# BOOTSTRAP SAMPLING RATE GREATER THAN 1.0 MAY IMPROVE RANDOM FOREST PERFORMANCE

#### Stanisław Kaźmierczak & Jacek Mańdziuk

Faculty of Mathematics and Information Science
Warsaw University of Technology
Koszykowa 75, 00-662 Warsaw, Poland
{stanislaw.kazmierczak, jacek.mandziuk}@pw.edu.pl

#### **ABSTRACT**

Random forests utilize bootstrap sampling to create an individual training set for each component tree. This involves sampling with replacement, with the number of instances equal to the size of the original training set (N). Research literature indicates that drawing fewer than N observations can also yield satisfactory results. The ratio of the number of observations in each bootstrap sample to the total number of training instances is called the bootstrap rate (BR). Sampling more than N observations (BR > 1) has been explored in the literature only to a limited extent and has generally proven ineffective. In this paper, we re-examine this approach using 36 diverse datasets and consider BR values ranging from 1.2 to 5.0. Contrary to previous findings, we show that such parameterization can result in statistically significant improvements in classification accuracy compared to standard settings (BR  $\leq$  1). Furthermore, we investigate what the optimal BR depends on and conclude that it is more a property of the dataset than a dependence on the random forest hyperparameters. Finally, we develop a binary classifier to predict whether the optimal BR is  $\leq 1$  or > 1 for a given dataset, achieving between 81.88% and 88.81% accuracy, depending on the experiment configuration.

#### 1 Introduction

Random forest (RF) algorithm, introduced by Breiman (2001), is an ensemble of decision trees (DTs) that collectively make decisions using either majority or soft voting. RF reduces variance, sometimes at the cost of slightly increasing bias, by introducing two sources of randomness. The first is the use of distinct subsets of features when selecting the best split at each node of the trees. The second is training each tree on a subset of observations drawn with replacement from the original training set, i.e., a bootstrap sample.

In this study, we analyze the bootstrap rate (BR), an RF hyperparameter that controls the training process and consequently affects the model's performance. BR is defined as the ratio of the number of observations in each bootstrap sample to the total number of training instances. In the literature, this parameter is also referred to as the sample rate, subsample size, bootstrap size ratio, or bag size. In his original work, Breiman (2001) used BR = 1. However, lower values have also been successfully applied (Martínez-Muñoz & Suárez, 2010; Adnan, 2014). When BR is low, each tree is trained on a more distinct subset of the data, which increases diversity among RF estimators. Naturally, the computational cost is reduced compared to BR = 1. On the other hand, the trees may become too weak, as they are trained on a relatively smaller portion of the data. For BR = 1, the expected fraction of unique observations from the entire dataset is 63.2%. When BR < 1, this fraction is even lower.

For BR = 1, we expect 36.8% of observations to be absent in each bootstrap sample. Intuitively, there is no obvious answer as to what would happen if BR > 1. A higher BR, on the one hand, causes subsets to be less diverse, but on the other hand, it includes more unique observations (i.e., more information) in each sample. We found this problem worth investigating. To our knowledge, Martínez-Muñoz & Suárez (2010) are the only ones who have analyzed BR > 1. However, they only considered BR = 1.2 and concluded that such parameterization is generally ineffective. In our

work, we not only analyze BR = 1.2 (and lower) but also explore higher values of 2, 3, 4, and 5. Additionally, we extend the experimental setup to 18 RF configurations, compared to what appears to be a single configuration (though this is not clearly specified) in the reference paper. Surprisingly, and in contrast to the findings of Martínez-Muñoz & Suárez (2010), we discover that BR > 1 often yields better results than conventional BR values in the range (0,1].

The primary contributions of this work can be summarized in four key points:

- To our knowledge, we are the first to analyze and shed light on what the optimal BR value depends on;
- To our knowledge, we are the first to suggest that testing BR > 1 is meaningful and often yields better results than the standard BR ≤ 1;
- We demonstrate that the optimal BR is only partially dependent on the RF configuration and is more a property of the dataset;
- We develop a binary classifier that, based on the class structure, predicts whether the optimal BR is  $\leq 1$  or > 1 for a given dataset, and achieves an accuracy between 81.88% and 88.81%, depending on the experimental configuration.

#### 2 Related Literature

Probst et al. (2019), in their survey on RF tuning, point out several hyperparameters that are commonly targeted by researchers when optimizing RF. The number of trees, which is the most extensively explored RF hyperparameter, was analyzed by Oshiro et al. (2012); Scornet (2017); Probst & Boulesteix (2018). The optimization of the number of attributes to consider when looking for the best split was addressed by Bernard et al. (2009); Goldstein et al. (2011). Additionally, Scornet (2017); Duroux, Roxane & Scornet, Erwan (2018) analyzed maximum tree depth.

We found the BR hyperparameter to be underresearched. Probst et al. (2019) consider it to have a minor influence on RF performance, while simultaneously stating that it is often worth tuning. Duroux, Roxane & Scornet, Erwan (2018) claim that due to the complexity of RFs, conducting a thorough theoretical analysis is challenging. As a result, most studies either overlook bootstrapping entirely (Biau et al., 2008; Ishwaran & Kogalur, 2010; Denil et al., 2013) or focus on simplified versions of RF, such as median forests (Scornet, 2017; Duroux, Roxane & Scornet, Erwan, 2018).

