unrelated to y to the design matrix. We call these noise variables. In simulations 1–3, noise variables were U(0, 1); for simulations 4–11, noise variables were U(−1, 1); for simulation 12, noise variables were N(0, 1). All features (strong and noisy) were simulated independently. Simulations 1–3 are the well known Friedman 1, 2, 3 simulations.[27,6] Simulations 4–11 were inspired from COBRA.[28] Simulation 12 is a pure noise model.

The sample size was set at $n = 250$. Subsampling was set at a rate of $b = n^{1/2}$, which in this case is $b = 15.8$. We can see that practically speaking this a very small sample size and allows subsampled VIMP to be rapidly computed. The value of d for the delete-d jackknife was always set to $d = n - b$. The number of bootstraps was set to 100 and the number of subsamples was set to 100. Note that this not a large number of bootstraps or subsampled replicates. However, they represent values practitioners are likely to use in practice, especially for big data, due to computational costs.

All RF calculations were implemented using the `randomForestSRC` R-package.[29] The package runs in OpenMP parallel processing mode, which allows for parallel processing on user desktops, as well as large scale computing clusters. The package now includes a dedicated function "`subsample`" which implements the three methods studied here. The `subsample` function was used for all calculations. All RF calculations used 250 trees. Tree nodesize was set to 5 and $p/3$ random feature selection used (these are default settings for regression). Each simulation was repeated independently 250 times. RF parameters were kept fixed over simulations. All calculations related to prediction error and VIMP were based on squared error loss, $\ell(Y, \hat{h}) = (Y - \hat{h})^2$.

4.2 | Estimating the true finite standard error and true finite VIMP

Each procedure provides an estimate of the $\text{Var}(I_{n,M}^{(j)})$. We took the square root of this obtain an estimate for the standard error of VIMP, $(\text{Var}(I_{n,M}^{(j)}))^{1/2}$. To assess performance in estimating the standard error we used the following strategy to approximate the uknown parameter $\text{Var}(I_{n,M}^{(j)})$. For each simulation model, we drew 1000 independent copies of the data, and for each of these copies, we calculated the finite forest VIMP, $I_{n,M}^{(j)}$. The same sample size of $n = 250$ was used and all forest tuning parameters were kept the same as outlined above. We used the variance of these 1000 values to estimate $\text{Var}(I_{n,M}^{(j)})$. We refer to the square root of this value as the true finite standard error. Additionally, we averaged the 1000 values to estimate $\mathbb{E}[I_{n,M}^{(j)}] = \mathbb{E}[I(X^{(j)}, \mathscr{L})]$. We call this the true finite VIMP.

4.3 | Results

Performance of methods was assessed by bias and standardized mean-squared-error (SMSE). The bias for a method was obtained by averaging its estimated standard error over the 250 replications and taking the difference between this and the true finite standard error. MSE was estimated by averaging the squared difference between a method's estimated value for the standard error and the true finite standard error. SMSE was defined by dividing MSE by the true finite standard error. In evaluating these performance values, we realized it was important to take into account signal strength of a variable. In our simulations there are

noisy variables with no signal. There are also variables with strong and moderately strong signal. Therefore, to better understand performance differences, results were stratified by size of a variable's true finite VIMP. In total, there were 240 variables to be dealt with (12 simulations, each with $p = 20$ variables). These 240 variables were stratified into 6 groups based on 10, 25, 50, 75, and 90th percentiles of true finite VIMP (standardized by the Y variance to make VIMP comparable across simulations). Bias and SMSE for the 6 groups are displayed in Figure 1.

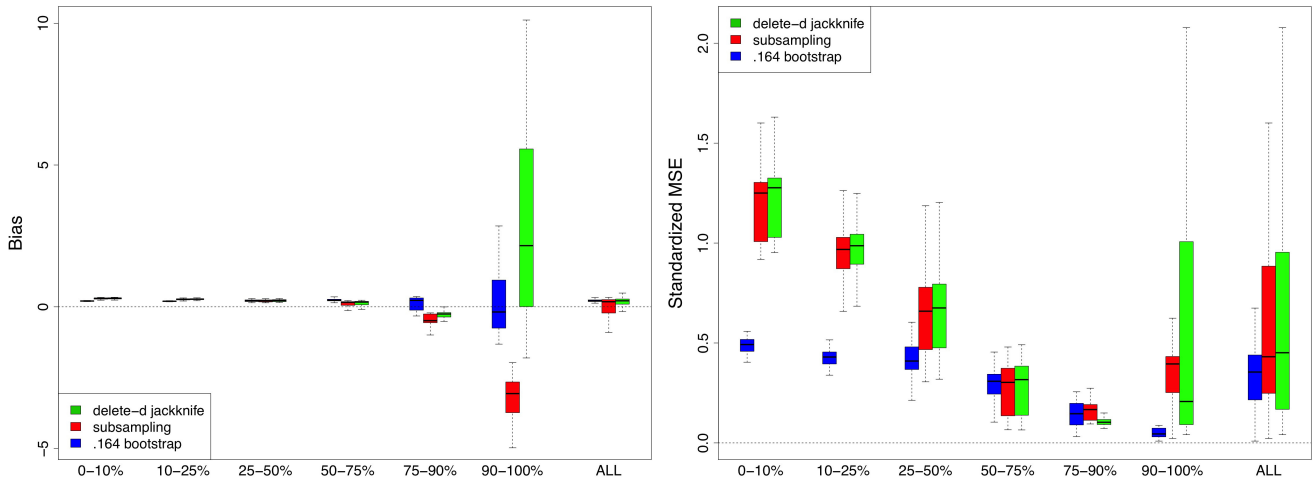

FIGURE 1: *Bias and standardized mean-squared-error (SMSE) performance for estimating VIMP standard error from RF-R (RF-regression) simulations. In total there are 240 variables (12 simulations, $p = 20$ variables in each simulation). These 240 variables have been stratified into 6 groups based on 10, 25, 50, 75, and 90th percentiles of true finite VIMP. Extreme right boxplots labeled "ALL" display performance for all 240 variables simultaneously.*

All methods exhibit low bias for small VIMP. As VIMP increases, corresponding to stronger variables, bias for the subsampling estimator increases. Its bias is negative showing that it underestimates variance. The delete-d estimator does much better. This is due to the bias correction factor discussed earlier (see (11)) which kicks in when signal increases. The pattern seen for bias is reflected in the results for SMSE: the delete-d is similar to the subsampling estimator except for large VIMP where it does better. Overall, the .164 estimator is the best of all three methods. On the other hand, it is hundreds of times slower.

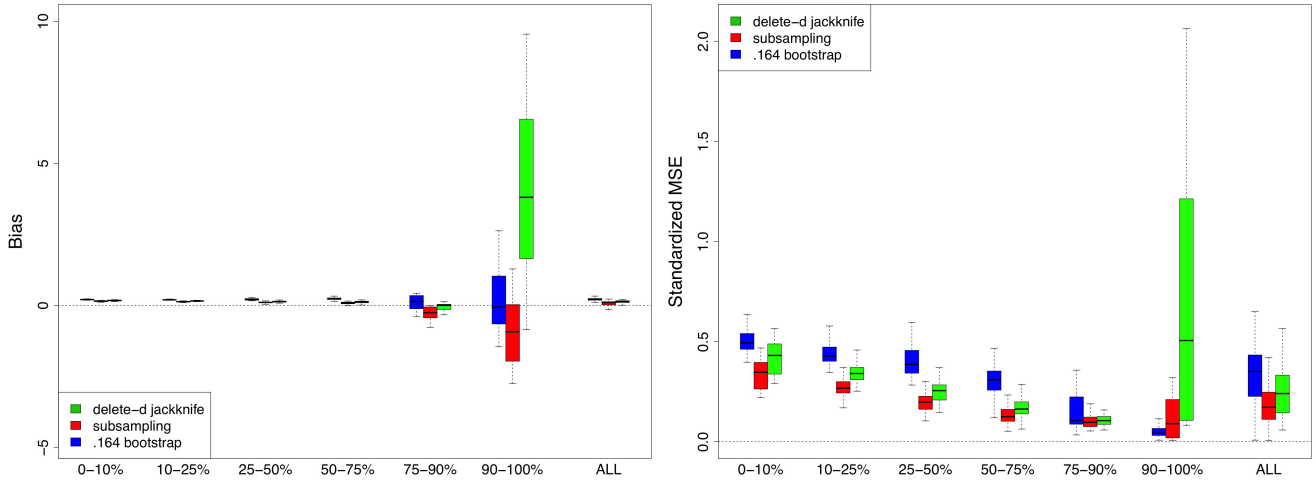

FIGURE 2: *Results from RF-R simulations but with increased subsampling rate $b = n^{3/4}$. Notice the improvement in bias and SMSE for the subsampling estimator.*

This shows that the delete-d estimator should be used when bias is an issue. However, bias of the subsampling estimator can be improved by increasing the subsampling rate. Figure 2 reports the results from the same set of simulations but using an

increased sampling rate $b = n^{3/4}$ ($b = 62.8$). Both estimators improve overall but note the improvement in bias and SMSE for the subsampling estimator relative to the delete-d estimator. Also notice that both estimators now outperform the .164 bootstrap.

4.4 | Confidence intervals for VIMP

The subsampling distribution (9) discussed in Section 3 can also be used to calculate nonparametric confidence intervals.[21] The general idea for constructing a confidence interval for a target parameter $\theta(\mathbf{P})$ is as follows. Define the $1-\alpha$ quantile for the subsampling distribution as $c_{n,b}(1-\alpha) = \inf\{x : \widetilde{U}_{n,b}(x) \geq 1-\alpha\}$. Similarly, define the $1-\alpha$ quantile for the limiting distribution $\mathbb{Q}$ of $n^{1/2}(\hat{\theta}_n - \theta(\mathbf{P}))$ as $c(1-\alpha) = \inf\{t : F(x) = \mathbb{Q}[-\infty, x] \geq 1-\alpha\}$. Then, assuming (8) and $b/n \to 0$, the interval

$$\left[\hat{\theta}_n - n^{-1/2} c_{n,b}(1-\alpha), \infty\right) \tag{12}$$

contains $\theta(\mathbf{P})$ with asymptotic probability $1-\alpha$ if $c(1-\alpha)$ is a continuity point of F.

