The Hedged Random Forest

Elliot Beck*
Department of Banking and Finance
University of Zurich
8032 Zurich, Switzerland
elliot.beck@bf.uzh.ch

Damian Kozbur
Department of Economics
University of Zurich
8001 Zurich, Switzerland
damian.kozbur@econ.uzh.ch

Michael Wolf†
Department of Economics
University of Zurich
8032 Zurich, Switzerland
michael.wolf@econ.uzh.ch

Abstract

The random forest is one of the most popular and widely employed tools for supervised machine learning. It can be used for both classification and regression tasks; in this paper, the focus will be on regression only. In its standard form, the crux of the random forest is to use an equal-weighted ensemble of tree-based predictors. Instead, we suggest a more general weighting scheme that borrows certain ideas from the related problem of financial portfolio selection and, in particular, allows for negative weights. Based on a benchmark collection of real-life data sets, we demonstrate the improved predictive performance of our method not only relative to the standard random forest but also relative to two previous proposals for (non-equal) weighting the tree-based predictors. It is noteworthy that our methodology is of a high-level nature and can also be applied to other forecast-combination problems, when forecast methods are of arbitrary nature and not necessarily tree-based.

KEY WORDS: Forecast combinations, machine learning, nonlinear shrinkage.

JEL classification codes: C21, C53.

*Second affiliation: Swiss National Bank, 8001 Zurich, Switzerland.
†Corresponding author.

1 Introduction

The random forest (Breiman, 2001) is one of the most popular tree-based methods for supervised machine learning, given its ease of use, simplicity, and prediction quality across a wide variety of applications. Grinsztajn et al. (2022) demonstrate that tree-based methods, particularly the random forest and ensembles enhanced by boosting, continue to excel as top-performing methods on medium-sized tabular data sets. Depending on the nature of the response variable, the random forest can be used for either *classification* or *regression*. If the response variable is categorical, the random forest is used for *classification*; if the variable is numerical, the random forest is used for *regression*. In this paper, the focus will be on regression only.

In the standard implementation of the random forest, one first grows an ensemble of (de-correlated) trees and then uses the simple average of the trees (also called the equal-weighted ensemble). This means the individual forecasts of the trees are simply averaged to arrive at a final, combined forecast. For a textbook treatment on the random forest the reader is referred to Hastie et al. (2017, Chapter 15), for example.

The goal of this paper is to deviate from equal weighting in a data-dependent fashion that outperforms, on balance, the standard random forest. One can consider this task to be a special case of the well-studied, general problem of *forecast combinations*. In this general problem, one has a fixed number of forecasting (or prediction) methods available and then assigns individual weights, which sum up to one, to them to obtain a combined forecast (or prediction). There exists an extensive literature on forecast combinations; for example, see Elliott and Timmermann (2016, Chapter 14), Wang et al. (2022), and the references therein.

We will propose a high-level methodology for the generic forecast-combination problem and then study in detail its use and its performance for the specific application to combining tree-based forecasts in the context of the random forest. It will be seen that the high-level problem formulation is related to the well-studied problem of portfolio selection from

finance. This insight allows us to borrow certain ideas from that strand of literature. For reasons that will become apparent below, we call the high-level solution *hedging forecast combinations* and the solution applied to combining tree-based forecasts *the hedged random forest*.

The remainder of the paper is organized as follows. Section 2 presents the general, high-level methodology which can be applied to a generic forecast-combination problem. Section 3 specializes to tree-based forecasting methods in the context of the random forest, giving specific guidance to implementation details. Section 4 provides an empirical application to real data, comparing the predictive performance of our method to the standard random forest as well as to two previous proposals for (non-equal) weighting the individual trees. An appendix contains various robustness checks.

2 Methodology

2.1 General Description

In this section we will detail our high-level methodology which can be applied to a generic forecast-combination problem. In the subsequent section we will then specialize to the problem of combining tree-based forecasts as an alternative the standard random forest, which simply gives equal weight to all trees.

The goal is to forecast (or predict)[1] a random variable $y \in \mathbb{R}$ based on a set of variables (or attributes) $x \in \mathbb{R}^d$. Denote a generic forecast by $\hat{f}$. Then its mean-squared error (MSE) is given by

$$\mathrm{MSE}(\hat{f}) \coloneqq \mathbb{E}\big(y - \hat{f}(x)\big)^2 \ .$$

Here and below the moments are, of course, obtained under the joint distribution governing

[1] In this paper, the terms "forecast" and "prediction" are used interchangeably. Arguably, some people associate a time-series setting with the term "forecast" but we do not.

the random vector $v \coloneqq (y, x')' \in \mathbb{R}^{1+d}$ and assumed to exist. Letting

$$\text{Bias}(\hat{f}) \coloneqq \mathbb{E}\big(y - \hat{f}(x)\big) \quad \text{and} \quad \text{Var}(\hat{f}) \coloneqq \mathbb{V}\text{ar}\big(y - \hat{f}(x)\big) = \mathbb{E}\big((y - \hat{f}(x)\big)^2 - \text{Bias}^2(\hat{f}) \ .$$

there exists the well-known decomposition

$$\text{MSE}(\hat{f}) = \text{Bias}^2(\hat{f}) + \text{Var}(\hat{f}) \ . \tag{2.1}$$

The oracle minimizing the MSE is given by the conditional expectation $\hat{f}_{\text{or}}(x) \coloneqq \mathbb{E}(y|x)$; for example, see Hayashi (2000, Section 2.9). However, the oracle being an oracle, it is not available in practice.

We assume that a set (or universe) of p forecast methods is available, denoted by $\{\mathcal{M}_j\}_{j=1}^p$. The number of methods, p, is assumed to be exogenous and fixed. Although we do not make this explicit in the notation, methods may be data-dependent in the sense that certain parameters may be fitted based on observed data; for example, the methods may correspond to linear regression models where each models postulates the identity (and the number) of regressors but not the values of the corresponding coefficients.

There exists an extensive literature on forecast combinations; for example, see Elliott and Timmermann (2016, Chapter 14), Wang et al. (2022), and the references therein. The consensus seems to be that equal weighting, given by

$$\hat{f}_{\text{EW}}(x) \coloneqq \frac{1}{p} \sum_{j=1}^{p} \mathcal{M}_j(x) \ ,$$

is hard to beat by more general linear combinations of the kind

$$\hat{f}_w(x) \coloneqq \sum_{j=1}^{p} w_j \mathcal{M}_j(x) \quad \text{with} \quad w \coloneqq (w_1, \ldots, w_p)' \quad \text{and} \quad \sum_{j=1}^{p} w_j = 1 \ . \tag{2.2}$$

Nevertheless, our aim is to find a method for selecting a set of weights w that does improve the (out-of-sample) MSE of equal weighting, on balance.

