Flexible Tree Ensembles with Data Adaptive Splitting Rules

by

Heshani A. T. Mendis

A Thesis submitted to the Faculty of Graduate and Postdoctoral Studies of
The University of Manitoba
in partial fulfilment of the requirements of the degree of

Master of Science

Department of Statistics
University of Manitoba
Winnipeg

Copyright © 2026 by Heshani A. T. Mendis

Abstract

Accurate split selection in regression trees and random forests is critical for predictive performance. In regression trees, splits are selected by minimizing within-node variances weighted by node proportions. These weights directly influence the impurity reduction criterion. We modify the standard weighting scheme by applying a power transformations to node proportions and examine how this modification influences split selection and predictive accuracy. Based on the proposed weighting framework, we develop four algorithmic variants of random forest, referred to as *FlexForest*, that introduce flexibility at the node, tree, and forest levels, along with a randomized weight-selection strategy. The proposed framework is evaluated via simulations across a range of data-generating mechanisms, including varying sample sizes, noise levels, functional relationships, and covariate dependence structures and distributions. Performance is assessed using relative prediction error against the default weighting scheme, and the results demonstrate considerable improvements, with the tree-level flexible method achieving improved or comparable predictive performance relative to the standard random forest in nearly all scenarios.

Acknowledgment Page

Foremost, I would like to express my sincere gratitude to my supervisors, Dr. Kevin McGregor and Dr. Elif Fidan Acar, for their unwavering guidance and support. Their encouragement has been invaluable, and I will always appreciate the knowledge gained during the past two years of collaboration.

I also extend my thanks to Dr. Carson Leung and Dr. Lei Ding for their valuable feedback. Their time and effort in reviewing this thesis are greatly appreciated.

Lastly, I would like to extend my thanks to everyone in the Department of Statistics who contributed to making my time in the master's program so meaningful. Whether through teaching or organizing student events, the experiences and memories I have gained will remain with me for years to come.

Dedication Page

To my family and loved ones, who stood by me through every challenge and celebrated each success.

Contents

Contents iii

List of Tables ix

List of Figures xi

1 Introduction 1

1.1 Background and Motivation 1

1.2 Decision Trees 3

1.3 Random Forests 6

1.4 Splitting Rules 7

1.5 Objectives and Contributions 10

1.6 Thesis Outline 11

2 Methodology **13**

2.1 Weighted Splitting Criterion 14

2.2 FlexForest 15

2.2.1 FlexForest with Forest-wise Variant (FF-F) 16

2.2.2 FlexForest with Tree-wise Variant (FF-T) 19

2.2.3 FlexForest with Optimal Node-wise Variant (FF-ON) 21

2.2.4 FlexForest with Random Node-wise Variant (FF-RN) 23

2.3 Selection of the Weighting Parameter γ 26

2.3.1 Learning-Based Selection 27

2.3.2 Independent Data-Based Selection 28

2.3.3 Comparing Learning-Based and Independent Data-Based Tuning 30

2.4 Summary 36

3 Simulation Study Design **37**

3.1 Covariate Generation 37

3.1.1 Covariate Distributions 38

3.1.2 Dependence Structures 41

3.2 Response Generation 42
3.2.1 Linear Model 44
3.2.2 Nonlinear Models 45
3.2.3 Heavily Nonlinear Model — Random Function Generator (RFG) 47
3.2.4 Illustrative Simulated Data Examples 49
3.3 Simulation Design and Procedure 50
3.4 Performance Evaluation 55
3.5 Simulation Results 57

4 Real Data Application **71**

4.1 Data Example 1: Concrete Compressive Strength 72
4.2 Data Example 2: Energy Efficiency 74
4.3 Data Example 3: Parkinson's Disease Telemonitoring 76
4.4 Evaluation Procedure 78
4.5 Results 79
4.5.1 Concrete Compressive Strength 80
4.5.2 Energy Efficiency 81
4.5.3 Parkinsons Telemonitoring 82
4.5.4 Summary Across Datasets 84

5 Conclusion and Future Work **87**

5.1 Summary 87

5.2 Limitations and Future Work 89

Bibliography **93**

Index **96**

A Proofs **97**

A.1 Comparison of Existing Child-Node Weighting Schemes 97

A.2 Recovery of Standard CART 101

A.3 Recovery of Standard Random Forest 102

B Weighting Parameter Tuning Extra Results **105**

B.1 Comparison of Training-Based and Validation-Based Tuning for Forest-wise Variant 105

B.2 Comparison of Training-Based and Validation-Based Tuning for Tree-wise Variant 107

C Simulation Results for Sample Size $n = 100$ **109**

C.1 Normal Linear 109

C.2 Normal Nonlinear 109

C.3 Exponential Linear 109

C.4 Exponential Linear 109

D Forest-Wise γ Selection **115**

D.1 Normal Linear 115

D.2 Normal Nonlinear 116

D.3 Exponential Linear 116

D.4 Exponential Nonlinear 117

D.5 Heavily Nonlinear 118

E Behavior of the Tree-wise Variant: Distribution of Selected γ Values **119**

List of Tables

1.1 Weighting rules for the child-node impurity criterion. 9

2.1 Average computation time (in mins) for each γ selection strategy. 35

3.1 Nonlinear regression functions considered in the simulation study. 46

3.2 Summary of simulation factors and levels for the full factorial design. 53

4.1 Prediction performance on the Concrete Compressive Strength dataset. 81

4.2 Prediction performance on the Energy Efficiency dataset . . . 82

4.3 Prediction performance on the Parkinsons Telemonitoring dataset 84

4.4 Average computation time (in seconds) for each method across the three real datasets 85

D.1 Selection frequency of the optimal γ value selected by the FF-F under the normal linear scenario 115

D.2 Selection frequency of the optimal γ value selected by the FF-F under the normal nonlinear scenario 116

D.3 Selection frequency of the optimal γ value selected by the FF-F under the exponential linear scenario 116

D.4 Selection frequency of the optimal γ value selected by the FF-F under the exponential nonlinear scenario 117

D.5 Selection frequency of the optimal γ value selected by the FF-F under the heavily nonlinear scenario 118

List of Figures

2.1 Flowchart of the FlexForest algorithm. 25

2.2 Data generation with normal covariates with a nonlinear relationship, forest-wise variant tuning comparison. 32

2.3 Data generation from normal covariates with a nonlinear relationship, tree-wise variant tuning comparison. 33

3.1 Scatter plot matrices illustrating the effect of covariate distribution and dependence structure 51

3.2 Scatter plot matrices illustrating the effect of functional form and noise level. 52

3.3 Scatter plot matrix of the heavily nonlinear scenario 53

3.4 RPE distributions for the normal linear scenario across different noise levels and response dependence structures. 60

3.5 RPE distributions for the normal nonlinear scenario across different noise levels and response dependence structures. 61

3.6 RPE distributions for the exponential linear scenario across different noise levels and response dependence structures. . . . 64

3.7 RPE distributions for the exponential nonlinear scenario across different noise levels and response dependence structures . . . 66

3.8 Heavily nonlinear relationship generated from the RGF function under different sample sizes. 68

4.1 Scatter plots of the concrete compressive strength against concrete mixture components predictors in the Concrete Compressive Strength dataset. 73

4.2 Scatter plots of the heating load against building characteristics predictors in the Energy Efficiency dataset. 75

4.3 Scatter plots of the motor UPDRS score against individual biomedical voice predictors in the Parkinson's disease dataset. 78

B.1 Data generation from exponential covariates with a nonlinear relationship, forest-wise variant tuning comparison. 106

B.2 Data generation from exponential covariates with a nonlinear relationship, tree-wise variant tuning comparison. 108

C.1 RPE distributions for the normal linear scenario across different noise levels and response dependence structures. 110

C.2 RPE distributions for the normal nonlinear scenario across different noise levels and response dependence structures. 111

C.3 RPE distributions for the exponential linear scenario across different noise levels and response dependence structures. . . . 112

C.4 RPE distributions for the exponential nonlinear scenario across different noise levels and response dependence structures. . . . 113

E.1 Distribution of selected γ values in the tree-wise variant for normally distributed covariates 120

E.2 Distribution of selected γ values in the tree-wise variant for exponentially distributed covariates 121

Chapter 1

Introduction

1.1 Background and Motivation

Random forests are among the most widely used ensemble learning methods for regression and classification, owing to their strong predictive performance, robustness to overfitting, and ability to handle nonlinear relationships and high-dimensional data (Breiman, 2001; Hastie, 2009; Boulesteix et al., 2012). Despite the emergence of many modern ensemble and deep learning methods, random forests continue to perform competitively across a wide range of scientific and industrial applications (Grinsztajn et al., 2022).

A key component of decision trees and random forests is the split selection criterion, which determines how the feature space is partitioned at each node during tree construction. In regression trees, splits are typically selected by minimizing the within-node variance, where the contributions of child nodes are weighted by their respective proportions of observations. This weighted

variance formulation plays a central role in guiding the tree construction process and ultimately affects predictive accuracy.

Although the standard weighting scheme based on node proportions has been shown to perform well across a variety of settings, existing studies suggest that different weighting strategies may be preferred depending on data characteristics such as signal strength and noise level (Ishwaran, 2015). Empirical evidence indicates that the choice of weighting scheme can influence the nature of selected splits; for instance, some schemes may favor splits in denser regions of the data, while others may be more sensitive to observations in sparser regions. However, these tendencies are data-dependent and needs further systematic investigation.

Despite the widespread adoption of this default proportion-based weighting scheme, relatively little attention has been given to how alternative weighting formulations influence the behavior of the splitting rule. Ishwaran (2015) suggests that different weighting schemes may favor signals of different strengths. However, existing weighting formulations are not directly comparable within a unified framework, and to the best of our knowledge, no systematic study has examined their comparative effects under controlled settings.

Several recent papers have addressed limitations of standard CART splitting from related but distinct angles. Ishwaran (2015) compared three discrete weighted splitting rules and showed that standard weighted variance splitting is most adaptive to both signal and noise. Zhang and Luo (2025) proposed

MinimaxSplit, which selects splits by minimizing the worst-case child risk, thereby avoiding highly unbalanced child nodes and mitigating the end-cut preference (ECP) problem. Su et al. (2024) introduced a smooth sigmoid surrogate as an alternative to greedy search, which also alleviates ECP by smoothing the split-point search. However, none of these approaches provides the unified framework identified as missing above, in particular, no existing work introduces a continuous tuning parameter that enables direct and systematic comparison of child-node weighting schemes under controlled settings.

Motivated by this, we propose a framework that introduces flexible weighting schemes into the split evaluation process. Specifically, we introduce a continuous tuning parameter that governs the adjusted weight assigned to each child node during split evaluation, such that the standard CART criterion is recovered as a special case. This leads to a new parameterization within the random forest framework, offering improved adaptability and potentially enhanced predictive performance.

The remainder of this chapter presents some background information on decision trees, random forests and split selection criteria.

1.2 Decision Trees

A decision tree is a nonparametric supervised learning method that approximates a target function by recursively partitioning the feature space into

disjoint sub-regions until each sub-region becomes homogeneous or nearly homogeneous with respect to the response variable (Breiman et al., 2017; Hastie, 2009). Decision trees are typically represented as a directed graph with a hierarchical structure, with the root node at the top and terminal nodes at the bottom.

A decision tree consists of three types of nodes. The root node is the topmost node, which contains all training observations and represents the first split applied to the full dataset. Internal nodes are intermediate nodes that result from splitting a parent node and are further split into child nodes based on the selected splitting variable and split point. Terminal nodes, also known as leaf nodes, are the final nodes of the tree from which predictions are made, no further splitting occurs at these nodes. Starting from the root node, the algorithm successively splits each node into two child nodes by selecting a splitting variable and a corresponding split point that optimally reduces node impurity. This process continues until a stopping criterion is met, such as a minimum node size or maximum tree depth, resulting in a set of terminal nodes each of which yields a prediction. For regression problems, the prediction at each terminal node is the mean of the response values of the observations contained within that node (Breiman et al., 2017).

Formally, let $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ denote a training sample where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip}) \in \mathbb{R}^p$ is a vector of predictors and y_i is the response variable. A decision tree partitions the covariate space into disjoint partitions $R_1, \ldots, R_M$, where

$R_m = \{i : \mathbf{x}_i \in \text{region } m\}$ denotes the set of training observations falling within region m, and $|R_m|$ is the size of the set. The tree produces a piecewise constant prediction of the form

$$\hat{f}(\mathbf{x}) = \sum_{m=1}^{M} \hat{c}_m \cdot \mathbf{I}(\mathbf{x} \in R_m) \, .$$

The value assigned to each region, $\hat{c}_m$, depends on the type of learning problem. In the case of regression, where $y_i \in \mathbb{R}$, $\hat{c}_m$ is typically defined as the mean of the responses for observations falling in region R_m, that is $\hat{c}_m = \frac{1}{|R_m|} \sum_{i:\mathbf{x}_i \in R_m} y_i$. In the case of classification, where $y_i \in \{1, \ldots, K\}$ represents class labels, $\hat{c}_m$ corresponds to the most frequent class label in region R_m, given by $\hat{c}_m = \arg\max_{k \in \{1,\ldots,K\}} \hat{p}_{mk}$, with $\hat{p}_{mk} = \frac{1}{|R_m|} \sum_{i:\mathbf{x}_i \in R_m} I(y_i = k)$ representing the proportion of observations in class k within that region. Thus, while the tree structure defines the partition of the predictor space, the predicted value within each region is determined by either the average response or the majority class, depending on whether the task is regression or classification (Breiman et al., 2017; Hastie, 2009).

Despite their interpretability and flexibility, individual decision trees suffer from high variance; small changes in the training data can produce substantially different tree structures (Hastie, 2009). This instability motivates the use of ensemble methods such as random forests, which aggregate predictions across

many trees to reduce variance and improve predictive performance (Breiman, 2001).

1.3 Random Forests

Random forests (Breiman, 2001) are one of the most widely used and successful machine learning methods for regression and classification. Despite the emergence of many modern ensemble and deep learning methods, random forests continue to perform competitively across a wide range of scientific and industrial applications (Grinsztajn et al., 2022), mainly owing to their strong predictive accuracy, robustness to overfitting, and ability to handle nonlinear relationships and high-dimensional data (Breiman, 2001; Hastie, 2009).

Random forests combine multiple randomized decision trees and aggregate their predictions by averaging (Breiman, 2001). Formally, let $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ denote a training sample where $\mathbf{x}_i \in \mathbb{R}^p$ is a vector of predictors and $y_i \in \mathbb{R}$ is the response. A random forest grows B randomized trees $\{T_1, \ldots, T_B\}$, where each tree T_b is constructed using a bootstrap sample $\mathcal{D}_b$ drawn with replacement from the original training data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$. Let $\hat{f}_b(\mathbf{x})$ denote the prediction of the b^{th} tree at a new point $\mathbf{x}$. The random forest prediction is then obtained by aggregating across all B trees

$$\hat{f}_{RF}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(\mathbf{x}),$$

for regression problems (Breiman, 2001; Biau and Scornet, 2016), where each $\hat{f}_b(\mathbf{x})$ is the mean response of the terminal node in tree T_b that contains $\mathbf{x}$. In classification problems, the final prediction is determined by majority voting across all trees (Breiman, 2001; Hastie, 2009)

$$\hat{f}_{RF}(\mathbf{x}) = \arg\max_k \sum_{b=1}^{B} \mathbf{I}\left(\hat{f}_b(\mathbf{x}) = k\right),$$

where k indexes the class labels. To reduce the correlation among trees, the algorithm introduces randomness in the selection of splitting variables, i.e., at each split, only a randomly selected subset of predictors is considered for splitting. This two-step randomization, decorrelates the trees and reduces variance, thereby improving predictive performance (Ishwaran, 2015; Hastie, 2009).

1.4 Splitting Rules

The trees in a random forest are typically constructed using the classification and regression tree (CART) splitting criterion (Breiman et al., 2017), which produces axis-aligned binary splits. That is, each split depends on the value of only one predictor variable. At each node of a tree, the optimal split is chosen by optimizing the CART criterion, which is based on the Gini impurity for classification problems or the mean squared error (equivalent to variance reduction) for regression problems (Breiman et al., 2017; Hastie, 2009).

Consider a node t containing n_t observations. In the random forest setting, at each node a random subset of h candidate predictors is selected from $\{1, \ldots, p\}$, where $h \leq p$ (Breiman, 2001). A candidate split s is defined by a splitting variable $j \in \{1, \ldots, h\}$ chosen from this random subset and a split point $c \in \mathbb{R}$, which partitions the observations in node t into two child nodes

$$t_L = \{i \in t : X_{ij} \leq c\} \quad \text{and} \quad t_R = \{i \in t : X_{ij} > c\},$$

with $n_L = |t_L|$ and $n_R = |t_R|$ observations respectively, where $n_L + n_R = n_t$.

For regression problems, the quality of a candidate split $s = (j, c)$ is evaluated using the variance reduction criterion. The impurity of a node t is measured by its within-node sample variance

$$\hat{\Delta}(s, t) = \hat{\sigma}^2(t) - \left[\frac{n_L}{n_t}\hat{\sigma}^2(t_L) + \frac{n_R}{n_t}\hat{\sigma}^2(t_R)\right],$$

where $\hat{\sigma}^2(t) = \dfrac{1}{n_t}\sum_{i \in t}(y_i - \overline{y}_t)^2$ is the sample variance of the response in node t, and $\overline{y}_t$ is the sample mean (Breiman et al., 2017; Ishwaran, 2015). Since $\hat{\sigma}^2(t)$ is constant for all candidate splits within a node, maximizing the variance reduction is equivalent to minimizing the weighted impurity of the resulting child nodes. The optimal split is selected as

$$(j^*, c^*) = \arg\min_{j,c}\left[\frac{n_L}{n_t}\hat{\sigma}^2(t_L) + \frac{n_R}{n_t}\hat{\sigma}^2(t_R)\right],$$

where the weights $\frac{n_L}{n_t}$ and $\frac{n_R}{n_t}$ are corresponding to the proportion of observations in the left and right child nodes respectively (Breiman et al., 2017; Ishwaran, 2015).

In addition to the standard CART weighting scheme, Ishwaran (2015) investigated two alternative weighting strategies that modify the contribution of child nodes in the variance-reduction criterion. These approaches differ only in the weighting assigned to the child-node impurities. The general form can be written as

$$D_w(s,t) = w_L\hat{\sigma}^2(t_L) + w_R\hat{\sigma}^2(t_R) \ ,$$

where w_L and w_R determine the weights of the left and right child nodes. The three weighting rules considered by Ishwaran (2015) are summarized in Table 1.1. This weighting scheme, proportional to the child node sizes, is the standard implicit assumption in CART-based methods, and forms the basis of the flexible weighting framework proposed in this work.

Table 1.1: Weighting rules for the child-node impurity criterion.

Weighting Rule	**Splitting Criterion**
Unweighted	$\hat{\sigma}^2(t_L) + \hat{\sigma}^2(t_R)$
Weighted	$\frac{n_L}{n_t}\hat{\sigma}^2(t_L) + \frac{n_R}{n_t}\hat{\sigma}^2(t_R)$
Heavy-weighted	$\left(\frac{n_L}{n_t}\right)^2\hat{\sigma}^2(t_L) + \left(\frac{n_R}{n_t}\right)^2\hat{\sigma}^2(t_R)$

1.5 Objectives and Contributions

The main objective of this thesis is to examine the effects of child-node weighting schemes on split selection in regression random forests and to develop methods that enhance predictive performance by increasing flexibility. The specific aims are as follows:

1. Understand how different child-node weighting strategies influence split selection and model accuracy under various settings.

2. Investigate flexible weighting strategies for split selection and their impact on predictive accuracy.

To address the limitations of the standard approach, this thesis proposes a flexible weighting framework that modifies the traditional variance reduction criterion. The framework introduces a tunable weighting mechanism that allows the contribution of child nodes to vary according to the data.

Based on this framework, four algorithmic variants are developed. These include methods that incorporate flexibility at the forest level, tree level, and node level, as well as a randomized variant. The node-wise approach allows weights to be selected at each split; however, due to the greedy nature of tree construction, this can lead to suboptimal decisions based on local information. To mitigate this limitation, a randomized strategy is introduced, where only one of the candidate weights is considered at each split. This approach is

inspired by the feature subsampling mechanism in standard random forests and is intended to improve robustness and generalization.

The methods are evaluated through extensive simulation studies under varying conditions, including different covariate distributions, noise levels, and dependence structures. In addition, real data applications are used to demonstrate practical effectiveness. The results show that the proposed approaches, particularly the tree-wise variant, achieve improved or comparable predictive performance relative to the standard random forest across a wide range of scenarios.

1.6 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 introduces the proposed methodology, including the modified weighted variance criterion and the four algorithmic variants, and explains why the weighting parameter is tuned using the training data rather than a validation set. Chapter 3 describes the simulation study design and presents the corresponding results. Chapter 4 evaluates the proposed methods using real-world data sets and discusses their performance. Finally, Chapter 5 summarizes the main findings, discusses the limitations of the study, and outlines directions for future research.

Chapter 2

Methodology

This chapter presents the proposed flexible weighting framework for random forest splitting. Building on the standard CART weighted variance criterion reviewed in Chapter 1, we introduce a continuous tuning parameter γ that governs the weight assigned to each child node during split evaluation, generalizing the standard proportion-based weighting scheme to a broader class of splitting rules. Based on this criterion, four algorithmic variants of a flexible random forest, collectively referred to as FlexForest, are developed, each incorporating flexibility at a different level of the ensemble: the forest level, the tree level, and the node level. The chapter is organized as follows. Section 2.1 introduces the proposed proposed weighted splitting criterion and establishes its connection to existing methods. Section 2.2 describes the four FlexForest variants and their corresponding algorithms. Section 2.3 discusses the strategy used to select the tuning parameter γ, including a comparison of learning-based and independent data-based tuning strategies.

2.1 Weighted Splitting Criterion

Since the existing weighting scheme does not allow meaningful comparisons across different weights, as heavy-weight always results in a smaller weighted variance (see Appendix A.1 for the full derivation), we modified the weighted variance formulation. Specifically, the formula was adjusted so that the left and right weights are normalized to sum to 1 for every value of γ. This normalization enables fair comparisons across different weighting schemes.

The Proposed Weighted Splitting Criterion

At a given node t, candidate splits are defined by (j, c, γ), where $j \in \{1, \ldots, h\}$ is the index of the splitting variable and $c \in S_j(t)$ is the split point in the set of admissible split points constructed from the observed values of variable j among samples in node t and the $\gamma \geq 0$ is the weighting parameter that determines the power applied to the node proportions. The function $m(j, c, \gamma)$ denotes the proposed modified child node variance criterion.

$$m(j,c,\gamma) = \underbrace{\left(\frac{\widehat{w}_{(t_L)}^{\gamma}}{\widehat{w}_{(t_L)}^{\gamma} + \widehat{w}_{(t_R)}^{\gamma}}\right)}_{\widehat{w}^{*}_{(t_L,\gamma)}} \hat{\sigma}^2(t_L) + \underbrace{\left(\frac{\widehat{w}_{(t_R)}^{\gamma}}{\widehat{w}_{(t_L)}^{\gamma} + \widehat{w}_{(t_R)}^{\gamma}}\right)}_{\widehat{w}^{*}_{(t_R,\gamma)}} \hat{\sigma}^2(t_R),$$

where the quantities $\widehat{w}_{(t_L)} = n_L/n$ and $\widehat{w}_{(t_R)} = n_R/n$ represent the proportions of observations in the left and right child nodes. Here t_L and t_R denote the left and right child nodes of the parent node t, respectively. Furthermore,

n, n_L, and n_R denote the number of observations in parent node, left child node, and right child node, respectively.

In addition,

$$\widehat{\sigma}(t_L) = \frac{1}{n_L} \sum_{i \in t_L} \left(y_i - \overline{y}_{t_L}\right)^2 \qquad \text{and} \quad \widehat{\sigma}(t_R) = \frac{1}{n_R} \sum_{i \in t_R} \left(y_i - \overline{y}_{t_R}\right)^2,$$

are sample variances of the left and right child nodes, where $\overline{y}_{t_L}$ and $\overline{y}_{t_R}$ denote the corresponding sample means. The quantities $\widehat{w}^*_{(t_L,\gamma)}$ and $\widehat{w}^*_{(t_R,\gamma)}$ denote the adjusted weights assigned to the left and right child nodes, respectively, and are defined such that

$$\widehat{w}^*_{(t_L,\gamma)} + \widehat{w}^*_{(t_R,\gamma)} = 1 \, .$$

This normalization ensures that the weights form a valid convex combination for every value of γ, allowing meaningful comparisons across different choices of the weighting parameter. Furthermore, when $\gamma = 1$, the proposed criterion coincides with the standard CART splitting criterion (see Appendix A.2 for the full derivation).

