*Naval Research Logistics (NRL)*

WILEY

RESEARCH ARTICLE OPEN ACCESS

# Improved Regression Tree Models Using Generalization Error-Based Splitting Criteria

Ying Yang[1] | Shuaian Wang[1] | Gilbert Laporte[2,3]

[1]Faculty of Business, The Hong Kong Polytechnic University, Hung Hom, Hong Kong | [2]Department of Decision Sciences, HEC Montréal, Montréal, Canada | [3]School of Management, University of Bath, Bath, UK

**Correspondence:** Shuaian Wang (hans.wang@polyu.edu.hk)

**Received:** 26 May 2024 | **Revised:** 9 May 2025 | **Accepted:** 16 May 2025

**Handling Editor:** Cong Shi

**Funding:** The authors received no specific funding for this work.

**Keywords:** generalization error | leave-one-out cross-validation | mean squared error | regression tree

## ABSTRACT

Despite the widespread application of machine learning (ML) approaches such as the regression tree (RT) in the field of data-driven optimization, overfitting may impair the effectiveness of ML models and thus hinder the deployment of ML for decision-making. In particular, we address the overfitting issue of the traditional RT splitting criterion with a limited sample size, which considers only the training mean squared error, and we accurately specify the mathematical formula for the generalization error. We introduce two novel splitting criteria based on generalization error, which offer higher-quality approximations of the generalization error than the traditional training error does. One criterion is formulated through a mathematical derivation based on the RT model, and the second is established through leave-one-out cross-validation (LOOCV). We construct RT models using our proposed generalization error-based splitting criteria from extensive ML benchmark instances and report the experimental results, including the models' computational efficiency, prediction accuracy, and robustness. Our findings endorse the superior efficacy and robustness of the RT model based on the refined LOOCV-informed splitting criterion, marking substantial improvements over those of the traditional RT model. Additionally, our tree structure analysis provides insights into how our proposed LOOCV-informed splitting criterion guides the model in striking a balance between a complex tree structure and accurate predictions.

## 1 | Introduction

Machine learning (ML) models have become an invaluable asset in operations research, offering sophisticated data-driven approaches to optimizing complex systems and assisting in decision-making processes (Bengio et al. 2021; Chou et al. 2023). For example, the predict-then-optimize framework is a powerful approach used in operations research to improve decision-making, in which an ML model is first employed to forecast key parameters that are subsequently fed as inputs into an optimization model to determine the most effective operational strategies (Mišić and Perakis 2020). However, inaccurate predictions from ML models can lead to suboptimal or even detrimental decisions that fail to meet operational objectives, as the resulting optimization models rely on inaccurate inputs.

Among the major impediments to the effective performance of ML models is *overfitting*, which is characterized by an excessive focus on the training data that is detrimental to a model's ability to generalize to new data (Ying 2019). Insufficient data is one of the primary causes of overfitting in ML, because deficient data may lead to limited exposure to variability and an inability to capture underlying patterns. In fact, in many practical application scenarios, there might be a lack of adequate data due to the high cost of data collection, privacy concerns, or accessibility issues. An overfitted model may exhibit remarkable accuracy on the training set, capturing potentially irrelevant fluctuations in the data. However, such specificity reduces the model's generalizability, that is, its capability to adapt to new data with traits different from those of the training set. Consequently, the performance of overfitted models on independent test sets is typically poor, as such models fail to capture the underlying patterns and structures that are essential for accurate predictions across diverse datasets.

An overfitted model often results from relying exclusively on *training error* as a measure for ML model construction (Hastie et al. 2009). During the training phase, models are often evaluated based on their ability to minimize errors or losses (e.g., mean squared error (MSE)) on the training data. When the minimization of training errors is the sole objective, there is a risk that the model will "memorize" the data, including noise and outliers, rather than "learn" from the data. In contrast, the *generalization error*, a fundamental ML concept, quantifies the ability of a model to make accurate predictions from new data, which is the ultimate aim of an ML model. Techniques such as cross-validation, regularization, and the use of a separate validation set to monitor and tune model complexity help to estimate a model's generalization error during construction. Among such methods, modifying the loss function to minimize the generalization error may be one of the most intuitive. For example, in linear regression, regularization is commonly applied to the loss function to train the model, typically in the form of lasso regression or ridge regression penalties (Tibshirani 1996; Hoerl and Kennard 1970). These penalties serve to constrain coefficient values, thereby mitigating overfitting and promoting model generalization. By focusing on minimizing generalization errors, researchers can develop models that are robust and reliable, thereby maximizing their performance in subsequent optimization processes.