The study most relevant to our research was conducted by Martínez-Muñoz & Suárez (2010). First, it is the only work we found that analyzed BR > 1, although it is limited to a BR value of 1.2. Second, they examine how RF performance depends on BR. Among the 30 datasets analyzed, four types of BR curves showing the relationship between BR and classification error were identified. However, the analysis is limited to just one RF configuration and does not explore why the optimal BR may differ significantly between datasets—there is no insight provided as to why a particular curve shape is associated with a given dataset.

#### 3 EXPERIMENT CONFIGURATION

Experiments were conducted on 36 diverse datasets, which underwent the following preprocessing steps: all duplicates and rows with classes occurring only once were removed. Columns with a single unique value were also dropped. Missing values in categorical features were replaced with a new category, while missing values in numerical attributes were imputed with the column mean. Finally, one-hot encoding of categorical attributes was applied before standardizing all features. Table 3 in Appendix A presents the characteristics of the datasets after the preprocessing. Our experiments include all 30 datasets used by Martínez-Muñoz & Suárez (2010) and six additional ones.

The following hyperparameters (along with BR) are considered to be the most important for RF performance (Scornet, 2017; Probst et al., 2019; Zhu et al., 2022): number of trees (nt); parameters controlling the size of the trees: maximum tree depth (md), the minimum number of instances required to split an internal node (mn), the minimum count of observations necessary to constitute a leaf node (ml); function measuring the quality of a split (qs); number of attributes to consider when looking for the best split (nf).

As the base values for these hyperparameters, we adopted the defaults from the scikit-learn 1.1.3 Python package: nt = 100, md = None (no depth limit), qs = "gini" (Gini impurity), mn = 2, ml = 1, nf = "sqrt" (square root of the number of features). We denote such a model as RF(base). Altogether, we tested RF(base) and 17 other configurations resulting from the following modifications of each single hyperparameter in RF(base):

- RF(nt\_200), RF(nt\_500): number of trees equals 200 or 500, respectively;
- RF(md\_10), RF(md\_15), RF(md\_20), RF(md\_25): maximum depth of a tree equals 10, 15, 20, or 25, respectively;
- RF(qs\_ent): split quality is measured using Shannon entropy (information gain);
- RF(mn\_3), RF(mn\_4), RF(mn\_6), RF(mn\_8): minimum number of observations required to split an internal node is equal to 3, 4, 6, or 8, respectively;
- RF(ml\_2), RF(ml\_3), RF(ml\_4), RF(ml\_5): minimum number of instances per leaf is 2, 3, 4, or 5, respectively;
- RF(nf\_log), RF(nf\_all): number of features considered in a node split equals the logarithm with base 2 of the number of attributes or all features are taken into account, respectively.

The following BRs were tested: 0.2, 0.4, 0.6, 0.8, 1.0, 1.2 (as analyzed by Martínez-Muñoz & Suárez (2010)), 2.0, 3.0, 4.0, and 5.0. For each configuration, 2-fold stratified cross-validation, repeated 200 times, was applied, yielding 400 results.

#### 4 RESULTS

For each dataset, we searched for the pair of RF configuration and BR that yielded the highest classification accuracy. Table 1 presents these results.

**Statistical significance.** The main observation is that BR > 1 constituted the best setup in 20 out of 36 datasets. To further compare standard BRs (BR  $\leq$  1, first group) with those greater than one (second group), we performed a paired *t*-test (with the alternative hypothesis that the first sample has a greater mean than the second one) on the results of the dataset winner (best performing configuration) and results from all configurations with the other BR group. So, if the best classification accuracy was achieved by RF with BR  $\leq$  1, we compared these results with all results related to configurations with BR > 1, and vice versa. The last column of the table shows the maximum *p*-value among all *t*-tests for each dataset. We analyzed several significance levels: 0.1, 0.05, 0.01, 0.001, 0.0001, and 0.00001. Considering only conclusive results (i.e., *p*-values lower than the specified significance level), the difference in the number of datasets with the best related model with BR > 1 versus those with BR  $\leq$  1 amounted to 5, 2, -2, -4, -2, and 0, respectively. This indicates that, depending on the chosen significance level, the number of datasets with the optimal solution involving BR  $\leq$  1 is roughly comparable to those with BR > 1.

**Number of winning configurations.** Among the 18 analyzed RF configurations, only seven achieved the highest classification accuracy in at least one dataset: RF(nt\_500) (20 datasets), RF(qs\_ent) (5 datasets), RF(ml\_5) (4 datasets), RF(mn\_8) (3 datasets), RF(ml\_4) (2 datasets), RF(mn\_4) (1 dataset), and RF(nf\_all) (1 dataset). Further analysis will concentrate on these setups.