2.2 FlexForest

Based on our proposed splitting criterion, we introduce additional flexibility into the tree-building mechanism and, consequently, into the random forest algorithm. This results in a flexible extension of the conventional random forest framework. Specifically, we propose four variants, which differ according to the

level at which this flexibility is incorporated. These are forest-wise, tree-wise, random node-wise, and optimal node-wise. The forest-wise variant applies a single flexibility parameter to the entire forest; the tree-wise variant allows the parameter to vary across trees; while the node-wise variants introduce flexibility at the node level through either randomly selected or optimally chosen parameters. A summary diagram of the four variants is provided in Figure 2.1

2.2.1 FlexForest with Forest-wise Variant (FF-F)

In the FlexForest with forest-wise variant (Algorithm 1), flexibility is introduced at the forest level by modifying the variance-based splitting criterion through the weighting parameter γ. In essence, this approach extends the standard random forest by running it multiple times under different γ values for split evaluation, resulting in a collection of forests constructed using different γ values. Specifically, a common γ is applied across all trees and nodes in the forest, resulting in a forest-specific proposed weighted variance measure used for split selection. Notably, when the γ is set to 1, the proposed method reduces to the standard CART-based random forest (see Appendix A.3 for the full derivation).

To select the weighting parameter γ, we adopt a data-driven approach based on predictive performance. Specifically, we define a set Γ of candidate γ values corresponding to different child-node weighting schemes. For each

candidate γ, a random forest is trained in which the splitting criterion at every node is computed using the corresponding weight to evaluate the weighted variance and determine the optimal split variable and split point. This results in a collection of fitted forests, one for each γ. We then evaluate the predictive performance of each forest on the training data set and select the value of γ that yields the lowest prediction error. This selected γ is taken to be the most appropriate weighting parameter for the given dataset. Further discussion of the γ selection procedure is provided in Section (2.3).

Algorithm 1 FlexForest with Forest-wise Variant.

Require: Training data $\mathcal{D}_{\text{train}}$; set of weighting parameters $\Gamma = \{\gamma_1, \ldots, \gamma_K\}$; number of trees B; number of predictors p; maximum depth d_{max} .

for each $\gamma_k \in \Gamma$ **do**
  Initialize forest $\mathcal{F}^{(k)} = \emptyset$.
  for $b = 1$ to B **do**
    Draw a bootstrap sample $\mathcal{D}_b$ from $\mathcal{D}_{\text{train}}$.
    Grow a regression tree $T_b^{(k)}$.
    while current depth $\leq d_{max}$ **do**
      Randomly select h predictors from the p features.
      for each selected predictor X_j **do**
        Sort unique values of X_j .
        Generate candidate split points (midpoint approach) c .
        for each candidate split c **do**
          Compute γ_k-weighted MSE:

$$m(j, c, \gamma_k) \; .$$

        end for
      end for
      Select best split:

$$(j^*, c^*) = \arg\min_{j,c} m(j, c, \gamma_k) \; .$$

      Split node using X_j^* at c^* and add node to tree $T_b^{(k)}$.
    end while
    Add tree $T_b^{(k)}$ to forest $\mathcal{F}^{(k)}$.
  end for
  Compute training error:

$$E_k = \text{Error}(\mathcal{F}^{(k)}, \mathcal{D}_{\text{train}}) \; .$$

end for
Select optimal parameter:

$$\gamma^* = \arg\min_{\gamma_k \in \Gamma} E_k \; .$$

Output: Final forest $\mathcal{F}^{(\gamma^*)}$.
Prediction: For a new observation $\mathbf{x}_{new}$,

$$\hat{f}_{FF}(\mathbf{x}_{new}) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b^{(\gamma^*)}(\mathbf{x}_{new}) \; .$$

2.2.2 FlexForest with Tree-wise Variant (FF-T)

In the FlexForest with tree-wise variant (Algorithm 2), flexibility is introduced at the tree level. For each bootstrap sample, an appropriate weighting parameter is selected based on the predictive ability and assigned as a tree-specific weight. In this approach, each tree in the forest is assigned an individual γ weight, allowing it to adapt to the most suitable weighting scheme. Consequently, each tree in the forest is constructed using a single γ, but different trees may use different weights. This results in a forest composed of heterogeneous trees, each governed by its own weighting scheme, while maintaining a single-weight structure within each tree.

In the tree-wise extension of the proposed framework, the weighting parameter γ is selected adaptively for each bootstrap sample. For a given bootstrap, we consider a predefined set Γ of candidate γ values. For each candidate, a decision tree is constructed using the corresponding weighted splitting criterion, where the split variable and split point at each node are determined by minimizing the proposed weighted variance. This yields a collection of candidate trees for each bootstrap sample, each associated with a different γ. The predictive performance of these trees is then evaluated using bootstrap samples, and the tree achieving the lowest prediction error is selected. The selected tree, together with its associated γ, is retained for that bootstrap sample. This procedure is repeated independently across all bootstrap replicates, resulting in a forest in which each tree is constructed using an adaptively chosen weighting parameter.

Algorithm 2 FlexForest with Tree-wise Variant.

Require: Training data $\mathcal{D}_{\text{train}}$; set of weighting parameters $\Gamma = \{\gamma_1, \ldots, \gamma_K\}$; number of trees B; maximum depth $d_{\max}$; number of predictors p .

for $b = 1$ to B **do**
  Draw a bootstrap sample $\mathcal{D}_b$ from $\mathcal{D}_{\text{train}}$.
  for each $\gamma_k \in \Gamma$ **do**
    Grow a regression tree $T_b^{(k)}$ on $\mathcal{D}_b$.
    while current depth $\leq d_{\max}$ **do**
      Randomly select h predictors from the p features.
      for each selected predictor X_j **do**
        Sort unique values of X_j .
        Generate candidate split points (midpoint approach) c .
        for each candidate split c **do**
          Compute γ_k-weighted MSE:

$$m(j, c, \gamma_k) \ .$$

        end for
      end for
      Select optimal split:

$$(j^*, c^*) = \arg\min_{j,c} m(j, c, \gamma_k) \ .$$

      Split node using X_j^* at c^* and add node to tree $T_b^{(k)}$.
    end while
    Compute prediction error on bootstrap data:

$$E_{b,k} = Error(T_b^{(k)}, \mathcal{D}_b) \ .$$

  end for
  Select best parameter:

$$k^* = \arg\min_{k} E_{b,k} \ .$$

  Set $T_b = T_b^{(k^*)}$ and store in the forest .
end for
Compute and store the proportion of trees in the forest that utilize each weighting parameter γ_k .
Output: Forest $\{T_b\}_{b=1}^B$ and rule distribution .
Prediction: For a new observation $\mathbf{x}_{new}$,

$$\hat{f}_{FF}(\mathbf{x}_{new}) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(\mathbf{x}_{new}) \ .$$

2.2.3 FlexForest with Optimal Node-wise Variant (FF-ON)

In the FlexForest with optimal node-wise variant (Algorithm 3), flexibility is introduced at the node level, making it the most flexible as well as complex form of the proposed framework. At each split, a set of candidate weighting parameters is evaluated, and the optimal weight is selected based on the resulting partition of observations into child nodes. This selected value is then used as the split-specific weighting parameter for that node.

As a result, different splits within the same tree may use different weighting parameters, leading to trees with heterogeneous weighting structures. Consequently, the forest consists of a collection of such trees, each incorporating a node-specific weighting schemes, thereby providing a high degree of modeling flexibility.

In the node-wise extension of the proposed framework, the weighting parameter γ is selected adaptively at each split during tree construction. As in the standard CART procedure, a random subset of predictor variables is first selected at each node. For each selected predictor, candidate split points are generated using midpoints between sorted observed values. For every combination of predictor, split point, and candidate γ, the weighted variance is computed. The optimal split is then determined by jointly selecting the predictor, split point, and γ that minimize the weighted variance criterion. The selected γ is assigned to that node, and the splitting procedure is repeated

recursively at each node until the stopping criterion is reached. Repeating this process across bootstrap samples yields a random forest composed of trees in which the weighting parameter varies adaptively at each node.

Algorithm 3 FlexForest with Optimal Node-wise Variant.

Require: Training data D_{train}; set of weighting parameters $\Gamma = \{\gamma_1, \ldots, \gamma_K\}$; number of trees B; maximum tree depth $d_{\max}$; number of predictors p .

for $b = 1$ to B **do**
  Draw a bootstrap sample D_b from D_{train} .
  while current depth $\leq d_{\max}$ **do**
    for each node N to split **do**
      Randomly select h predictors from the p features .
      for each selected predictor X_j **do**
        Sort unique values of X_j .
        Generate candidate split points (midpoint approach) c .
        for each candidate split c **do**
          for each $\gamma_k \in \Gamma$ **do**
            Compute weighted variance $m(j, c, \gamma_k)$ at node N .
          end for
        end for
      end for
      Select optimal split for this node:

$$(j^*, c^*, \gamma^*) = \arg\min_{j,c,k} m(j, c, \gamma_k) \; .$$

      Split node N using predictor X_j^* at c^* and add node to tree $T_b^{(k)}$.
    end for
  end while
  Store the fitted tree T_b in the forest $\mathcal{F}$.
end for
Output: Forest $\mathcal{F} = \{T_b\}_{b=1}^{B}$.
Prediction: For a new observation $\mathbf{x}_{new}$,

$$\hat{f}_{FF}(\mathbf{x}_{new}) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(\mathbf{x}_{new}) \; .$$

2.2.4 FlexForest with Random Node-wise Variant (FF-RN)

In the FlexForest with random node-wise variant (Algorithm 4) flexibility is introduced at the node level. This variant further increases the randomness of the random forest by randomly selecting the weighting parameter at each split. Specifically, at every split, one weighting parameter is randomly chosen from the set of candidate weighting parameters and is used to evaluate the splitting criterion. Consequently, different splits within the same tree may employ different weighting parameters, resulting in trees with heterogeneous weighting structures.

This variant is closely related to the optimal node-wise variant, as both introduce flexibility at the node level and allow different splits within a tree to use different weighting parameters. The key difference is that, in the FF-RN, the weighting parameter is selected randomly, whereas in the optimal node-wise variant, it is chosen by optimizing the proposed weighted variance criterion.

Algorithm 4 FlexForest with Random Node-wise Variant.

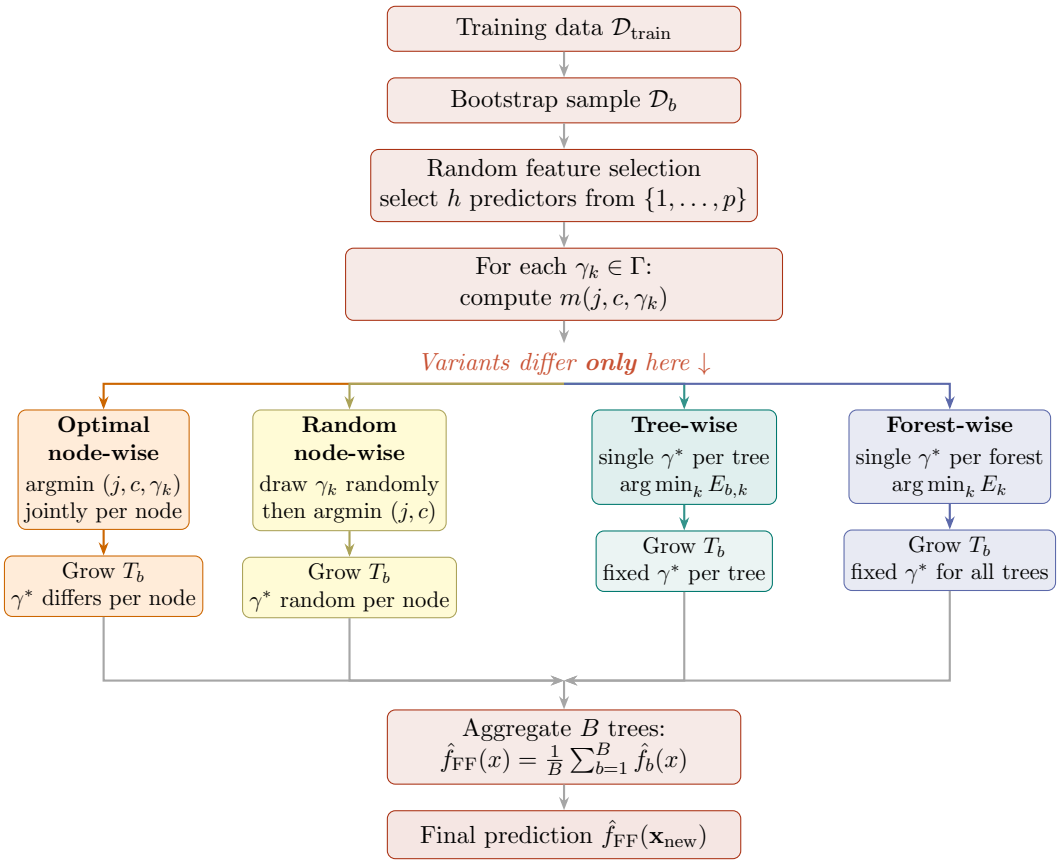
Require: Training data D_{train}; set of weighting parameters $\Gamma = \{\gamma_1, \ldots, \gamma_K\}$; number of trees B; maximum tree depth $d_{\max}$; number of predictors p .

for $b = 1$ to B **do**
  Draw a bootstrap sample D_b from D_{train} .
  while current depth $\leq d_{\max}$ **do**
    for each node N to split **do**
      Randomly select h predictors from the p features.
      Randomly select a weight γ^* from the Γ .
      for each selected predictor X_j **do**
        Sort unique values of X_j .
        Generate candidate split points (midpoint approach) c .
        for each candidate split c **do**
          Compute weighted variance $m(j, c, \gamma^*)$ at node N .
        end for
      end for
      Select optimal split for this node:

$$(j^*, c^*, \gamma^*) = \arg\min_{j,c} m(j, c, \gamma^*) \; .$$

      Split node N using predictor X_j^* at c^* and add node to tree $T_b^{(k)}$.
    end for
  end while
  Store the fitted tree T_b in the forest $\mathcal{F}$.
end for
Output: Forest $\mathcal{F} = \{T_b\}_{b=1}^{B}$.
Prediction: For a new observation $\mathbf{x}_{new}$,

$$\hat{f}_{FF}(\mathbf{x}_{new}) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(\mathbf{x}_{new}) \; .$$



Note: the standard random forest is a special case of the forest-wise variant when $\gamma = 1$.

Figure 2.1: Flowchart of the FlexForest algorithm. All four variants share the same training data, bootstrap sampling, random feature selection, and proposed weighted splitting criterion. The variants differ only in where the optimal γ^* is selected: at the node, tree, or forest level.

2.3 Selection of the Weighting Parameter γ

A key practical consideration in the FlexForest framework is the strategy used to select the weighting parameter γ from the pre-specified candidate set $\Gamma = \{\gamma_1, \ldots, \gamma_k\}$. As described in Section 2.2, the forest-wise and tree-wise variants select γ by minimizing prediction error computed on the same data used to fit the model, the training data for the forest-wise variant and the bootstrap sample for the tree-wise variant. A natural concern is whether this approach leads to overfitting, since the same data are used for both model fitting and parameter selection.

An alternative approach would be to use an independent dataset for γ selection, by ensuring that the tuning criterion is evaluated on data not seen during model fitting. Although this is conceptually preferable, it requires a portion of the available data to be used for training.

We investigate whether the training-based and bootstrap-based selection strategies used in the forest-wise and tree-wise variants, respectively, lead to overfitting relative to validation-based approaches. The comparison is carried out empirically using small simulation studies under nonlinear settings. Full details of the simulation design are provided in Chapter 3; here we briefly describe the settings used for this specific comparison.

We considered $p = 10$ predictors under a nonlinear functional relationship between the predictors and the response, as this represents the setting most favorable to the proposed FlexForest framework and therefore the setting in

which overfitting from training-based tuning is most likely to manifest. Two covariate distributions were considered, multivariate normal and exponential and two dependence structures independent and correlated covariates. Two noise levels were considered: low noise ($\sigma = 1$) and high noise ($\sigma = 4$). The nonlinear relationship between the predictors and the response was specified using a combination of transformations with varying coefficients; the full specification is given in Section 3.2. The sample size was fixed at $n = 500$. Each random forest consisted of 500 trees with a maximum tree depth of 4, and 4 predictors were randomly selected as candidate splitting variables at each node. The weighting parameter was selected from the candidate set $\Gamma = \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$. See Section 2.3.3 for the results.

2.3.1 Learning-Based Selection

In the forest-wise variant, a single γ is selected for the entire forest by minimizing the training prediction error. For each candidate $\gamma_k \in \Gamma$, a forest $\mathcal{F}^{(k)}$ is grown using the γ_k-weighted splitting criterion. The training error is computed as

$$E_k = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \widehat{\mathcal{F}}^{(k)}(\mathbf{x}_i) \right)^2 ,$$

where $\widehat{\mathcal{F}}^{(k)}(\mathbf{x}_i)$ is the prediction of forest $\mathcal{F}^{(k)}$ for observation i. The optimal parameter is then

$$\gamma^* = \arg\min_{\gamma_k \in \Gamma} E_k \, .$$

In the tree-wise variant, a single γ_b^* is selected per tree T_b by minimizing the prediction error on the bootstrap sample $\mathcal{D}_b$

$$E_{b,k} = \frac{1}{|\mathcal{D}_b|} \sum_{i \in \mathcal{D}_b} \left(y_i - \hat{f}_b^{(k)}(\mathbf{x}_i) \right)^2,$$

where $\hat{f}_b^{(k)}$ is the prediction of tree $T_b^{(k)}$ grown with parameter γ_k. The optimal parameter for tree T_b is

$$\gamma_b^* = \arg\min_{\gamma_k \in \Gamma} E_{b,k} \ .$$

2.3.2 Independent Data-Based Selection

As a comparison, we consider two independent data-based approaches, where an independent validation set $\mathcal{D}_{\text{valid}}$ or an out-of-bag (OOB) dataset $\mathcal{D}_{\text{OOB}}$ is used for tuning. In the validation-based approach, $\mathcal{D}_{\text{valid}}$ is held out from the original training dataset $\mathcal{D}_{\text{train}}$ and is not used during model fitting. In contrast, the OOB approach uses the observations that are not included in the bootstrap sample used to construct each tree.

In random forests, each tree is fitted using a bootstrap sample drawn from $\mathcal{D}_{\text{train}}$. The remaining observations, which are not selected in the bootstrap sample, form the OOB sample and can be used as an independent dataset for evaluating the fitted tree. Approximately one-third of the observations are expected to be excluded from each bootstrap sample and therefore be available as observations (Smith et al., 2024). The OOB-based tuning approach

is considered only for the tree-wise variant, as OOB observations are naturally associated with individual trees.

For the forest-wise variant, forests $\mathcal{F}^{(k)}$ are grown on $\mathcal{D}_{\text{train}}$ and the validation error is computed on $\mathcal{D}_{\text{valid}}$

$$E_k^{\text{valid}} = \frac{1}{|\mathcal{D}_{\text{valid}}|} \sum_{i \in \mathcal{D}_{\text{valid}}} \left(y_i - \widehat{\mathcal{F}}^{(k)}(\mathbf{x}_i) \right)^2 .$$

The optimal parameter is selected as

$$\gamma^* = \arg\min_{\gamma_k \in \Gamma} E_k^{\text{valid}} .$$

For the tree-wise variant, two independent datasets are considered: the validation set and the OOB sample. The only difference here is that we have two independent datasets, OOB and the validation dataset. Each tree $T_b^{(k)}$ is grown on the bootstrap sample $\mathcal{D}_b$ and evaluated on independent dataset $\mathcal{D}_{\text{ind}}$, where $\mathcal{D}_{\text{ind}} \in \{\mathcal{D}_{\text{OOB}}, \mathcal{D}_{\text{valid}}\}$.

$$E_{b,k}^{\text{ind}} = \frac{1}{|\mathcal{D}_{\text{ind}}|} \sum_{i \in \mathcal{D}_{\text{ind}}} \left(y_i - \hat{f}_b^{(k)}(\mathbf{x}_i) \right)^2 .$$

The optimal parameter for tree T_b is

$$\gamma_b^* = \arg\min_{\gamma_k \in \Gamma} E_{b,k}^{ind} .$$

Independent data-based tuning is not considered for the node-wise variant. Unlike the forest-wise and tree-wise approaches, node-wise optimization simultaneously selects the splitting variable, splitting point, and γ during tree construction. Since the validation mean squared error is defined only after a complete tree or forest has been grown, it does not provide a meaningful objective for evaluating candidate splits at an individual node. Consequently, the node-wise approach selects these quantities by minimizing the weighted variance criterion described in Sections 2.2.4 and 2.2.3, rather than using an independent validation dataset.

2.3.3 Comparing Learning-Based and Independent Data-Based Tuning

To assess whether training-based selection leads to overfitting relative to validation-based selection, a simulation study was conducted under nonlinear settings as described in earlier.

For this comparison, the available data were split into training (70%), validation (15%), and test (15%) sets. The training-based approach used the training set alone for both model fitting and γ selection, while the validation-based approach used the validation set for γ selection. Both approaches were evaluated on the same test set, and performance was compared using the Mean Squared Error (MSE), over 100 replications. Only the results for the nonlinear setting with normally distributed covariates are presented in the main text.

The corresponding results for the exponential nonlinear simulation settings are provided in the Appendix B. The comparison was assessed primarily through visualization: for each pair of tuning approaches, the test errors were plotted against each other, with the x-axis representing the learning-based tuning error and the y-axis representing the independent data-based tuning error, together with the 45° diagonal line. Points lying on the diagonal indicate that the two tuning approaches produced similar test error for that replication. In contrast, points falling systematically below the diagonal would indicate that the learning-based approach achieved a higher test error than the independent data-based approach on the same replication. Importantly, a systematic pattern of points falling below the diagonal can be a signature of overfitting. Thus, visual evidence of points lying below the diagonal provides a hint of potential overfitting by the learning-based tuning method.

For the forest-wise variant, the test errors obtained under both training-based and validation-based approached were plotted against each other across replications (Figure 2.2), and the correlation between them was computed. A strong positive correlation of approximately 0.98 was observed, and most points fall close to the diagonal line with no systematic pattern of points falling below it. This indicates that the test errors under the two tuning approaches tend to vary similarly across simulation replications.

Similarly, for the tree-wise variant (Figure 2.3) , the test errors obtained under bootstrap-based tuning were compared against those obtained under the

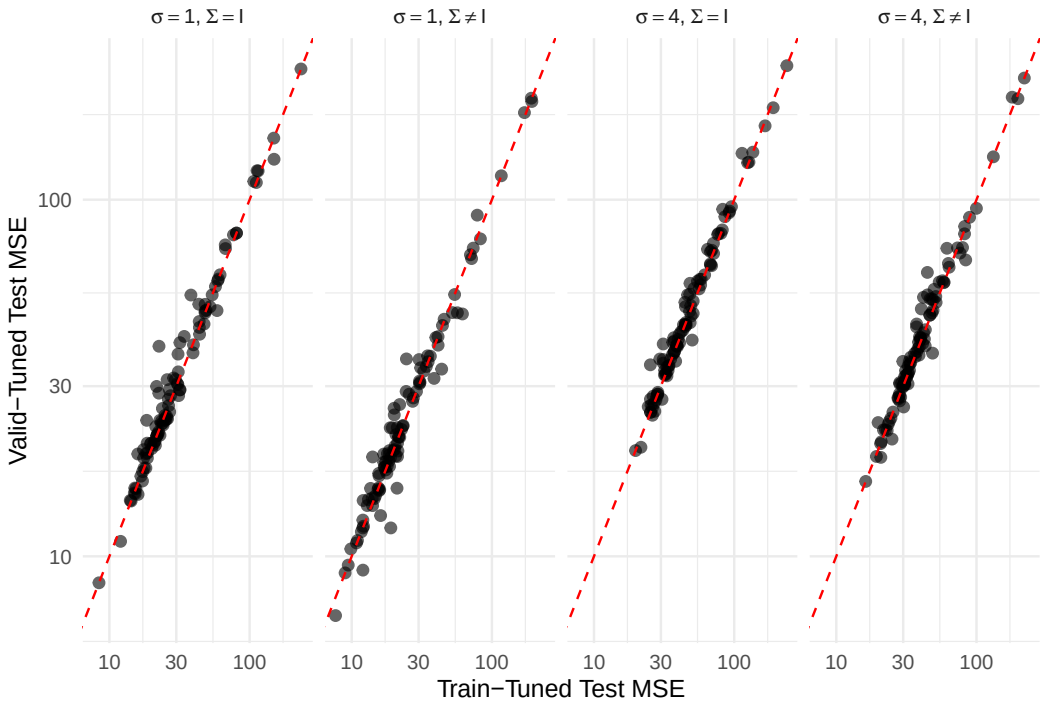


Figure 2.2: Data generation with normal covariates with a nonlinear relationship, forest-wise variant tuning comparison.

two independent data based approaches (OOB-based and validation-based), yielding a strong positive correlation of approximately 0.96. The bootstrap-tuned and OOB-tuned test errors are closely aligned along the diagonal across replications. The comparison between bootstrap-tuned and validation-tuned test errors shows a similar overall pattern, though some points lie above the diagonal, indicating higher test MSE under validation-based tuning for those replications.