Denote by $e_j \coloneqq y - \mathcal{M}_j(x)$ the forecast error made by model $\mathcal{M}_j$ and collect these errors into the vector $e \coloneqq (e_1, \ldots, e_p)'$ with expectation (vector) and covariance matrix

$$\mu \coloneqq \mathbb{E}(e) \quad \text{and} \quad \Sigma \coloneqq \mathbb{V}\text{ar}(e) \ .$$

The MSE of the forecast (2.2) is then given by

$$\text{MSE}(\hat{f}_w) = (w'\mu)^2 + w'\Sigma w \ .$$

Therefore, the optimal (in terms of the MSE) forecast in the class (2.2) is the solution to the following optimization problem:

$$\min_w \ (w'\mu)^2 + w'\Sigma w \tag{2.3}$$

$$\text{s.t.} \quad w'\mathbb{1} = 1 \ , \tag{2.4}$$

where $\mathbb{1}$ denotes a conformable vector of ones. Problem (2.3)–(2.4) is a convex optimization problem and can be solved easily and quickly with readily available software, even for large dimensions p.

The problem in practice is that the inputs μ and Σ are unknown. A feasible solution is to replace them with sample-based estimates $\hat{\mu}$ and $\hat{\Sigma}$, which is an application of the general "plug-in method".

Being agnostic, for the time being, about the nature of the estimators $\hat{\mu}$ and $\hat{\Sigma}$, we then solve the feasible optimization problem

$$\min_w \ (w'\hat{\mu})^2 + w'\hat{\Sigma} w \tag{2.5}$$

$$\text{s.t.} \quad w'\mathbb{1} = 1 \quad \text{and} \tag{2.6}$$

$$||w||_1 \leq \kappa \ , \tag{2.7}$$

where $||w||_1 \coloneqq \sum_{j=1}^{p} |w_j|$ denotes the L_1 norm of w and $\kappa \in [1, \infty]$ is a constant chosen by

the user. Assuming succinctly that the estimator $\hat{\Sigma}$ is symmetric and positive semi-definite, the optimization problem (2.5)–(2.7) is still of convex nature and can be solved easily and quickly, even for large dimensions p. We shall denote the solution to this optimization problem by $\hat{w}$.

The addition of the constraint (2.7) is motivated by the related problem of *portfolio selection* in finance, in which context the constraint is called a "gross-exposure constraint". Adding this type of constraint to the infeasible problem (2.3)–(2.4) clearly would result in a (weakly) worse solution for any value $\kappa \in [1, \infty)$. But in the feasible problem, which must use estimated instead of true inputs, the constraint typically helps. The intuition here is that replacing μ and Σ with respective estimates $\hat{\mu}$ and $\hat{\Sigma}$ can lead to unstable and underdiversified solutions that look good in sample (or in the training set) but perform badly out of sample, especially when the number of models, p, is not (exceedingly) small relative to the sample size relevant to the estimation of μ and Σ; for example, see Jobson and Korkie (1980), Michaud (1989), Jagannathan and Ma (2003), In applications where the number of forecast methods, p, is not very small relative to the sample size of training data, n, the addition of the gross-exposure constraint is particularly helpful when $\hat{\Sigma}$ is given by the sample covariance matrix, an estimator that is unbiased but contains too much estimation error (unless $p \ll n$).

In the extreme case $\kappa = 1$, the weights are forced to be non-negative, that is, $w_j \geq 0$, which is called a "no-short-sales constraint" in finance. Imposing this constraint is standard in the forecast-combination literature but it might well lead to sup-optimal performance because of not giving enough flexibility to the solution of the problem (2.5)–(2.7). At the other end of the spectrum, the choice $\kappa = \infty$ corresponds to effectively removing the constraint (2.7), which may also lead to sub-optimal performance for the reasons mentioned above. Staying away from either extreme, there is ample evidence in the finance literature that choosing $\kappa \in [1.5, 2.5]$ typically results in improved forecasting performance, and that the exact choice in this interval is not overly critical; for example, see DeMiguel et al. (2009).

Because the constraint (2.7) protects the user against extreme "positions", that is, against weights $\hat{w}_j$ that are unduly large in absolute value, we call our approach "hedging forecast combinations".[2]

2.2 Theory

The solution to the convex optimization problem (2.5)–(2.7) is continuous in the inputs $\hat{\mu}$ and $\hat{\Sigma}$. Therefore, with the choice $\kappa := \infty$, its solution $\hat{w}$ would lead to an asymptotically optimal forecast combination $\hat{f}_{\hat{w}}$ based on consistent estimators $\hat{\mu}$ and $\hat{\Sigma}$. Stating this fact as formal result is possible, but this is a routine matter and also has been recognized before. Furthermore, in practical application, the relevant property is the finite-sample performance of the forecast $\hat{f}_{\hat{w}}$ and, so far, the evidence based on simulation studies and empirical applications to real-life data sets indicates that such forecast combinations, on balance, do not outperform $\hat{f}_{\mathrm{EW}}$, that is, equal weighting; again, see Elliott and Timmermann (2016, Chapter 14), Wang et al. (2022), and the references therein.

Our preceding high-level description is agnostic about the estimation of the mean (vector) μ and the covariance matrix Σ of the corresponding vector of forecast errors $e \in \mathbb{R}^p$. In practice, we assume the existence of a training data set $\{v_i\}_{i=1}^n$ with $v_i := (y_i, x_i')'$. One first constructs a set of (artificial) forecast errors which are then used as inputs for the estimation of μ and Σ. How one goes about this 'best' in detail depends on both the nature of the data and the nature of the individual forecasting methods, so at this stage we need to be necessarily somewhat vague.

When $\{v_i\}_{i=1}^n$ is an independent and identically distributed (i.i.d.) sample where the distribution of v_i is equal to the distribution of v. In this case there exists a well-established literature on how to generate pseudo-out-of-sample forecast errors, with the most popular technique being cross-validation; for example, see Efron and Hastie (2022, Chapter 12) and Hastie et al. (2017, Chapter 7). Another option is to use in-sample errors, or residuals;

[2]For example, according to Merriam-Webster (online version) the verb "to hedge against" means "to protect oneself from (something)".

this option has a bad reputation because in-sample errors tend to have a smaller variance than (actual) out-of-sample errors because of the well-known phenomenon of "overfitting". However, for our purposes this need not be a serious problem; see Remark 2.1 below. Whether one computes pseudo-out-of-sample or in-sample errors, they form the basis on which one estimates μ and Σ. To this end, one can use sample counterparts (that is, sample mean and sample covariance matrix), shrinkage methods, penalized estimation schemes, etc; Having said this, we shall restrict attention to shrinkage methods as an alternative to sample counterparts. For shrinkage estimation of a mean vector, see Hansen (2016), Bodnar et al. (2019), and the references therein; for shrinkage estimation of a covariance matrix, see Ledoit and Wolf (2022a) and the references therein.

Note that other data settings are also possible, for instance when $\{v_1, \ldots, v_n, v\}$ is a stationary time series. Also in this case the recommendation is to base the estimation of μ and Σ on pseudo-out-of-sample or in-sample forecast errors. How to generate those is less well established compared to the i.i.d. case but proposals do exist; for example, see Bergmeir and Benítez (2012) and Bergmeir et al. (2018). If one prefers shrinkage estimation over sample counterparts, ideally, one should use methods designed for time-series data; for example, see Sancetta (2008) and Engle et al. (2019) for shrinkage estimation of the covariance matrix. Having said this, the case of stationary time series is, arguably, more difficult in practice. Compared to an i.i.d. sample it would take a larger sample size, generally, to have a similar chance of outperforming $\hat{f}_{\mathrm{EW}}$, that is, equal weighting; but especially macroeconomic time series only have relatively small sample sizes. Furthermore, many real-life time series (even after detrending and deseasonalizing) may still not be stationary because of structural breaks, for example.