**Frequence of winning BRs.** Fig. 1 depicts the frequency of winning BRs, both globally and for each RF parameter setting. The histogram related to RF(nt\_500) is the most similar to the global one, as this model achieved the best score in 20 out of 36 datasets. For RF(ml\_4) and RF(ml\_5), models that restrictively control the size of the tree, BR > 1 constituted the best setup for as many as 26 datasets. It stems from the fact that, in many cases, a low number of training instances combined with a relatively high minimum number of samples required to create a leaf led to underfitted trees. Thus, high BR served as a remedy, enabling the construction of more complex models. RF(nf\_all) exhibited different behavior compared to the other models. The higher the BR, the less frequently it was optimal. The key sources of diversity among the individual trees are the distinct subsets of attributes to consider when looking for the best split in each node, along with the unique bootstrap sample used in training. When all features are analyzed in a node splitting, the first source of diversity ceases to exist. Thus, to maintain an overall level of diversity, RF(nf\_all) preferred lower BRs,

Table 1: Classification results. The consecutive columns present the dataset name, the optimal RF configuration, the achieved accuracy, the best BR, and the *p*-value from the conducted *t*-test.

Dataset	Best model	Acc. [%]	Bootstrap rate	<i>p</i> -value
Abalone	$RF(ml_5)$	26.801	0.2	$< 10^{-6}$
Adult	$RF(ml_5)$	86.484	4.0	$< 10^{-6}$
Arrhythmia	RF(nf_all)	76.161	1.2	0.305022
Audiology (Standardized)	$RF(mn_8)$	75.338	5.0	0.013121
Australian Credit Approval	RF(nt_500)	87.225	0.6	0.132623
Balance Scale	RF(nt_500)	85.972	0.2	$< 10^{-6}$
Breast Cancer Wisc. (Diag.)	RF(qs_ent)	95.898	5.0	$< 10^{-6}$
Breast Cancer Wisc. (Orig.)	RF(nt_500)	95.506	0.4	0.001910
Congressional Voting Rec.	RF(mn_8)	94.795	2.0	0.029394
Echocardiogram	$RF(ml_5)$	73.113	2.0	0.035933
Ecoli	RF(nt_500)	85.835	0.6	0.000097
German Credit Data	RF(nt_500)	75.467	1.2	0.079419
Glass Identification	RF(qs_ent)	75.596	2.0	0.002702
Heart	$RF(ml_4)$	83.324	0.2	0.000004
Hepatitis	RF(nt_500)	84.726	0.4	0.000644
Horse Colic	$RF(nt_{-}500)$	86.516	1.0	0.485986
Image Segmentation (Stat.)	RF(qs_ent)	97.133	5.0	$< 10^{-6}$
Ionosphere	RF(nt_500)	93.254	1.2	0.192406
Iris	$RF(mn_4)$	95.232	0.4	0.105220
Labor Relations	RF(nt_500)	93.608	1.2	0.004194
Liver Disorders	$RF(ml_4)$	59.714	0.2	$< 10^{-6}$
Optical Recognition (Digits)	$RF(nt_500)$	97.413	4.0	$< 10^{-6}$
Parkinsons	RF(nt_500)	89.306	5.0	$< 10^{-6}$
Pima Indians Diabetes	RF(nt_500)	76.344	0.2	0.000018
Sonar, Mines vs. Rocks	RF(qs_ent)	81.627	4.0	$< 10^{-6}$
Soybean (Large)	RF(mn8)	92.712	4.0	$< 10^{-6}$
Tic-Tac-Toe Endgame	RF(nt_500)	97.264	5.0	0.021006
Thyroid Disease	RF(qs_ent)	95.840	1.2	0.059261
Vehicle Silhouettes	RF(nt_500)	74.583	5.0	0.001911
Vowel Recognition	$RF(nt_500)$	92.285	3.0	$< 10^{-6}$
Wine	RF(nt_500)	97.809	1.2	0.072172
Ringnorm	RF(nt_500)	92.717	0.6	$< 10^{-6}$
Threenorm	RF(nt_500)	80.050	0.4	0.000654
Twonorm	RF(nt_500)	96.002	0.2	$< 10^{-6}$
Waveform	RF(nt_500)	86.165	0.2	$< 10^{-6}$
LED Display Domain	RF(ml_5)	66.590	1.0	$< 10^{-6}$

which created more varied sets of samples drawn and made the trees less correlated. Looking at the BR histograms, extreme BR values (0.2 and 5.0) constituted the best solutions, both overall and for all analyzed RF configurations, the greatest number of times (the highest bar in each histogram corresponds either to 0.2 or 5.0). This suggests that the optimal BR may often be lower than 0.2 or higher than 5.0, indicating that even a broader range should be tested when tuning RF. Finally, BR = 1, defined in the original formulation of the bootstrapping procedure and the most frequently used value, performed relatively poorly. Overall, it was optimal for only two datasets. When analyzing individual RF configurations, for three of them, BR = 1 was not able to win even in one dataset. Averaging over all seven parameter settings, it was optimal for only 1.43 out of 36 datasets. Interestingly, adjacent to 1, the nonstandard BR = 1.2 was, with the exception of RF(nf\_all), always better, often substantially.