This is consistent with learning-based tuning tracking selection strategies based on independent data reasonably closely. These results provide an

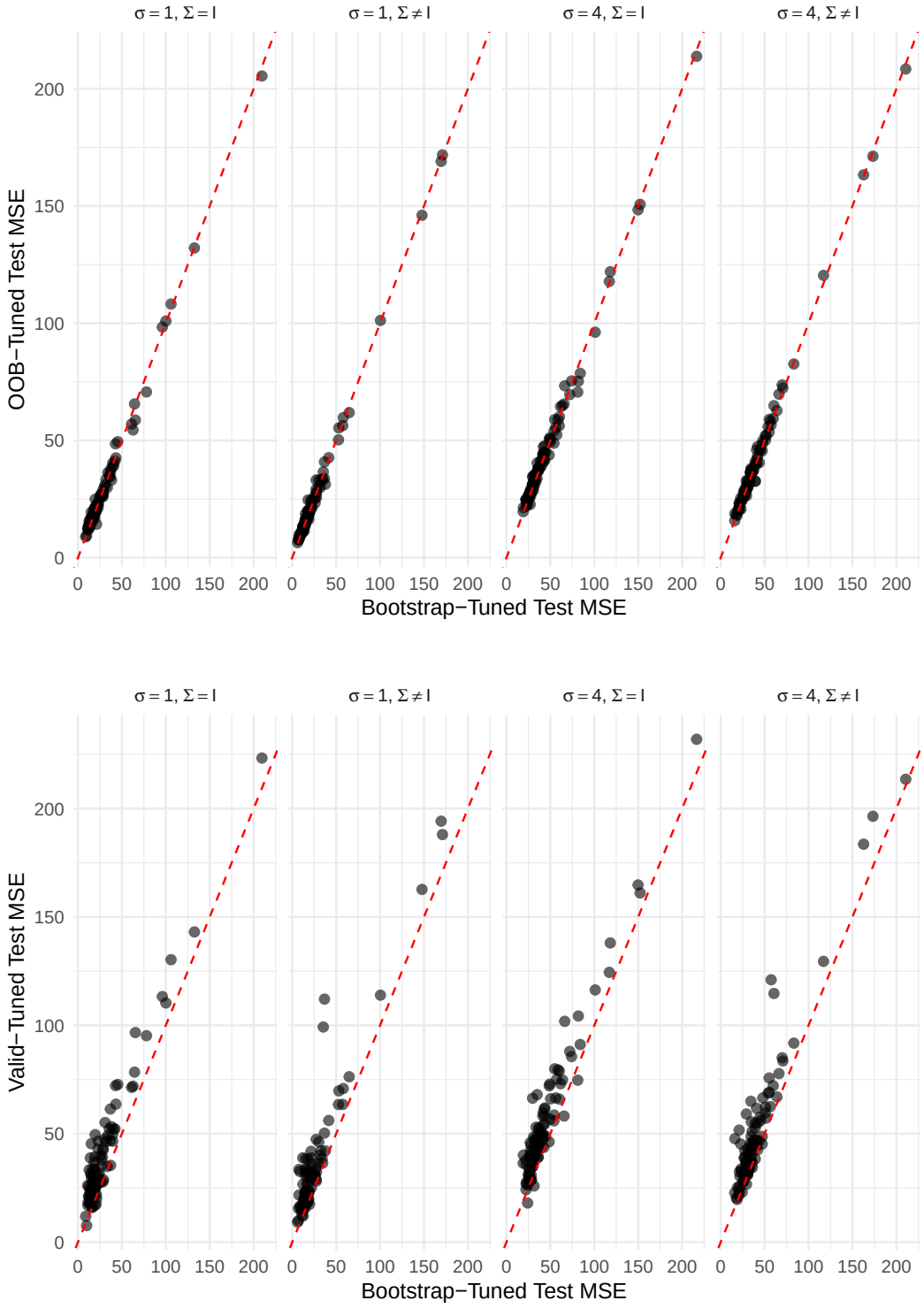


Figure 2.3: Data generation from normal covariates with a nonlinear relationship, tree-wise variant tuning comparison.

important practical benefit. Since training-based and bootstrap-based (learning-based) γ selection do not require holding out a separate validation set, a larger proportion of the available data can be allocated to model fitting, with the remainder reserved for testing, rather than being split further for validation purposes. This is particularly advantageous when the available sample size is limited.

However, learning-based selection requires additional computational cost relative to validation-based selection, since the dataset used to evaluate prediction error for γ selection, the full training set for the forest-wise variant and the bootstrap sample for the tree-wise variant is substantially larger than a held-out validation set, which typically comprises a small proportion of the available data. Consequently, computing predictions and evaluating the γ selection criterion takes longer under the learning-based approach than under the validation-based approach. Table 2.1 summarizes the computational time associated with each γ selection strategy, evaluated under normal independently distributed covariates with a nonlinear predictor–response relationship and low noise ($\sigma = 1$). The last column expresses each method's computation time as a multiple of standard random forest (RF) time.

These results show that training-based and validation-based γ selection yield comparable predictive performance on the held-out test set for both variants, with no systematic upward bias under training-based selection. This is consistent with the earlier pattern of high correlation and no systematic

Table 2.1: Average computation time (in minutes) for each γ-selection strategy, averaged over 100 simulation replications. Results are based on datasets with sample size $n = 500$, where each strategy evaluates the candidate set $\Gamma = \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$.

Variant	γ **Selection**	**Mean Time (mins)**	**Relative to RF**
Forest-wise	Training-based	51.27	6.29
	Validation-based	39.80	4.88
Tree-wise	Bootstrap-based	36.82	4.52
	OOB-based	36.02	4.42
	Validation-based	35.60	4.37

downward bias, and does not suggest that the improvement in predictive performance over the standard random forest is an artifact of overfitting to the training or bootstrap data. Moreover, training-based and bootstrap-based selection avoid the need to allocate data to a separate validation set.

Based on these findings, training-based selection for the forest-wise variant and bootstrap-based selection for the tree-wise variant are used throughout the remainder of this thesis. For the node-wise variants, γ is determined during the node-splitting process. In the optimal node-wise variant γ, the splitting variable, and the splitting point are selected jointly by minimizing the proposed weighted variance criterion. In the random node-wise variant, γ is selected randomly from the candidate set before determining the optimal split. Validation-based γ selection are not well-defined at the node level (see Section 2.3.2).

2.4 Summary

This chapter introduced the proposed flexible weighted splitting criterion and the four FlexForest algorithmic variants: optimal node-wise, random node-wise, tree-wise, and forest-wise, that incorporate this flexibility at different levels of the random forest. The strategy used to select the weighting parameter γ was also described and empirically validated. The performance of these methods is evaluated through simulation studies in Chapter 3 and demonstrated in real data applications in Chapter 4.

Chapter 3

Simulation Study Design

This chapter presents a simulation study to evaluate the performance of the proposed methods under a variety of data-generating scenarios. The simulation framework investigates the impact of several factors, including the number of predictors, sample size, covariate distribution, noise level, dependence structure, and the functional form of the relationship between the response and predictors. The chapter first describes the data-generating mechanisms and simulation settings, followed by the evaluation criteria and the simulation results.

3.1 Covariate Generation

Let $\mathbf{X} = (X_1, \ldots, X_p)^\top$ denote a p-dimensional vector of covariates, where $p \in \{3, 5, 10\}$. The simulation study considered two marginal distributions for the covariates and two dependence structures, resulting in a fully crossed design.

3.1.1 Covariate Distributions

In this simulation study, two types of covariate distributions were considered to represent qualitatively different data characteristics that may influence the behavior of the splitting criterion. The normal distribution was selected as a benchmark symmetric distribution, while the exponential distribution was chosen to represent a positively skewed distribution with non-negative support (Ishwaran, 2015; Schmid et al., 2023). Together, these two distributions allow evaluation of the proposed FlexForest variants under substantially different distributional characteristics and provide insight into whether the performance improvements achieved by flexible child-node weighting are robust to departures from normality. For both distributional settings, the covariate vector $\mathbf{X} = (X_1, \ldots, X_p)^\top$ was generated with mean vector $\boldsymbol{\mu} = \mathbf{1}$, where $\mathbf{1} = (1, 1, \ldots, 1)^\top \in \mathbb{R}^p$ denotes the p-dimensional vector of ones. A mean of one was chosen rather than zero for the following reason: for the exponential distribution, the rate parameter λ is the reciprocal of the mean, $\lambda_j = 1/\mu_j$. If $\mu_j = 0$, the rate parameter would be undefined ($\lambda_j \to \infty$), and the exponential distribution would not be well-defined. Using $\mu_j = 1$ ensures $\lambda_j = 1$ for all j, yielding a standard exponential distribution with mean one for each covariate, and allows the same mean vectors to be used consistently across both distributional settings.

The two marginal distributions considered are as follows:

1. **Normally-distribution covariates**

This provides a natural baseline against which the behavior of the proposed FlexForest variants can be assessed under idealized, symmetric conditions. Covariates were generated as:

$$\mathbf{X} \sim \mathcal{N}_p(\mathbf{1}\ , \mathbf{\Sigma}),$$

where $\mathbf{\Sigma}$ is the covariance matrix governing the dependence structure among covariates. Two choices of $\mathbf{\Sigma}$ are considered, corresponding to independent and correlated covariate structures, as described in Section 3.1.2.

2. **Exponentially-distributed covariates**

   For the exponential settings, each covariate X_j follows a marginal exponential distribution:

$$X_j \sim \text{Exp}(\lambda_j), \quad \lambda_j = \frac{1}{\mu_j} = 1, \quad j = 1, \ldots, p,$$

   where $\lambda_j = 1$ is the rate parameter, chosen so that each X_j has mean $\mu_j = 1$.

   While simulating independent exponential variables is straightforward, introducing a specified dependence structure among them is not. Unlike the multivariate normal distribution, there is no standard multivariate extension of the exponential distribution that directly adapts an arbitrary correlation matrix $\mathbf{\Sigma}$. A Gaussian copula approach was therefore used to introduce the desired dependence structure among the exponential

marginals while preserving their specified marginal distribution (Joe, 2014).

The construction proceeds in three steps. First, multivariate normal variables $\mathbf{X}^{(N)} \sim \mathcal{N}_p(\mathbf{1}_p, \boldsymbol{\Sigma})$ were generated, where $\boldsymbol{\Sigma}$ specifies the desired correlation structure. Second, these are transformed to uniform marginals via the probability integral transform, which maps any continuous random variable to uniform distribution on $[0, 1]$:

$$U_j = \Phi\left(\frac{X_j^{(N)} - \mu_j}{\sigma_j}\right) \sim \text{Unif}(0, 1),$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function and σ_j is the standard deviation of $X_j^{(N)}$. Since the transformation is monotone and applied marginally, the dependence structure introduced by $\boldsymbol{\Sigma}$ is preserved in the uniform variables $U_1, \ldots, U_p$. Third, the correlated uniform variables are transformed to exponential distribution:

$$X_j = F^{-1}_{\text{Exp}(\lambda_j)}(U_j) = -\frac{1}{\lambda_j}\log(1 - U_j), \quad \lambda_j = 1.$$

By the probability integral transform, each X_j has the desired exponential marginal distribution, while the dependence structure among $X_1, \ldots, X_p$ is inherited from the Gaussian copula induced by $\boldsymbol{\Sigma}$. The same covariance matrix $\boldsymbol{\Sigma}$ is used for both the normal and exponential settings, allowing

the effect of the marginal distribution to be assessed independently of the dependence structure.

3.1.2 Dependence Structures

To gauge the effect on dependence structure in the underlying covariates on the performance of the FlexForest variants, two covariate dependence structures were considered. The same covariance matrix $\mathbf{\Sigma}$ were used for both marginal distributions described in Section 3.1.1, allowing the effect of dependence to be evaluated independently of choice of marginal distribution.

(i) Independent structure We start with the simplest case, an independent covariance structure, where

$$\mathbf{\Sigma} = \mathbf{I}_p.$$

(ii) Dependent structure Next, we investigate different strengths of dependence among the covariates. For all settings, the variances of all variables were fixed at one. The covariance structure depended on the dimension p. For both $p = 3$ and $p = 5$, a single correlated pair was introduced, with $\text{Cov}(X_1, X_2) = 0.85$, and all other covariances equal to zero. For $p = 10$, three correlated pairs were introduced: $\text{Cov}(X_1, X_2) = 0.85$, $\text{Cov}(X_3, X_4) = 0.75$, and $\text{Cov}(X_5, X_6) = 0.90$, with all other covariances set to zero. The correlation values 0.75, 0.85, and 0.90 were chosen to represent strong to very strong dependence.

3.2 Response Generation

This section describes how the response variable y was generated for each simulation scenario. The primary goal of the response generation scheme is to evaluate the performance of the proposed FlexForest variants across a range of conditions that are known to influence the behavior of tree-based methods, including the functional form of the predictor-response relationship and the signal-to-noise ratio, while maintaining heterogeneous predictor effect structure (Breiman, 2001; Ishwaran, 2015; Schmid et al., 2023). Specifically, both linear and nonlinear functional relationships between the covariates and the response are considered to assess whether the proposed flexible weighting framework offers advantages over the standard random forest under different structural assumptions.

In both the linear and nonlinear settings, the response vector is generated according to the model

$$\mathbf{Y} = f(\mathbf{X}) + \boldsymbol{\varepsilon},$$

where $f(\mathbf{X})$ denotes the systematic component describing the functional relationship between the predictors and the response, and $\boldsymbol{\varepsilon} = \{\varepsilon_1, \ldots, \varepsilon_n\}^\top$ represent the random error vector. The errors are assumed to be independent and identically distributed with

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2), \qquad i = 1, \ldots, n,$$

where $\sigma > 0$ controls the spread of the noise around the true function $f(\mathbf{X})$.

The systematic component $f(\mathbf{X})$ depends on a regression coefficient vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$, which was held fixed across all simulation scenarios to ensure a consistent underlying signal. To reflect heterogeneous predictor importance, the predictors were partitioned into three approximately equal groups corresponding to strong, moderate, and weak effects. The associated regression coefficients were assigned values c_s, c_m, and c_w, respectively, where $c_s > c_m > c_w \geq 0$. Consequently, predictors with strong effects received the largest coefficients, those with moderate effects received intermediate coefficients, and those with weak effects received the smallest coefficients. The specific values of the regression coefficients c_s, c_m, and c_w were determined through an iterative visualization process. For each candidate set of coefficient values, the resulting simulated datasets were plotted and visually inspected to verify that the generated data exhibited the intended signal characteristics. This heterogeneous coefficient structure reflects realistic data-generating mechanisms in which predictors contribute unequally to the response.

To investigate the influence of the signal-to-noise ratio (SNR), the regression coefficients were kept fixed while the error standard deviation was varied. Specifically, two noise levels, $\sigma \in \{1, 4\}$, were considered, representing low- and high-noise settings, respectively. Because the regression coefficients and the underlying functional relationship were held fixed across all simulation scenarios, the signal component remained unchanged. Consequently, varying σ altered only the SNR. This design allows the performance of the proposed

FlexForest variants to be evaluated under both favorable and more challenging prediction settings, while ensuring that any observed differences in performance can be attributable to changes in the noise level rather than changes in the underlying signal.

The following subsections describe the linear and nonlinear functional forms of $f(\mathbf{X})$ in detail.

3.2.1 Linear Model

First we consider the simplest simulation setting, which assumes a typical multiple linear regression model. This scenario allows us to investigate the performance of the FlexForest under a simple linear model and verify that the methods do not overfit.

For the linear setting, the response variable was generated according to

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon$$

The predictor matrix $\mathbf{X}$ is an $n \times p$ matrix of covariates generated according to one of the simulation schemes described in Section 3.1. Depending on the simulation scenario, the covariates were generated from either multivariate normal or multivariate exponential distributions under the specified covariance structures, with $p \in \{3, 5, 10\}$.

3.2.2 Nonlinear Models

Next, we investigate a model in which the covariate effects are nonlinear. This scenario allows us to evaluate the performance of the proposed FlexForest variants under a more complex data-generating mechanism involving nonlinear effects and interactions among the predictors.

For the nonlinear setting, the response variable was generated as

$$\mathbf{Y} = f(\mathbf{X}) + \varepsilon,$$

where $f(\mathbf{X})$ represents the nonlinear relationship between the response and predictors. Three nonlinear regression functions with increasing complexity were considered by varying the number of predictors $p \in \{3, 5, 10\}$. The nonlinear functions used in the simulation study are summarized in Table 3.1.

The nonlinear regression functions were selected to represent a range of predictor-response relationships of increasing complexity. Rather than relying on a single form of nonlinearity, the functions incorporate smooth nonlinear transformations, including sine, cosine, and exponential functions, as well as interaction effects between predictors. These components were chosen because they capture qualitatively distinct nonlinear behaviors commonly encountered in practice and provide a more comprehensive assessment of the proposed FlexForest variants. Furthermore, the complexity of the regression function increases with the number of predictors, allowing the proposed methods to be evaluated under progressively more challenging prediction settings.

Table 3.1: Nonlinear regression functions considered in the simulation study.

Number of predictors	**Regression function**
$p = 3$	$f(\mathbf{X}) = \beta_1 \sin(\pi X_1) + \beta_2 X_2 + \beta_3 X_3.$
$p = 5$	$f(\mathbf{X}) = \beta_1 \sin(\pi X_1) + \beta_2 \cos(\pi X_2) + \beta_3 X_3 + \beta_4 X_1 X_4 + \beta_5 X_5.$
$p = 10$	$f(\mathbf{X}) = \beta_1 \sin(\pi X_1) + \beta_2 \exp(X_2) + \sum_{j=3}^{5} \beta_j X_j + \beta_6 X_1 X_2 + \beta_7 X_4 X_5 + \beta_8 \sin(\pi X_6) + \sum_{j=7}^{8} \beta_{j+2} X_j + \beta_{11} \cos(\pi X_9) + \beta_{12} X_{10}.$

Each model term is assigned its own regression coefficient. Consequently, the number of regression coefficients does not necessarily equal the number of predictors. For the $p = 5$ model, the total number of coefficients remains equal to the number of predictors because one interaction term replaces a main effect. In contrast, the $p = 10$ model includes both main effects and additional nonlinear and interaction terms, resulting in more regression coefficients than predictors.

3.2.3 Heavily Nonlinear Model — Random Function Generator (RFG)

To assess the performance of the proposed methods under a more challenging and complex data-generating mechanism, an additional scenario was considered based on the Random Function Generator (RFG) framework of Friedman (2001). The RFG model generates highly complex regression functions exhibiting nonlinearity and higher-order interactions among predictors, making it particularly suitable for benchmarking fully nonparametric methods such as random forests.

The response was generated according to

$$\mathbf{Y} = f(\mathbf{X}) + \varepsilon,$$

where the regression function takes the form

$$f(\mathbf{X}) = \sum_{l=1}^{20} a_l \, g_l(\mathbf{z}_l) \; .$$

The coefficients $\{a_l\}_{l=1}^{20}$ are independently drawn from a uniform distribution

$$a_l \sim \mathcal{U}[-1, 1] \, .$$

Each function $g_l(\mathbf{z}_l)$ is a p_l-dimensional Gaussian basis function

$$g_l(\mathbf{z}_l) = \exp\left(-\frac{1}{2}(\mathbf{z}_l - \boldsymbol{\mu}_l)^\top \mathbf{V}_l (\mathbf{z}_l - \boldsymbol{\mu}_l)\right),$$

where $\mathbf{z}_l$ is a randomly selected p_l-dimensional sub-vector of $\mathbf{x}$, defined as

$$\mathbf{z}_l = \{x_{W_l(j)}\}_{j=1}^{p_l} \, ,$$

with W_l an independent random permutation of $\{1, \ldots, p\}$.

The subset size p_l is randomly determined by

$$p_l = \min(\lfloor 1.5 + r \rfloor, \, p) \, , \quad r \sim \text{Exp}(0.5) \, .$$

The mean vector $\boldsymbol{\mu}_l$ is drawn from the same distribution as the input covariates. The $p_l \times p_l$ precision matrix $\mathbf{V}_l$ is randomly generated as

$$\mathbf{V}_l = \mathbf{U}_l \mathbf{D}_l \mathbf{U}_l^\top,$$

where $\mathbf{U}_l$ is a uniformly distributed random orthonormal matrix and $\mathbf{D}_l = \text{diag}(d_{1l}, \ldots, d_{p_l l})$, with entries independently drawn from

$$\sqrt{d_{jl}} \sim \mathcal{U}[0.1, \, 2.0] \, .$$

The input covariates were generated from a multivariate standard normal distribution

$$\mathbf{X} \sim \mathcal{N}_p(\mathbf{0}, \mathbf{I}_p), \quad p = 10 \ .$$

This scenario was evaluated separately from the full factorial design, using sample sizes of $n = 100$ and $n = 500$, with 100 replications, as it does not correspond to a specific level of any factor in the factorial design.

3.2.4 Illustrative Simulated Data Examples

To provide a visual illustration of the data-generating mechanisms described in Sections 3.1 and 3.2, scatter plot matrices were constructed for a representative set of scenarios using $p = 3$ predictors (for RGF scenario $p = 10$). These figures are intended to give a general sense of the distributional characteristics, dependence structures, and predictor–response relationships considered in the simulation study.

Figure 3.1 illustrates the effect of the covariate distributions and dependence structures on the joint distribution of the predictors. Each panel presents a scatter plot matrix for a single replication with $p = 3$ and $n = 500$, with the diagonal panels showing the marginal distribution of each predictor, the off-diagonal panels showing pairwise scatter plots, and the upper triangle showing pairwise correlations.

Figure 3.2 illustrates the effect of the functional form of the predictor-response relationship and the noise level on the observed data. Each panel

presents a scatter plot for a single replication with $p = 3$, $n = 500$, and normally distributed independent covariates. Since $X1$ has the largest contribution for responser generation, it is displayed on the x-axis, while the generated response is shown on the y-axis. The rows correspond to linear and nonlinear functional forms, and the columns correspond to low noise ($\sigma_\varepsilon = 1$) and high noise ($\sigma_\varepsilon = 4$).

Figure 3.3 presents a scatter plot matrix for a single replication of the heavily nonlinear Random Function Generator (RFG) scenario, with $p = 10$ and $n \in \{100, 500\}$. The scatter plots of the response against each predictor show weak, highly irregular marginal relationships with no clear linear or smooth nonlinear trend, reflecting the random higher-order interactions and complex functional form generated by the RFG mechanism. For clarity of visualization, only a subset of the predictors is shown.

3.3 Simulation Design and Procedure

A full factorial design was used, crossing all simulation factors to produce a comprehensive set of scenarios.

For each scenario, 100 replications were performed. In each replication, data were generated according to the specified mechanism and randomly partitioned into a training set (75%) and a test set (25%). The same training-test split was used for all competing methods within each replication to ensure a fair comparison. Table 3.2 summarizes the factors and levels considered.