While (12) can be used to calculate a nonparametric confidence interval for VIMP, we have found that a more stable solution can be obtained if we are willing to strengthen our asssumptions to include asympotic normality. Let $\hat{\theta}_n^{(j)} = I_{n,M}^{(j)}$ denote the finite forest estimator for VIMP. We call the limit of $\hat{\theta}_n^{(j)}$ as $n, M \to \infty$ the true VIMP and denote this value by $\theta_0^{(j)}$,

$$\theta_0^{(j)} = \lim_{n,M\to\infty} \hat{\theta}_n^{(j)} := \lim_{n,M\to\infty} I_{n,M}^{(j)}.$$

Let $\hat{v}_n^{(j)}$ be an estimator for $\text{Var}(I_{n,M}^{(j)})$. Assuming asymptotic normality,

$$\frac{\hat{\theta}_n^{(j)} - \theta_0^{(j)}}{\sqrt{\hat{v}_n^{(j)}}} \overset{\text{d}}{\rightsquigarrow} \text{N}(0,1), \tag{13}$$

an asymptotic $100(1-\alpha)$ confidence region for $\theta_0^{(j)}$, the true VIMP, can be defined as

$$\hat{\theta}_n^{(j)} \pm z_{\alpha/2}\sqrt{\hat{v}_n^{(j)}},$$

where $z_{\alpha/2}$ is the $1-\alpha/2$-quantile from a standard normal, $\Pr\{\text{N}(0,1) \leq z_{\alpha/2}\} = 1-\alpha/2$.

4.5 | Justification of normality

To provide justification for assumption (13), we re-ran our previous simulations 1000 times independently. For each Monte Carlo replication and each simulation model we calculated the finite forest VIMP for a variable and centered it by its mean value from the 1000 simulations and divided this centered value by the standard deviation of the 1000 VIMP values. All experimental parameters were held at the same values as before except for the sample size which was increased to $n = 2500$. The left-hand side of Figure 3 displays normal quantile plots for standardized VIMP for each of the 12 simulation models. On the y-axis are the quantiles for the standardized VIMP while the x-axis are corresponding N(0, 1) quantiles. The right-hand side displays quantile bias defined as the difference between the quantile for standardized VIMP to the quantile for a standard normal. Values are displayed for 5,10,25,50,75,90,95 percentile values. The results generally confirm that (13) holds. Deviations from normality occur primarily in the tails but these are reasonable and expected in finite sample settings. For example, the median bias is about 0.02 for the 95th percentile. Thus the standardized VIMP quantile differs from the true standard normal value of 1.645 by only a value of 0.02. Also, observe that overall mean bias is near zero (far right boxplot).

Coverage probabilities for 90% confidence intervals are provided in Figure 4. The left and right figures correspond to simulations of Figures 1 and 2 respectively (recall Figure 2 is based on a larger subsampling rate $b = n^{3/4}$). Confidence intervals for the subsampling and delete-d estimators use asymptotic normality. For direct comparison, asymptotic normal bootstrap confidence intervals are also provided. All procedures tend to produce confidence intervals that are too large when VIMP is small (reflected by coverage probabilities exceeding the targeted 90% level). This is actually a good feature as it implies they tend to over-estimate confidence intervals for noisy variables, thereby making them less likely to be selected. For larger VIMP, in the left figure, the subsampling estimator tends to produce intervals that are too small. As mentioned earlier this is because it tends to underestimate the variance. The delete-d estimator performs much better. However, when the subsampling rate is increased (right figure), the subsampling estimator is generally superior to the delete-d estimator. Its overall mean coverage rate is 92 which is much better than the delete-d and bootstrap which achieve coverage rates of 97 which are too high.

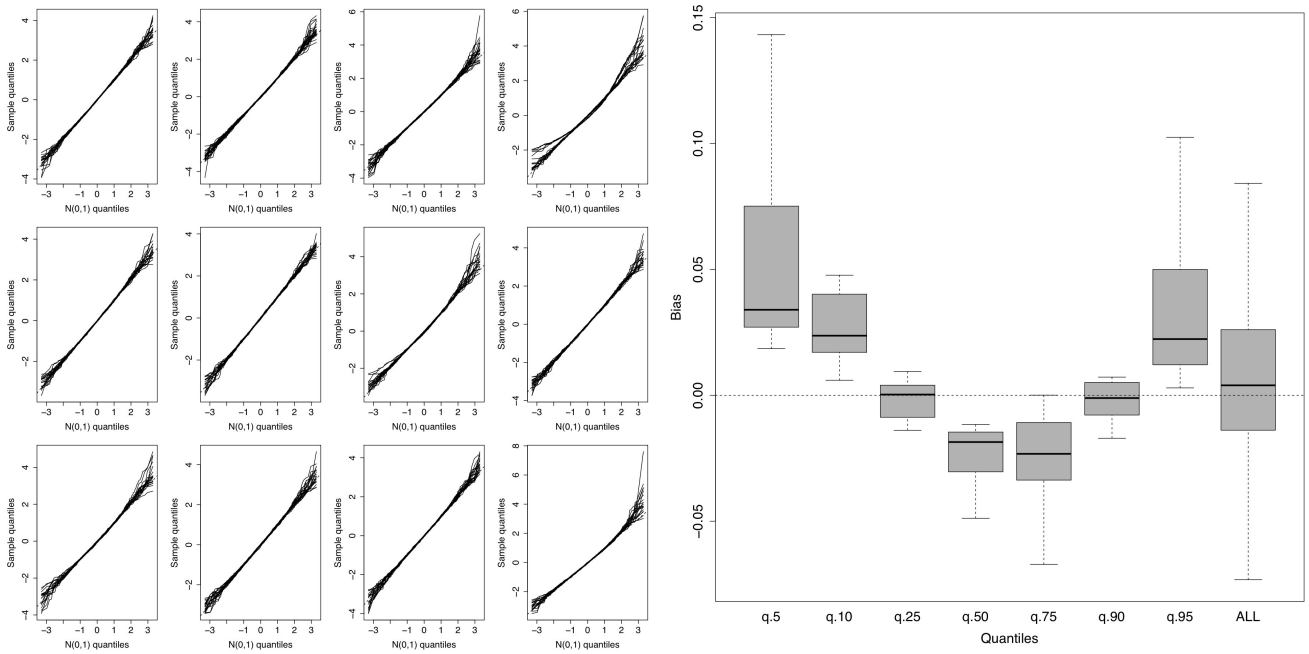

FIGURE 3: *Assessing asymptotic normality of VIMP from RF-R simulations. Left-hand figure displays normal quantile plots for standardized VIMP for each of the 12 simulations. Right-hand figure displays bias of VIMP quantiles compared to standard normal quantiles for all 240 variables from all 12 simulations.*

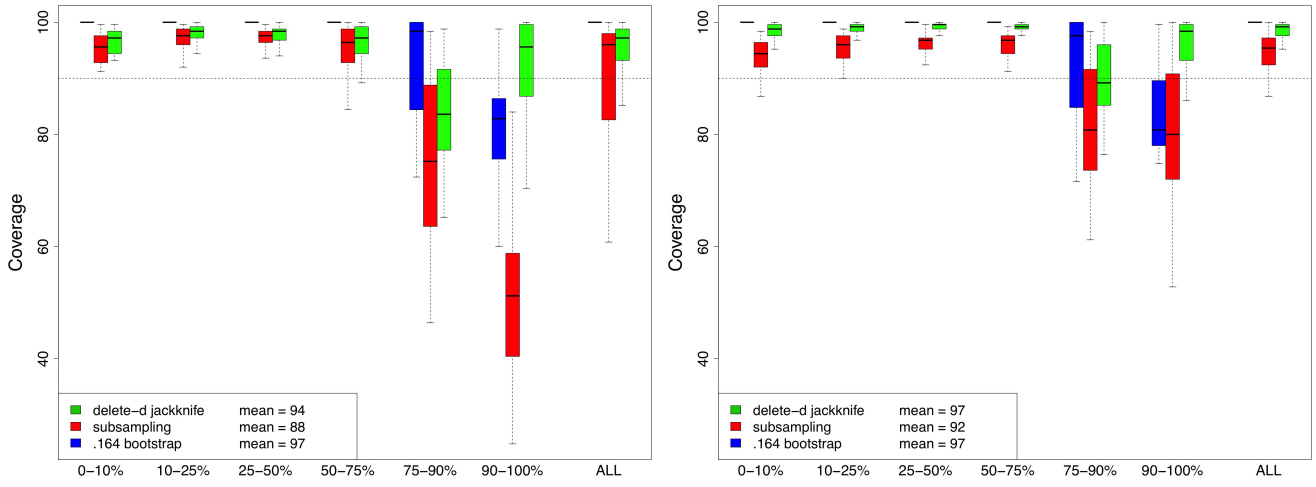

FIGURE 4: *Coverage of VIMP 90% asymptotic normal confidence intervals from RF-R simulations. Left and right hand side figures based on subsampling rates $b = n^{1/2}$ and $b = n^{3/4}$ respectively. Confidence regions for the 240 variables from the 12 simulation experiments have been stratified into 6 groups based on 10, 25, 50, 75, and 90th percentiles of true finite VIMP values.*

5 | RANDOM FOREST CLASSIFICATION, RF-C

5.1 | Simulations

The following simulations were used to study performance of methods in the classification problem.

1. Threenorm simulation using `mlbench.threenorm` from the `mlbench` R-package.[30]
2. Two class simulation using `twoClassSim` from the `caret` R-package[31] with 2 factors, 5 linear and 3 non-linear variables.

3. Same as 2, but with a $\rho = .75$ exchangeable correlation.
4. Same as 2, but with 15 linear variables.
5. Same as 2, but with 15 linear variables and a $\rho = .75$ exchangeable correlation.
6. RF-R simulation 6 with y discretized into two classes based on its median.
7. RF-R simulation 8 with y discretized into two classes based on its median.
8. RF-R simulation 9 with y discretized into three classes based on its 20 and 75th quantiles.
9. RF-R simulation 10 with y discretized into three classes based on its 20 and 75th quantiles.
10. RF-R simulation 11 with y discretized into three classes based on its 20 and 75th quantiles.

In simulation 1, the feature space dimension was $p = 20$. Simulations 2–4 added $d = 10$ noise variables (see the `caret` package for details). Simulations 6–10 added $d = 10$ noise variables from a U$[-1, 1]$ distribution. Experimental parameters were set as in RF-R simulations: $n = 250$; $b = \{n^{1/2}, n^{3/4}\}$; 100 bootstrap samples; 100 subsample draws. Parameters for `randomForestSRC` were set as in RF-R except for random feature selection which used $p^{1/2}$ random features (default setting). The entire procedure was repeated 250 times.