Remark 2.1 (Scale invariance)**.** The solution $\hat{w}$ to the optimization problem (2.5)–(2.7) remains unchanged if $\hat{\mu}$ and $\hat{\Sigma}$ are replaced by $c\hat{\mu}$ and $c^2\hat{\Sigma}$, respectively, for any constant $c \in (0, \infty)$. Therefore, it is not important that the estimators $\hat{\mu}$ and $\hat{\Sigma}$ get the 'levels' of the true quantities μ and Σ right. In particular, the use of in-sample (or training-set) errors in the construction of $\hat{\mu}$ and $\hat{\Sigma}$ can still lead to favorable performance of the forecast

combination $\hat{f}_{\hat{w}}$ even if such errors may have smaller variance than out-of-sample errors because of in-sample (or training-set) overfitting. Instead of approximating the actual entries of μ and Σ, the corresponding estimators $\hat{\mu}$ and $\hat{\Sigma}$ only need to approximate the entries relative to each other in order for $\hat{f}_{\hat{w}}$ to outperform $\hat{f}_{\text{EW}}$. That may still not be a trivial task, but it is certainly an easier task. ■

3 Hedging the Random Forest

Given the high-level description in the previous section, we only have to detail (i) how we compute forecast errors from training data and (ii) how we estimate μ and Σ in the context of tree-based forecasting methods $\mathcal{M}_j$.

We assume the existence of a training data set $\{v_i\}_{i=1}^n$ with $v_i := (y_i, x_i')'$ and only consider the case when the $\{v_i\}$ are i.i.d. We train a random forest consisting of p trees on this data set using the "ranger library" implemented in the programming language R, where the various hyperparameter are set to the defaults recommended by Wright and Ziegler (2017) for regression problems. After training the random forest, we extract the forecasts of each tree $\mathcal{M}_j$ on the entire training set and thus obtain an in-sample error (or residual) matrix of size $n \times p$. We denote this matrix by R to indicate that it is not made up of 'true' forecast errors but of in-sample forecast errors (or residuals). It is important to point out that we do not, for a given tree, extract forecasts on the corresponding out-of-bag observations only (that is, on the subset of the training set not used in pruning the particular tree) because in this way we would not obtain a full $n \times p$ matrix of residuals but instead a matrix that would contain a large number of missing values.[3] But as explained below, we need a full matrix R for the estimation of Σ, if not for the estimation of μ necessarily.

The various inputs to the feasible optimization problem (2.5)–(2.7) are chosen as follows. First, for the estimation of μ, we use the (column-wise) sample mean of R;

[3] On average, there would be about $1 - 1/e \approx 63.2\%$ of missing values

we also experimented with some shrinkage estimators instead but the results remained virtually unchanged. Second, for the estimation of Σ, we apply nonlinear shrinkage to R; in particular, we use the quadratic inverse shrinkage (QIS) estimator of Ledoit and Wolf (2022b). Note here that nonlinear shrinkage requires a full matrix R as an input; a matrix with missing values would not be feasible. Third, for the gross-exposure constraint κ in (2.7), we use $\kappa := 2$. Appendix A runs robustness checks that consider (i) the sample covariance matrix rather than nonlinear shrinkage as the estimator $\hat{\Sigma}$ and (ii) alternative values of the gross-exposure constraint κ.

Thus all inputs to the optimization problem (2.5)–(2.7) are now in place. The solution $\hat{w}$ assigns weight $\hat{w}_j$ to tree $\mathcal{M}_j$ rather than weight $1/p$ as for the standard random forest (RF). We call this weighted random forest the "hedged random forest" (HRF).

Remark 3.1 (Absence of cross-validation)**.** At this point, some readers may wonder why we do not use cross-validation to build up a $n \times p$ matrix of pseudo-out-of-sample errors. The reason lies in the special nature of the random forest, since the individual forecast methods, namely the trees, depend on the underlying training set. If we used ten-fold cross-validation, say, we would obtain ten different tree ensembles, none of which would coincide with the ensemble used at the end for forecasting, namely the ensemble based on the entire training set. This problem would not arise with other forecast methods, such as linear regression models, say; of course, estimated parameters in a given regression model would change as a function of the underlying training set, but not the 'characteristics' of that model, such as the identity (and the number) of the regressors.

A leading motivation for the use of cross-validation is that the resulting pseudo-out-of-sample errors have similar variance as out-of-sample errors whereas in-sample errors may have smaller variance. However, as outlined in Remark 2.1, this need not be a serious concern in our case; the proof will be in the pudding in the form of the empirical application below. ■

4 Empirical Application

4.1 Data

We use the 17 numerical regression benchmark data sets proposed by Grinsztajn et al. (2022) to evaluate the performance the hedged random forest (HRF) in comparison with the standard equal-weighted random forest (RF). Each data set is accessible on the official openML website `www.openml.org`, which also offers comprehensive metadata and descriptions. To ensure the relevance and quality of these data sets, Grinsztajn et al. (2022) employ the following criteria.

- **Heterogeneous columns:** Columns correspond to attributes of different nature. For example, this criterion excludes images or signal data sets where each column corresponds to the same signal on different sensors, for example.
- **Not high dimensional:** Only data sets with a d/n ratio below 1/10 are included, where d denotes the numbers of attributes.
- **Well-documented data sets:** Only data sets with sufficient metadata are included.
- **I.I.D. data:** Stream-like data sets and time-series date sets are excluded.
- **Real-world data:** Artificial data sets are excluded.
- **Not too small:** Data sets with fewer than four attributes or fewer than 3,000 observations are excluded.
- **Not too easy:** For a data set to be included, linear regression must perform 'sufficiently worse' than certain machine-learning methods; for the details see Grinsztajn et al. (2022).
- **Not deterministic:** Data sets where the response variable is a deterministic function of the attributes are excluded. This mostly means removing data sets on games such as poker and chess.
- **No missing data:** All observations containing missing values are excluded from the data sets. No imputations are performed.

Name	# Observations	# Attributes
ailerons	13,750	734
Bike_Sharing_Demand	17,379	12
brazilian_houses	10,692	10
cpu_act	8,192	22
diamonds	53,940	10
elevators	16,599	12
house_16H	22,784	17
house_sales	21,613	18
houses	20,640	9
isolet	7,797	614
medical_charges	163,065	4
MiamiHousing2016	13,932	14
nyc-taxi-green-dec-2016	581,835	19
pol	15,000	49
sulfur	10,081	7
superconduct	21,263	80
wine_quality	6,497	12

Table 1: Data sets used.

For large data sets, we follow Grinsztajn et al. (2022) by selecting 10,000 observations at random. To avoid interference with the data, we refrain from any attribute-preprocessing. Table 1 lists the data sets used in this analysis together with the corresponding numbers of observations and numbers of attributes.