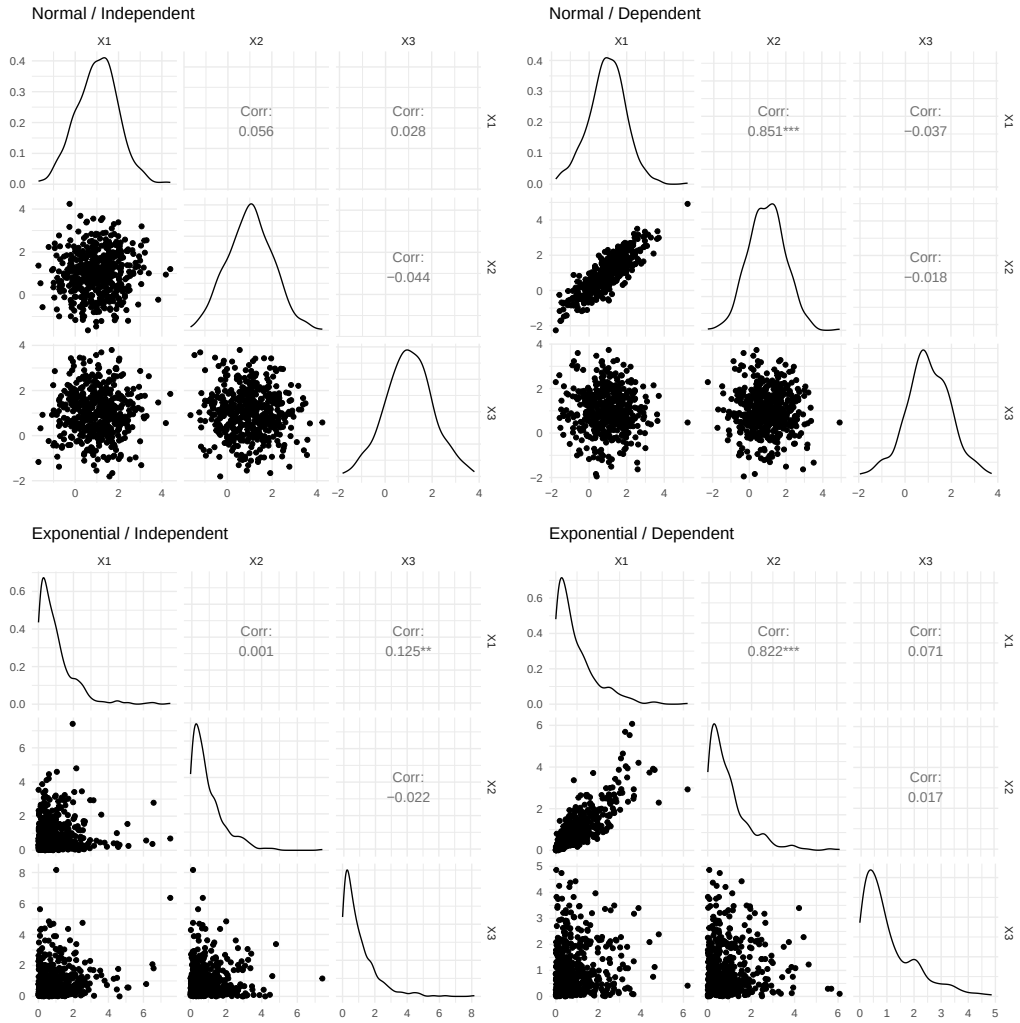


Figure 3.1: Scatter plot matrices illustrating the effect of covariate distribution and dependence structure on the joint predictor distribution, for $p = 3$ and $n = 500$. Panels correspond to (top left) normal independent, (top right) normal correlated, (bottom left) exponential independent, and (bottom right) exponential correlated settings.

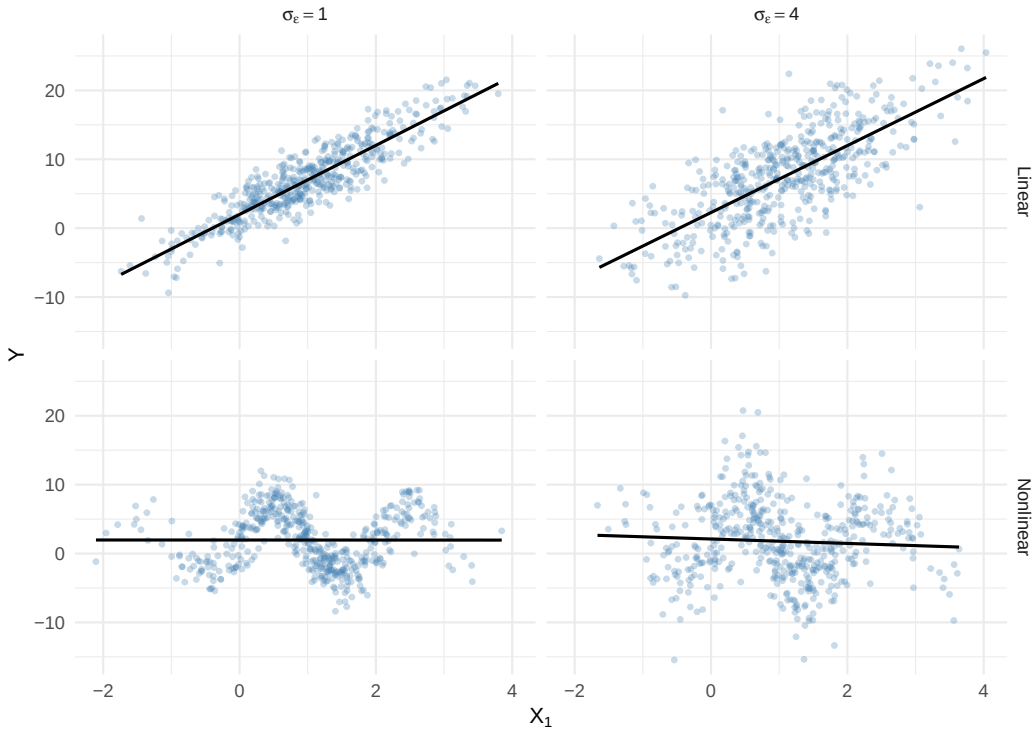


Figure 3.2: Scatter plot matrices illustrating the effect of functional form and noise level on the observed data, for $p = 3$, $n = 500$, and normally distributed independent covariates. Rows correspond to linear (top) and nonlinear (bottom) predictor-response relationships; columns correspond to low noise ($\sigma_\varepsilon = 1$, left) and high noise ($\sigma_\varepsilon = 4$, right).

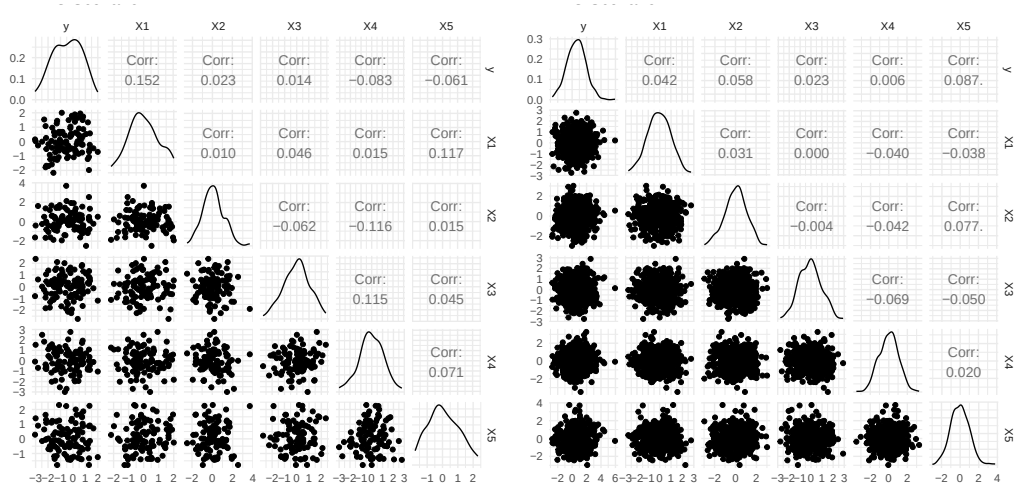


Figure 3.3: Scatter plot matrix of the heavily nonlinear (RFG) scenario, with $p = 10$ predictors. Left plot correspond to $n = 100$, right plot to $n = 500$ observations.

Table 3.2: Summary of simulation factors and levels for the full factorial design.

Factor	**Levels**	**Number of Scenarios**
Number of predictors	$p \in \{3, 5, 10\}$	3
Sample size	$n \in \{100, 500\}$	2
Covariate distribution	Normal, Exponential	2
Dependence structure	Independent, Correlated	2
Noise level	$\sigma \in \{1, 4\}$	2
Relationship type	Linear, Nonlinear	2
Total (factorial)	$3 \times 2 \times 2 \times 2 \times 2$	**96**
Additional scenario	Heavily nonlinear (Friedman, 2001) $p = 10, n \in \{100, 500\}$	2

This yields 96 scenarios in the full factorial design plus 2 additional heavily nonlinear scenarios, for 98 total simulation scenarios. Signal strength was not

treated as a separate crossed factor; instead, within each scenario, predictors were divided into three equal groups reflecting strong, moderate, and weak effects, with one third of predictors assigned to each group. In addition to the factorial design, a separate heavily nonlinear scenario based on the Random Function Generator of Friedman (2001) was evaluated separately for $p = 10$ and $n \in \{100, 500\}$ as described in Section 3.2.3.

All FlexForest variants and the standard random forest baseline were fitted using identical hyperparameter settings across all simulation scenarios to ensure a fair and consistent comparison. Each forest consistent of $B = 500$ trees, grown to a maximum depth $d_{max} = 4$. At each node, a random subset of h candidate predictors was selected for split evaluation, with h set as a function of p to approximate the commonly recommended default of $\lfloor p/3 \rfloor$ (Breiman, 2001): $m = 2$ for $p = 3$, $m = 3$ for $p = 5$, and $m = 4$ for $p = 10$. For $p = 3$, $m = 2$ was used rather than $\lfloor 3/3 \rfloor = 1$ to ensure that at least two predictors were considered at each split. The candidate set of weighting parameters was fixed as $\Gamma = \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$ across all scenarios.

The standard random forest baseline was obtained by setting $\gamma = 1$ in the FF-F, which recovers the standard CART-based random forest as established in Proposition 3, with all other hyperparameters identical to those used for the FlexForest variants. Each method was fitted using the training data, and the evaluation procedure is described in the Section 3.4.

All simulation code, including the implementation of the FlexForest variants,

the data-generating mechanisms, and the analysis scripts used to produce the results presented in this thesis, is available in a GitHub repository at https://github.com/HeshaniAchinthika/Flexsplit.git. The repository includes the R package *FlexForest*, all simulation scripts, and the coefficient values and parameter settings used in the simulation study.

3.4 Performance Evaluation

In order to compare FlexForest's predictive abilities, we consider a number of difference performance metrics. Each metric attempts to measure a different aspect of predictive accuracy. The mean squared error provides an absolute measure of prediction error by quantifying the average squared difference between the observed and predicted responses. The relative prediction error measures the prediction error of each method relative to that of the standard random forest, thereby facilitating direct comparisons across competing methods. Predictive performance is evaluated on the independent test set using these performance metrics.

Mean Squared Error (MSE)

Prediction accuracy was evaluated using the mean MSE, defined as

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2,$$

where Y_i and $\hat{Y}_i$ denote the true and predicted responses, respectively. The MSE is non-negative, with smaller values indicating greater predictive accuracy.

Relative Prediction Error (RPE)

To facilitate comparison across methods, the predictive performance of each FlexForest variant was assessed relative to the standard random forest baseline using the Relative Prediction Error (RPE).

In this study, the standard random forest serves as the baseline method. Rather than using an external package, the standard random forest was obtained directly from the proposed FlexForest framework. This approach ensures a fair and consistent comparison: since the standard random forest is obtained from the same implementation as the FlexForest variants, all aspects of the algorithm, including the candidate split point identification method (midpoint approach) and the stopping criterion (maximum depth) are held identical across all methods. These implementation details can vary between software packages, and using the same codebase for all methods eliminates any performance differences attributable to such implementation choices rather than to the weighting scheme itself. The RPE for the FlexForest variant $FF*$:

$$\text{RPE}_{FF*} = \frac{\text{MSE}_{FF*}}{\text{MSE}_{RF}} ,$$

where MSE_{FF*} is the MSE of FlexForest variant $FF*$ on the test set and MSE_{RF} represents the MSE of the standard random forest on the same test

set. A RPE value less than 1 indicates improved predictive performance over the standard random forest, while values greater than 1 indicate inferior performance.

3.5 Simulation Results

Based on the simulation study described in Section 3.3, we present the key findings across all simulation scenarios. The patterns observed were largely consistent across different sample sizes. Therefore, for brevity, only the results for the sample size $n = 500$ are presented here, while the corresponding results for $n = 100$ are provided in Appendix C. Results for the additional heavily nonlinear scenario are presented separately for both sample sizes. For the remaining scenarios, and for ease of comparison, results are organized into four scenario groups based on two factors: the covariate distribution (normal or exponential) and the functional form of the relationship between the response variable and the covariates (linear or nonlinear). This yields four scenario groups: normal linear, normal nonlinear, exponential linear, and exponential nonlinear. Within each scenario group, results are presented across four conditions corresponding to the column facets of each figure: low noise with independent covariates ($\sigma_\epsilon = 1, \boldsymbol{\Sigma_X} = \boldsymbol{I}$), low noise with correlated covariates ($\sigma_\epsilon = 1, \boldsymbol{\Sigma_X} \neq \boldsymbol{I}$), high noise with independent covariates ($\sigma_\epsilon = 4, \boldsymbol{\Sigma_X} = \boldsymbol{I}$), and high noise with correlated covariates ($\sigma_\epsilon = 4, \boldsymbol{\Sigma_X} \neq \boldsymbol{I}$). In all figures, the boxplots summarize the RPE computed on the independent test data across

the simulation replications. The blue boxplots represent the forest-wise variant (FF-F), green the tree-wise variant (FF-T), yellow the random node-wise variant (FF-RN), and orange the optimal node-wise variant (FF-ON). The red dashed line at RPE = 1 serves as the reference line corresponding to the performance of the standard random forest; RPE values below this line indicate that the proposed FlexForest variant outperforms the standard random forest.

Normal Linear Setting

Figure 3.4 presents the RPE distributions for the normal linear scenario. Overall, no variant emerges as a clear winner across all conditions, consistent with the standard random forest already being well-suited to linear data-generating mechanisms.

Across most conditions, the RPE distributions for the FF-F (blue) variant are centered close to 1, indicating predictive performance comparable to the standard random forest. This is consistent with the empirical observation that, across all simulation settings, the forest-wise selection procedure chose $\gamma = 1$ for approximately 92% of the fitted forests (see Appendix D.1). The node-wise variants, both FF-RN (yellow) and FF-ON (orange), tend to perform worse than the standard random forest across most conditions, suggesting that introducing flexibility at the node level is not beneficial in this setting, where the standard proportion-based weighting is already well-tuned.

The FF-T (green) shows a more mixed pattern: it underperformed the

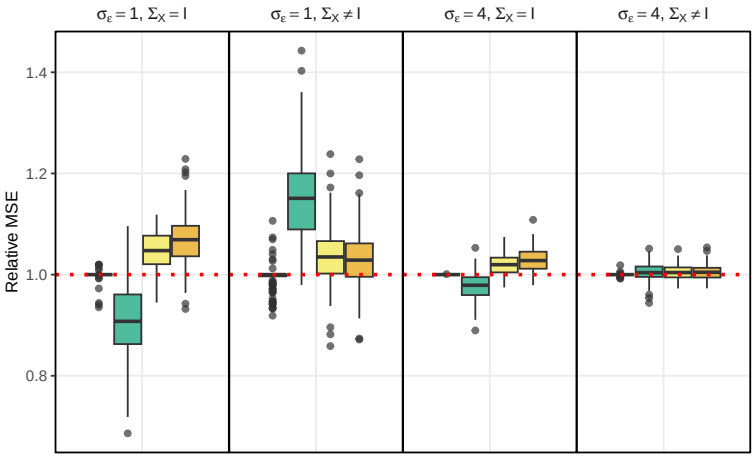
standard random forest under low noise with correlated covariates, particularly for $p = 3$ and $p = 5$, but performs comparably to or slightly better under independent covariates , with the difference narrowing further at high noise and larger p. Overall, FF-T remains a reasonable alternative in linear settings, particularly as model complexity increases.

Normal Nonlinear Setting

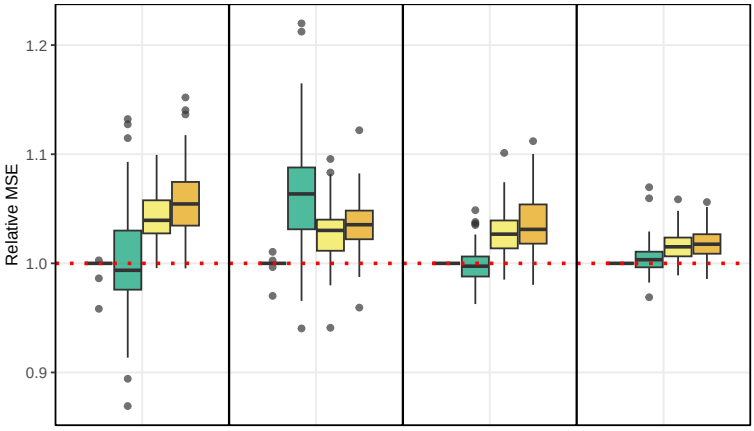
Figure 3.5 presents the RPE distributions for the normal nonlinear scenario. Unlike the linear setting, the FF-T consistently achieves RPE values below one across the majority of replications in all four conditions, indicating a clear improvement over the standard random forest. This improvement is observed regardless of the noise level or covariate dependence structure, suggesting that the advantage of the FF-T in nonlinear settings is robust to these factors.

The FF-F (blue) also shows noticeable improvement, particularly for $p = 3$. The empirical results show that the forest-wise selection procedure selected $\gamma > 1$ in approximately 71% of the fitted forests (see Appendix D.2). In contrast, the node-wise variants (yellow and orange) tend to perform worse than the standard random forest, with RPE distributions above one.

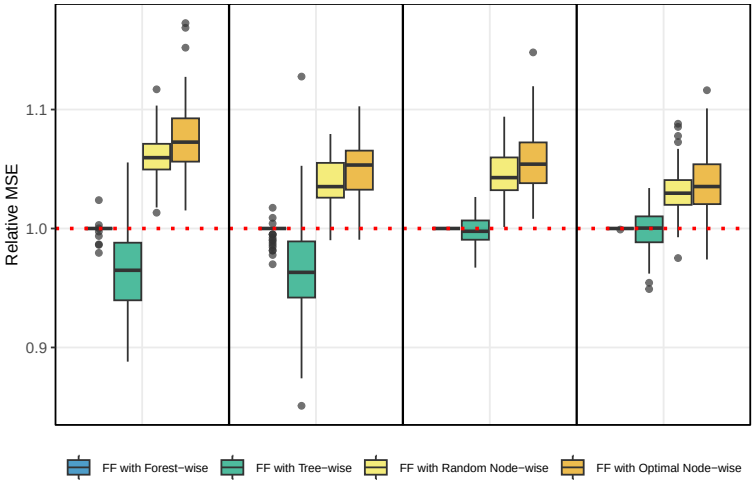
These results indicate that the tree-wise variant's advantage is specific to nonlinear data-generating mechanisms, consistent with its role in capturing local heterogeneity that a single global weighting parameter cannot.



(a) Three predictors ($p = 3$).

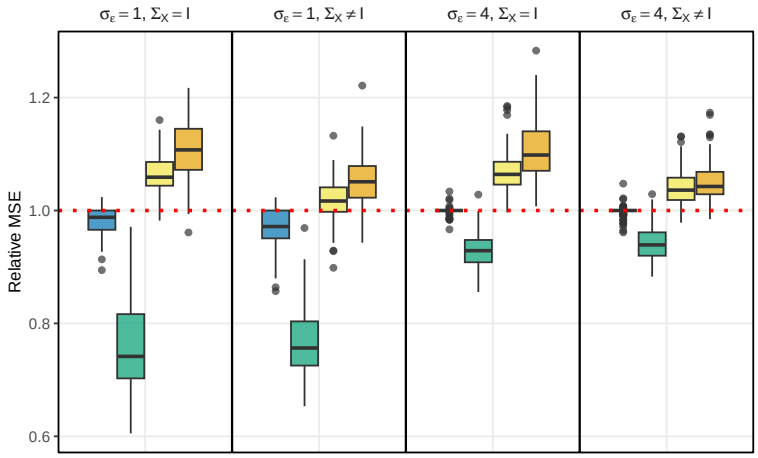


(b) Five predictors ($p = 5$).

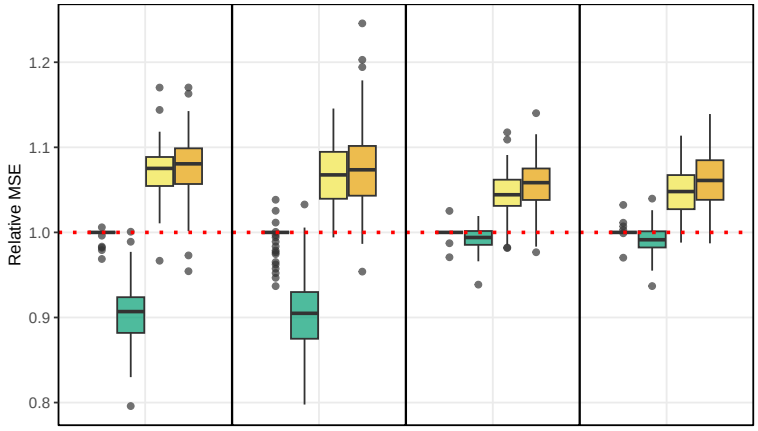


(c) Ten predictors ($p = 10$).

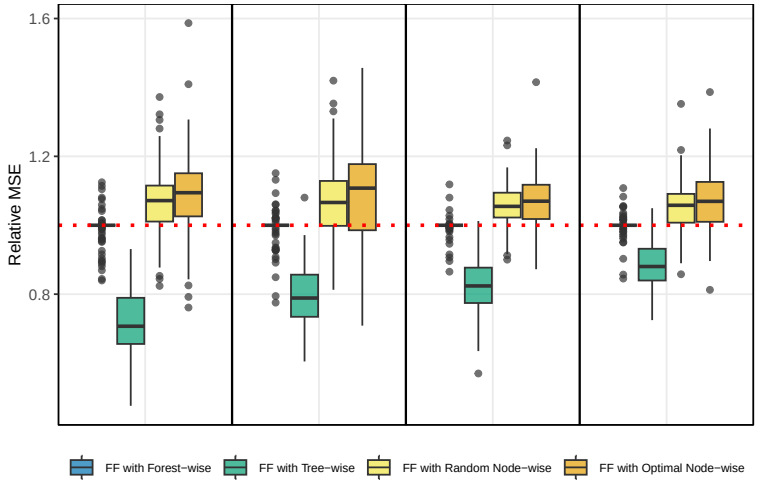
Figure 3.4: RPE distributions for the **normal linear** scenario across different noise levels and response dependence structures, based on 100 simulation replications. Each panel compares the four FlexForest variants, with the dashed horizontal line indicating RPE = 1. The three subfigures correspond to datasets with $p = 3$, 5, and 10 predictors, respectively.



(a) Three predictors ($p = 3$).



(b) Five predictors ($p = 5$).



(c) Ten predictors ($p = 10$).

Figure 3.5: RPE distributions for the **normal nonlinear** scenario across different noise levels and response dependence structures, based on 100 simulation replications. Each panel compares the four FlexForest variants, with the dashed horizontal line indicating RPE = 1. The three subfigures correspond to datasets with $p = 3$, 5, and 10 predictors, respectively.

Exponential Linear Setting

Figure 3.6 presents the RPE distributions for the exponential linear scenario. The overall pattern closely mirrors that observed in the normal linear setting (Section 3.5), confirming that FlexForest variants' performance in linear settings is not specific to the covariate distribution and persists under the positively skewed exponential distribution.

The FF-F (blue) performs almost identically to the standard random forest across all simulation settings, with boxplots collapsing to a line at RPE $= 1$. This reflects the forest-wise procedure's strong preference for $\gamma = 1$ (approximately 83% of the fitted forests; see Appendix D.3), which recovers the standard CART-based random forest. The node-wise variants (yellow and orange) show median RPE typically above one, although under low noise with correlated covariates, particularly at $p = 3$ and $p = 5$, a notable proportion of replications fall below one. Under other conditions, they consistently perform worse than the standard random forest.