5.2 | Brier score

Error performance was assessed using the normalized Brier score. Let $Y \in \{1, \ldots, K\}$ be the response. If $0 \leq \hat{p}_k \leq 1$ denotes the predicted probability that Y equals class k, $k = 1, \ldots, K$, the normalized Brier score is defined as

$$\text{BS}^* = \frac{100K}{K-1} \sum_{k=1}^{K} \left(1\{Y = k\} - \hat{p}_k\right)^2.$$

Note that the normalizing constant $100K/(K-1)$ used here is different than the value $1/K$ typically used for the Brier score. We multiply the traditional Brier score by $100K^2/(K-1)$ because we have noticed that the value for the Brier score under random guessing depends on the number of classes, K. If K increases, the Brier score under random guessing converges to 1. The normalizing constant used here resolves this problem and yields a value of 100 for random guessing, regardless of K. Thus, anything below 100 signifies a classifier that is better than pure guessing. A perfect classifier has value 0.

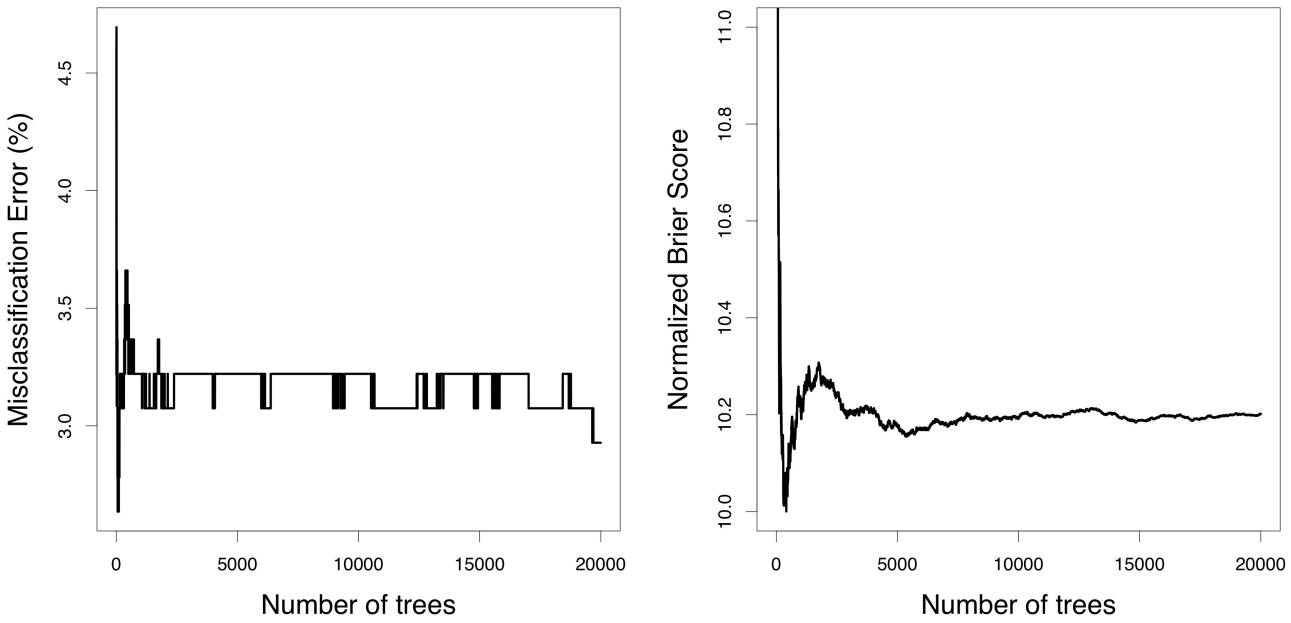

FIGURE 5: *OOB misclassification error rate versus OOB normalized Brier score for Wisconsin breast cancer data (obtained from the* `mlbench` *R-package). Note the fluctuations in misclassification error even after 20,000 trees in contrast to the stable behavior of the Brier score.*

Although misclassification error is the commonly used performance measure in classification problems, it cannot be overstated how much stabler the Brier score is. This stability will naturally lend itself to stabler estimates of VIMP which is why we have

chosen to use it. As a simple illustration of this, consider Figure 5 which displays the OOB misclassification rate and OOB normalized Brier score for the Wisconsin breast cancer data from the `mlbench` R-package. Observe that misclassification error remains unstable even with $M = 20,000$ trees.

5.3 | Results

Figure 6 displays the results from the RF-C simulations. Values are displayed as in RF-R. Left and right hand side figures are based on subsampling rates $b = n^{1/2}$ and $b = n^{3/4}$. The top and middle figures show bias and standardized MSE for estimating the standard error. The bottom figures are coverage probabilities for 90% confidence intervals. VIMP confidence intervals were calculated using asymptotic normality as in RF-R (see the Appendix for justification of normality). The conclusions from Figure 6 are similar to those for RF-R. The delete-d jackknife is more accurate in estimating the standard error for strong variables when the subsampling rate is small, but as b increases, the subsampling estimator improves. Both estimators generally improve with increased b. Note unlike RF-R, coverage probability for the delete-d jackknife is better than the subsampling estimator. This is probably because there are more variables with moderate signal in these simulations.

6 | RANDOM SURVIVAL FORESTS, RSF

Now we consider the survival setting. We begin by first defining the survival framework using the notation of Section 2. Following this we discuss two different methods that can be used for measuring prediction error in survival settings. Following this are illustrative examples.

6.1 | Notation

We assume a traditional right-censoring framework. The response is $Y = (T, \delta)$, where $T = \min(T^o, C^o)$ is the observed survival time and $\delta = 1\{T^o \leq C^o\}$ is the right-censoring indicator. Here (T^o, C^o) denote the unobserved event and censoring times. Thus $\delta = 1$ denotes an event such as death, while $\delta = 0$ denotes a right-censored case. The target function h is the conditional survival function $h(\mathbf{x}) = \mathbb{P}\{T^o > t | \mathbf{X} = \mathbf{x}\}$, where t is some selected time point.

6.2 | Weighted Brier score

Let $\hat{h}$ be an estimator of h. One method for measuring performance of $\hat{h}$ is the weighted Brier score,[23, 24] defined as

$$\text{wBS}(t) = \left(1\{T > t\} - \hat{h}\right)^2 w(t, Y, G),$$

where $w(t, Y, G)$ is the weight defined by

$$w(t, Y, G) = \frac{1\{T \leq t\}\delta}{G(T-)} + \frac{1\{T > t\}}{G(t)},$$

and $G(t) = \mathbb{P}\{C^o > t\}$ is the survival function of the censoring variable C^o. Using the notation of Section 2, the loss function ℓ under the weighted Brier score can be written as

$$\ell(Y, \hat{h}) = \left(1\{T > t\} - \hat{h}\right)^2 w(t, Y, G).$$

This assumes G is a known function, but in practice G must be estimated.[23, 24] Thus if $\hat{G}$ is an estimator of G, $w(t, Y, G)$ is replaced by $w(t, Y, \hat{G})$.

6.3 | Concordance index

Harrell's concordance index[32] is another measure of prediction performance that can be used in survival settings. The concordance index estimates the accuracy of the predictor $\hat{h}$ in ranking two individuals in terms of their survival. A value of 1 represents an estimator that has perfect discrimination, whereas a value of 0.5 indicates performance on par with a random coin toss. This intuitive interpretation of performance has made Harrell's concordance index very popular and for this reason we will base our analysis on it. Note that because the concordance index is calculated by comparing discordant pairs to concordant pairs, and therefore is very complex, it is not possible to express it in terms of the ℓ-loss function of Section 2. However, this just means that VIMP based on Harrell's concordance index is not easily described notationally in terms of a formal loss, but this does not pose any problems to the application of our methodology. Permutation VIMP based on Harrell's concordance index is well defined and can be readily calculated.[2]

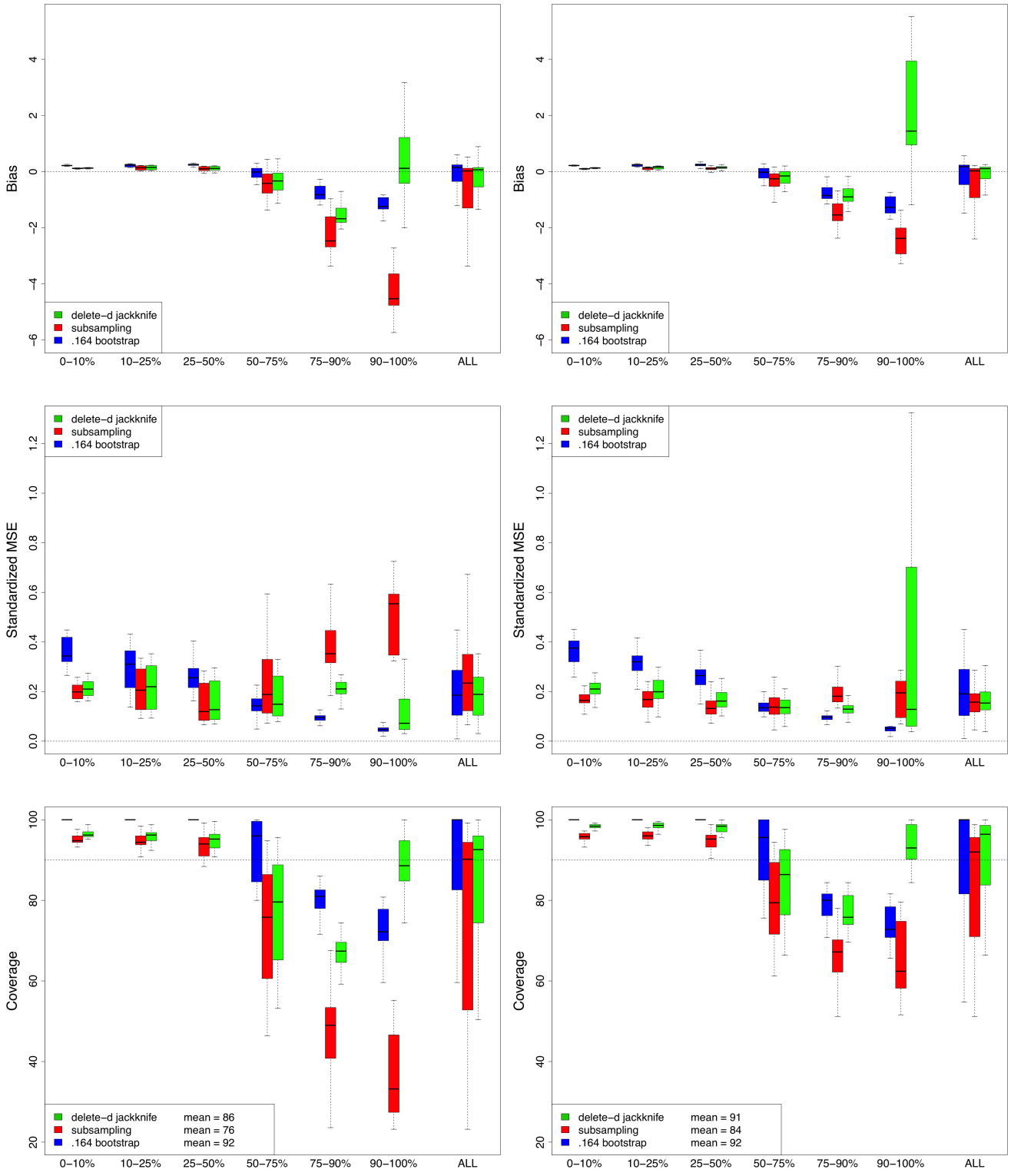

FIGURE 6: *Results from RF-C (RF-classification) simulations showing performance of subsampling, delete-d jackknife, and .164 bootstrap. Left and right hand side figures are based on subsampling rates* $b = n^{1/2}$ *and* $b = n^{3/4}$*. Top and middle figures display bias and standardized MSE for estimating VIMP standard error. Bottom figure displays coverage for VIMP 90% asymptotic normal confidence intervals. Results have been stratified into 6 groups based on 10, 25, 50, 75, and 90th percentiles of true finite VIMP.*