4.2 Results

Any data set is partitioned into a training set and a test set by selecting n observations at random, which constitute the training set, and then using the remainder of the observations as the test set. For each method, RF and HRF, fitted on the training set we obtain a MSE on the test set, denoted by MSE_{RF} and MSE_{HRF}, respectively. In order to guard against spurious results due to the random partitioning of the data, we then repeat this process (independently) B times. As the final performance measure, we report the following root-mean-squared-error (RMSE) ratio:

$$\mathrm{RMSE}_{\mathrm{HRF/RF}} := \frac{\sqrt{\frac{1}{B}\sum_{b=1}^{B} \mathrm{MSE}_{\mathrm{HRF,b}}}}{\sqrt{\frac{1}{B}\sum_{b=1}^{B} \mathrm{MSE}_{\mathrm{RF,b}}}} \,. \tag{4.1}$$

This means that, for each method, we average the MSE values over the B repetitions and then take the root to arrive at individual RMSE values. Finally, we take the ratio of the two individual RMSE values. Ratios greater than one speak in favor of RF whereas ratios smaller than one speak in favor of HRF. Our results below are all based on $B = 100$ repetitions; larger values of B leave the results virtually unchanged.

In this way, for any training-set size n, we obtain 17 RMSE ratios (4.1), one for each data set listed in Table 1. We convert the 17 ratios into a boxplot and then line up the boxplots for $n \in \{200, 400, 600, 800, 1000, 2000, 3000, 4000, 5000\}$ in Figure 1. The results can be summarized as follows:

- On balance, HRF clearly outperforms RF.
- The gains are most pronounced for small training-set sizes n but 'live on' up to the largest size considered, $n = 5000$.
- Out of the 17 data sets, there are five on which HRF performs worse than RF, but the loss is never more than 8% and always below 5% for $n \geq 600$.
- On the other hand, there are two data sets for which HRF reduces the RMSE compared to RF by more than 10% for $n \leq 1000$; for one of these data sets, HRF actually reduces the RMSE by more than 10% for all n.
- Summing up, based on the 17 data sets considered, there is little to lose but potentially much to gain by upgrading from RF to HRF.

HRF outperforms RF particularly convincingly for smaller n, and thus for larger d relative to n. The intuition for this finding is that both RF and HRF are consistent forecast methods for sequences of data with d independent of n (when data are i.i.d.) and hence the difference between the two forecasts tends to decrease as n increases.

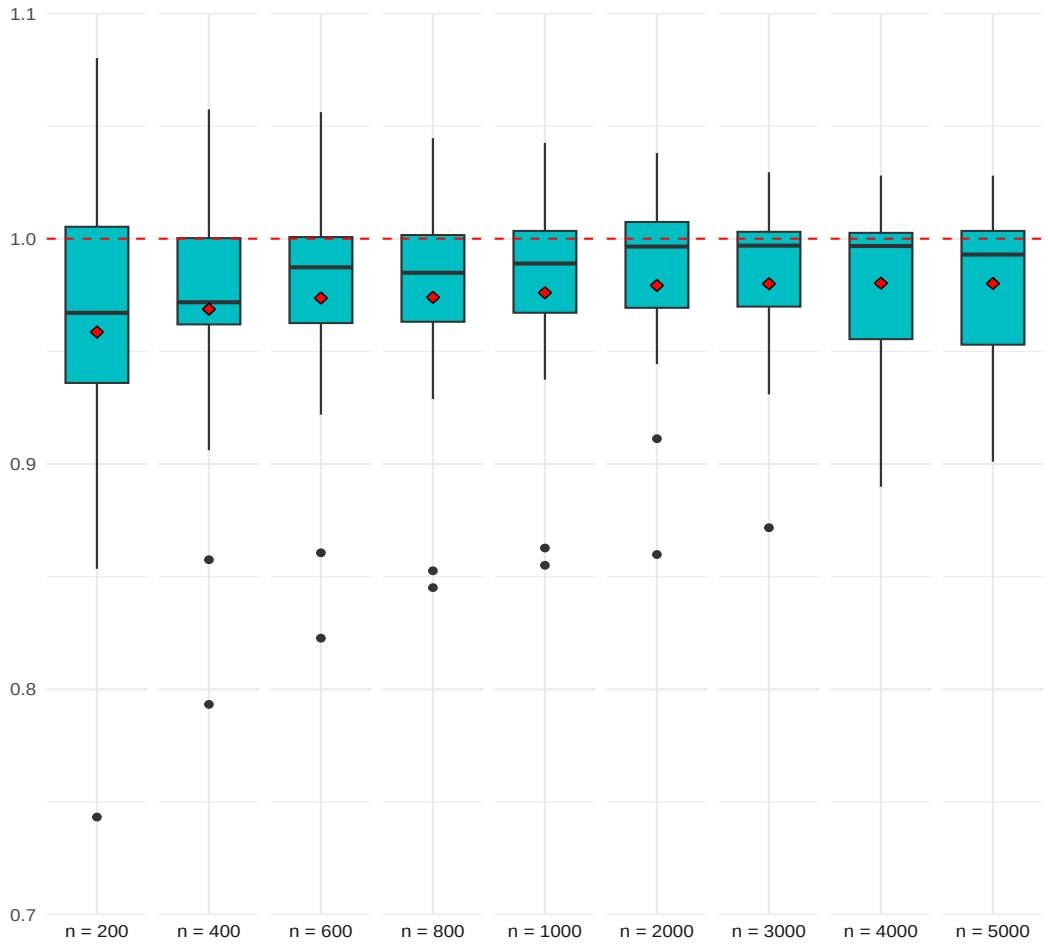

Figure 1: Boxplots of RMSE ratios (4.1). For each training-set size n, the boxplot is based on the 17 ratios corresponding to the data sets listed in Table 1.

4.3 Negative Weights and Shrinkage

All previous proposals for weighting the random forest that we are aware of, not only in the context of regression but also in the context of classification, impose the "no-short-sales constraint" $\kappa = 1$, that is, $w_j \geq 0$. As shown in the robustness checks of Appendix A, allowing for negative weights typically improves performance, and for certain data sets by a pronounced margin, when Σ is estimated with nonlinear shrinkage. On the other hand, this is not really the case when the sample covariance matrix is used instead. To illustrate this finding, we consider the following two RMSE ratios:

$$\text{RMSE}^{\text{sample}}_{\kappa=2/\kappa=1} := \frac{\sqrt{\frac{1}{B}\sum_{b=1}^{B} \text{MSE}_{\text{HRF,b},\kappa\,=\,2,\text{sample}}}}{\sqrt{\frac{1}{B}\sum_{b=1}^{B} \text{MSE}_{\text{HRF,b},\kappa\,=\,1,\text{sample}}}} \tag{4.2}$$

and

$$\text{RMSE}^{\text{shrinkage}}_{\kappa=2/\kappa=1} := \frac{\sqrt{\frac{1}{B}\sum_{b=1}^{B} \text{MSE}_{\text{HRF,b},\kappa\,=\,2,\text{shrinkage}}}}{\sqrt{\frac{1}{B}\sum_{b=1}^{B} \text{MSE}_{\text{HRF,b},\kappa\,=\,1,\text{shrinkage}}}} \,. \tag{4.3}$$

These ratios measure the effect of allowing for negative weights ($\kappa = 2$) relative to imposing non-negative weights ($\kappa = 1$): ratio (4.2) does so when Σ is estimated with the sample covariance matrix whereas ratio (4.3) does so when Σ is estimated with nonlinear shrinkage. Figure 2 shows that allowing for negative weights, on balance, leads to worse performance when Σ is estimated with the sample covariance matrix in contrast to using nonlinear shrinkage where it leads to better performance.