The FF-T (green) shows the strongest performance under low noise with independent covariates, achieving median RPE below one across all p. Under low noise with correlated covariates, FF-T underperformed at $p = 3$ and $p = 5$ but recovers at $p = 10$, consistent with the pattern observed in the normal linear setting. Under high noise, the difference between FF-T and the standard random forest are minimal regardless of the covariates structure or p.

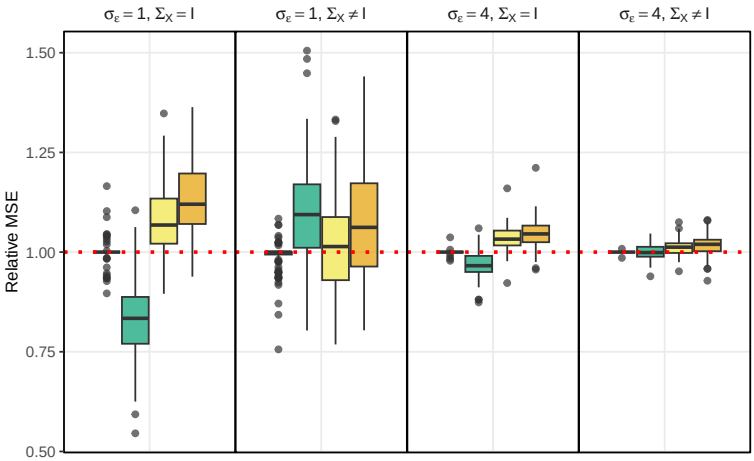
Overall, these results indicate that the standard random forest remains

a strong baseline in linear settings. Nonetheless, FF-T offers a reasonable alternative, performing comparably to or slightly better than the standard random forest across most conditions, with its performance advantage becoming more consistent as p increases and noise diminishes any remaining differences at high noise levels.

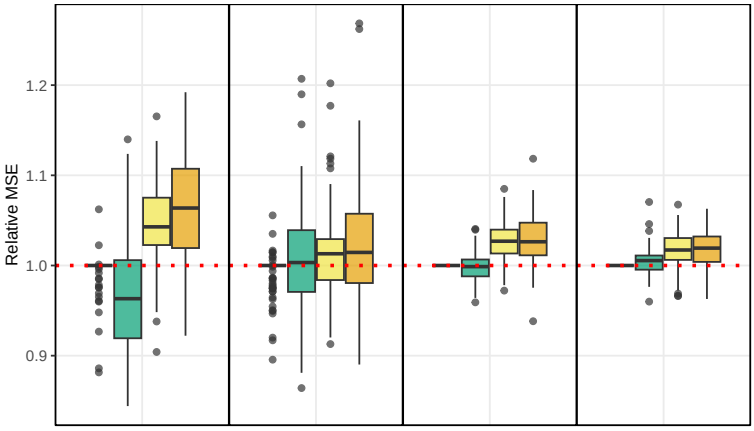
Exponential Nonlinear Setting

Figure 3.7 presents the RPE distributions for the exponential nonlinear scenario. The overall pattern is consistent with the normal nonlinear setting: the FF-T (green) consistently achieves RPE values below one across the majority of replications, confirming that its advantage over the standard random forest in nonlinear settings is not specific to the normal covariate distribution but persists under the skewed exponential distribution. We therefore focus here on findings that differ from or extend those of the normal nonlinear setting.

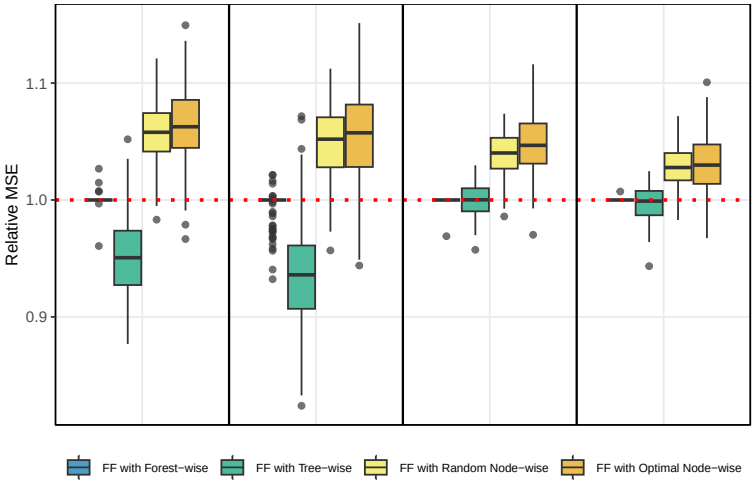
Two new patterns emerge in this scenario. First, FF-F (blue) shows improved performance at $p = 3$ and $p = 5$ under low noise, achieves median RPE values below one, a pattern not clearly observed in the normal nonlinear scenario. Empirical results show that the FF-F selected $\gamma > 1$ in approximately 83% of fitted forests in this scenario (see Appendix D.4). This suggests that the asymmetric, skewed nature of the exponential distribution creates conditions where a single global γ can effectively improve upon the standard weighting. Second, the node-wise variants (yellow and orange) show some improvement at



(a) Three predictors ($p = 3$).



(b) Five predictors ($p = 5$).



(c) Ten predictors ($p = 10$).

Figure 3.6: RPE distributions for the **exponential linear** scenario across different noise levels and response dependence structures, based on 100 simulation replications. Each panel compares the four FlexForest variants, with the dashed horizontal line indicating RPE = 1. The three subfigures correspond to datasets with $p = 3$, 5, and 10 predictors, respectively.

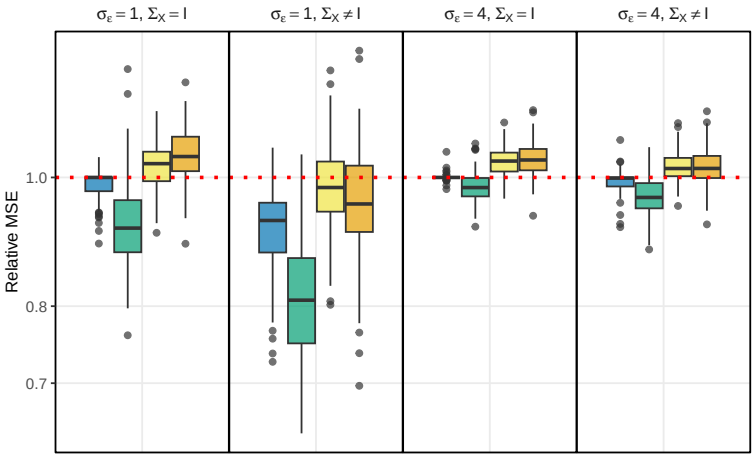
$p = 3$ under low-noise correlated covariates, the first scenario in the factorial design where they outperformed standard random forest.

The most prominent feature of this scenario is the extreme variability at $p = 10$, where the y-axis spans a logarithmic scale from approximately 0.1 to beyond 10.0. This variability arises from the combined difficulty of high-dimensional nonlinear prediction and the skewed exponential distribution, and the $\exp(X_2)$ term in the nonlinear predictor–response relationship, which can produce extremely large response values when X_2 falls in the upper tail of its distribution. This affects all methods equally, indicating it is a property of the data-generating mechanism rather than of any specific variant. Despite this, the median RPE of the FF-T remains below one across all four conditions at $p = 10$.

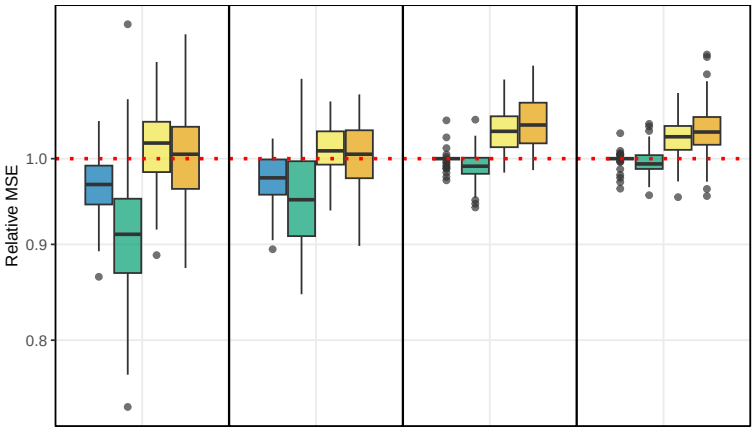
Overall, these results confirm that FF-T's advantage in nonlinear settings is robust to covariate distribution, holding under both symmetric normal and skewed exponential covariates, with the improvement most pronounced under low noise.

Heavily Nonlinear Setting

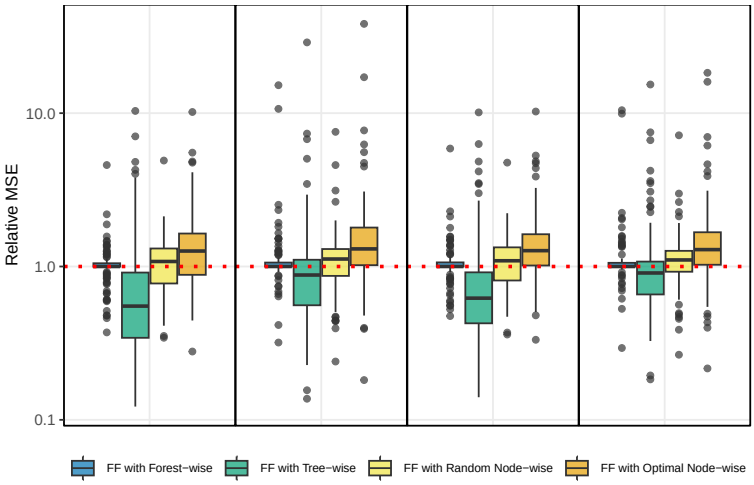
Given the consistent performance improvement observed for the FF-T across nonlinear settings, we further investigated all FlexForest variants under a more challenging, maximally complex data-generating mechanism based on the Random Function Generator (RFG) framework of Friedman (2001). Figure 3.8



(a) Three predictors ($p = 3$).



(b) Five predictors ($p = 5$).



(c) Ten predictors ($p = 10$).

Figure 3.7: RPE distributions for the **exponential nonlinear** scenario across different noise levels and response dependence structures, based on 100 simulation replications. Each panel compares the four FlexForest variants, with the dashed horizontal line indicating RPE = 1. The three subfigures correspond to datasets with $p = 3$, 5, and 10 predictors, respectively.

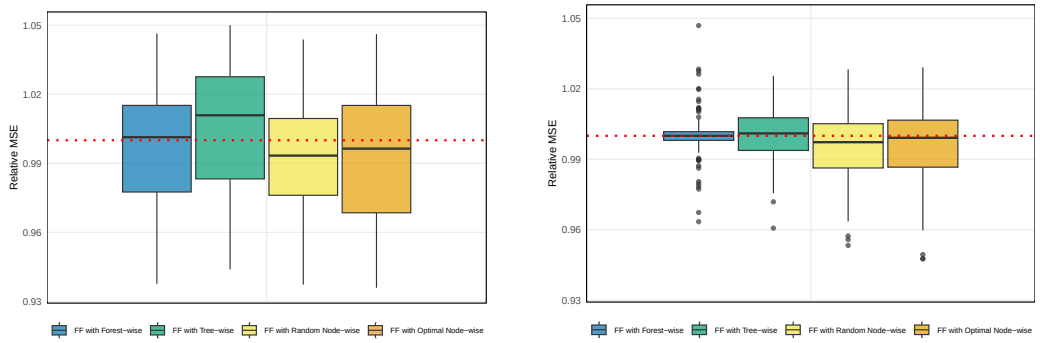
presents the RPE distributions for this heavily nonlinear scenario under $n = 100$ and $n = 500$.

Under the smaller sample size ($n = 100$), all four variants show RPE distributions centered close to one with substantial variability and considerable overlap between methods, indicating that performance differences are relatively small when the sample size is limited. FF-T has a median RPE slightly above one, FF-F is comparable to the standard random forest, and the node-wise variants show median RPE slightly below one, indicating modest improvement.

As the sample size increases to $n = 500$, variability decreases considerably across all methods. FF-F (blue) shows the narrowest interquartile range, with a median RPE essentially equal to one, indicating that the middle 50% of replications produce predictive performance nearly identical to the standard random forest. This behavior is consistent with the empirical selection frequencies, where approximately 45% of the fitted forests selected $\gamma = 1$ and the remaining 55% selected $\gamma = 1.5$. The FF-T (green) also shows reduced variability and a median RPE slightly closer to one, indicating a small but consistent improvement over the standard random forest. The node-wise variants (yellow and orange) have median RPE values below or at one, indicating improved predictive performance on average. However, their distributions are wider, with several lower outliers, suggesting that while they occasionally achieve substantially larger gains, their performance is less stable across replications.

Overall, these results suggest that when the underlying data structure is

maximally complex, as generated by the RFG mechanism, node-level flexibility tends to offer the greatest predictive benefit among the FlexForest variants, suggesting that the benefit of local adaptivity becomes more relevant as the complexity of the underlying data-generating mechanism increases.



(a) $n = 100$ (b) $n = 500$

Figure 3.8: Heavily nonlinear relationship generated from the RGF function under different sample sizes.

Distribution of Selected Weighting Parameters in the Tree-Wise Variant

To gain further insight into the mechanism underlying the performance of the tree-wise FlexForest variant, the distribution of the weighting parameter γ^* selected by individual trees was examined across replications. For each replication, the proportion of trees selecting each candidate $\gamma_k \in \Gamma$ was recorded across the $B = 500$ trees of each fitted forest. Results are presented for the four main scenario groups (normal linear, normal nonlinear, exponential linear, exponential nonlinear), each considered under low/high noise and indepen-

dent/dependent covariate structures, for the $p = 10$, $n = 500$ case. The corresponding distribution across all scenarios is presented in Appendix E; here we summarize the key patterns.

Three qualitatively distinct patterns of γ^* selection emerge, which correspond closely to the RPE results across scenarios. In linear settings, both normal and exponential the distribution of selected γ^* values is strongly concentrated, with $\gamma = 1$ ($\gamma = 1$, corresponding to the standard CART weighted variance criterion) accounting for approximately 25-45% of tree selections, with the concentration increasing under high noise. This indicates that in linear settings, individual trees systematically identify the standard CART weighting as optimal, reducing the diversity of weighting schemes across the ensemble and explaining the limited improvement of the tree-wise variant over the standard random forest in these scenarios.

In the normal nonlinear setting, the distribution is substantially more uniform, with all five candidate values (except $\gamma = 0.5$) receiving at least 12% proportions of tree selections under low noise and no single weighting parameter strongly dominating. This diversity reflects genuine heterogeneity in the locally optimal weighting across trees and is the mechanism by which the tree-wise variant achieves improved predictive performance under nonlinear data-generating mechanisms. The exponential nonlinear setting exhibits an intermediate pattern, with $\gamma = 1.5$ slightly elevated relative to the other rules but without the strong dominance observed in the linear settings, consistent

with the modest but consistent improvement of the tree-wise variant in this scenario.

Summary of Simulation Findings

The simulation results lead to two main conclusions. First, in nonlinear settings, the tree-wise FlexForest variant consistently outperforms the standard random forest, regardless of the covariate distribution or noise level. Second, when the data-generating mechanism is maximally complex, as in the RFG scenario, the node-wise variants, which allow γ to vary at every split, provide the greatest improvement, demonstrating the benefit of local flexibility in child-node weighting under highly complex predictor-response relationships.

Chapter 4

Real Data Application

This chapter evaluates the empirical performance of the proposed methods on three real world regression datasets, providing a complementary assessment under realistic data conditions where the true underlying relationship between predictors and response is unknown. Figures 4.1, 4.2, and 4.3 present scatter plots of the response variable against each predictor for the three datasets, with a nonparametric smoother overlaid to indicate the marginal relationship between each predictor and the response.

For each dataset, we considered multiple train-test splits and fitted all four FlexForest variants (optimal node-wise, random node-wise, tree-wise, and forest-wise) on the training data, alongside the standard random forest. Performance was then assessed on the test data using the MSE and RPE, defined in Section 3.4. The final reported results are the average MSE and RPE across all train-test splits considered.

4.1 Data Example 1: Concrete Compressive Strength

The Concrete Compressive Strength dataset, obtained from the UCI Machine Learning Repository (Yeh, 1998). It contains n = 1,030 observations and $p = 8$ continuous and integer predictors describing the composition of concrete mixtures. The response variable is the compressive strength of concrete, measured in megapascals (MPa). The predictors include the quantities of cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate (all measured in kg/m^3), as well as the age of the concrete sample in days. And there are no missing values in this dataset.

The scatter plots in Figure 4.1 reveal several notably nonlinear marginal relationships between the concrete mixture components and compressive strength. The relationship with age shows a clear nonlinear pattern, with compressive strength increasing sharply over the first 100 days before levelling off. Cement content shows a positive approximately linear trend. Water content shows a nonlinear U-shaped pattern, and superplasticizer shows a positive trend. The aggregate predictors (coarse and fine) show relatively flat relationships with the response. Overall, the plots provide clear visual evidence of nonlinear predictor–response relationships in this dataset, which is consistent with the improved performance of the tree-wise FlexForest variant relative to the standard random forest reported in Table 4.1.

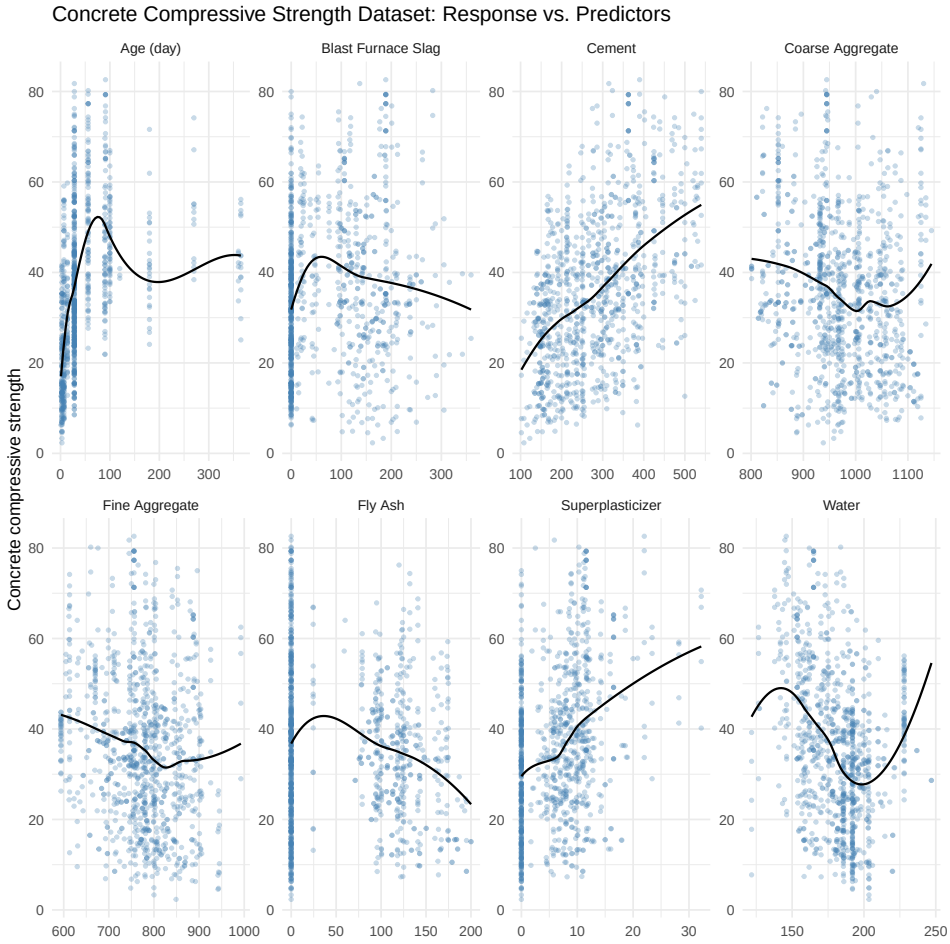


Figure 4.1: Scatter plots of the concrete compressive strength against concrete mixture components predictors in the Concrete Compressive Strength dataset.

4.2 Data Example 2: Energy Efficiency

The Energy Efficiency dataset was obtained from the UCI Machine Learning Repository (Tsanas and Xifara, 2012). It contains $n = 768$ observations and $p = 8$ numerical predictors, comprising both continuous and integer-valued variables, describing the structural characteristics of residential buildings. The predictors include relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area, and glazing area distribution. The dataset includes two response variables: heating load (HL) and cooling load (CL), both measuring the energy efficiency of the building. In this analysis, heating load is used as the response variable. The dataset contains no missing values.

The scatter plots in Figure 4.2 reveal strong nonlinear marginal relationships between the building characteristics and heating load. Predictors such as relative compactness, surface area, and roof area exhibit clear nonlinear patterns. Glazing area, orientation, and overall height take only a small number of distinct values, reflecting their discrete nature. Among these, glazing area and orientation exhibit relatively flat smoothed relationships. Note that the smoother for glazing area is unreliable because of the discrete and sparse nature of its values.

The nonlinear structure of this dataset is consistent with the strong improvement achieved by the tree-wise FlexForest variant (the largest improvement observed across all three datasets) and helps explain the clear advantage of

flexible child-node weighting in this application.

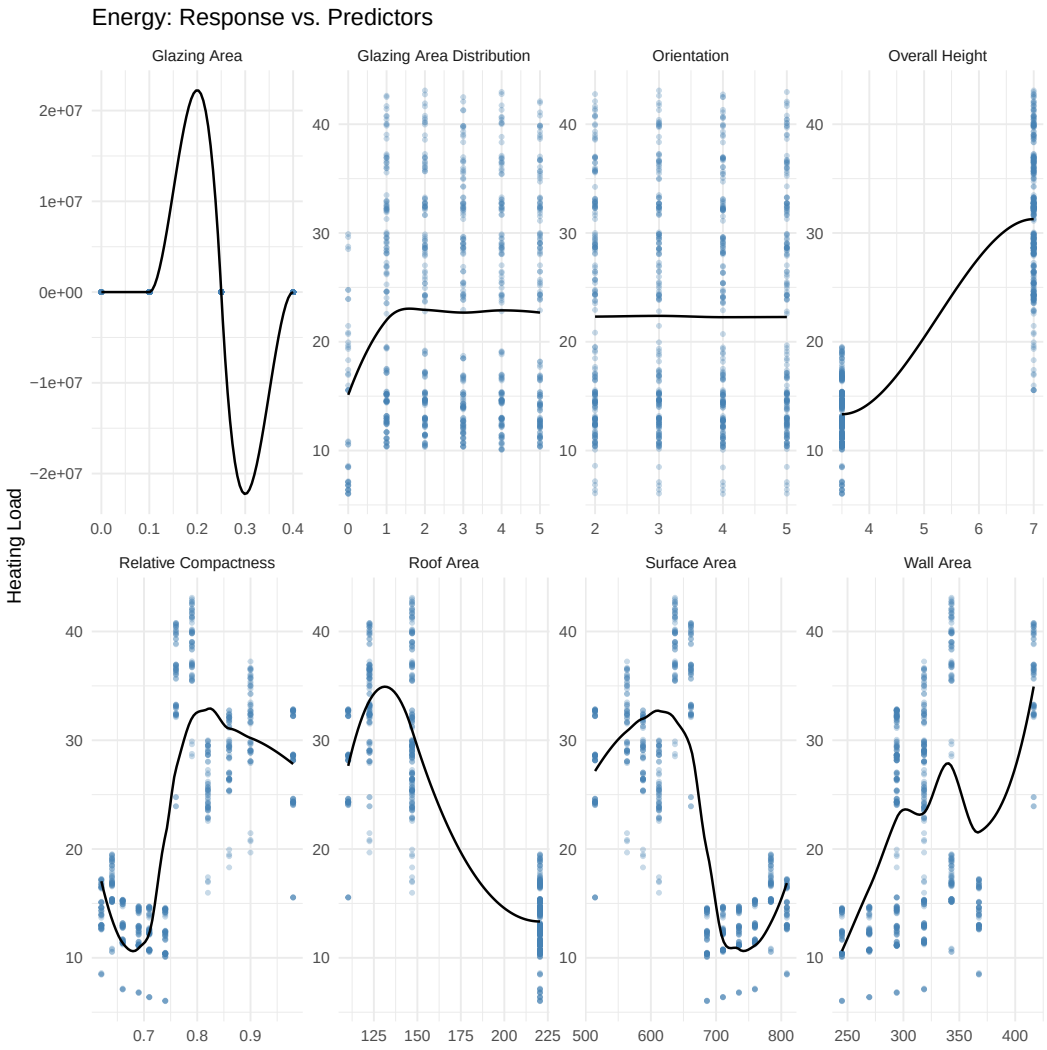


Figure 4.2: Scatter plots of the heating load against building characteristics predictors in the Energy Efficiency dataset.