**Individual dataset analysis.** Fig. 2 illustrates the relationship between the performance and BR for the analyzed RF configurations across a selected group of diverse datasets. Charts for the remaining datasets are provided in Appendix B. Our first observation is that RF(nf\_all) behaves differently from the other models. In almost all cases, it reaches optimal accuracy with a lower (or equal) BR

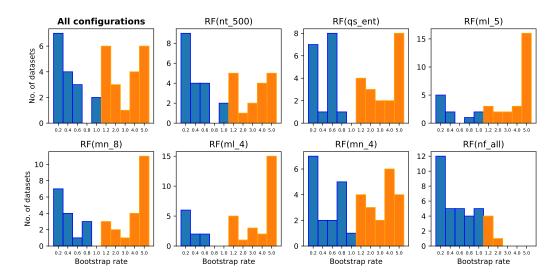


Figure 1: Distribution of winning BR across all RF configurations (top left) and among individual RF parameterizations.

compared to other RFs. This observation is consistent with the trends shown in Fig. 1 and the explanation provided in the previous paragraph. In most cases, the best accuracy achieved by RF(nf\_all) is substantially worse than that of the other models, and after reaching the optimum, its performance declines rapidly. There are several datasets for which RF(nf\_all) performs well. It achieved the best accuracy across all models on the Arrhythmia dataset and was comparable to the best on Audiology (Standardized), Tic-Tac-Toe Endgame, Pima Indians Diabetes, and Iris. The high performance of RF(nf\_all) is undoubtedly related to the characteristics of the features, e.g., the abovementioned Arrhythmia has the highest number of features among all datasets. However, the relationship is more complex for the other four datasets. We hypothesize that the importance of features needs to be further assessed to gain deeper insights. Presumably, RF(nf\_all) will perform well on datasets with a high proportion of insignificant or less significant features, as it may avoid building trees primarily based on these features.

**Typical BR curve shapes.** All RF configurations other than RF(nf\_all) are generally similar in terms of the characteristics of their BR curves. We identified three categories that describe how the set of curves appears:

- (a) In the first and most common pattern, all curves increase to at least BR = 1.2, indicating that the optimal BR is at least 1.2. The curves then either continue to rise (usually more smoothly)/reach a plateau (first subpattern) or they oscillate/gradually decrease (second subpattern). The first subpattern can be observed in the Arrhythmia, Audiology (Standardized), Parkinsons, Breast Cancer Wisc. (Diag.), Optical Recognition (Digits), Ionosphere, Image Segmentation (Stat.), Sonar, Mines vs. Rocks, Soybean (Large), Tic-Tac-Toe Endgame, Vowel, and Recognition datasets. The second subpattern is seen in the Wine, German Credit Data, Glass Identification, Labor Relations, Thyroid Disease, Vehicle Silhouettes, and Congressional Voting Rec. datasets.
- (b) In the second pattern, all curves either decrease from the very beginning (BR = 0.2) or rise to a BR in the range [0.4, 1.0] and then decline. The overall shape of the curves may be fairly smooth, as seen in the Abalone, Balance Scale, Breast Cancer Wisc. (Orig.), Heart, Liver Disorders, Twonorm, Waveform, and LED Display Domain datasets, or it may exhibit some irregularities, as observed in the Iris and Pima Indians Diabetes datasets.
- (c) The third pattern is a mixture of the first and second patterns. Curves associated with some RF configurations, mainly RF(ml\_4) and RF(ml\_5), behave similarly to those in the first pattern, while others resemble the curves seen in the second pattern. This third pattern is present in the Adult, Australian Credit Approval, Ecoli, Hepatitis, Ringnorm, Threenorm, Horse Colic, and Echocardiogram datasets. In the case of the last two, some additional irregularities in the BR curves may be observed.

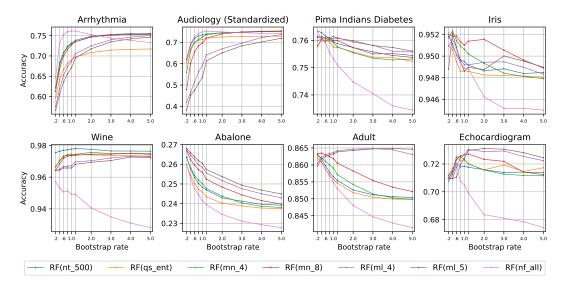


Figure 2: Characteristics of bootstrap rate curves for selected datasets.

The main observation from the above analysis is that BR curves associated with all RF configurations, except RF(nf\_all), are fairly consistent. The first and second patterns, within which all curves exhibit similar behavior, were observed in 28 out of 36 datasets. This leads to the conclusion that the optimal BR is merely dependent on RF parameterization and is closely related to the dataset.

Naturally, the procedure for testing high BR values follows a typical 'no free lunch' scenario—while we may find RF configuration yielding better results, it comes at the cost of slower execution, as it involves sampling more observations and building trees on a larger number of instances. We did not analyze issues related to time performance, which may represent potential direction for further research.

#### 5 TOWARDS UNDERSTANDING THE OPTIMAL BOOTSTRAP RATE

**Higher level approaches.** While searching for the reasons why the BR curve differs so significantly between datasets, we began with analyzing the general properties of these datasets, such as the number of features (divided into continuous and binary) and the number of training instances. We also created new features reflecting interactions by applying arithmetic operations to the aforementioned attributes. However, neither approach helped us to understand the problem better. Next, we took a more local approach and examined whether the BR curve was associated with the number of clusters present in the data. Unfortunately, this research direction was also inconclusive. In the meantime, we observed that even small changes in the data could lead to significant changes in the shape of the BR curve and the optimal value of BR. Fig. 3 provides an example. This observation prompted us to go even more local and analyze the neighborhood of individual instances.