This finding might explain to some extent why the forecast-combination literature, so far, has generally shied away from negative weights, since it is still the norm in this literature to use the sample covariance matrix, a practice that should be abandoned.

In real-life finance applications, a "no-short-sales constraint" can be motivated by legislation (for example, mutual funds are not allowed to short stocks) or by practical considerations (for example, shorting certain assets may not be possible or may be prohibitively expensive). On the other hand, "short-selling" individual forecast methods $\mathcal{M}_j$ by assigning them a negative weight does not violate any laws and does not incur any monetary costs. We, therefore, hope that our paper will serve as motivation to the scientific community to allow for negative weights not only in the random forest but also in other forecast-combination settings. Just remember to abandon the sample covariance matrix in favor of shrinkage estimation.

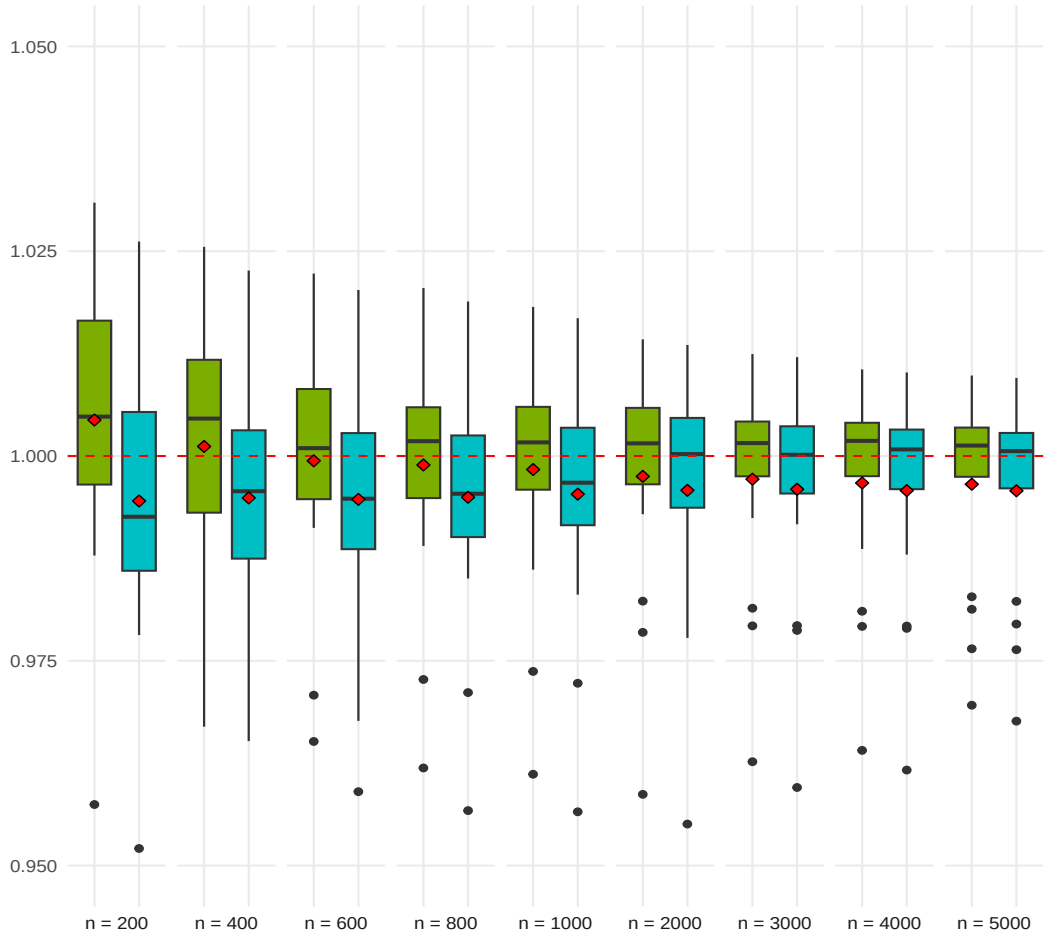

Figure 2: Boxplots of RMSE ratios (4.2) in green and RMSE ratios (4.3) in blue. For each training-set size n, the boxplots are based on the 17 ratios corresponding to the data sets listed in Table 1.

4.4 Related Literature

There have been previous proposals in the literature for weighting the random forest.

In the context of regression, we are only aware of Chen et al. (2023). It would have been interesting to compare our hedged random forest with their proposal but the authors (so far) have not made corresponding code available and our efforts to implement their proposal did not succeed.[4] At least, it can be pointed out that their proposal involves the choice of tuning parameters which also requires a validation set, in addition to a training set. resulting in a loss of information (meaning a smaller training set) in practice; see their

[4]We also contacted the authors by email about code but did not receive any response.

Section 4.

On the other hand, we can compare with two weighted-random-forest proposals originally intended for classification but easily adaptable for regression, namely the weighted random forest (WRF) by Winham et al. (2013) and the Césaro random forest (CRF) by Pham and Olafsson (2020). As will be seen below, the weights $\hat{w}_j$ in both proposals are always non-negative; actually, they are always strictly positive.

For the weighted random forest of Winham et al. (2013) one first computes, for $j = 1, \ldots p$, the tree-accuracy measure

$$t\mathrm{PE}_j \coloneqq \frac{1}{|\mathrm{OBB}_j|} \sum_{i \in \mathrm{OOB}_j} \left| y_i - \mathcal{M}_j(x_i) \right| , \tag{4.4}$$

where $\mathrm{OOB}_j \subset \{1, \ldots, n\}$ denotes the out-of-bag set corresponding to tree $\mathcal{M}_j$, that is, the subset of the training set not used in the pruning of the tree. Next, one computes relative weights according to one of the three following formulas:

$$\hat{w}_{j,\mathrm{rel}} \coloneqq 1 - t\mathrm{PE}_j \tag{4.5}$$

$$\hat{w}_{j,\mathrm{rel}} \coloneqq \exp\left(\frac{1}{t\mathrm{PE}_j}\right) \tag{4.6}$$

$$\hat{w}_{j,\mathrm{rel}} \coloneqq \left(\frac{1}{t\mathrm{PE}_j}\right)^{\lambda} \quad \text{for some } \lambda > 0 \tag{4.7}$$

Finally, the weights $\{\hat{w}_j\}$ are forced to sum up to one by setting

$$\hat{w}_j \coloneqq \frac{\hat{w}_{j,\mathrm{rel}}}{\sum_{l=1}^{p} \hat{w}_{l,\mathrm{rel}}} .$$

Following the advice of Winham et al. (2013), we tried as the leading candidates version (4.6) and version (4.7) with $\lambda \coloneqq 5$, of which the former performed somewhat better and is thus used below. We can construct as an analog to (4.1) the ratio

$$\mathrm{RMSE}_{\mathrm{HRF/WRF}} := \frac{\sqrt{\frac{1}{B}\sum_{b=1}^{B}\mathrm{MSE}_{\mathrm{HRF,b}}}}{\sqrt{\frac{1}{B}\sum_{b=1}^{B}\mathrm{MSE}_{\mathrm{WRF,b}}}} \ . \tag{4.8}$$