4.3 Data Example 3: Parkinson's Disease Telemonitoring

The Parkinson's Disease Telemonitoring dataset, obtained from the UCI Machine Learning Repository (Tsanas and Little, 2009). It contains $n = 5{,}875$ biomedical voice recordings from 42 patients with early-stage Parkinson's disease, collected over a six-month trial of a remote symptom-monitoring device; each patient contributes approximately 100 repeated recordings. Each observation includes 16 dysphonia (voice) measures, including jitter, shimmer, noise-to-harmonic ratio, and nonlinear signal complexity measures, along with patient age, sex, and time since recruitment (test time). The binary variable sex was retained and treated as a numerical binary predictor taking values 0 and 1. The clinician-rated motor UPDRS score (the motor component of the Unified Parkinson's Disease Rating Scale) was used as the response variable. The total UPDRS score, which is highly correlated with motor UPDRS by construction, was excluded from the predictor set to avoid leakage. The variable test time, which records the number of days since trail recruitment for each voice recording, was excluded, as it reflects the longitudinal structure of the data rather than a clinically informative biomedical predictor of the disease severity.

Because the data consist of repeated measurements within subjects, the train-test partition for each replication was constructed at the subject level, with a randomly selected 75% of the 42 subjects assigned to training and the

remaining subjects held out entirely for testing, rather than at the observation level, to avoid inflating predictive accuracy through within-subject correlation between training and test recordings. However, when fitting the standard random forest and the FlexForest models, repeated recordings within each subject were treated as independent observations.

The scatter plots in Figure 4.3 show the relationship between motor UPDRS and all 20 variables recorded in the dataset: the 16 voice measures, age, sex, test time, and total UPDRS. Note that test time and total UPDRS were excluded from the predictor set used in model fitting and are shown here for descriptive purposes only. Most of the 16 voice measures show weak, noisy marginal relationships with motor UPDRS, with dense clustering near low predictor values and considerable scatter around the smoothed curves. Age shows a non-monotonic, clustered pattern. The clustering arises from repeated recordings for the same patient, for whom age remains fixed. Sex, being binary, appears as two vertical strips with no visible association with motor UPDRS. The test time panel shows a broad, banded spread, reflecting repeated recordings collected over each patient's monitoring window, again with no clear marginal trend. In contrast, total UPDRS shows a strong, near-linear positive association with motor UPDRS, consistent with the close clinical relationship between the two severity scores. The weak and noisy marginal relationships in this dataset, combined with the presence of higher-order interactions among the 16 voice measures, suggest a complex data-generating mechanism, consistent with the finding that the node-wise FlexForest variants achieved the best performance

on this dataset.

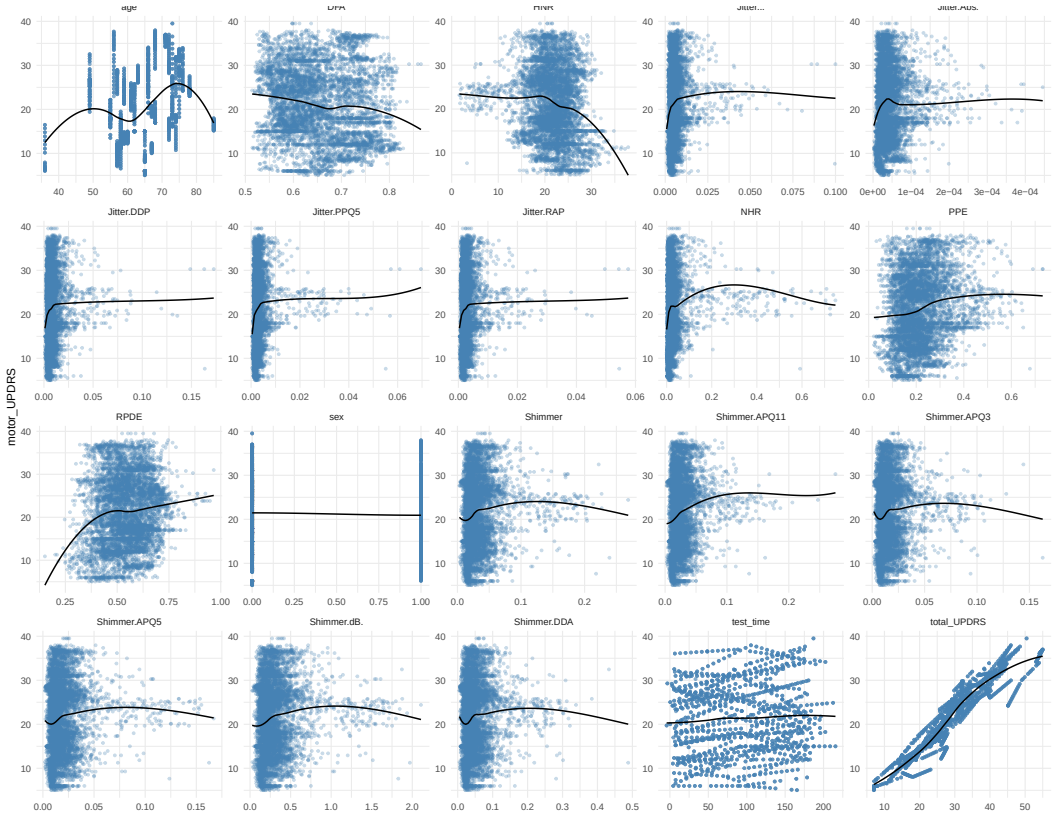


Figure 4.3: Scatter plots of the motor UPDRS score against individual biomedical voice predictors in the Parkinson's disease dataset.

4.4 Evaluation Procedure

For each dataset, the following procedure was applied to obtain the performance measure across repeated random splits.

1. The dataset was randomly split into training (75%) and test (25%) sets

(for the Parkinson's dataset, this split was performed at the subject level, as described in Section 4.3).

2. The standard random forest and the four FlexForest variants were fitted on the training dataset.

3. For each method, prediction were obtained on the test dataset, and the MSE was calculated.

4. The relative prediction error was computed for each FlexForest variant.

5. Steps 1 to 4 were repeated over 30 random train-test splits to obtain the variability and average prediction performance of each variant.

All forests were grown with ntree = 100 trees, a maximum tree depth of 6 (for the Parkinson's dataset 8), and mtry = 4 candidate predictors considered at each split; these settings were held fixes across the standard random forest and all four FlexForest variants to ensure fair comparison. For the FlexForest variants, the weighting parameter γ was varied over the candidate set $\Gamma = \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$.

4.5 Results

Tables 4.1, 4.2, and 4.3 present the MSE and RPE for each method, averaged over 30 random train-test splits, with standard deviations (SD) in parentheses. All performance metrics reported here are evaluated on the held-out test set,

not the training set, ensuring that the reported results reflect the out-of-sample predictive performance of each method.

4.5.1 Concrete Compressive Strength

Table 4.1 presents the results for the Concrete Compressive Strength dataset. The standard random forest achieves a mean MSE of 49.98 (SD = 5.52) on the test set. Among the FlexForest variants, the tree-wise variant achieves the best predictive performance, with a mean MSE of 48.75 (SD = 5.20) and a mean RPE of 0.976 (SD = 0.028). Since this RPE is below one, the tree-wise variant achieves improved predictive accuracy relative to the standard random forest, corresponding to an approximate 2.4% reduction in mean squared error. The forest-wise variant performs comparably to the standard random forest, with a mean RPE of 0.999 (SD = 0.014), indicating negligible difference from the baseline.

In contrast, the node-wise variants perform worse than the standard random forest on this dataset, with mean RPE values of 1.115 (SD = 0.041) for the random node-wise variant and 1.130 (SD = 0.014) for the optimal node-wise variant, indicating approximately 11–13% higher prediction error than the standard random forest. This finding is consistent with the simulation results of Chapter 3, where the node-wise variants did not consistently outperform the standard random forest except under maximally complex nonlinear data-generating mechanisms

Table 4.1: MSE and RPE for the Concrete Compressive Strength dataset, averaged over 30 random train–test splits. Standard deviations are shown in parentheses.

Method	**MSE**	**Relative MSE**
Standard Random Forest	49.98 (5.52)	1.000
FlexForest with Forest-wise	49.92 (5.49)	0.999 (0.014)
FlexForest with Tree-wise	**48.75 (5.20)**	**0.976 (0.028)**
FlexForest with Random Node-wise	55.67 (6.05)	1.115 (0.041)
FlexForest with Optimal Node-wise	56.43 (6.35)	1.130 (0.014)

4.5.2 Energy Efficiency

Table 4.2 presents the results for the Energy Efficiency dataset. The standard random forest achieves a mean MSE of 1.117 (SD = 0.172) on the test set. The tree-wise variant again achieves the best predictive performance among all methods, with a mean MSE of 0.895 (SD = 0.120) and a mean RPE of 0.806 (SD = 0.075). This represents a substantial improvement over the standard random forest, corresponding to approximately 19.4% reduction in mean squared error, the largest improvement observed across all three datasets considered in this chapter. The forest-wise variant also shows notable improvement on this dataset, with a mean RPE of 0.920 (SD = 0.064), indicating an approximate 8% reduction in MSE relative to the standard random forest.

As with the Concrete dataset, the node-wise variants perform comparably to or slightly worse than the standard random forest. The strong performance of

the tree-wise and forest-wise variants on the Energy Efficiency dataset suggests that this dataset exhibits a nonlinear predictor-response relationship, for which flexible child-node weighting at the tree and forest level is beneficial, consistent with the simulation findings of Chapter 3.

Table 4.2: MSE and RPE for the Energy Efficiency dataset, averaged over 30 random train–test splits. Standard deviations are shown in parentheses.

Method	**MSE**	**Relative MSE**
Standard Random Forest	1.117 (0.172)	1.000
FlexForest with Forest-wise	1.025 (0.159)	0.920 (0.064)
FlexForest with Tree-wise	**0.895 (0.120)**	**0.806 (0.075)**
FlexForest with Random Node-wise	1.118 (0.164)	1.003 (0.057)
FlexForest with Optimal Node-wise	1.134 (0.164)	1.018 (0.064)

4.5.3 Parkinsons Telemonitoring

Table 4.3 presents the results for the Parkinsons Telemonitoring dataset. The standard random forest achieves a mean MSE of 75.22 (SD = 20.33) on the test set. In contrast to the previously discussed datasets, the pattern of results for the Parkinson's dataset differs notably. The random node-wise variant achieves the best predictive performance, with a mean MSE of 73.36 (SD = 20.25) and a mean RPE of 0.975 (SD = 0.041), corresponding to an approximate 2.5% reduction in MSE relative to the standard random forest. The optimal node-wise variant also achieves a mean RPE below one (0.976, SD = 0.053),

indicating a comparable improvement over the standard random forest.

In contrast, the tree-wise variant underperforms relative to the standard random forest on this dataset, with a mean RPE of 1.047 (SD = 0.084), indicating approximately 4.7% higher prediction error than the baseline. The forest-wise variant performs comparably to the standard random forest, with a mean RPE of 0.998 (SD = 0.037).

The reversal in the relative performance of the variants on the Parkinson's dataset, where the node-wise variants outperform the tree-wise variant, in contrast to the Concrete and Energy Efficiency datasets, is consistent with the simulation findings for the heavily nonlinear RFG scenario (Section 3.5), where the node-wise variants also achieved the strongest improvement. Parkinson's Telemonitoring dataset involves 16 biomedical voice measurements and exhibits complex nonlinear relationships between the predictors and disease severity. This may indicate a more complex underlying relationship than those present in the other two datasets and may help explain the stronger performance of the node-wise variants. Under such conditions, the additional flexibility introduced by varying γ at the node level, rather than at the tree or forest level appears to be beneficial.

Table 4.3: MSE and RPE for the Parkinsons Telemonitoring dataset, averaged over 30 random train–test splits. Standard deviations are shown in parentheses.

Method	**MSE**	**Relative MSE**
Standard Random Forest	75.22 (20.33)	1.000
FlexForest with Forest-wise	75.15 (20.72)	0.998 (0.037)
FlexForest with Tree-wise	78.96 (23.01)	1.047 (0.084)
FlexForest with Random Node-wise	**73.36 (20.25)**	**0.975 (0.041)**
FlexForest with Optimal Node-wise	73.28 (19.90)	0.976 (0.053)

4.5.4 Summary Across Datasets

Across all three datasets, at least one FlexForest variant achieves improved predictive performance relative to the standard random forest on the held-out test set. The tree-wise variant performs best on the two engineering datasets, Concrete Compressive Strength and Energy Efficiency, with the most substantial improvement observed for the Energy Efficiency dataset. For the Parkinson's Telemonitoring dataset, the random node-wise variant achieves the best performance, consistent with the simulation finding that node-level flexibility provides the greatest benefit under maximally complex nonlinear data-generating mechanisms.

Table 4.4 presents the average computation time, in seconds, for each method across the three datasets, based on $B = 100$ trees and averaged over the 30 random train–test splits described above.

As shown in Table 4.4, the node-wise variants required less computation

Table 4.4: Average computation time (in seconds) for each method across the three real datasets, based on $B = 100$ bootstrap samples (trees) and averaged over 30 random train–test splits. The value in parentheses for each FlexForest variant expresses its computation time as a multiple of the standard random forest time for that dataset.

Method	**Concrete** $n = 1{,}030$ max depth = 6	**Energy** $n = 768$ max depth = 6	**Parkinson's** $n = 5{,}875$ max depth = 8
Standard Random Forest	99.61	9.07	1649.38
FF- Forest-wise	586.70 (5.9×)	54.21 (6.0×)	9913.34 (6.0×)
FF- Tree-wise	558.56 (5.6×)	64.49 (7.1×)	9292.42 (5.6×)
FF- Random Node-wise	456.40 (4.6×)	30.03 (3.3×)	8033.43 (4.9×)
FF- Optimal Node-wise	516.28 (5.2×)	32.36 (3.6×)	8826.40 (5.4×)

time than the forest-wise and tree-wise variants, and random node-wise was consistently the fastest of the four. This is likely because the forest-wise and tree-wise variants evaluate multiple full candidate trees (one per candidate γ value) before selecting the best one, so the total number of node-splitting operations performed is a multiple of $|\Gamma|$ times that of a single tree. In contrast, each node-wise tree performs only one splitting search per node, regardless of $|\Gamma|$, and although the search at each node considers all candidate split variables, split points, and γ values jointly, this is carried out on the – typically much smaller – subset of observations contained in that node rather than on the full bootstrap sample. As splitting proceeds down the tree, node sizes shrink rapidly, so the computational cost of each successive split decreases even though the search space per split is larger than in the forest-wise and tree-wise variants. The optimal node-wise variant is somewhat slower than random node-wise,

consistent with the fact that it performs a joint search over split variable, split point, and γ at each node, whereas random node-wise simply draws γ at random and then searches only over the split variable and split point.

These results demonstrate that the optimal FlexForest variant is data-dependent, and suggest that the complexity of the underlying predictor-response relationship is an important factor in determining which level of flexibility such as tree, forest, or node provides the greatest predictive benefit. Motivated by this, future work could explore data-adaptive selection of the appropriate flexibility level, allowing the degree of flexibility to be chosen automatically based on the complexity of the data.

Chapter 5

Conclusion and Future Work

In this chapter, we provide a summary of the work presented in this thesis. We then discuss possible directions for future work to expand on the findings of this thesis.

5.1 Summary

In this thesis, we proposed a new proposed weighted splitting criterion, governed by a weighting parameter γ. An important feature of this formulation is that it enables direct comparison across different values of γ, since the left and right child-node weights are constructed to sum to one. When $\gamma = 1$, the proposed criterion reduces exactly to the standard CART splitting criterion. The proposed framework is therefore a generalization of the existing CART splitting criterion.

Building on this criterion, we introduced four algorithmic variants of a flexible random forest, which we call FlexForest. These variants differ according to the level at which flexibility is introduced, that is, the level at which the weighting parameter γ is tuned. The four proposed variants are forest-wise, tree-wise, optimal node-wise, and random node-wise. When $\gamma = 1$, the forest-wise FlexForest variant recovers the standard CART-based random forest.

We further compare training-based and bootstrap-based tuning of γ, used in the forest-wise and tree-wise variants respectively, with tuning based on an independent validation set (and, for the tree-wise variant, OOB sample). Through a small simulation study, we found that test errors under training-based and bootstrap-based tuning were highly correlated with those under independent validation-based tuning and did not show a systematic tendency for the learning-based approach to produce higher test errors than the independent-data-based approach. This suggests that, in the settings examined, learning-based tuning does not exhibit a clear systematic bias, offering a practical advantage: it allows a larger proportion of the available data to be allocated to model fitting rather than withholding a separate portion for parameter tuning.

In Chapter 3, we evaluated the proposed FlexForest variants through an extensive simulation study, varying the number of predictors, covariate distribution, covariate dependence structure, the functional form of the predictor and response relationship (linear versus nonlinear), and the noise level. We found that under nonlinear relationships regardless of the covariate distribution or noise level the tree-wise FlexForest variant consistently outperformed the stan-

dard random forest. We further considered a highly complex data-generating mechanism based on the Random Function Generator framework (Friedman, 2001). Under this setting, the most notable performance gains were achieved by the variants that introduce flexibility at the node level, indicating that when the underlying data structure is maximally complex, allowing γ to vary at the node level provides a better fit than fixing γ at the tree or forest level.

In Chapter 4, we evaluated the proposed methods on real data applications, using three publicly available datasets with only numerical predictors: the Concrete Compressive Strength dataset, the Energy Efficiency dataset and the Parkinson's Telemonitoring dataset. For the concrete compressive data set and the energy efficiency data set, the tree-wise FlexForest variant achieved the best predictive performance among the proposed variants, consistent with the simulation findings for nonlinear settings and for the Parkinson's Telemonitoring dataset the node-wise FlexForest variant achieved the best predictive performance among the proposed variants, consistent with the simulation findings for complex nonlinear setting.

5.2 Limitations and Future Work

Despite this thesis demonstrates the potential benefits of the proposed FlexForest framework, several limitations should be acknowledged, along with directions for addressing them in future work.

The proposed framework was developed and evaluated exclusively for regression problems, where all response and predictor variables are continuous and the squared loss serves as the splitting criterion. The classification setting, which would require an analogous γ-weighted formulation of the Gini impurity or entropy criterion, was not addressed. Extending the framework to classification represents a natural direction for future work.

The framework was developed exclusively for the univariate regression setting. Extending the γ-weighted splitting criterion to multi-response or multivariate random forests would considerably broaden its applicability and provide another promising research direction.

In the tree construction procedure, node splitting was controlled solely through a maximum depth criterion. Investigating a minimum node size stopping criterion, which is commonly used in standard random forest implementations could be investigated in the future.

The performance improvements of the proposed FlexForest variants have been established empirically through simulation studies and real data applications. However, a formal theoretical analysis of the γ-weighted splitting criterion has not yet been developed. In particular, theoretical properties such as consistency and convergence remain to be established, and the upper bound or optimal range of the tuning parameter γ has not been theoretically characterized. The choice of candidate set Γ was determined empirically. Developing a theoretical framework for selecting or bounding γ would strengthen

the proposed methodology.

A further direction concerns variable importance measures for the FlexForest framework. Variable importance is one of the most practically valued features of the standard random forest, providing a data-driven measure of each predictor's contribution to predictive performance (Breiman, 2001). In the FlexForest framework, the γ-weighted splitting criterion modifies the weighted variance used to evaluate candidate splits, which may influence the frequency and location of predictor selections during tree construction and therefore affect the validity of standard impurity-based importance measures. A potential future direction is to develop a FlexForest-specific variable importance measure. For example, a weighted selection frequency approach that accounts for node depth and the number of observations in each node could better reflect the contribution of predictors under the adaptive splitting mechanism of FlexForest.

All computations were carried out sequentially: trees were grown one at a time, and candidate γ values were evaluated one at a time, with parallelization used only across replications and train-test splits via job arrays. Since trees within a forest, and candidate γ values within a variant, are largely independent computations, substantial reductions in running time could be achieved by parallelizing across them. For the forest-wise variant in particular, this could be done at two levels simultaneously: across the candidate forests corresponding to different γ values, and across the trees within each forest. For the node-wise variants, an additional sequential dependency exists between parent and child nodes within a tree, but the search over candidate splits and γ values at each

individual node is itself parallelizable. Implementing and benchmarking these optimizations is left for future work.

Finally, although the proposed method was evaluated using three publicly available real-world datasets with varying sample sizes and predictor dimensions, its practical performance has been assessed on only a limited number of applications. Future work should evaluate the proposed method on a broader collection of real-world datasets spanning a wider range of sample sizes and numbers of predictors. Such an evaluation would provide a more comprehensive assessment of the method's robustness, scalability, and generalizability across diverse application domains.

Bibliography

Biau, G. and E. Scornet (2016). A random forest guided tour. *Test 25*(2), 197–227. (Cited on page 7.)

Boulesteix, A.-L., S. Janitza, J. Kruppa, and I. R. König (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2*(6), 493–507. (Cited on page 1.)

Breiman, L. (2001). Random forests. *Machine learning 45*(1), 5–32. (Cited on pages 1, 6, 7, 8, 42, 54, 91, 102 and 103.)

Breiman, L., J. Friedman, R. A. Olshen, and C. J. Stone (2017). *Classification and regression trees*. Chapman and Hall/CRC. (Cited on pages 4, 5, 7, 8 and 9.)

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232. (Cited on pages 47, 53, 54, 65 and 89.)

Grinsztajn, L., E. Oyallon, and G. Varoquaux (2022). Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems 35*, 507–520. (Cited on pages 1 and 6.)

Hastie, T. (2009). The elements of statistical learning: data mining, inference, and prediction. (Cited on pages 1, 4, 5, 6 and 7.)

Ishwaran, H. (2015). The effect of splitting on random forests. *Machine learning 99*(1), 75–118. (Cited on pages 2, 7, 8, 9, 38, 42, 97 and 98.)

Joe, H. (2014). *Dependence modeling with copulas*. CRC press. (Cited on page 40.)

Schmid, L., A. Gerharz, A. Groll, and M. Pauly (2023). Tree-based ensembles for multi-output regression: Comparing multivariate approaches with separate univariate ones. *Computational Statistics & Data Analysis 179*, 107628. (Cited on pages 38 and 42.)

Smith, H. L., P. J. Biggs, N. P. French, A. N. Smith, and J. C. Marshall (2024). Out of (the) bag—encoding categorical predictors impacts out-of-bag samples. *PeerJ Computer Science 10*, e2445. (Cited on page 28.)

Su, X., G. E. Quaye, Y. Wei, J. Kang, L. Liu, Q. Yang, J. Fan, and R. A. Levine (2024). Smooth sigmoid surrogate (sss): An alternative to greedy search in decision trees. *Mathematics 12*(20), 3190. (Cited on page 3.)

Tsanas, A. and M. Little (2009). Parkinsons Telemonitoring. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5ZS3N. (Cited on page 76.)

Tsanas, A. and A. Xifara (2012). Energy Efficiency. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C51307. (Cited on page 74.)

Yeh, I.-C. (1998). Concrete Compressive Strength. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5PK67. (Cited on page 72.)

Zhang, Z. and H. Luo (2025). Stabilizing the splits through minimax decision trees. *arXiv preprint arXiv:2502.16758*. (Cited on page 2.)

Index

acknowledgment, i

dedication, ii

thesis regulations, i, ii

    acknowledgment, i

    dedication, ii

Appendix A

Proofs

A.1 Comparison of Existing Child-Node Weighting Schemes

In this section, we present a mathematical proof that, for a fixed candidate split, Ishwaran (2015)'s heavy-weighted scheme yields the smallest weighted variance among the unweighted, weighted, and heavy-weighted schemes. Since the weighting scheme only rescales the weighted variance for a given candidate split, it does not alter the resulting split selected under the unadjusted weighting framework.