6.4 | Systolic heart failure

For our first illustration we consider a survival data set of $n = 2231$ cardiovascular patients. All patients suffered from systolic heart failure and all underwent cardiopulmonary stress testing. The outcome was defined as all cause mortality. Over a mean follow-up of 5 years, 742 of the patients died. Patient variables included baseline characteristics and exercise stress test results ($p = 39$). More detailed information regarding the data can be found in Hsich et al.[33]

A RSF analysis was run on the data. A total of 250 survival trees were grown using a nodesize value of 30 with all other parameters set to default values used by RSF in `randomForestSRC` software. Performance was measured using the C-index defined as one minus the Harrell concordance index.[2] The delete-d jackknife estimator was calculated using 1000 subsampled values using a $b = n^{1/2}$ subsampling rate ($b = 47.2$). We preferred to use the delete-d jackknife rather than the subsampling estimator because of the low subsampling rate. Also, we did not use the .164 estimator because it was too slow.

The 95% asymptotic normal confidence intervals are given in Figure 7. VIMP values have been multiplied by 100 for convenient interpretation as percentage. BUN (blood urea nitrogen), exercise time, and peak VO_2 have the largest VIMP with confidence intervals well bounded away from zero. All three variables are known to be highly predictive of heart failure and these findings are not surprising. More interesting, however, are several variables with moderate sized VIMP which have confidence regions bounded away from zero. Some examples are creatinine clearance, sex, LVEF (left ventricular ejection fraction) and use of beta-blockers. The finding for sex is especially interesting because sex is often under appreciated for predicting heart failure.

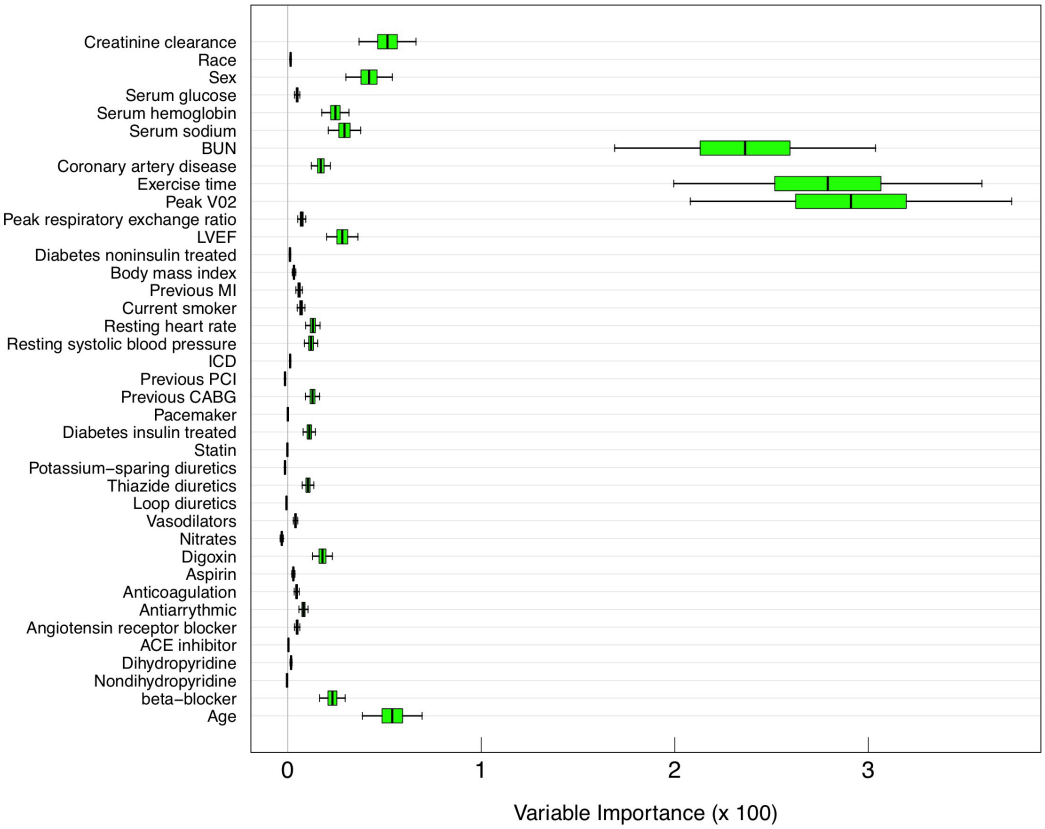

FIGURE 7: *Delete-d jackknife 95% asymptotic normal confidence intervals from RSF analysis of systolic heart failure data.*

6.5 | Survival simulations

Next we study peformance using simulations. We use the three survival simulation models described by Breiman in his 2002 Wald lectures.[34] Let $h(t, \mathbf{x})$ be the hazard function for covariate $\mathbf{x}$ at time t. Simulations are as follows.

1. $h(t, \mathbf{x}) = \exp(x_1 - 2x_4 + 2x_5)$.
2. If $x_1 \leq .5$, $h(t, \mathbf{x}) = \exp(x_2)1\{t \notin [.5, 2.5]\}$. If $x_1 > .5$, $h(t, \mathbf{x}) = \exp(x_3)1\{t \notin [2.5, 4.5]\}$.
3. $h(t, \mathbf{x}) = (1 + z_2 t)\exp(z_1 + z_2 t)$, where $z_1 = .5x_1$ and $z_2 = x_4 + x_5$.

In all simulations, covariates were independently sampled from a U[0, 1] distribution. Noise variables were added to increase the dimension to $p = 10$. Simulation 1 corresponds to a Cox model. Simulations 2 and 3 are non-proportional hazards. Censoring was simulated independently of time in all simulations. Censoring rates were 19%, 15%, and 29% respectively.

RSF was fit using `randomForestSRC` using the same tuning values as in RF-C simulations (nodesize: 5, random feature selection: $p^{1/2}$) Experimental parameters were kept the same as previous simulations. Experiments were repeated 250 times. Results are displayed using the same format as RF-R and RF-C and are provided in Figure 8. The results generally mirror our earlier findings: bias for the subsampling estimator improves relative to the delete-d jackknife with increasing subsampling rate.

6.6 | Competing risk simulations

Here we study performance of the methods in a competing risk setting. For our analysis we use the competing risk simulations from Ishwaran et al.[3] Simulations were based on a Cox-exponential hazards model with two competing events. Covariates had differing effects on the hazards. Models included covariates common to both hazards as well as covariates unique to only one hazard. We considered three of the simulations from Section 6.1 of Ishwaran et al.[3]

1. Linear model. All p covariate effects are linear.
2. Quadratic model. A subset of the p covariate effects are quadratic.
3. Interaction model. Same as 1, but interactions between certain p variables were included.

The feature dimension was $p = 12$ for simulations 1 and 2, and $p = 17$ for simulation 3 (i.e. 5 interaction terms were added). Covariates were sampled from both continuous and discrete distributions. Performance was measured using the time truncated concordanace index.[3] Without loss of generality we record performance for variables related to event 1 only. RSF competing risk trees were constructed using log-rank splitting with weight 1 on event 1 and weight 0 on event 2 (this ensures VIMP identifies only those variables affecting the event 1 cause). RSF parameters and experimental parameters were identical to the previous simulations. For brevity, results are given in the Appendix in Figure 11. The results mirror our previous findings.

7 | DISCUSSION

7.1 | Summary

One widely used tool for peering inside the RF "black box" is variable importance (VIMP). But analyzing VIMP is difficult because of the complex nature of RF. Given the difficulties of theoretical analysis, our strategy was to approximate the distribution of VIMP through the use of subsampling, a general methodology for approximating distributions of complex statistics. We described a general procedure for estimating the variance of VIMP and for constructing confidence intervals.

We compared our subsampling estimator, and also the closely related delete-d jackknife,[22] to the .164 bootstrap estimator, a modified bootstrap procedure designed to address ties in OOB data. Using extensive simulations involving regression, classification, and survival data, a consistent pattern of performance emerged for the three estimators. All procedures tended to under estimate variance for strong variables and over estimate variance for weak variables. This was especially problematic for the subsamping estimator in low subsampling rate scenarios. The delete-d jackknife did much better in this case due to its bias correction. Both of these methods improved with increasing subsampling rate, eventually outperforming the .164 bootstrap.

7.2 | Computational speed

Overall, we generally prefer the delete-d jackknife because of its better peformance under low subsampling rates, which we feel will be the bulk of applications due to the computational complexity of VIMP. Consider survival with concordance error rates, the most computationally expensive setting for VIMP. The concordance index measures concordance and discordance over pairs of points, a $O(n^2)$ operation. With M trees, the number of computations is $O(n^2 M)$ for a method like the .164 bootstrap. On the other hand, employing a subsampling rate of $b = n^{1/2}$ reduces this to $O(nM)$, a factor of n times smaller. The resulting increase in speed will be of tremendous advantage in big data settings.