For the Césaro random forest of Pham and Olafsson (2020), one uses the tree-accuracy measures $t\mathrm{PE}_j$ defined in (4.4) to construct the weights, but in a different scheme: One sorts the measures $\{t\mathrm{PE}_1, \ldots, t\mathrm{PE}_p\}$ from smallest to largest and defines r_j as the order of $t\mathrm{PE}_j$ in the sorted sequence. Hence, if $t\mathrm{PE}_{(1)} \leq t\mathrm{PE}_{(2)} \leq \cdots \leq t\mathrm{PE}_{(p)}$, it holds that $t\mathrm{PE}_j = t\mathrm{PE}_{(r_j)}$. Then the relative Césaro weights are defined as

$$\hat{w}_{j,\mathrm{rel}} := \sum_{l=r_j}^{p} \frac{1}{l} \ . \tag{4.9}$$

Finally, the weights $\{\hat{w}_j\}$ are forced to sum up to one again by setting

$$\hat{w}_j := \frac{\hat{w}_{j,\mathrm{rel}}}{\sum_{l=1}^{p}\hat{w}_{l,\mathrm{rel}}} \ .$$

We can construct as an analog to (4.1) the ratio

$$\mathrm{RMSE}_{\mathrm{HRF/CRF}} := \frac{\sqrt{\frac{1}{B}\sum_{b=1}^{B}\mathrm{MSE}_{\mathrm{HRF,b}}}}{\sqrt{\frac{1}{B}\sum_{b=1}^{B}\mathrm{MSE}_{\mathrm{CRF,b}}}} \ . \tag{4.10}$$

As an analog to Figure 1 we now obtain Figure 3 which demonstrates that HRF clearly outperforms WRF, on balance. The comparison of HRF with CRF is a bit more subtle: Although the two methods exhibit comparable performance in terms of the median RMSE ratio, HRF consistently outperforms CRF in terms of the mean RMSE ratio for all training-set sizes n. Hence, taking everything into account, HRF also outperforms CRF, on balance.

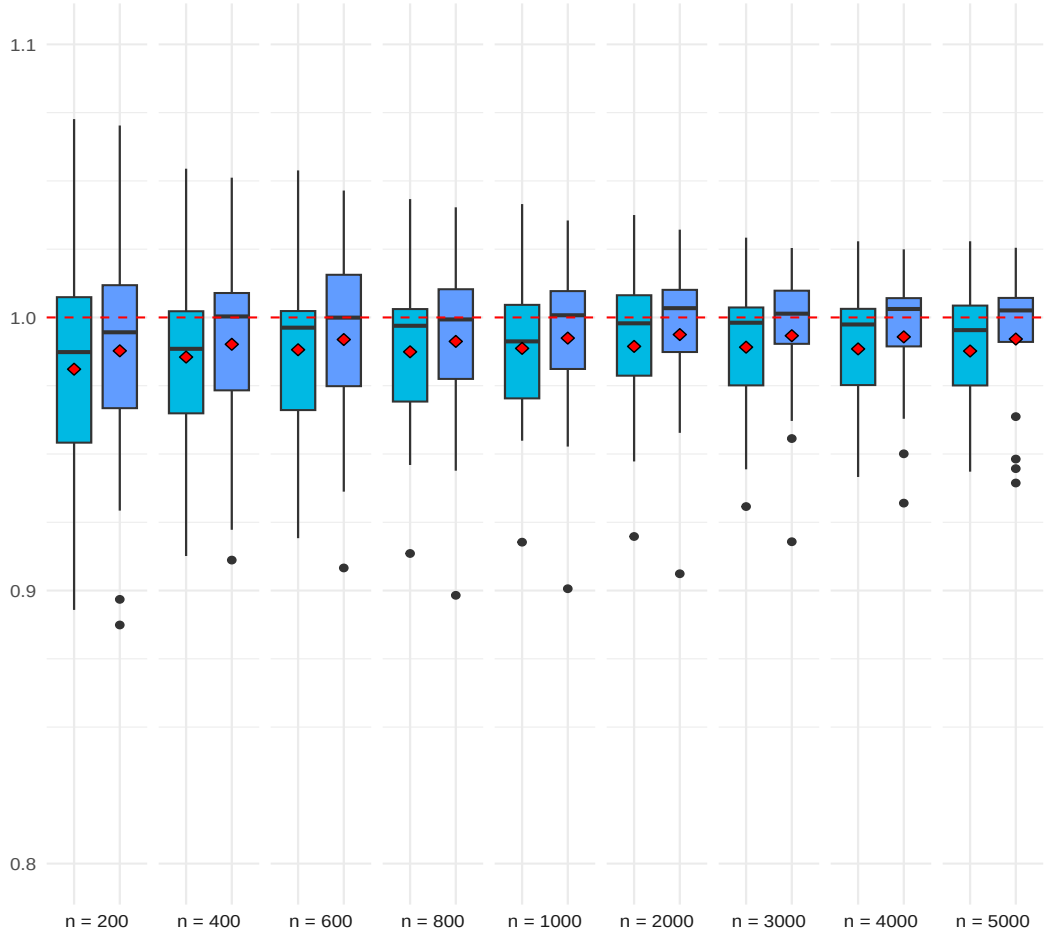

Figure 3: Boxplots of RMSE ratios (4.8) in light blue and MSE ratios (4.10) in dark blue. For each training-set size n, the boxplots are based on the 17 ratios corresponding to the data sets listed in Table 1.

4.5 Time-Series Data

We have previously demonstrated that the hedged random forest, on balance, provides superior forecasting performance compared to the standard random forest (based on equal-weighted trees). The evidence we presented has been for cross-sectional (or i.i.d.) data sets.

We also ran some limited, preliminary experiments for time-series data sets but did not find consistent outperformance of the hedged random forest over the standard random forest. There could be several reasons for this finding. On the one hand, with time-series data, estimating μ and Σ presents new challenges due to inherent dependence in

the data. In addition, many time-series data sets are not stationary but suffer from trends, seasonalities, time-varying parameters, or structural breaks; in such cases, the estimates of μ and Σ based on observations in a past window (including today) could simply be noticeably off-target for what is actually coming in the future, and thus not weighting the trees would be more robust.

This topic is an important and practical direction for future research.

A Robustness Checks

The goal of this appendix is to provide robustness checks with respect to the nature of the estimator $\hat{\Sigma}$ and the choice of the gross-exposure constraint κ.

As an alternative estimator $\hat{\Sigma}$ to nonlinear shrinkage, we will consider the canonical choice, the sample covariance matrix. As far as κ is concerned, we will consider the choices $\kappa \in \{1, 1.5, 2, 2.5, \infty\}$., the canonical choice being $\kappa := 1$.

Figure 4 displays the results for all choices of κ when $\hat{\Sigma}$ is given by nonlinear shrinkage whereas Figure 5 displays analogous results when nonlinear shrinkage is replaced by the sample covariance matrix instead.