**Lower level approaches.** In brief, RF is composed of DTs that cut the feature hyperspace into decision regions defined by the path leading from the root to the corresponding leaf. In a single tree, the prediction for a sample located in a particular leaf region is based on the majority class of the instances that reached that leaf during training. This means that the neighbors (specifically, their classes) of the predicted instance affect the predicted label. The same applies to RF, as it performs majority voting on predictions made by the component DTs. To analyze the structure of neighbors in each dataset, we standardized all continuous features and mapped all binary attributes to -1 and 1. We employed the Manhattan metric, which considers the distance along each feature (axis) independently, to measure the distance between observations. The Manhattan distance is generally a good choice in the context of a DT (and RF), which also defines a decision boundary that, at any moment, moves along only one feature.

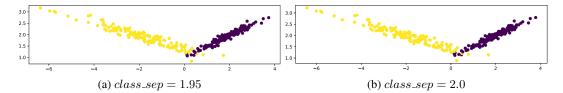


Figure 3: An example illustrating how even small differences in the data can significantly affect the optimal BR value. Both figures (a) and (b) depict synthetically generated data using scikit-learn's  $make\_classification$  method with the following parameters:  $n\_samples = 300$ ,  $n\_features = 2$ ,  $n\_classes = 2$ ,  $n\_classes = 2$ ,  $n\_classes = 1$ , and  $random\_state = 1$ . The only difference is the value of the  $class\_sep$  parameter, which controls the separation between the classes. In (a), it is set to 1.95, while in (b), it is 2.0. As a result of this slight difference, the optimal BR in (a) equals 5.0, while in (b), it amounts to 0.2. All other parameters of the  $make\_classification$  method remain at their default values.

Let us introduce the notation  $k\_l$ , which is the number of observations for which, among k nearest neighbors, l samples belong to the same class as the considered instance. Intuitively, high  $k\_l$  values for low l, relative to k, indicate inhomogeneity in the data and possibly a relatively high number of outliers. For each dataset, we calculated the  $k\_l$  statistics for  $k \in \{1, 2, \dots, 10\}$  and  $l \in \{0, \dots, k\}$  for each k. Then, we performed normalization so that for each k, the sum  $k\_0$ , ...,  $k\_k$  equaled 100, making those values comparable between datasets of varying sizes.

For each  $k\_l$  and each RF's optimal BR, including the overall best, we calculated the Spearman rank-order correlation coefficient. Table 2 presents the results for  $k \in \{1, 2, \ldots, 6\}$ . Our first observation is that the overall best BR is always positively correlated with  $k\_k$ . The highest correlation corresponds to k=1,2,3, after which it gradually decreases. Second, for each  $k\_l$  where  $l \neq k$ , the correlation is negative; the lower l is, the stronger the correlation becomes in terms of absolute value. This means that the optimal BR for inhomogeneous datasets (high  $k\_l$  values for low l) tends to be lower than for uniform datasets (high  $k\_l$  values for l close to k). A low BR leads to ambiguous observations being drawn less frequently. Therefore, fewer decision trees have leaves affected by such instances, and the remaining RF trees may mitigate incorrect decisions in the majority voting scheme. Conversely, for more uniform data, we hypothesize that a higher BR yields better results because it creates a bootstrap sample with more unique instances, thereby providing more information while maintaining diversity through varying the number of occurrences of these instances. Finally, an excessively high BR does not yield good results because it reduces diversity; as a consequence of the law of large numbers, the number of occurrences of individual observations converges towards each other.

Optimal BRs related to individual RF configurations share properties similar to those of the best overall BR. For all k, the correlation between  $k\_k$  and the optimal BR is positive, gradually decreasing after 2\_2 (or 1\_1 in the case of RF(nf\_all)). The rest of the  $k\_l$  values ( $l \neq k$ ), for  $k \leq 5$ , are negative, with some exceptions for 5\_4. The trend between  $k\_0$  and  $k\_k$  is generally upward, but unlike the best overall BR, it is non-monotonic. RF(nf\_all) exhibits this behavior as well, but the range of  $k\_l$  values is visibly narrower, and some irregularity occurs for k=5.

For the increasing k, starting from k=6, the general ascending trend from  $k\_0$  to  $k\_k$  is maintained. However, the range  $[k\_0, k\_k]$  narrows, the changes are not perfectly monotonic, and more positive values, other than  $k\_k$ , appear. We suppose that considering k>5 becomes too general (causing the above irregularities) but still reflects the uniformity of the data (hence, an overall trend is maintained).