7.3 | Practical Guidelines

One of the major applications of our methodolgy will be variable selection. Below we provide some practical guidelines for this setting:

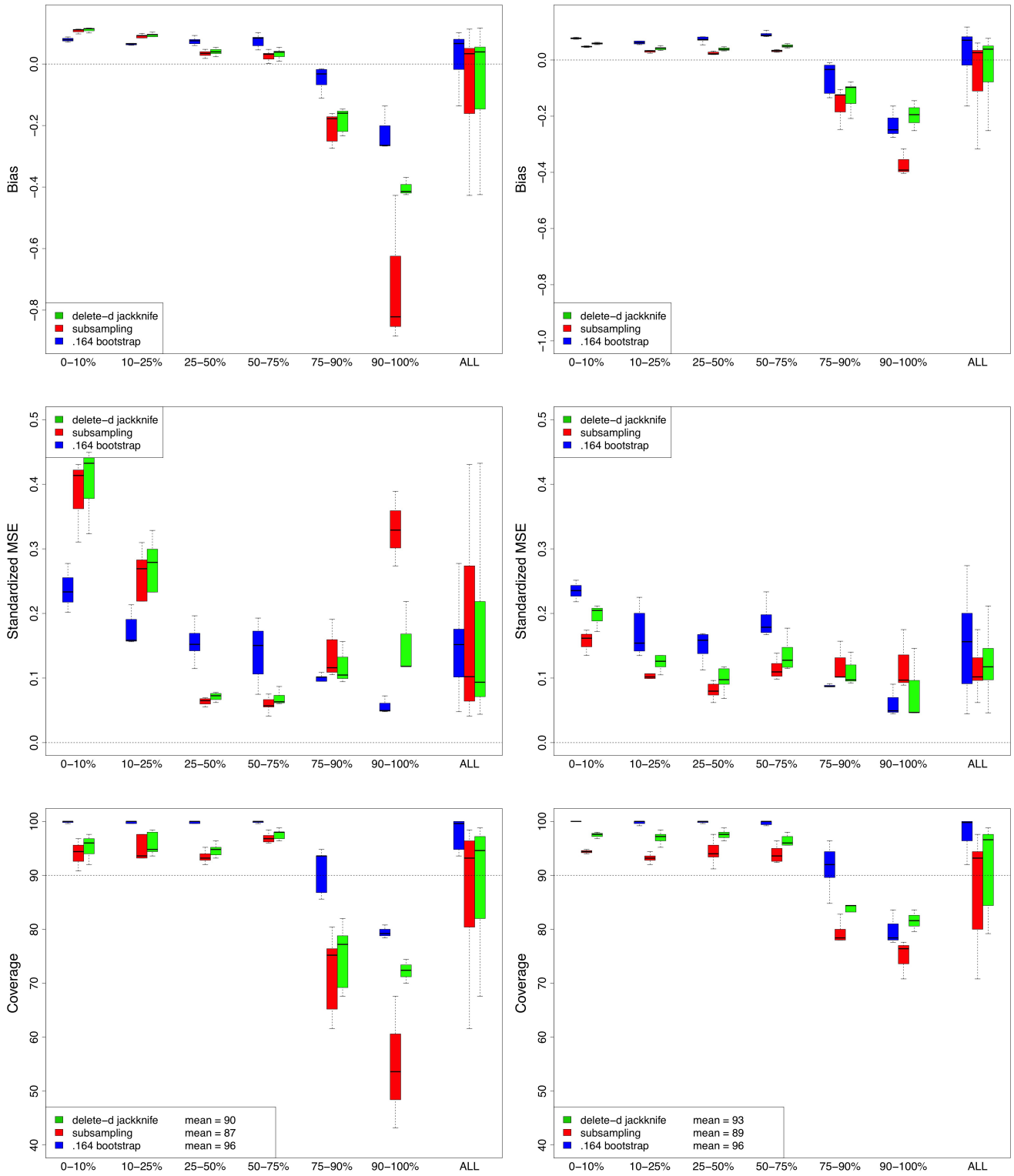

FIGURE 8: *Results from RSF simulations showing performance of subsampling, delete-d jackknife, and .164 bootstrap. Left and right hand side figures are based on subsampling rates $b = n^{1/2}$ and $b = n^{3/4}$. Top and middle figures display bias and standardized MSE for estimating VIMP standard error. Bottom figure displays coverage for VIMP 90% asymptotic normal confidence intervals. Results have been stratified into 6 groups based on 10, 25, 50, 75, and 90th percentiles of true finite VIMP.*

1. Use asymptotic normal confidence intervals derived from the delete-d jackknife variance estimator.
2. A good default subsampling rate is $b = n^{1/2}$. As mentioned, this will substantially reduce computational costs in big n problems. In small n problems, while this might seem overly aggressive leading to small subsamples, our results have shown solid performance even when $n = 250$.
3. The α value for the confidence region should be chosen using typical values such as $\alpha = .1$ or $\alpha = .05$. Outside of extreme settings such as high-dimensional problems, our experience suggests this should work well.

The above guidelines *are only meant to be starting points for the analyst* and obviously there will be exceptions to the rules. However, as way of support for these recommendations we did the following variable selection experiment. We re-ran the RF-R simulations of Section 4. Variables were selected using $100(1-\alpha)\%$ delete-d jackknife asymptotic normal confidence intervals. The true positive rate (TPR) and true negative rate (TNR) was calculated for each simulation and results averaged over 250 independent runs. TPR was defined as the fraction of true signal variables identified. TNR was the fraction of noisy signal variables identified.

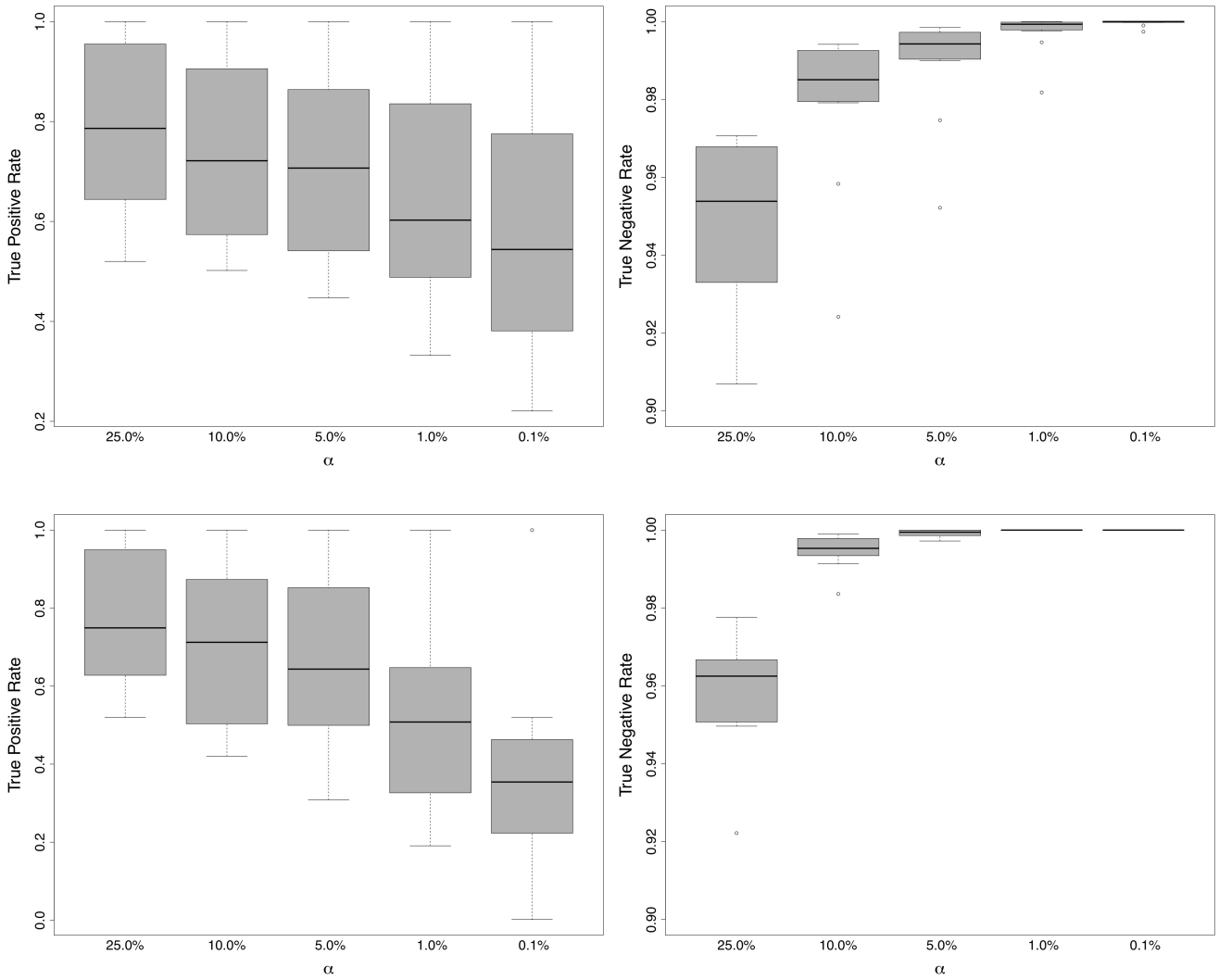

FIGURE 9: *Results from variable selection experiment using RF-R simulations of Section 4. Displayed are the true positive rate (TPR) and true negative rate (TNR) for variables selected using* $100(1-\alpha)\%$ *delete-d jackknife confidence regions where* $\alpha = .25,.1,.01,.001$. *Top and bottom figures are based on subsampling rates* $b = n^{1/2}$ *and* $b = n^{3/4}$.

Figure 9 displays the averaged TPR and TNR for each of the 12 simulation experiments under levels of significance $\alpha = .25,.1,.01,.001$. The top panel was calculated under a subsampling rate $b = n^{1/2}$, while the bottom panel used $b = n^{3/4}$. A similar

pattern for TPR and TNR values is generally observed for both sampling rates: TPR decreases with increasing α; TNR increases with α. For $b = n^{3/4}$, TNR rates are slightly better, however for $b = n^{1/2}$, TPR rates are slightly better. Thus if the goal is finding true signal the edge goes to $b = n^{1/2}$.

Focusing on the top panel corresponding to the recommended $n^{1/2}$ subsampling rate, a striking pattern for TPR is that while TPR decreases with α, the decline is fairly slow. This is interesting given the wide range of α values from 25% to 0.1%. These values are extreme and unlikely to be used in practice and yet TPR results remain quite robust. Values for TNR are also fairly robust to α, although TNR values appear relatively too small when $\alpha = .25$. The value $\alpha = .25$ is too extreme and creates overly narrow confidence regions causing noisy variables to be misclassified as signal variables. Generally, however, values $\alpha = .1, .05, .01$ perform very well under both TNR and TPR.

7.4 | Theoretical considerations

The key assumption underlying subsampling is the existence of a limiting distribution (8) for the estimator. However, as discussed earlier, theoretical results for VIMP are difficult to come by and establishing a result like (8) for something as complicated as permuation importance is not easy. As a token, we would like to offer some partial insight into VIMP for the regression case (RF-R), perhaps pointing the way for more work in this area. As shown in the Appendix (see Theorem 1), assuming an additive model $h(\mathbf{X}) = \sum_{j=1}^{p} h_j(X_i^{(j)})$, the population mean for VIMP equals

$$\mathbb{E}\left[I(X^{(j)}, \Theta, \mathscr{L})\right] = \mathbb{E}\left[(h_j(\tilde{X}^{(j)}) - h_j(X^{(j)}))^2\right] + 2\sigma^2(1 - \rho_j) + o(1),$$

where ρ_j is a correlation coefficient and h_j is the additive expansion of h attributed to $X^{(j)}$. For noisy variables, $h_j = 0$ and $\rho_j = 1$; thus VIMP will converge to zero. For strong variables, $h_j \neq 0$. Our theory suggests that the value of ρ_j will be the same for all strong variables. Therefore for strong variables, except for some constant, VIMP equals the amount that h_j changes when $X^{(j)}$ is permuted, thus showing that VIMP correctly isolates the effect of $X^{(j)}$ in the model.