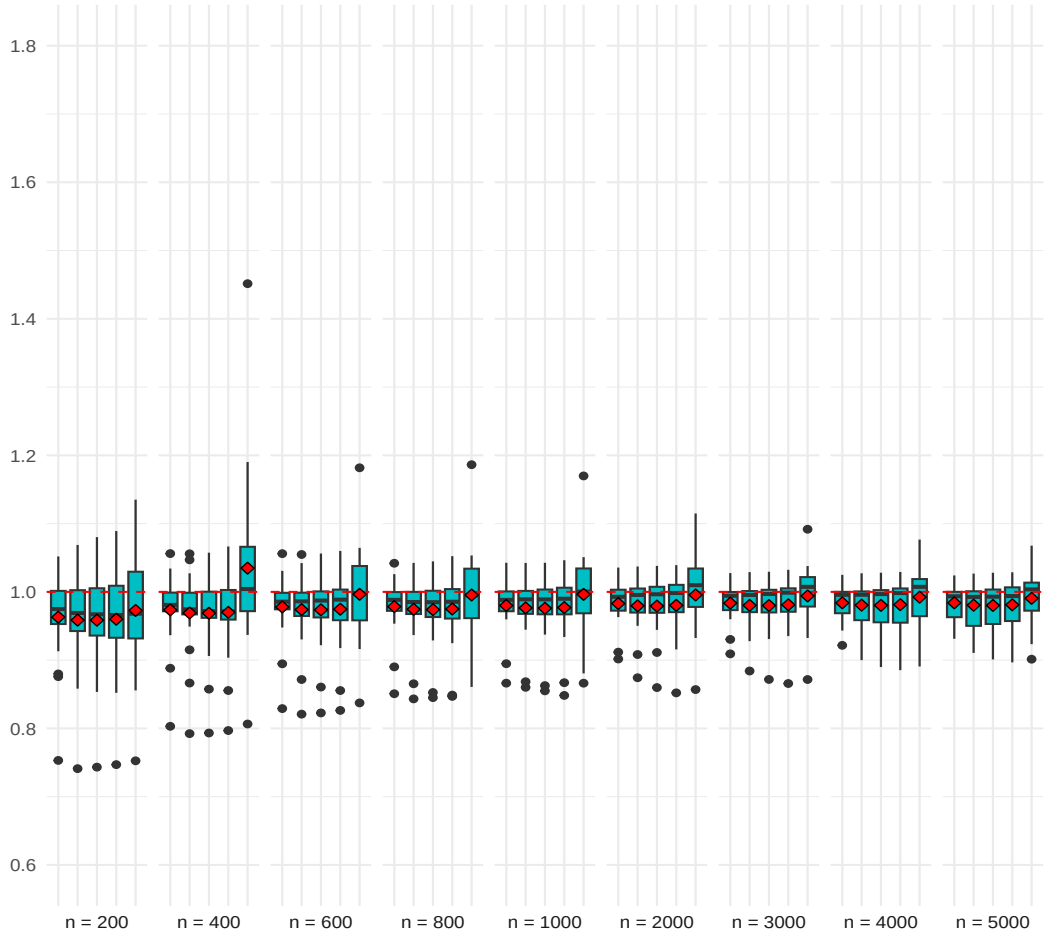

Figure 4: Boxplots of RMSE ratios (4.1). For each training-set size n, there are five boxplots corresponding to $\kappa \in \{1, 1.5, 2, 2.5, \infty\}$ where each boxplot is based on the 17 ratios corresponding to the data sets listed in Table 1. The estimator $\hat{\Sigma}$ is given by nonlinear shrinkage.

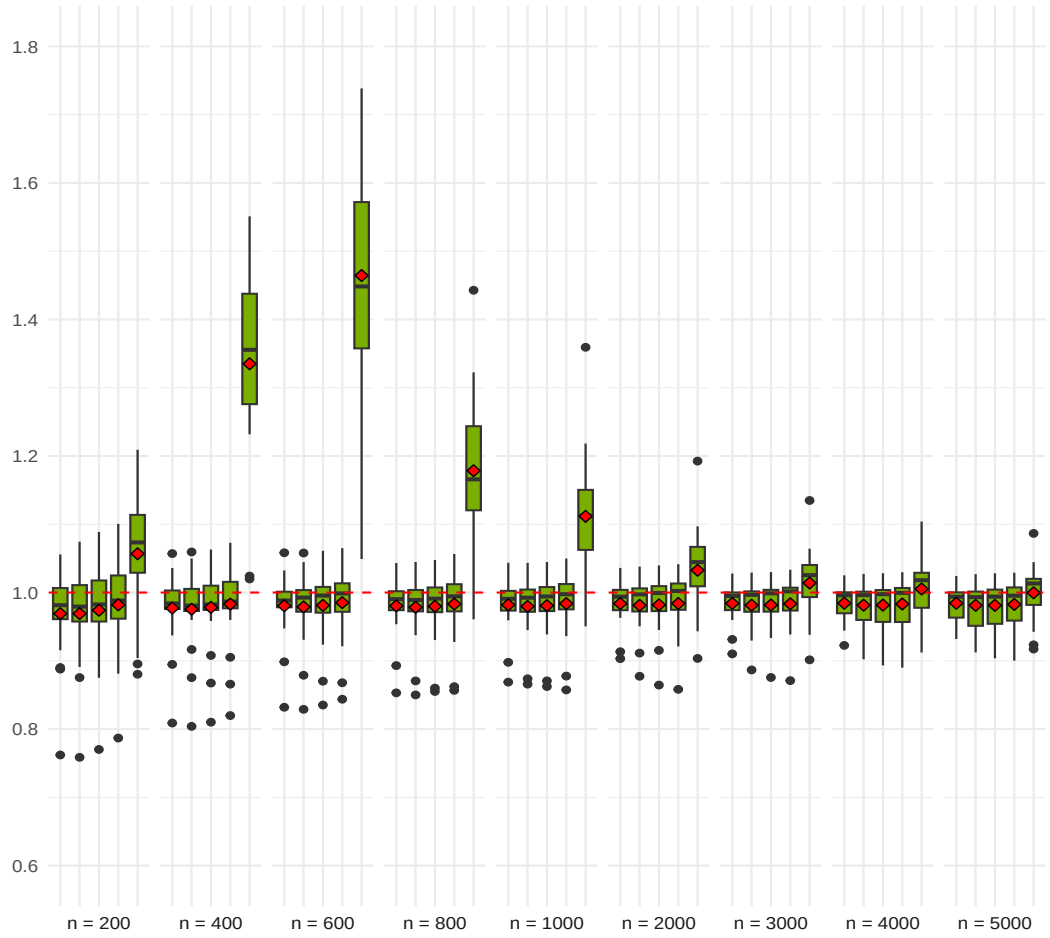

Figure 5: Boxplots of RMSE ratios (4.1). For each training-set size n, there are five boxplots corresponding to $\kappa \in \{1, 1.5, 2, 2.5, \infty\}$ where each boxplot is based on the 17 ratios corresponding to the data sets listed in Table 1. The estimator $\hat{\Sigma}$ is given by the sample covariance matrix.

When $\hat{\Sigma}$ is given by nonlinear shrinkage, the choice $\kappa \in \{1.5, 2, 2.5\}$ matters very little, and all three choices perform better than the canonical choice $\kappa := 1$. Not imposing a gross-exposure constraint (that is, the choice $\kappa := \infty$) performs worst overall, though it still outperforms RF, on balance.

When $\hat{\Sigma}$ is given by the sample covariance matrix instead, the choice of κ matters more. In particular, not imposing a gross-exposure constraint (that is, the choice $\kappa := \infty$) can lead to grave underperformance of the HRF when the training-set size, n, is close to the number of trees, p. On balance, the canonical choice $\kappa := 0$ seems to perform best.