Proposition 1. *Let s denote a fixed candidate split of node t, defined by a splitting variable and a split point. Among the unweighted, weighted, and heavy-weighted schemes, the heavy-weighted scheme yields the minimum weighted variance.*

Proof. To facilitate the comparison of the existing weighting schemes, let $\widehat{D}(s, t : \alpha)$ denote the weighted variance criterion associated with a candidate split s at node t, where α controls the child-node weighting scheme. Specifically, $\alpha = 0$, $\alpha = 1$, $\alpha = 2$ correspond to the Ishwaran (2015)'s unweighted, proportion-weighted, and heavy-weighted schemes, respectively. For a candidate split s at node t, resulting in left and right child nodes t_L and t_R, define the weighted split criterion as

$$\widehat{D}(s, t; \alpha) = w_L^{\alpha}\widehat{\sigma}(t_L) + w_R^{\alpha}\widehat{\sigma}(t_R),$$

where w_L and w_R are the corresponding proportions of the left and right child nodes. The parameter α controls the weighting scheme applied to the child nodes.

Let consider the unweighted weighting scheme,

$$\widehat{D}\left(s, t; \alpha = 0\right) = \widehat{\sigma}(t_L) + \widehat{\sigma}(t_R)\ .$$

This is equivalent to,

$$\widehat{D}\left(s, t; 0\right) = \frac{1}{n_L}\sum_{i \in t_R}\left(y_i - \bar{y}_{t_L}\right)^2 + \frac{1}{n_R}\sum_{i \in t_R}\left(y_i - \bar{y}_{t_R}\right)^2.$$

Similarly, the normal weighted or porportion-weighted scheme as follows,

$$\widehat{D}\left(s, t, \alpha = 1\right) = \widehat{w}(t_L)\widehat{\sigma}(t_L) + \widehat{w}(t_R)\widehat{\sigma}(t_R)\ .$$

This is equivalent to,

$$\widehat{D}\left(s,t;\ 1\right) = \frac{n_L}{n}\frac{1}{n_L}\sum_{i\in t_R}\left(y_i - \bar{y}_{t_L}\right)^2 + \frac{n_R}{n}\frac{1}{n_R}\sum_{i\in t_R}\left(y_i - \bar{y}_{t_R}\right)^2$$

$$= \frac{1}{n}\sum_{i\in t_R}\left(y_i - \bar{y}_{t_L}\right)^2 + \frac{1}{n}\sum_{i\in t_R}\left(y_i - \bar{y}_{t_R}\right)^2$$

Since $n > n_L$ and $n > n_R$ and also $\sum_{i\in t_R}\left(y_i - \bar{y}_{t_L}\right)^2$ and $\sum_{i\in t_R}\left(y_i - \bar{y}_{t_R}\right)^2$ is fixed for particular split point s.

Then,

$$\frac{1}{n}\sum_{i\in t_R}\left(y_i - \bar{y}_{t_L}\right)^2 < \frac{1}{n_L}\sum_{i\in t_R}\left(y_i - \bar{y}_{t_L}\right)^2$$

and similarly,

$$\frac{1}{n}\sum_{i\in t_R}\left(y_i - \bar{y}_{t_R}\right)^2 < \frac{1}{n_L}\sum_{i\in t_R}\left(y_i - \bar{y}_{t_R}\right)^2$$

Therefore, $\widehat{D}\left(s,t;\ 1\right) < \widehat{D}\left(s,t;\ 0\right)$.

Let us consider the heavily weighted scheme,

$$\widehat{D}\left(s,t;\alpha = 2\right) = \widehat{w}(t_L)^2\widehat{\Delta}(t_L) + \widehat{w}(t_R)^2\widehat{\Delta}(t_R)$$

This is equivalent to,

$$\widehat{D}\left(s,t;\ 2\right) = \left(\frac{n_L}{n}\right)^2 \frac{1}{n_L} \sum_{i \in t_R} \left(y_i - \bar{y}_{t_L}\right)^2 + \left(\frac{n_R}{n}\right)^2 \frac{1}{n_R} \sum_{i \in t_R} \left(y_i - \bar{y}_{t_L}\right)^2$$

$$= \frac{n_L}{n^2} \sum_{i \in t_R} \left(y_i - \bar{y}_{t_L}\right)^2 + \frac{n_R}{n^2} \sum_{i \in t_R} \left(y_i - \bar{y}_{t_L}\right)^2$$

Let's consider,

$$\widehat{D}\left(s,t;\ 1\right) - \widehat{D}\left(s,t;\ 2\right) = \frac{1}{n} \sum_{i \in t_R} \left(y_i - \bar{y}_{t_L}\right)^2 + \frac{1}{n} \sum_{i \in t_R} \left(y_i - \bar{y}_{t_R}\right)^2$$

$$- \left[\frac{n_L}{n^2} \sum_{i \in t_R} \left(y_i - \bar{y}_{t_L}\right)^2 + \frac{n_R}{n^2} \sum_{i \in t_R} \left(y_i - \bar{y}_{t_L}\right)^2\right]$$

$$= \sum_{i \in t_R} \left(y_i - \bar{y}_{t_R}\right)^2 \left(\frac{1}{n} - \frac{n_L}{n^2}\right) + \sum_{i \in t_R} \left(y_i - \bar{y}_{t_L}\right)^2 \left(\frac{1}{n} - \frac{n_R}{n^2}\right)$$

Let's consider,

$$\frac{1}{n} - \frac{n_L}{n^2} = \frac{n - n_L}{n^2} = \frac{n_L + n_R - n_L}{n^2} = \frac{n_R}{n^2} > 0$$

Similarly,

$$\frac{1}{n} - \frac{n_R}{n^2} = \frac{n - n_R}{n^2} = \frac{n_L + n_R - n_R}{n^2} = \frac{n_L}{n^2} > 0$$

Therefore,

$$\widehat{D}\left(s,t;\ 1\right) - \widehat{D}\left(s,t;\ 2\right) > 0$$

This implies $\widehat{D}\,(s,t;\ 1) > \widehat{D}\,(s,t;\ 2)$.

Hence, a heavily weighted scheme always gives the lowest weighted variance.

□

A.2 Recovery of Standard CART

In this section, we show that the proposed proposed weighted splitting criterion reduces to the standard CART splitting criterion when $\gamma = 1$.

Recall from Chapter 2 (Equation (2.1)) that, for a splitting variable j and split point c at node t, the γ-weighted splitting criterion is denoted by $m(j, c, \gamma)$. Setting $\gamma = 1$ yields the standard CART splitting criterion, as established in the following proposition.

Proposition 2. *For any candidate split defined by splitting variable j and split point c, the proposed weighted splitting criterion satisfies*

$$m(j, c, 1) = m_{\text{CART}}(j, c) = \left(\frac{n_L}{n}\right)\hat{\sigma}^2(t_L) + \left(\frac{n_R}{n}\right)\hat{\sigma}^2(t_R),$$

where $m_{\text{CART}}(j, c)$ denotes the standard CART splitting criterion.

Proof. Let's consider $m(j, c, \gamma)$ when $\gamma = 1$. Where $\widehat{w}^{\gamma}_{(t_L)}$, and $\widehat{w}^{\gamma}_{(t_R)}$ are the γ power of the left node proportion and right node proportions, respectively . Then,

$$
\begin{aligned}
m(j, c, \gamma = 1) &= \underbrace{\left(\frac{\widehat{w}_{(t_L)}}{\widehat{w}_{(t_L)} + \widehat{w}_{(t_R)}} \right)}_{\widehat{w}^*_{(t_L, \gamma=1)}} \hat{\sigma}^2(t_L) + \underbrace{\left(\frac{\widehat{w}_{(t_R)}}{\widehat{w}_{(t_L)} + \widehat{w}_{(t_R)}} \right)}_{\widehat{w}^*_{(t_R, \gamma=1)}} \hat{\sigma}^2(t_R) \\
&= \left(\frac{n_L/n}{n_L/n + n_R/n} \right) \hat{\sigma}^2(t_L) + \left(\frac{n_R/n}{n_L/n + n_R/n} \right) \hat{\sigma}^2(t_R) \\
&= \left(\frac{n_L}{n_L + n_R} \right) \hat{\sigma}^2(t_L) + \left(\frac{n_R}{n_L + n_R} \right) \hat{\sigma}^2(t_R) \\
&= \left(\frac{n_L}{n} \right) \hat{\sigma}^2(t_L) + \left(\frac{n_R}{n} \right) \hat{\sigma}^2(t_R)
\end{aligned}
$$

□

A.3 Recovery of Standard Random Forest

In this section, we show that the proposed forest-wise FlexForest variant reduces to the standard CART Random Forest when $\gamma = 1$.

Proposition 3. *When $\gamma = 1$, the forest-wise FlexForest variant reduces to the standard random forest of Breiman (2001).*

Proof. In the forest-wise FlexForest variant, a single value $\gamma^* \in \Gamma$ is selected for the entire forest, and all trees are grown using the γ^*-weighted splitting criterion. It was established in Proposition 2 that when $\gamma = 1$, the proposed proposed weighted splitting criterion:

$$
m(j, c, \gamma) = \hat{w}^*_{t_L, \gamma} \, \hat{\sigma}^2(t_L) + \hat{w}^*_{t_R, \gamma} \, \hat{\sigma}^2(t_R),
$$

reduces to the standard CART weighted variance criterion:

$$\frac{n_L}{n_t}\hat{\sigma}^2(t_L) + \frac{n_R}{n_t}\hat{\sigma}^2(t_R).$$

Therefore, when $\gamma^* = 1$, every split in every tree T_b, $b = 1, \ldots, B$, is selected by minimizing the standard CART criterion. Since each tree T_b is grown on a bootstrap sample $\mathcal{D}_b$ drawn with replacement from the training data, with a randomly selected subset of h candidate predictors considered at each node, the resulting ensemble is identical in construction to the standard random forest of Breiman (2001). The forest-wise FlexForest prediction:

$$\hat{f}_{\mathrm{FF}}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}_b(\mathbf{x})$$

then coincides with the standard random forest prediction:

$$\hat{f}_{\mathrm{RF}}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}_b(\mathbf{x}),$$

since all B trees are grown identically under the standard CART criterion. Hence, the forest-wise FlexForest variant with $\gamma = 1$ is equivalent to the standard random forest. □

Appendix B

Weighting Parameter Tuning Extra Results

B.1 Comparison of Training-Based and Validation-Based Tuning for Forest-wise Variant

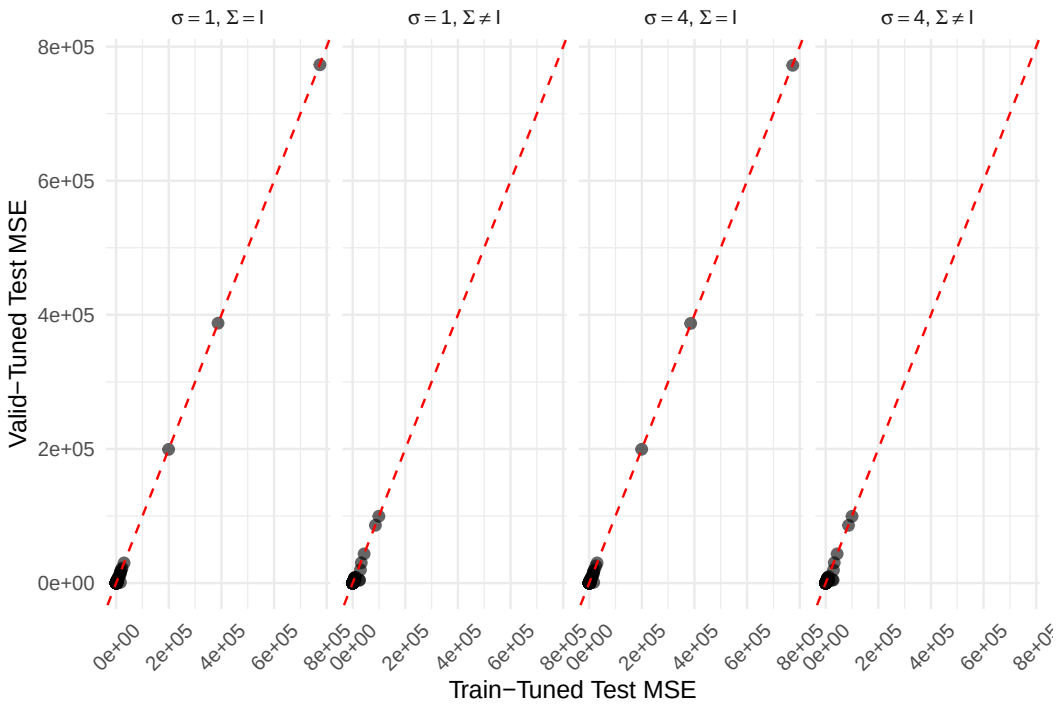


Figure B.1: Data generation from exponential covariates with a nonlinear relationship, forest-wise variant tuning comparison.

B.2 Comparison of Training-Based and Validation-Based Tuning for Tree-wise Variant

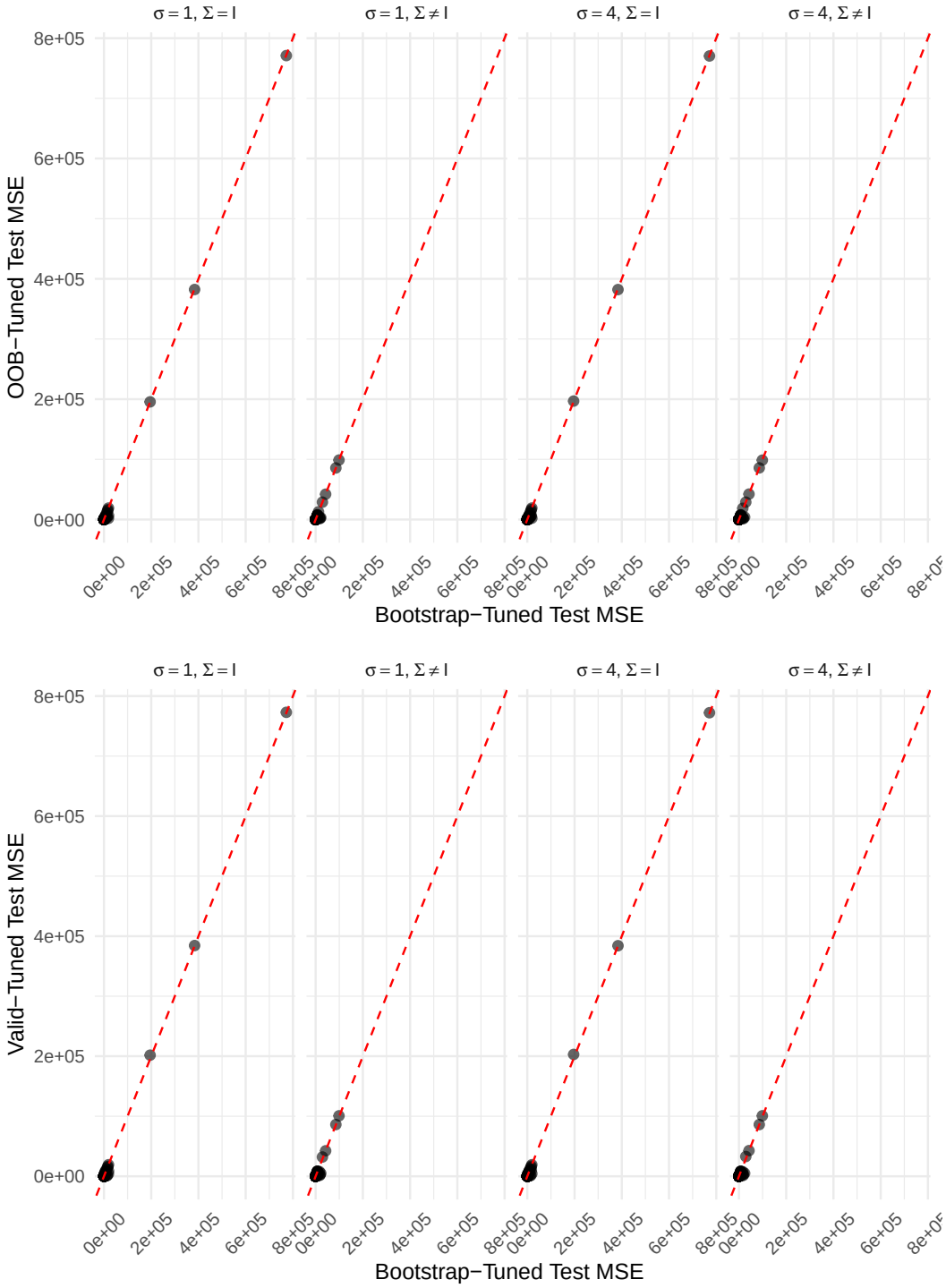


Figure B.2: Data generation from exponential covariates with a nonlinear relationship, tree-wise variant tuning comparison.

Appendix C

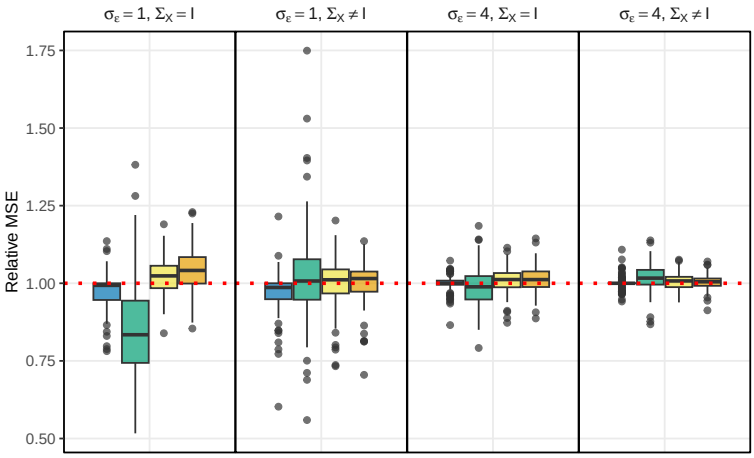
Simulation Results for Sample Size $n = 100$

C.1 Normal Linear

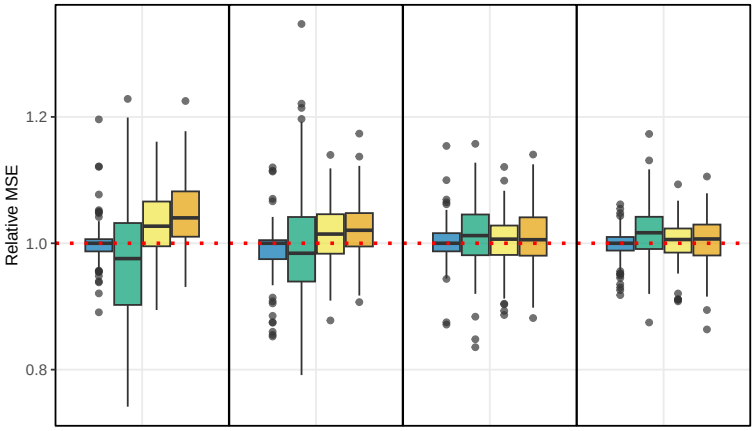
C.2 Normal Nonlinear

C.3 Exponential Linear

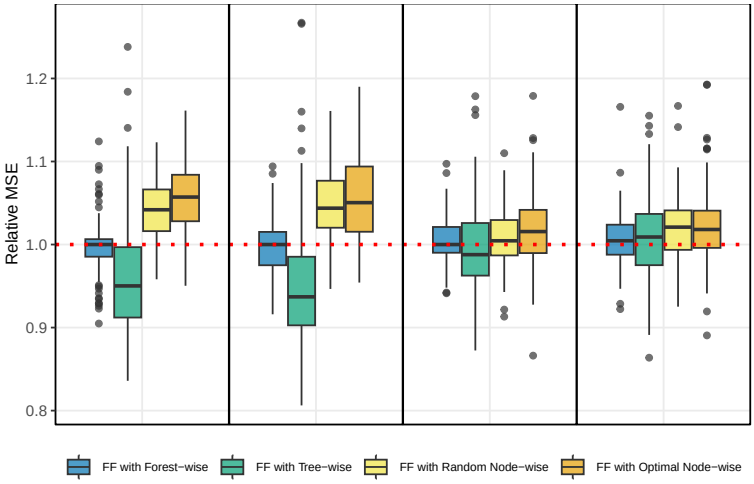
C.4 Exponential Linear



(a) Three predictors ($p = 3$).

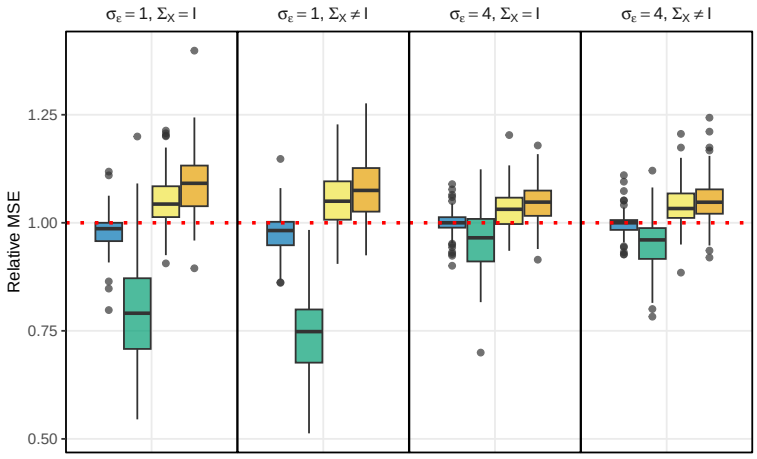


(b) Five predictors ($p = 5$).

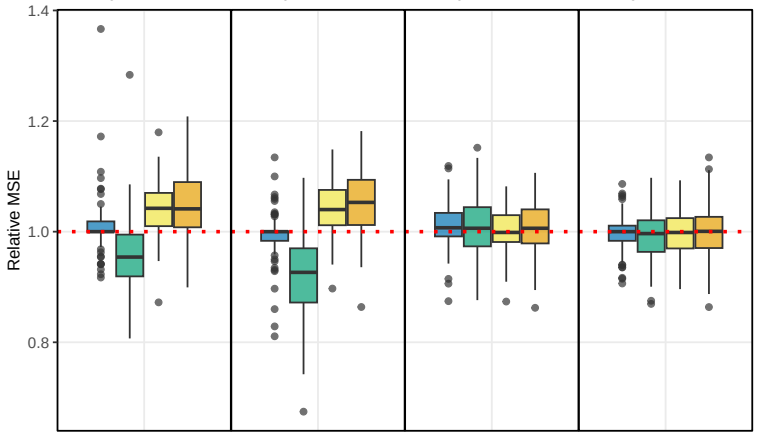


(c) Ten predictors ($p = 10$).

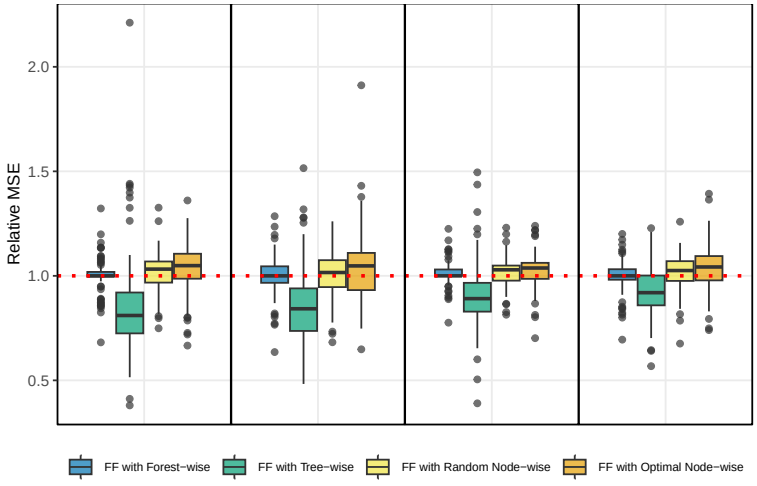
Figure C.1: RPE distributions for the **normal linear** scenario across different noise levels and response dependence structures, based on 100 simulation replications. Each panel compares the four FlexForest variants, with the dashed horizontal line indicating RPE = 1. The three subfigures correspond to datasets with $p = 3$, 5, and 10 predictors, respectively.



(a) Three predictors ($p = 3$).