The technical assumptions required by Theorem 1 are provided in the Appendix, however there are two key conditions worth briefly mentioning. One is the use of deep trees in which terminal nodes contain exactly one unique value (replicated values due to bootstrapping are allowed). A second condition is that the forest predictor is L_2-consistent for h. As discussed in the Appendix, this latter assumption is reasonable in our setting and has been proven by Scornet et al.[35]

It is interesting that the above property for VIMP is tied to the consistency of the forest. We believe in general that properties for VIMP, such as its limiting distribution, will rely on analogous results for the RF predictor. Hopefully in the future these results for VIMP will be proven. At least in the case of RF-R we know that distributional results exist for the predictor. Wager[19] established asymptotic normality of the infinite forest predictor (assuming one observation per terminal node). Mentch and Hooker[20] established a similar result for the finite forest predictor. See Biau and Scornet[36] for a comprehensive discussion of known theoretical results for RF.

References

1. Breiman, L. Random forests. *Machine Learning*. 2001. 45:5–32.
2. Ishwaran, H., Kogalur, U. B., Blackstone, E. H., and Lauer, M. S. Random survival forests. *The Annals of Applied Statistics*. 2008. 2(3):841–860.
3. Ishwaran, H., Gerds, T. A., Kogalur, U. B., Moore, R. D., Gange, S. J., and Lau, B. M. Random survival forests for competing risks. *Biostatistics*. 2014. 15(4):757–773.
4. Hothorn, T., Hornik, K., and Zeileis, A. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*. 2006. 15(3):651–674.
5. Zhu, R. and Kosorok, M. R. Recursively imputed survival trees. *Journal of the American Statistical Association*. 2012. 107(497):331–340.
6. Breiman, L. Bagging predictors. *Machine Learning*. 1996. 24(2):123–140.
7. Ishwaran, H. Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*. 2007. 1:519–537.
8. Grömping, U. Variable importance assessment in regression: linear regression versus random forest. *The American Statistician*. 2009. 63(4):308–319.
9. Genuer, R., Poggi, J.-M., and Tuleau-Malot, C. Variable selection using random forests. *Pattern Recognition Letters*. 2010. 31(14):2225–2236.
10. Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. *Classification and Regression Trees*. CRC press. 1984.
11. Breiman, L. Manual on setting up, using, and understanding random forests v3.1. 2002.
12. Louppe, G., Wehenkel, L., Sutera, A., and Geurts, P. Understanding variable importances in forests of randomized trees. In *Advances in Neural Information Processing Systems*. pages 431–439. 2013.
13. Breiman, L. Out-of-bag estimation. 1996. Technical report, Statistics Department, University of California Berkeley, Berkeley CA 94708, 1996b. 33, 34.

14. Archer, K. J. and Kimes, R. V. Empirical characterization of random forest variable importance measures. *Computational Statistics and Data Analysis*. 2008. 52(4):2249–2260.
15. Nicodemus, K. K., Malley, J. D., Strobl, C., and Ziegler, A. The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC Bioinformatics*. 2010. 11(1):110.
16. Gregorutti, B., Michel, B., and Saint-Pierre, P. Correlation and variable importance in random forests. *Statistics and Computing*. 2017. 27(3):659–678.
17. Zhu, R., Zeng, D., and Kosorok, M. R. Reinforcement learning trees. *Journal of the American Statistical Association*. 2015. 110(512):1770–1784.
18. Sexton, J. and Laake, P. Standard errors for bagged and random forest estimators. *Computational Statistics & Data Analysis*. 2009. 53(3):801–811.
19. Wager, S., Hastie, T., and Efron, B. Confidence intervals for random forests: the jackknife and the infinitesimal jackknife. *Journal of Machine Learning Research*. 2014. 15(1):1625–1651.
20. Mentch, L. and Hooker, G. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *The Journal of Machine Learning Research*. 2016. 17(1):841–881.
21. Politis, D. N. and Romano, J. P. Large sample confidence regions based on subsamples under minimal assumptions. *The Annals of Statistics*. 1994. 22(4):2031–2050.
22. Shao, J. and Wu, C. J. A general theory for jackknife variance estimation. *The Annals of Statistics*. 1989. 17(3):1176–1197.
23. Graf, E., Schmoor, C., Sauerbrei, W., and Schumacher, M. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine*. 1999. 18(17-18):2529–2545.
24. Gerds, T. A. and Schumacher, M. Consistent estimation of the expected brier score in general survival models with right-censored event times. *Biometrical Journal*. 2006. 48(6):1029–1040.
25. Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*. 1963. 58(301):13–30.
26. Radulović, D. On the subsample bootstrap variance estimation. *Test*. 1998. 7(2):295–306.
27. Friedman, J. H. Multivariate adaptive regression splines. *The Annals of Statistics*. 1991. pages 1–67.
28. Biau, G., Fischer, A., Guedj, B., and Malley, J. D. COBRA: A combined regression strategy. *Journal of Multivariate Analysis*. 2016. 146:18–28.
29. Ishwaran, H. and Kogalur, U. B. Random Forests for Survival, Regression, and Classification (RF-SRC). https://cran.r-project.org/web/packages/randomForestSRC. 2017. R package version 2.5.0.
30. Leisch, F. and Dimitriadou, E. mlbench: Machine Learning Benchmark Problems. https://cran.r-project.org/web/packages/mlbench. 2010. R package version 2.1-1.
31. Kuhn, M. caret: Classification and Regression Training. https://cran.r-project.org/web/packages/caret. 2017. R package version 6.0-77.
32. Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L., and Rosati, R. A. Evaluating the yield of medical tests. *JAMA*. 1982. 247(18):2543–2546.
33. Hsich, E., Gorodeski, E. Z., Blackstone, E. H., Ishwaran, H., and Lauer, M. S. Identifying important risk factors for survival in patient with systolic heart failure using random survival forests. *Circulation: Cardiovascular Quality and Outcomes*. 2011. 4(1):39–45.
34. Breiman, L. Software for the masses. Lecture III, IMS Wald Lectures, Banff, Alberta, Canada. 2002.
35. Scornet, E., Biau, G., and Vert, J.-P. Consistency of random forests. *The Annals of Statistics*. 2015. 43(4):1716–1741.
36. Biau, G. and Scornet, E. A random forest guided tour. *Test*. 2016. 25(2):197–227.

8 | APPENDIX

8.1 | Assessing normality for classification, survival, and competing risk

We applied the same strategy as in RF-R simulations to assess normality of VIMP for RF-C, RSF, and RSF competing risk simulations. Specifically, for each setting we ran simulations 1000 times independently. Experimental parameters were set as before with $n = 2500$. The finite forest VIMP for a variable was centered by its averaged value from the 1000 simulations. This centered value was then divided by the standard deviation of the 1000 VIMP values. Quantile bias was calculated by taking the difference between the quantile for standardized VIMP to that of a standard normal quantile. Quantile bias is displayed in Figure 10 for the three families.

8.2 | Performance results from competing risk simulations

Competing risk simulations from Ishwaran et al.[3] were used to assess performance of the subsampling, delete-d jackknife, and .164 bootstrap estimators. Performance was measured using the time truncated concordanace index.[3] Analysis focused on variables affecting cause 1 event. RSF competing risk trees were constructed using log-rank splitting with weight 1 on event 1 and weight 0 on event 2. This ensured VIMP identified only those variables affecting event 1. Results are displayed in Figure 11.

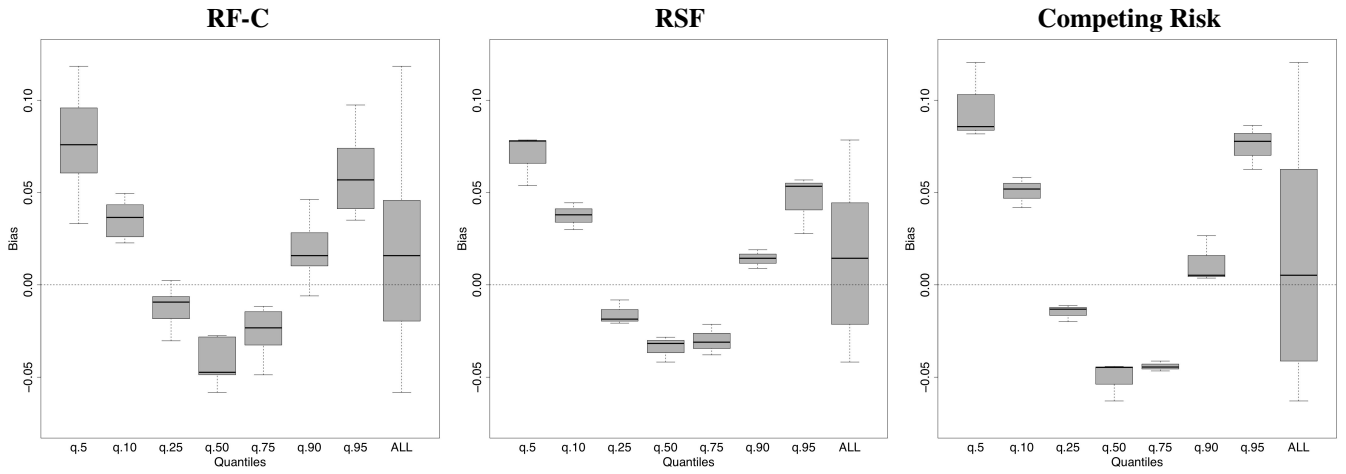

FIGURE 10: *Assessing asymptotic normality of VIMP from RF-C, RSF, and RSF competing risk simulations. Figure displays bias of standardized VIMP quantiles compared to standard normal quantiles. Values are displayed for 5,10,25,50,75,90,95 percentile values.*