The absolute values of the correlation coefficients are not high. For the overall best BR, the highest Spearman rank-order correlation coefficient amounts to 0.330. This is because the function modeling the optimal BR is complex and not dependent on just one predictor. We found two ways to build attributes that are more highly correlated with the target. The first is by multiplying each feature (k - l) by the number of classes in a particular dataset. Intuitively, the higher the number of classes, the lower the probability that outliers (ambiguous observations) from the same class, potentially forming a decision leaf in a tree, are drawn. Indeed, the correlation coefficients move upward. The negative ones become closer to zero or even turn positive. For example, for the overall best BR,

Table 2: Spearman rank-order correlation coefficient between  $k \rfloor$  and the best BR—overall (second column) and respective RF configurations (columns 3–9).

$k_{l}$	Best RF	nt_500	qs_ent	ml_5	mn_8	ml_4	mn_4	nf_all
1_0	-0.311	-0.299	-0.345	-0.319	-0.312	-0.332	-0.387	-0.173
$1_{-}1$	0.311	0.299	0.345	0.319	0.312	0.332	0.387	0.173
2_0	-0.292	-0.252	-0.292	-0.263	-0.258	-0.280	-0.354	-0.156
$2_{-1}$	-0.252	-0.264	-0.298	-0.255	-0.241	-0.277	-0.317	-0.164
2_2	0.330	0.301	0.347	0.320	0.320	0.332	0.379	0.163
3_0	-0.264	-0.258	-0.275	-0.242	-0.263	-0.256	-0.350	-0.139
3_1	-0.250	-0.238	-0.311	-0.264	-0.250	-0.292	-0.331	-0.142
3_2	-0.239	-0.213	-0.230	-0.164	-0.163	-0.192	-0.261	-0.183
3_3	0.323	0.280	0.341	0.307	0.294	0.320	0.365	0.151
4_0	-0.292	-0.266	-0.278	-0.254	-0.268	-0.258	-0.351	-0.134
4_1	-0.261	-0.233	-0.283	-0.255	-0.249	-0.274	-0.325	-0.159
4_2	-0.213	-0.208	-0.264	-0.209	-0.179	-0.235	-0.280	-0.147
4_3	-0.114	-0.116	-0.134	-0.031	-0.056	-0.067	-0.158	-0.090
$4_{-}4$	0.299	0.261	0.319	0.286	0.269	0.301	0.346	0.146
5_0	-0.238	-0.221	-0.218	-0.198	-0.185	-0.217	-0.285	-0.099
$5_{-1}$	-0.227	-0.178	-0.224	-0.183	-0.201	-0.205	-0.267	-0.034
5_2	-0.223	-0.232	-0.298	-0.230	-0.220	-0.264	-0.321	-0.134
5_3	-0.204	-0.148	-0.181	-0.114	-0.113	-0.143	-0.211	-0.127
5_4	-0.084	-0.027	-0.035	0.096	0.056	0.055	-0.058	0.003
5_5	0.302	0.244	0.301	0.269	0.245	0.285	0.318	0.125
6_0	-0.213	-0.186	-0.170	-0.165	-0.149	-0.182	-0.251	-0.080
6_1	-0.156	-0.134	-0.183	-0.127	-0.165	-0.156	-0.223	0.031
6_2	-0.234	-0.239	-0.292	-0.225	-0.214	-0.261	-0.328	-0.154
6_3	-0.158	-0.128	-0.199	-0.167	-0.136	-0.198	-0.213	-0.015
6_4	-0.109	-0.041	-0.062	0.039	0.039	0.008	-0.086	-0.083
6_5	-0.013	0.030	-0.001	0.169	0.125	0.129	0.030	0.080
6_6	0.265	0.204	0.261	0.220	0.190	0.235	0.260	0.080

2\_0, 3\_2, and 4\_3 increase from -0.292, -0.239, and -0.114 to -0.191, -0.041, and 0.038, respectively. Similarly, the positive values rise even higher. For instance, 2\_2, 3\_3, and 4\_4 increase from 0.330, 0.323, and 0.299 to 0.456, 0.493, and 0.454, respectively.

The second way to create predictors with a higher correlation to the target is to introduce features that represent interactions between existing attributes  $k \rfloor l$ , where  $k \in \{1, 2, \ldots, 10\}$  and  $l \in \{0, \ldots, k\}$  for each k. More specifically, for all pairs of distinct features, we perform division, subtraction, multiplication, and addition (the first two in both directions). Additionally, each attribute is multiplied by itself and added to itself, creating two additional features. In this way, 12620 new attributes are created. The features most positively correlated with the target are  $9 \rfloor 2/2 \rfloor 0$  ( $9 \rfloor 2$  divided by  $2 \rfloor 0$ ),  $10 \rfloor 2/3 \rfloor 0$ , and  $10 \rfloor 2/4 \rfloor 0$ . The respective correlation coefficients are 0.607, 0.595, and 0.595.

**Bootstrap rate prediction.** To further assess how well the above set of attributes describes the problem, we used them (along with base  $k\_l$  statistics) to build a binary classifier predicting whether the optimal BR across all RF configurations is  $\leq 1$  or > 1 for a given dataset. As in the main experiments, we tested all 18 RF configurations and 10 BR values. Due to the limited number of observations—each corresponding to one of 36 datasets—we performed Leave-Two-Out Cross-Validation in all possible variants, yielding 320 train-validation splits. Given the high dimensionality (12 685 features), the initial results were poor. To address this, we reduced the number of input features to the k highest correlated with the target, with k ranging from 1 to 10, based on the absolute value of the Spearman rank-order coefficient, calculated separately for each run on the training instances. The best classification accuracy, averaged over all 320 runs, equaled 81.88% and was achieved by RF(nt\_200) with BR = 0.4, using seven attributes.