Finally, we look at the combined benefit of our version of HRF compared to the canonical

choice in the literature: (i) use nonlinear shrinkage for $\hat{\Sigma}$ instead of the sample covariance matrix and (ii) use $\kappa := 2$ instead of $\kappa := 1$. Calling the two versions HRFour and HRFcan (for "our" and "canonical"), we can construct as an analog to (4.1) the ratio

$$\mathrm{RMSE}_{\mathrm{HRFour/HRFcan}} := \frac{\sqrt{\frac{1}{B}\sum_{b=1}^{B} \mathrm{MSE}_{\mathrm{HRFour,b}}}}{\sqrt{\frac{1}{B}\sum_{b=1}^{B} \mathrm{MSE}_{\mathrm{HRFcan,b}}}} \,. \tag{A.1}$$

As an analog to Figure 1 we then obtain Figure 6,

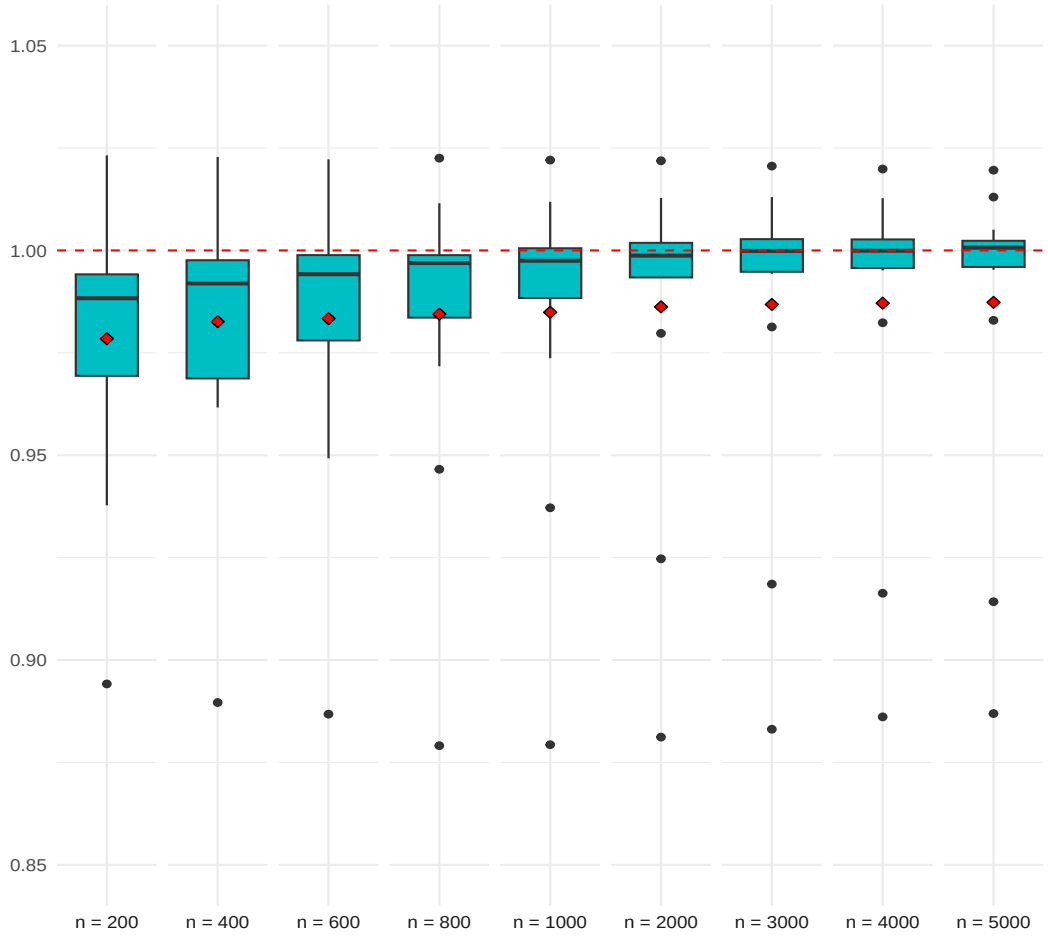

Figure 6: Boxplots of RMSE ratios (A.1). For each training-set size n, the boxplot is based on the 17 ratios corresponding to the data sets listed in Table 1.

One can see that in terms of the median RMSE ratio, our version outperforms the canonical version for $n \leq 2000$, after which the competition becomes a tie, whereas in terms of the mean RMSE ratio, our version outperforms the canonical version for all n.

References

Bergmeir, C. and Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213.

Bergmeir, C., Hyndman, R. J., and Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120:70–83.

Bodnar, T., Okhrin, O., and Parolya, N. (2019). Optimal shrinkage estimator for high-dimensional mean vector. *Journal of Multivariate Analysis*, 170:63–79.

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

Chen, X., Yu, D., and Zhang, X. (2023). Optimal weighted random forests. arXiv Preprint arXiv:2305.10042.

DeMiguel, V., Garlappi, L., Nogales, F. J., and Uppal, R. (2009). A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms. *Management Science*, 55(5):798–812.

Efron, B. and Hastie, T. J. (2022). *Computer Age Statistical Inference*. Cambridge University Press, Cambridge.

Elliott, G. and Timmermann, A. (2016). *Economic Forecasting*. Princeton University Press, Princeton.

Engle, R. F., Ledoit, O., and Wolf, M. (2019). Large dynamic covariance matrices. *Journal of Business & Economic Statistics*, 37(2):363–375.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, 35:507–520.

Hansen, B. E. (2016). The risk of James-Stein and Lasso shrinkage. *Econometric Reviews*, 35(8-10):1456–1470.

Hastie, T. J., Tibshirani, R., and Freedman, J. H. (2017). *The Elements of Statistical Learning*. Springer, New York, second edition.

Hayashi, F. (2000). *Econometrics*. Princeton University Press, Princeton, New Jersey.

Jagannathan, R. and Ma, T. (2003). Risk reduction in large portfolios: Why imposing the wrong constraints helps. *Journal of Finance*, 54(4):1651–1684.

Jobson, J. D. and Korkie, B. M. (1980). Estimation for Markowitz efficient portfolios. *Journal of the American Statistical Association*, 75:544–554.

Ledoit, O. and Wolf, M. (2022a). The power of (non-)linear shrinking: A review and guide to covariance matrix estimation. *Journal of Financial Econometrics*, 20(1):187–218.

Ledoit, O. and Wolf, M. (2022b). Quadratic shrinkage for large covariance matrices. *Bernoulli*, 28(3):1519–1547.

Michaud, R. (1989). The Markowitz optimization enigma: Is optimized optimal? *Financial Analysts Journal*, 45:31–42.

Pham, H. and Olafsson, S. (2020). On Césaro averages for weighted trees in the random forest. *Journal of Classification*, 37(1):223–236.

Sancetta, A. (2008). Sample covariance shrinkage for high dimensional dependent data. *Journal of Multivariate Analysis*, 99(5):949–967.

Wang, X., Hyndman, R. J., Li, F., and Kang, Y. (2022). Forecast combinations: An over 50-year review. *International Journal of Forecasting*.

Winham, S. J., Freimuth, R. R., and Biernacka, J. M. (2013). A weighted random forests approach to improve predictive performance. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 6(6):496–505.

Wright, M. N. and Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1):1–17.