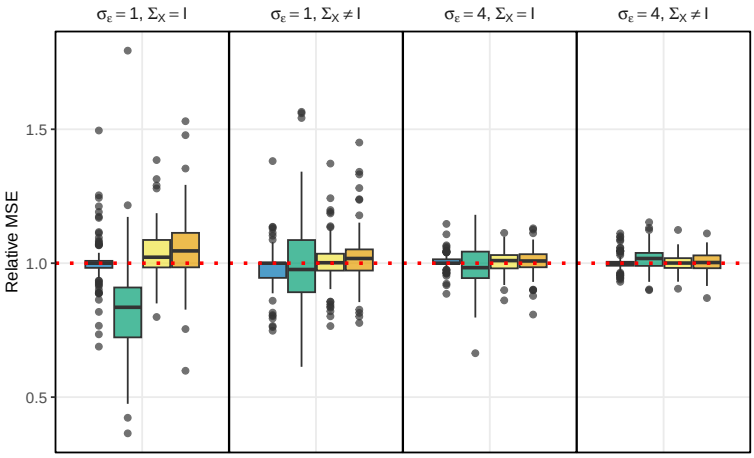


(b) Five predictors ($p = 5$).

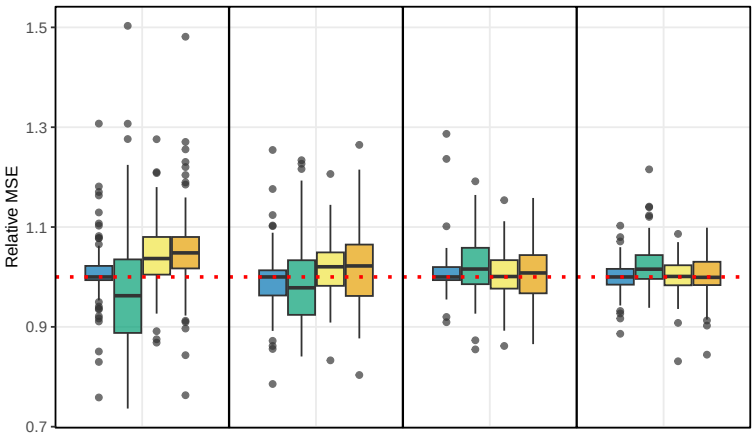


(c) Ten predictors ($p = 10$).

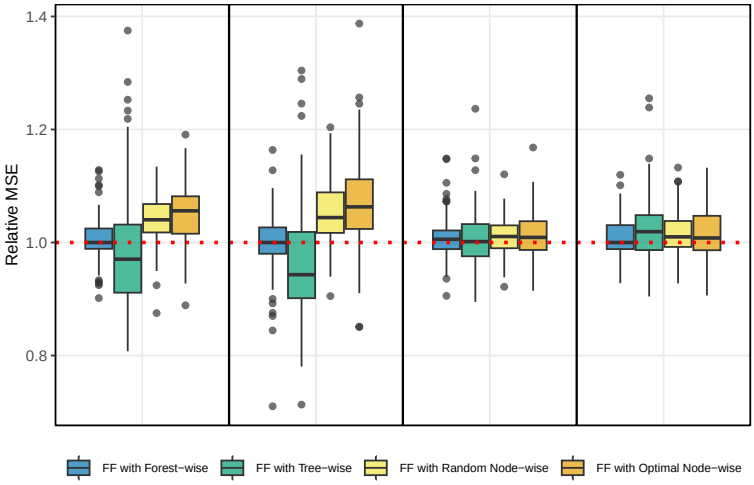
Figure C.2: RPE distributions for the **normal nonlinear** scenario across different noise levels and response dependence structures, based on 100 simulation replications. Each panel compares the four FlexForest variants, with the dashed horizontal line indicating RPE = 1. The three subfigures correspond to datasets with $p = 3$, 5, and 10 predictors, respectively.



(a) Three predictors ($p = 3$).

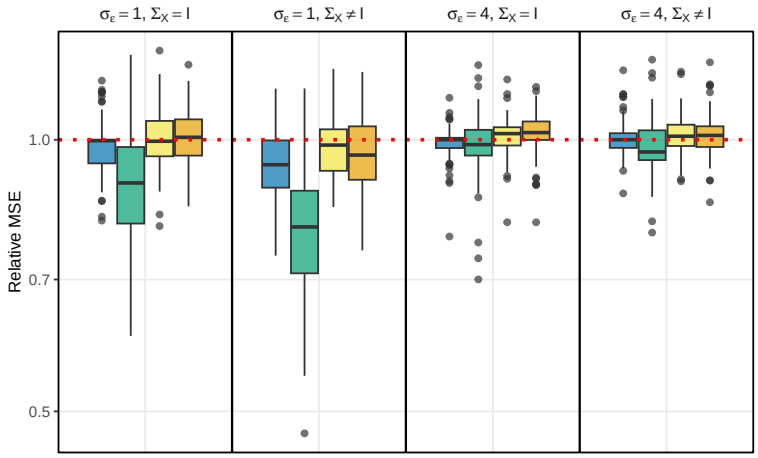


(b) Five predictors ($p = 5$).

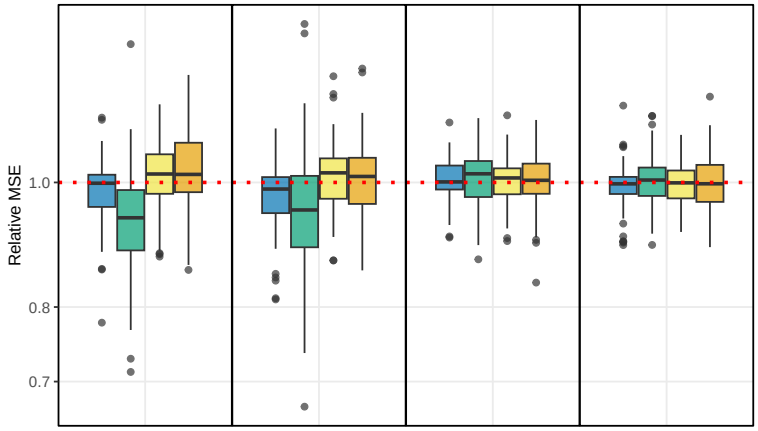


(c) Ten predictors ($p = 10$).

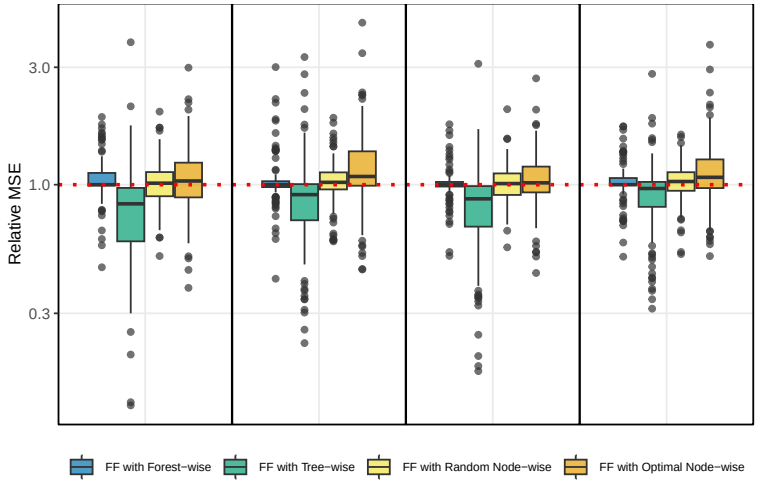
Figure C.3: RPE distributions for the **exponential linear** scenario across different noise levels and response dependence structures, based on 100 simulation replications. Each panel compares the four FlexForest variants, with the dashed horizontal line indicating RPE = 1. The three subfigures correspond to datasets with $p = 3$, 5, and 10 predictors, respectively.



(a) Three predictors ($p = 3$).



(b) Five predictors ($p = 5$).



(c) Ten predictors ($p = 10$).

Figure C.4: RPE distributions for the **exponential nonlinear** scenario across different noise levels and response dependence structures, based on 100 simulation replications. Each panel compares the four FlexForest variants, with the dashed horizontal line indicating RPE = 1. The three subfigures correspond to datasets with $p = 3$, 5, and 10 predictors, respectively.

Appendix D

Forest-Wise γ Selection

D.1 Normal Linear

Table D.1: Selection frequency (out of 100 replications) of the optimal γ value selected by the forest-wise FlexForest (FF-F) under the normal linear scenario.

p	**Scenario**	**0.5**	**1**	**1.5**	**2**	**2.5**	**3**
3	$\sigma_\varepsilon = 1,\ \Sigma \neq \mathbf{I}$	19	63	15	3	0	0
	$\sigma_\varepsilon = 1,\ \Sigma = \mathbf{1}$	0	86	12	2	0	0
	$\sigma_\varepsilon = 4,\ \Sigma \neq \mathbf{I}$	9	91	0	0	0	0
	$\sigma_\varepsilon = 4,\ \Sigma = \mathbf{1}$	0	4	95	1	0	0
5	$\sigma_\varepsilon = 1,\ \Sigma \neq \mathbf{I}$	0	96	4	0	0	0
	$\sigma_\varepsilon = 1,\ \Sigma = \mathbf{1}$	0	97	3	0	0	0
	$\sigma_\varepsilon = 4,\ \Sigma \neq \mathbf{I}$	0	0	100	0	0	0
	$\sigma_\varepsilon = 4,\ \Sigma = \mathbf{1}$	0	0	100	0	0	0
10	$\sigma_\varepsilon = 1,\ \Sigma \neq \mathbf{I}$	0	84	15	1	0	0
	$\sigma_\varepsilon = 1,\ \Sigma = \mathbf{1}$	0	92	8	0	0	0
	$\sigma_\varepsilon = 4,\ \Sigma \neq \mathbf{I}$	0	99	1	0	0	0
	$\sigma_\varepsilon = 4,\ \Sigma = \mathbf{1}$	0	0	100	0	0	0

D.2 Normal Nonlinear

Table D.2: Selection frequency (out of 100 replications) of the optimal γ value selected by the forest-wise FlexForest (FF-F) under the normal nonlinear scenario.

p	**Scenario**	**0.5**	**1**	**1.5**	**2**	**2.5**	**3**
3	$\sigma_\varepsilon = 1,\ \Sigma \neq \mathbf{I}$	0	23	38	20	9	10
	$\sigma_\varepsilon = 1,\ \Sigma = \mathbf{1}$	0	35	42	19	4	0
	$\sigma_\varepsilon = 4,\ \Sigma \neq \mathbf{I}$	0	70	29	1	0	0
	$\sigma_\varepsilon = 4,\ \Sigma = \mathbf{1}$	0	0	85	15	0	0
5	$\sigma_\varepsilon = 1,\ \Sigma \neq \mathbf{I}$	0	83	17	0	0	0
	$\sigma_\varepsilon = 1,\ \Sigma = \mathbf{1}$	0	94	6	0	0	0
	$\sigma_\varepsilon = 4,\ \Sigma \neq \mathbf{I}$	0	94	6	0	0	0
	$\sigma_\varepsilon = 4,\ \Sigma = \mathbf{1}$	0	97	3	0	0	0
10	$\sigma_\varepsilon = 1,\ \Sigma \neq \mathbf{I}$	0	67	21	7	4	1
	$\sigma_\varepsilon = 1,\ \Sigma = \mathbf{1}$	0	64	29	4	3	0
	$\sigma_\varepsilon = 4,\ \Sigma \neq \mathbf{I}$	0	73	21	4	2	0
	$\sigma_\varepsilon = 4,\ \Sigma = \mathbf{1}$	0	77	21	2	0	0

D.3 Exponential Linear

Table D.3: Selection frequency (out of 100 replications) of the optimal γ value selected by the forest-wise FlexForest (FF-F) under the exponential linear scenario.

p	**Scenario**	**0.5**	**1**	**1.5**	**2**	**2.5**	**3**
3	$\sigma_\varepsilon = 1,\ \Sigma \neq \mathbf{I}$	5	60	28	6	1	0
	$\sigma_\varepsilon = 1,\ \Sigma = \mathbf{1}$	0	78	19	3	0	0
	$\sigma_\varepsilon = 4,\ \Sigma \neq \mathbf{I}$	1	98	1	0	0	0
	$\sigma_\varepsilon = 4,\ \Sigma = \mathbf{1}$	0	94	6	0	0	0
5	$\sigma_\varepsilon = 1,\ \Sigma \neq \mathbf{I}$	0	68	28	4	0	0

p	**Scenario**	**0.5**	**1**	**1.5**	**2**	**2.5**	**3**
	$\sigma_\varepsilon = 1,\ \Sigma = \mathbf{1}$	0	81	16	3	0	0
	$\sigma_\varepsilon = 4,\ \Sigma \neq \mathbf{I}$	0	100	0	0	0	0
	$\sigma_\varepsilon = 4,\ \Sigma = \mathbf{1}$	0	100	0	0	0	0
10	$\sigma_\varepsilon = 1,\ \Sigma \neq \mathbf{I}$	0	74	26	0	0	0
	$\sigma_\varepsilon = 1,\ \Sigma = \mathbf{1}$	0	94	6	0	0	0
	$\sigma_\varepsilon = 4,\ \Sigma \neq \mathbf{I}$	0	99	1	0	0	0
	$\sigma_\varepsilon = 4,\ \Sigma = \mathbf{1}$	0	99	1	0	0	0

D.4 Exponential Nonlinear

Table D.4: Selection frequency (out of 100 replications) of the optimal γ value selected by the forest-wise FlexForest (FF-F) under the exponential nonlinear scenario.

p	**Scenario**	**0.5**	**1**	**1.5**	**2**	**2.5**	**3**
3	$\sigma_\varepsilon = 1,\ \Sigma \neq \mathbf{I}$	0	3	31	58	8	0
	$\sigma_\varepsilon = 1,\ \Sigma = \mathbf{1}$	0	48	47	5	0	0
	$\sigma_\varepsilon = 4,\ \Sigma \neq \mathbf{I}$	0	25	71	4	0	0
	$\sigma_\varepsilon = 4,\ \Sigma = \mathbf{1}$	0	86	14	0	0	0
5	$\sigma_\varepsilon = 1,\ \Sigma \neq \mathbf{I}$	0	11	81	8	0	0
	$\sigma_\varepsilon = 1,\ \Sigma = \mathbf{1}$	0	7	85	8	0	0
	$\sigma_\varepsilon = 4,\ \Sigma \neq \mathbf{I}$	0	84	16	0	0	0
	$\sigma_\varepsilon = 4,\ \Sigma = \mathbf{1}$	0	85	15	0	0	0
10	$\sigma_\varepsilon = 1,\ \Sigma \neq \mathbf{I}$	0	53	29	9	6	3
	$\sigma_\varepsilon = 1,\ \Sigma = \mathbf{1}$	0	45	30	16	7	2
	$\sigma_\varepsilon = 4,\ \Sigma \neq \mathbf{I}$	0	53	29	8	7	3
	$\sigma_\varepsilon = 4,\ \Sigma = \mathbf{1}$	0	47	31	13	6	3

D.5 Heavily Nonlinear

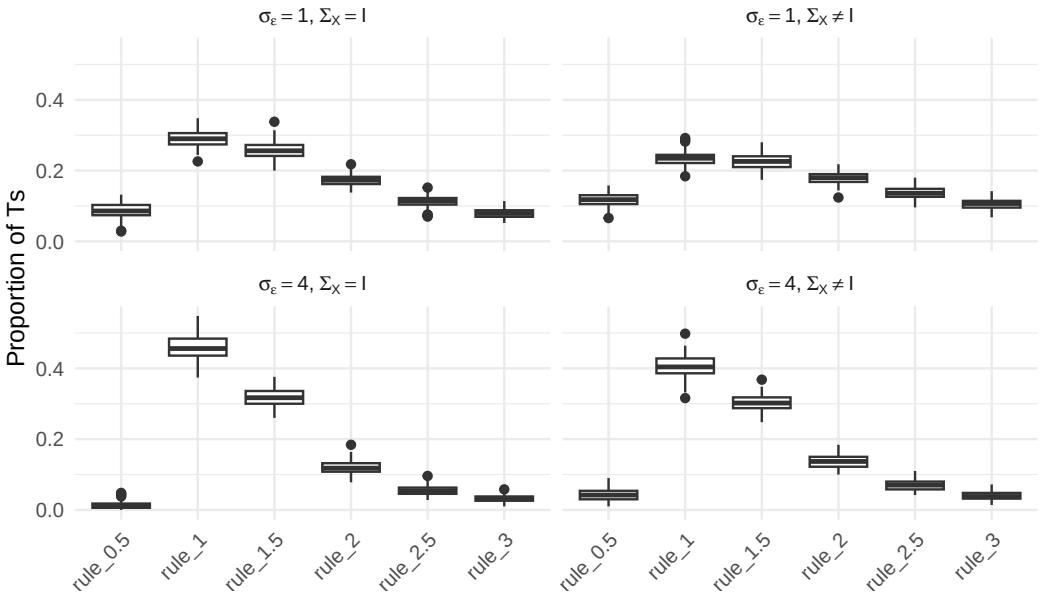
Table D.5: Selection frequency (out of 100 replications) of the optimal γ value selected by the forest-wise FlexForest (FF-F) under the heavily nonlinear scenario.

Scenario	**0.5**	**1**	**1.5**	**2**	**2.5**	**3**
$n = 100$, $p = 10$	0	45	35	20	0	0
$n = 500$, $p = 10$	0	45	55	0	0	0

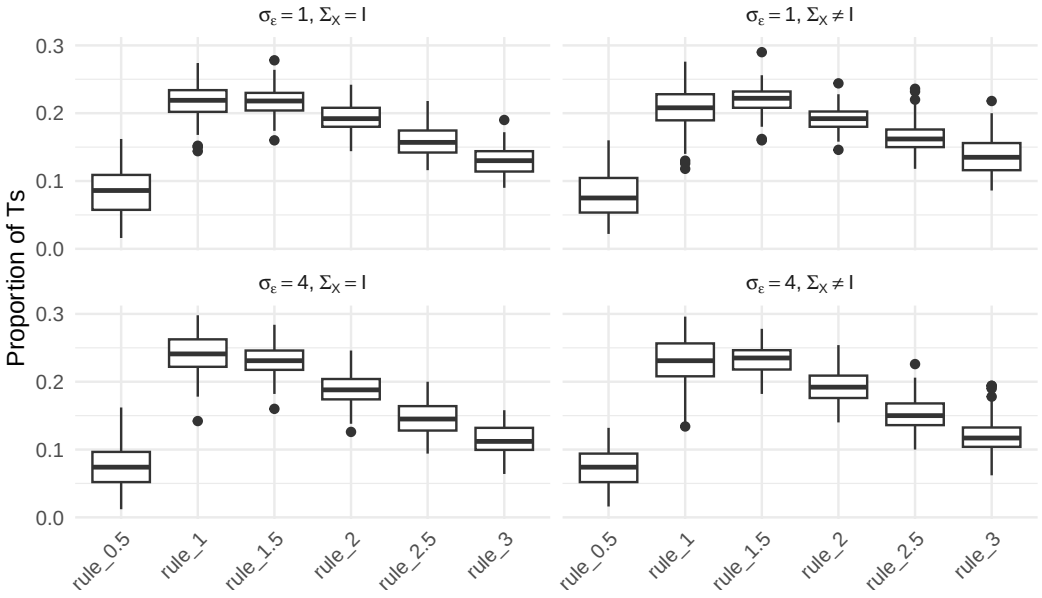
Appendix E

Behavior of the Tree-wise Variant: Distribution of Selected γ Values

To gain further insight into the mechanism by which the tree-wise variant achieves improved predictive performance, we examined the distribution of the weighting parameter γ^* selected by individual trees across replications. Recall that in the tree-wise variant, each tree independently selects its own optimal γ^* from the candidate set $\Gamma = \{\gamma_1, \ldots, \gamma_K\}$ by minimizing the bootstrap prediction error. For each replication, the proportion of trees selecting each candidate γ_k was recorded, yielding a distribution of weighting parameter selection proportions across the $B = 500$ trees.

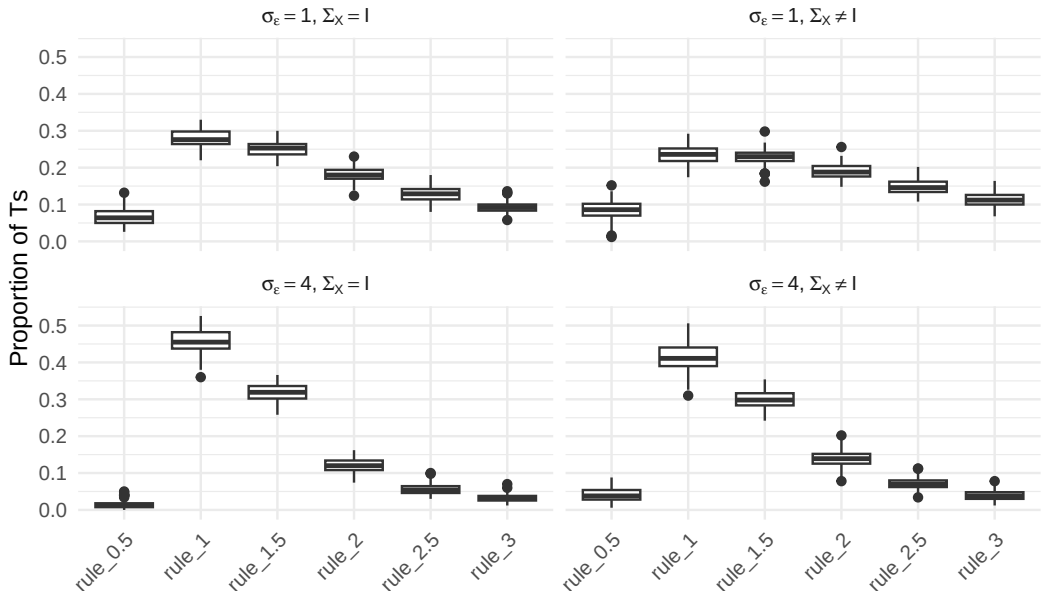


(a) Linear Relationship

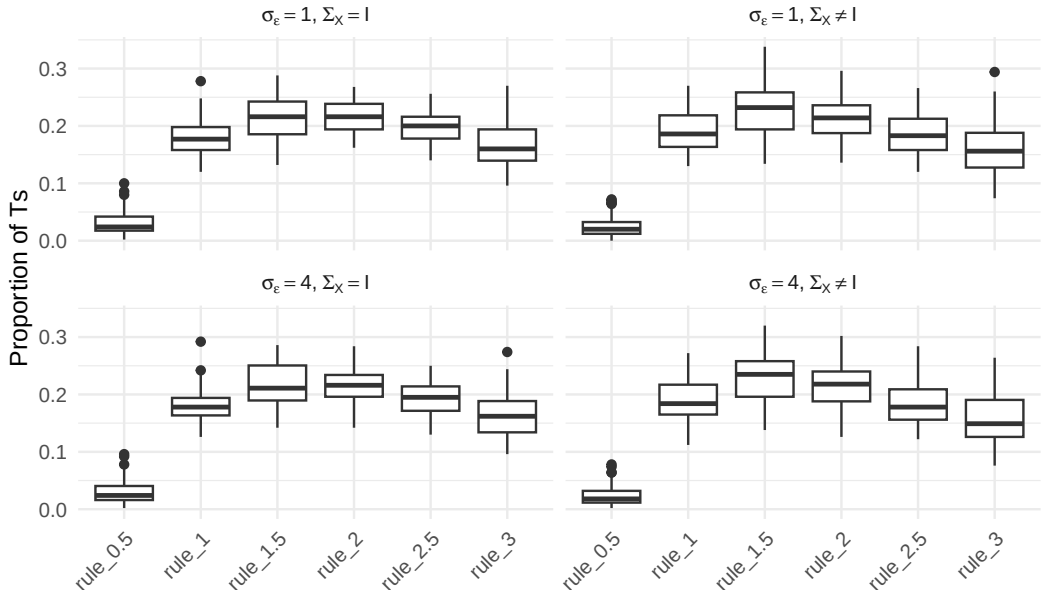


(b) Nonlinear Relationship.

Figure E.1: Distribution of the selected γ values for the **normal linear and nonlinear** scenario under different noise levels and covariance structures, based on 100 simulation replications. The y-axis represents the proportion of trees assigned to each splitting rule, and the x-axis represents the candidate γ values. Results are shown for $p = 10$.



(a) Linear Relationship



(b) Nonlinear Relationship

Figure E.2: Distribution of the selected γ values for the **exponential linear and nonlinear** scenario under different noise levels and covariance structures, based on 100 simulation replications. The y-axis represents the proportion of trees assigned to each splitting rule, and the x-axis represents the candidate γ values. Results are shown for $p = 10$.