8.3 | Some theoretical results for VIMP in RF-R

Let $\hat{\theta}_n^{(j)} = I(X^{(j)}, \mathscr{L})$ be the infinite forest estimator for VIMP (5). We assume the following additive regression model holds

$$Y_i = \sum_{j=1}^{p} h_j(X_i^{(j)}) + \varepsilon_i, \quad i = 1, \ldots, n, \tag{14}$$

where $(\mathbf{X}_i, \varepsilon_i)$ are i.i.d. with distribution $\mathbb{P}$ such that $\mathbf{X}_i$ and ε_i are independent and $\mathbb{E}(\varepsilon_i) = 0$, $\text{Var}(\varepsilon_i) = \sigma^2 < \infty$. Notice (14) implies that the target function h has an additive expansion $h(\mathbf{X}) = \sum_{j=1}^{p} h_j(X^{(j)})$. This is a useful assumption because it will allow us to isolate the effect of VIMP. Also, there are known consistency results for RF in additive models[35] which we will use later in establishing our results. Assuming squared error loss, $\ell(Y, \hat{h}) = (Y - \hat{h})^2$, we have

$$\hat{\theta}_n^{(j)} = \mathbb{E}_\Theta \left[\frac{1}{N(\Theta)} \sum_{i \in \mathscr{L}^{**}(\Theta)} \left[(Y_i - h(\tilde{\mathbf{X}}_i^{(j)}, \Theta, \mathscr{L}))^2 - (Y_i - h(\mathbf{X}_i, \Theta, \mathscr{L}))^2 \right] \right].$$

We will assume the number of OOB cases $N(\Theta)$ is always fixed at $\text{Round}(n e^{-1})$, where $\text{Round}(\cdot)$ is the nearest integer function. Write N_n for $N(\Theta)$. Because this is a fixed value,

$$\hat{\theta}_n^{(j)} = \frac{1}{N_n} \mathbb{E}_\Theta \left[\sum_{i \in \mathscr{L}^{**}(\Theta)} \left[(Y_i - h(\tilde{\mathbf{X}}_i^{(j)}, \Theta, \mathscr{L}))^2 - (Y_i - h(\mathbf{X}_i, \Theta, \mathscr{L}))^2 \right] \right].$$

To study $\hat{\theta}_n^{(j)}$ we will evaluate its mean $\theta_{n,0}^{(j)} = \mathbb{E}_{\mathscr{L}}[\hat{\theta}_n^{(j)}]$. For ease of notation, write $h_{n,i} = h(\mathbf{X}_i, \Theta, \mathscr{L})$ and $\tilde{h}_{n,i} = h(\tilde{\mathbf{X}}_i^{(j)}, \Theta, \mathscr{L})$. Likewise, let $h_n = h(\mathbf{X}, \Theta, \mathscr{L})$ and $\tilde{h}_n = h(\tilde{\mathbf{X}}^{(j)}, \Theta, \mathscr{L})$. We have

$$\theta_{n,0}^{(j)} = \frac{1}{N_n} \mathbb{E} \left[\sum_{i \in \mathscr{L}^{**}(\Theta)} \left[(Y_i - \tilde{h}_{n,i})^2 - (Y_i - h_{n,i})^2 \right] \right] = \mathbb{E}\left[(Y - \tilde{h}_n)^2 - (Y - h_n)^2 \right],$$

where the right hand side follows because $(\mathbf{X}, Y, h_n, \tilde{h}_n) \stackrel{\text{d}}{=} (\mathbf{X}_i, Y_i, h_{n,i}, \tilde{h}_{n,i})$ if i is OOB (i.e., because the tree does not use information about $(\mathbf{X}_i, Y_i)$ in its construction, we can replace $(\mathbf{X}_i, Y_i)$ with $(\mathbf{X}, Y)$). Now making using of the representation $Y = h(\mathbf{X}) + \varepsilon$, which holds by the assumed regression model (14), and writing h for $h(\mathbf{X})$ and $\tilde{\Delta}_n = \tilde{h}_n - h_n$,

$$\theta_{n,0}^{(j)} = \mathbb{E}\left[(Y - \tilde{h}_n)^2 - (Y - h_n)^2 \right] = \mathbb{E}\left[-2\varepsilon\tilde{\Delta}_n + \tilde{\Delta}_n^2 + 2\tilde{\Delta}_n(h_n - h) \right] = \mathbb{E}\left[\tilde{\Delta}_n^2 \right] + 2\mathbb{E}\left[\tilde{\Delta}_n(h_n - h) \right], \tag{15}$$

where in the last line we have used $\mathbb{E}(\varepsilon) = 0$ and that ε is independent of $\{\tilde{\Delta}_n, h_n, h\}$.

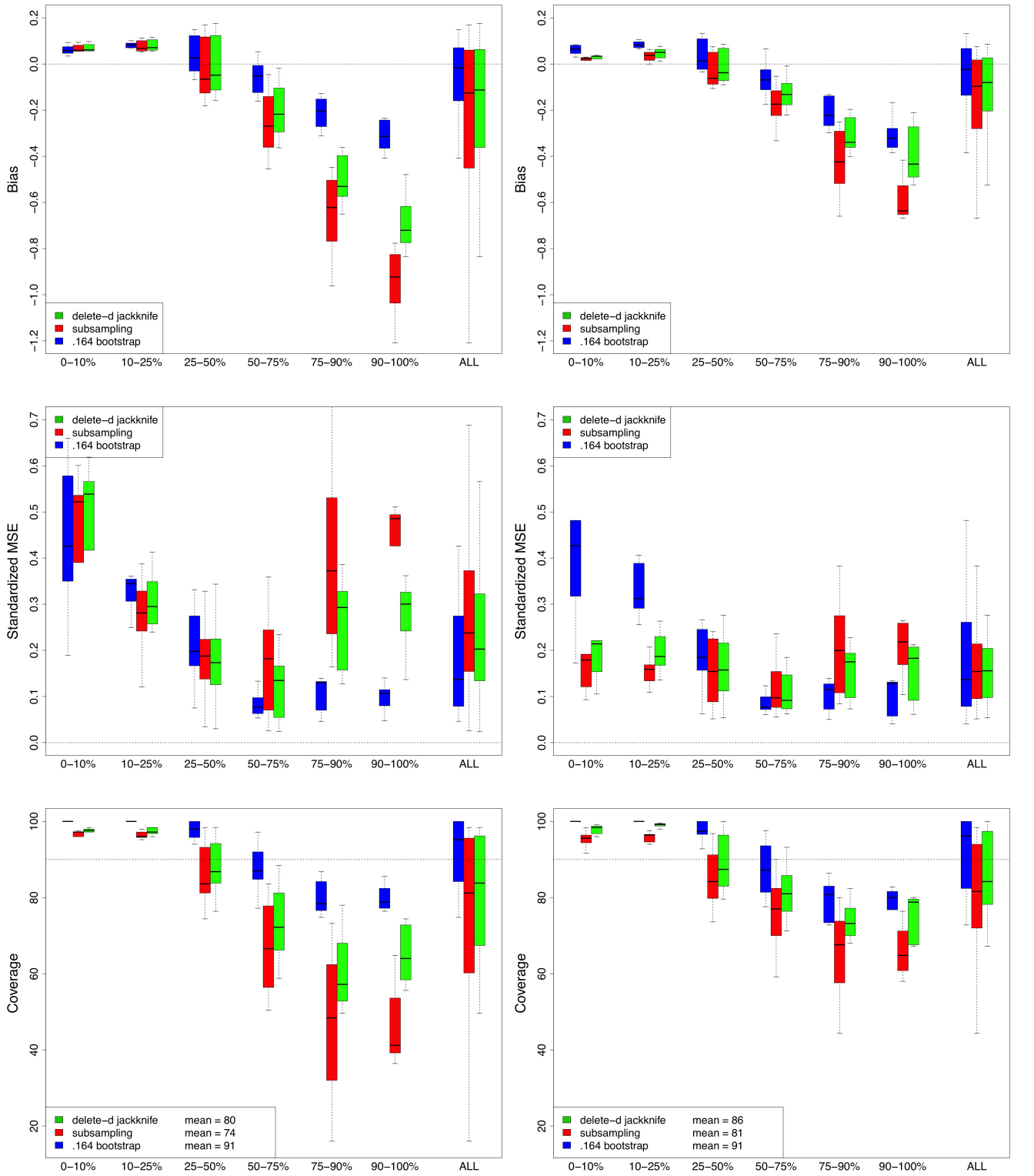

FIGURE 11: *Results from competing risk simulations showing performance of subsampling, delete-d jackknife, and .164 bootstrap. Left and right hand side figures are based on subsampling rates $b = n^{1/2}$ and $b = n^{3/4}$. Top and middle figures display bias and standardized MSE for estimating VIMP standard error. Bottom figure displays coverage for VIMP 90% asymptotic normal confidence intervals. Results have been stratified into 6 groups based on 10, 25, 50, 75, and 90th percentiles of true finite VIMP.*

We can see that (15) is driven by the two terms: $\tilde{\Delta}_n$ and $h_n - h$. Define integer values $n_i := n_i(\Theta) \geq 0$ recording the bootstrap frequency of case $i = 1, \ldots, n$ in $\mathscr{L}^*(\Theta)$ (notice that $n_i = 0$ implies case i is OOB). By the definition of a RF-R tree, we have

$$h_n = h(\mathbf{X}, \Theta, \mathscr{L}) = \sum_{i=1}^{n} W_i(\mathbf{X}, \Theta) Y_i$$

where $\{W_i(\mathbf{X}, \Theta)\}_1^n$ are the forest weights defined as

$$W_i(\mathbf{X}, \Theta) = \frac{n_i 1\{\mathbf{X}_i \in R(\mathbf{X}, \Theta)\}}{|R(\mathbf{X}, \Theta)|}$$

where $R(\mathbf{X}, \Theta)$ is the tree terminal node containing $\mathbf{X}$ and $|R(\mathbf{X}, \Theta)|$ is the cardinality equal to the number of bootstrap cases in $R(\mathbf{X}, \Theta)$. Notice that the weights are convex since $0 \leq W_i(\mathbf{X}, \Theta) \leq 1$ and

$$\sum_{i=1}^{n} W_i(\mathbf{X}, \Theta) = \sum_{i=1}^{n} \frac{n_i 1\{\mathbf{X}_i \in R(\mathbf{X}, \Theta)\}}{|R(\mathbf{X}, \Theta)|} = \sum_{i=1}^{n} \frac{n_i 1\{\mathbf{X}_i \in R(\mathbf{X}, \Theta)\}}{\sum_{i'=1}^{n} n_{i'} 1\{\mathbf{X}_{i'} \in R(\mathbf{X}, \Theta)\}} = 1.$$

Similarly, we have

$$\tilde{h}_n = h(\tilde{\mathbf{X}}^{(j)}, \Theta, \mathscr{L}) = \sum_{i=1}^{n} W_i(\tilde{\mathbf{X}}^{(j)}, \Theta) Y_i, \quad \text{where } W_i(\tilde{\mathbf{X}}^{(j)}, \Theta) = \frac{n_i 1\{\mathbf{X}_i \in R(\tilde{\mathbf{X}}^{(j)}, \Theta)\}}{|R(\tilde{\mathbf{X}}^{(j)}, \Theta)|}.$$

Therefore,

$$\tilde{\Delta}_n(\mathbf{X}) = h(\tilde{\mathbf{X}}^{(j)}, \Theta, \mathscr{L}) - h(\mathbf{X}, \Theta, \mathscr{L}) = \sum_{i=1}^{n} W_i(\tilde{\mathbf{X}}^{(j)}, \Theta) Y_i - \sum_{i=1}^{n} W_i(\mathbf{X}, \Theta) Y_i.$$

In order to study $\tilde{\Delta}_n$ in more detail we will assume deep trees containing one unique case per terminal node.