Differences in classification accuracies between different BRs are sometimes marginal. Therefore, another experiment was conducted, this time focusing only on observations with undisputed labels. We assumed these are datasets for which the corresponding p-value in Table 1 is at most 0.01. A total of 24 observations met this condition. The remaining experimental configuration was the same as in the previous experiment. Leave-Two-Out Cross-Validation was performed on all possible 143 train-validation splits. The highest accuracy, 88.81%, was achieved by RF(nf\_all) with BR = 0.8, using only three features.

In both of the above experiments, the number of training instances was low: 36 and 24, respectively. We believe that a simple increase in these numbers may lead to further improvements in performance. Additionally, both datasets were well-balanced, with the majority class constituting 55.56% and 54.17%, respectively. Thus, we conclude that the proposed attributes, which enabled us to achieve accuracies of 81.88% and 88.81%, can be considered as effective descriptors of the analyzed problem.

### 6 CONCLUSIONS AND FUTURE WORK

In this paper, we analyze the BR hyperparameter in RF. To the best of our knowledge, this work is the first to shed light on what the optimal BR value depends on and to demonstrate that it is often greater than 1, thus exceeds the standard values within the (0,1] range. We also show that the optimal BR value is largely independent of the other hyperparameters of the RF. In fact, most RF configurations are highly correlated in terms of the BR curve, which makes the optimal BR value more of a property of the dataset.

Our main conclusion stating that BR > 1 often yields superior results and is worth considering contradicts the findings of the baseline reference paper (Martínez-Muñoz & Suárez, 2010). We identify two main reasons for this. First, the authors of (Martínez-Muñoz & Suárez, 2010) stopped their analysis at BR = 1.2 and did not explore higher values. Second, they most likely tested only one RF configuration. In fact, they did not provide any details regarding the RF hyperparameters, other than stating that the ensemble was composed of 200 unpruned CART trees (Breiman et al., 1984).

Prediction of the optimal BR value is a complex task, highly dependent on the local class structure. In this work, we propose to calculate  $k\_l$  statistics, which reflect the number of observations from the same class as the instances considered in their neighborhood and use them to calculate the correlations with the optimal BR value. While this approach works generally well, we believe that it cas still be improved through describing the local class structure in a different, possibly more precise way.

Considering nearest neighbors assumes that the analyzed observation is located at the center of the decision regions and that these regions form a hypercube, meaning all decision hyperplanes are equally distant from the analyzed sample. This is a simplification, as in real-world scenarios, the range of features corresponding to a decision subspace is usually unequal, and no point lies exactly at the center of all feature ranges. Therefore, analyzing an instance's neighborhood by sampling each feature range to reflect the different decision subspaces to which a particular instance may belong may be a viable approach.

Another research direction worth exploring is to extend  $k\_l$  to more detailed statistics that specify the number of neighbors from each individual class.  $k\_l$  can be interpreted as the number of instances for which, among the k nearest neighbors,  $k\_l$  observations belong to classes other than the one under consideration. We believe that the distribution of these classes may be useful in predicting the optimal BR. The more uniform the distribution (with no dominant class), the easier it is for an ambiguous example to outvote the correct class in a majority or soft voting scheme.

Finally, we examined three well-established ML libraries: scikit-learn, Weka, and H2O.ai. In all of them, values of the BR hyperparameter greater than one are disabled in their RF implementations. Based on our findings, we recommend that the developers of ML libraries consider making this feature available.

## REFERENCES

- Md Nasim Adnan. Improving the Random Forest Algorithm by Randomly Varying the Size of the Bootstrap Samples. In *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration, IEEE IRI 2014*, pp. 303–308, 08 2014.
- Simon Bernard, Laurent Heutte, and Sébastien Adam. Influence of Hyperparameters on Random Forest Accuracy. In *Multiple Classifier Systems*, pp. 171–180. Springer Berlin Heidelberg, 2009.
- Gérard Biau, Luc Devroye, and Gábor Lugosi. Consistency of Random Forests and Other Averaging Classifiers. *Journal of Machine Learning Research*, 9(66):2015–2033, 2008.
- Leo Breiman. Arcing Classifiers. *The Annals of Statistics*, 26(3):801–824, 1998.
- Leo Breiman. Random Forests. Machine Learning, 45:5–32, 2001.
- Leo Breiman, Jerome Friedman, Charles J. Stone, and Richard A. Olshen. Classification and Regression Trees. Taylor & Francis, 1984.
- Misha Denil, David Matheson, and Nando Freitas. Consistency of Online Random Forests. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pp. 1256–1264. PMLR, 2013.
- Duroux, Roxane and Scornet, Erwan. Impact of subsampling and tree depth on random forests. *ESAIM: PS*, 22:96–128, 2018.
- Benjamin A Goldstein, Eric C Polley, and Farren B. S. Briggs. Random Forests for Genetic Association Studies. *Statistical Applications in Genetics and Molecular Biology*, 10(1), 2011.
- Hemant Ishwaran and Udaya B. Kogalur. Consistency of random survival forests. *Statistics & Probability Letters*, 80(13):1056–1064, 2010.
- Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The UCI Machine Learning Repository. https://archive.ics.uci.edu, 2023. Accessed: 2024-07-11.
- Gonzalo Martínez-Muñoz and Alberto Suárez. Out-of-bag estimation of the optimal sample size in bagging. *Pattern Recognition*, 43(1):143–152, 2010.
- Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas. How Many Trees in a Random Forest? In *Machine Learning and Data Mining in Pattern Recognition*, pp. 154–168. Springer Berlin Heidelberg, 2012.
- Philipp Probst and Anne-Laure Boulesteix. To Tune or Not to Tune the Number of Trees in Random Forest. *Journal of Machine Learning Research*, 18(181):1–18, 2018.
- Philipp Probst, Marvin N. Wright, and Anne-Laure Boulesteix. Hyperparameters and tuning strategies for random forest. WIREs Data Mining and Knowledge Discovery, 9, 2019.
- Erwan Scornet. Tuning parameters in random forests. ESAIM: Procs, 60:144–162, 2017.
- Ningyuan Zhu, Chaoyang Zhu, Liang Zhou, Yayun Zhu, and Xiaojuan Zhang. Optimization of the Random Forest Hyperparameters for Power Industrial Control Systems Intrusion Detection Using an Improved Grid Search Algorithm. *Applied Sciences*, 12(20), 2022.

# **Appendices**

# A DATASETS

Table 3: Dataset characteristics. The subsequent columns refer to the dataset name, the number of numerical and binary features, the number of observations, and the count of classes. The first 31 datasets presented in the table come from the UCI Machine Learning Repository (Kelly et al., 2023). The next four are from Breiman (1998), and the last one is from Breiman et al. (1984).

Dataset	Numerical features	Binary features	Observations	Classes
Abalone	7	3	4172	23
Adult	6	85	48790	2
Arrhythmia	194	64	420	12
Audiology (Standardized)	0	89	171	18
Australian Credit Approval	6	32	690	2
Balance Scale	0	20	625	3
Breast Cancer Wisc. (Diag.)	30	0	569	2
Breast Cancer Wisc. (Orig.)	9	0	449	2
Congressional Voting Rec.	0	48	342	2
Echocardiogram	6	1	62	2
Ecoli	5	1	336	8
German Credit Data	6	53	1000	2
Glass Identification	9	0	213	6
Heart	7	13	270	2
Hepatitis	6	27	148	2
Horse Colic	7	140	368	2
Image Segmentation (Stat.)	18	0	2086	7
Ionosphere	32	1	350	2
Iris	4	0	149	3
Labor Relations	8	29	57	2
Liver Disorders	5	0	341	2
Optical Recognition (Digits)	61	0	1797	10
Parkinsons	22	0	195	2
Pima Indians Diabetes	8	0	768	2
Sonar, Mines vs. Rocks	60	0	208	2
Soybean (Large)	0	132	631	19
Tic-Tac-Toe Endgame	0	27	958	2
Thyroid Disease	5	0	215	3
Vehicle Silhouettes	18	0	845	4
Vowel Recognition	10	0	990	11
Wine	13	0	178	3
Ringnorm	20	0	300	2
Threenorm	20	0	300	2
Twonorm	20	0	300	2
Waveform	21	0	300	3
LED Display Domain	0	24	200	10

# B BOOTSTRAP RATE CURVES

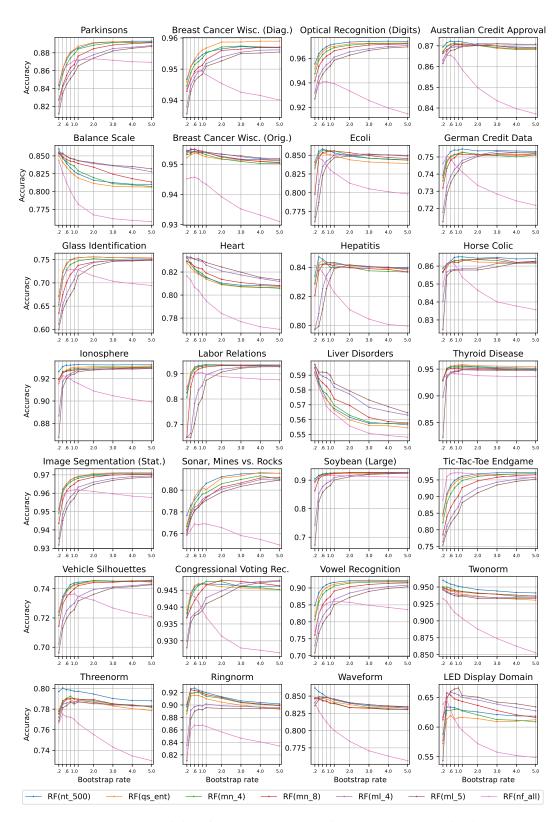


Figure 4: Characteristics of bootstrap rate curves for datasets not shown in Fig. 2.