Assumption 1. *We assume each terminal node contains exactly one unique value. That is, each terminal node contains the bootstrap copies of a unique data point.*

Assumption 1 results in the following useful simplification. For notational ease, write $\tilde{R} = R(\tilde{\mathbf{X}}^{(j)}, \Theta)$ and $R = R(\mathbf{X}, \Theta)$. Then

$$\begin{aligned}
\tilde{\Delta}_n(\mathbf{X}) &= \frac{1}{|\tilde{R}|} \sum_{i \in \tilde{R}} n_i Y_i - \frac{1}{|R|} \sum_{i \in R} n_i Y_i \\
&= \frac{1}{|\tilde{R}|} \sum_{i \in \tilde{R}} n_i h(\mathbf{X}_i) - \frac{1}{|R|} \sum_{i \in R} n_i h(\mathbf{X}_i) + \frac{1}{|\tilde{R}|} \sum_{i \in \tilde{R}} n_i \varepsilon_i - \frac{1}{|R|} \sum_{i \in R} n_i \varepsilon_i \\
&= h(\mathbf{X}_{i(\tilde{R})}) - h(\mathbf{X}_{i(R)}) + \varepsilon_{i(\tilde{R})} - \varepsilon_{i(R)},
\end{aligned}$$

where $i(\tilde{R})$ and $i(R)$ identify the index for the bootstrap case in $\tilde{R} = R(\tilde{\mathbf{X}}^{(j)}, \Theta)$ and $R = R(\mathbf{X}, \Theta)$ respectively (note that $i(\tilde{R})$ and $i(R)$ are functions of $\mathbf{X}$ and j but this is suppressed for notational simplicity). We can see that the information in the target function h is captured by the first two terms in the last line and therefore will be crucial to understanding VIMP. Notice that if $h(\mathbf{X}_{i(\tilde{R})}) \asymp h(\tilde{\mathbf{X}}^{(j)})$ and $h(\mathbf{X}_{i(R)}) \asymp h(\mathbf{X})$, which is what we would expect asymptotically with a deep tree, then

$$h(\mathbf{X}_{i(\tilde{R})}) - h(\mathbf{X}_{i(R)}) \asymp h(\tilde{\mathbf{X}}^{(j)}) - h(\mathbf{X}) = h_j(\tilde{X}^{(j)}) - h_j(X^{(j)}),$$

where the right-hand side follows by our assumption of an additive model (14). This shows that VIMP for $X^{(j)}$ is assessed by how much its contribution to the additive expansion, h_j, changes when $X^{(j)}$ is permuted. This motivates the following assumption.

Assumption 2. *Assuming a deep tree with one unique value in a terminal node,*

$$h(\tilde{\mathbf{X}}^{(j)}) = h(\mathbf{X}_{i(\tilde{R})}) + \tilde{\zeta}_n(\mathbf{X}), \qquad h(\mathbf{X}) = h(\mathbf{X}_{i(R)}) + \zeta_n(\mathbf{X}),$$

where $\mathbb{E}(\tilde{\zeta}_n^2) = o(1)$ *and* $\mathbb{E}(\zeta_n^2) = o(1)$.

This deals with the first two terms in the expansion of $\tilde{\Delta}_n(\mathbf{X})$. We also need to deal with the remaining term involving the measurement errors, $\varepsilon_{i(\tilde{R})} - \varepsilon_{i(R)}$. For this we will rely on a fairly mild exchangeability assumption.

Assumption 3. $\varepsilon_{i(\tilde{R})}, \varepsilon_{i(R)}$ *is a finite exchangeable sequence with variance* σ^2.

Finally, a further assumption we will need is consistency of the forest predictor.

Assumption 4. *The forest predictor is* L_2*-consistent,* $\mathbb{E}\left[(h_n - h)^2\right] \to 0$ *where* $\mathbb{E}[h^2] < \infty$.

Putting all of the above together, we can now state our main result.

Theorem 1. *If Assumptions 1, 2, 3, and 4 hold, then*

$$\theta_{n,0}^{(j)} = \mathbb{E}\left[(h_j(\tilde{X}^{(j)}) - h_j(X^{(j)}))^2\right] + 2\sigma^2(1-\rho_j) + o(1),$$

where $\rho_j = corr(\varepsilon_{i(\tilde{R})}, \varepsilon_{i(R)})$.

Note that the asymptotic limit will be heavily dependent on the strength of the variable. Consider when $X^{(j)}$ is a noisy variable. Ideally this means the tree is split without ever using $X^{(j)}$. Therefore, if we drop $\mathbf{X}$ and $\tilde{\mathbf{X}}^{(j)}$ down the tree they will occupy the same terminal node. Hence, $\tilde{R} = R$ and $\varepsilon_{i(\tilde{R})} = \varepsilon_{i(R)}$ and therefore $\rho_j = 1$. Furthermore, because h_j must be zero for a noisy variable, it follows that $\theta_{n,0}^{(j)} = o(1)$. Thus, the limit is zero for a noisy variable. Obviously, this is much different than the limit of a strong variable which must be strictly positive because $h_j \neq 0$ and $\rho_j < 1$ for strong variables.

Proof. By (15), we have $\theta_{n,0}^{(j)} = \mathbb{E}[\tilde{\Delta}_n^2] + 2\mathbb{E}[\tilde{\Delta}_n(h_n - h)]$. We start by dealing with the second term, $\mathbb{E}[\tilde{\Delta}_n(h_n - h)]$. By the Cauchy-Schwartz inequality,

$$\mathbb{E}\left[\tilde{\Delta}_n(h_n - h)\right] \leq \mathbb{E}\left[|\tilde{\Delta}_n||h_n - h|\right] \leq \sqrt{\mathbb{E}\left[\tilde{\Delta}_n^2\right]}\sqrt{\mathbb{E}\left[(h_n - h)^2\right]}.$$

By Assumption 4, the right-hand side converges to zero if $\mathbb{E}[\tilde{\Delta}_n^2]$ remains bounded. By Assumption 2, and the assumption of an additive model (14),

$$\tilde{\Delta}_n(\mathbf{X}) = h_j(\tilde{X}^{(j)}) - h_j(X^{(j)}) - \tilde{\zeta}_n(\mathbf{X}) + \zeta_n(\mathbf{X}) + \varepsilon_{i(\tilde{R})} - \varepsilon_{i(R)}.$$

Assumption 4 implies h (and therefore h_j) is square-integrable. Assumption 3 implies that $\varepsilon_{i(\tilde{R})}, \varepsilon_{i(R)}$ have finite second moment and are square-integrable. Therefore squaring and taking expectations, and using $\mathbb{E}(\tilde{\zeta}_n^2) = o(1)$ and $\mathbb{E}(\zeta_n^2) = o(1)$, deduce that

$$\mathbb{E}[\tilde{\Delta}_n(\mathbf{X})^2] = \mathbb{E}\left[(h_j(\tilde{X}^{(j)}) - h_j(X^{(j)}))^2\right] + 2\mathbb{E}\left[(h_j(\tilde{X}^{(j)}) - h_j(X^{(j)}))(\varepsilon_{i(\tilde{R})} - \varepsilon_{i(R)})\right] + \mathbb{E}\left[(\varepsilon_{i(\tilde{R})} - \varepsilon_{i(R)})^2\right] + o(1).$$

By exchangeability, $\mathbb{E}[g(\mathbf{X})\varepsilon_{i(\tilde{R})}] = \mathbb{E}[g(\mathbf{X})\varepsilon_{i(R)}]$ for any function $g(\mathbf{X})$. Hence,

$$0 = \mathbb{E}\left[(h_j(\tilde{X}^{(j)}) - h_j(X^{(j)}))\varepsilon_{i(\tilde{R})}\right] - \mathbb{E}\left[(h_j(\tilde{X}^{(j)}) - h_j(X^{(j)}))\varepsilon_{i(R)}\right] = \mathbb{E}\left[(h_j(\tilde{X}^{(j)}) - h_j(X^{(j)}))(\varepsilon_{i(\tilde{R})} - \varepsilon_{i(R)})\right].$$

Appealing to exchangeability once more, we have

$$\mathbb{E}[(\varepsilon_{i(\tilde{R})} - \varepsilon_{i(R)})^2] = \mathbb{E}[\varepsilon_{i(\tilde{R})}^2] + \mathbb{E}[\varepsilon_{i(R)}^2] - 2\mathbb{E}[\varepsilon_{i(\tilde{R})}\varepsilon_{i(R)}] = 2\sigma^2(1-\rho_j),$$

where $\rho_j = \text{corr}(\varepsilon_{i(\tilde{R})}, \varepsilon_{i(R)})$. Therefore we have shown

$$\mathbb{E}[\tilde{\Delta}_n(\mathbf{X})^2] = \mathbb{E}\left[(h_j(\tilde{X}^{(j)}) - h_j(X^{(j)}))^2\right] + 2\sigma^2(1-\rho_j) + o(1),$$

which verifies boundedness of $\mathbb{E}[\tilde{\Delta}_n^2]$ and that $\theta_{n,0}^{(j)} = \mathbb{E}[\tilde{\Delta}_n^2] + o(1)$. □

The conditions needed to establish Theorem 1 are fairly reasonable. Assumption 2 can be viewed as a type of continuity condition for h. However, it is also an assertion about the approximating behavior of the forest predictor h_n. It asserts that all features $\mathbf{X}$ within a terminal node have $h(\mathbf{X})$ values close to one another, which can be seen as an indirect way of asserting good local prediction behavior for the tree. Thus, Assumption 2 is very similar to Assumption 4. The latter assumption of consistency is reasonable for deep trees under an additive model assumption. Scornet et al.[35] established L_2-consistency of RF-R for additive models allowing for the number of terminal nodes to grow at rate of n (see Theorem 2 of their paper). For technical reasons their proof replaced bootstrapping with subsampling and required $\mathbf{X}$ to be uniformly distributed, but other than this their result can be seen as strong support for our assumptions.