Despite the many successful applications of regularization to reduce the generalization error of many ML models, similar techniques have not been applied to the loss function in the training of regression tree (RT) models (Wu et al. 2008), which are widely used in various industries for their interpretability and flexibility in modeling complex and nonlinear relationships within data (Sun et al. 2020; Bandi and Bertsimas 2021; Bertsimas et al. 2022; Salari et al. 2022). Some efforts have been devoted to improving the effectiveness of the tree-based models from other aspects, including Kao and Tang (2014), which adds the label-dependent "late constraints" to enrich the decision tree induction problem; Aouad et al. (2023), which uses market segmentation trees to explicitly learn market segmentations by recognizing variations in user response patterns; Liu (2022), which develops a new mixed-integer programming model to optimize split rule selection in the decision tree.

In this article, we focus on enhancing the extensively used classification and regression tree (CART) model by considering the generalization error. In the traditional CART model, all training data are input to the root node and split into child nodes based on the features and feature values that result in the greatest reduction of a chosen measure of error or impurity, such as MSE. This process is recursively applied in each split until a stopping criterion is met, producing a tree structure where each leaf node corresponds to a predicted value that is equal to the mean of the outputs of all training samples contained in the node. The choice of splitting criterion is crucial for determining how the input space is partitioned.

Typical splitting rules include the Chi-squared test, Gini impurity, entropy and information gain, F-test-based rules, and MSE. To be specific, the Chi-squared test is commonly used in classification trees, particularly in the Chi-square automatic interaction detection (CHAID) algorithm (Kass 1980; Biggs et al. 1991). The Chi-squared test evaluates the independence between categorical input variables and the target variable, identifying splits that maximize statistical significance. It is especially popular in fields such as market research (Baron and Phillips 1994; Kumar and Kaur 2023) and medical diagnostics (Kobayashi et al. 2013; Miller et al. 2014), where categorical data is prevalent. Gini impurity is another widely used criterion, particularly for classification tasks within the CART framework (Breiman 2017). It calculates the probability of misclassifying a randomly selected sample if it were labeled according to the class distribution at a given node of the tree. Gini impurity is computationally efficient and works well with balanced datasets and finds applications in areas such as finance (Yilgör et al. 2011), healthcare (Fonarow et al. 2005), and marketing (Karim and Rahman 2013). Entropy and information gain serve as foundational metrics for the ID3 and C4.5 classification tree algorithms (Shannon 1948; Quinlan 1986). Entropy quantifies the level of disorder or uncertainty in a dataset, while information gain measures the reduction in entropy achieved by a split. These criteria are widely utilized in ML applications such as natural language processing (Kuhn and De Mori 2002) and bioinformatics (Che et al. 2011). In contrast, F-test-based rules and MSE are primarily used for regression tasks in CART (Breiman 2017; Fisher et al. 1966). F-test-based rules assess the significance of potential splits by comparing the variance between groups created by a split to the variance within those groups, enabling the selection of the most statistically significant splits. Compared to F-test-based rules, training MSE is widely used for the construction of CART (Loh 2011) and is adopted by various state-of-the-art ML libraries, such as scikit-learn (Scikit-learn 2025) and fitrtree (MathWorks 2023). However, using the training MSE as the splitting criterion, which ensures that the resulting branches of the tree capture the patterns within the training data as accurately as possible, may create a tendency to overfit, especially as models become deeper and more complex. Without constraints or pruning, such models may exhibit excellent accuracy on the training set but perform poorly on new data.

Therefore, we consider intrinsically preventing the overfitting of RT models by modifying the loss function to minimize the generalization error instead of the training error. We explore the fundamental objective of splitting in RT models and derive the accurate formula for the generalization error for each split. Based on this analysis, we propose two enhancements for the traditional splitting criterion: One that approximates the generalization error by estimating the variance of the predicted targets, and one that approaches the out-of-sample error through leave-one-out cross-validation (LOOCV) (James et al. 2013). We point out that while the predicted output of these different splitting methods converges to the true (conditional) mean in probability, considering the limited available data in practical scenarios, our proposed enhancements can provide a better prediction of the true target. To validate our proposed RT models with generalization error-based splitting criteria, we implement them and compare their effectiveness, efficiency, and robustness with those of the traditional CART model and F-test-based model on 12 common ML datasets. The scientific contributions of our paper are as follows:

- *Clarification of splitting objective in CART*. We demonstrate the benefit of the fundamental objective of splitting in the context of our RT models, which is to maximize prediction accuracy on new data by minimizing generalization errors. We precisely formulate the generalization error for each split, offering a clear mathematical representation of a generalization error-based splitting criterion that can improve splitting decisions.
- *Generalization error-based splitting criteria development*. We introduce two innovative enhancements to the traditional splitting criterion. The first employs an estimate of the variance of predicted targets to approximate the generalization error, while the second utilizes LOOCV to closely approximate the out-of-sample error. These methods are proven to be able to better capture the predictive performance of potential splits on new data.
- *Comprehensive model performance assessment*. Our proposed RT models that incorporate generalization error-based splitting criteria are rigorously tested against the two classic RT models on 12 benchmark ML datasets. The results indicate the superior effectiveness, computational efficiency, and robustness of our LOOCV-based RT models, demonstrating significant improvements over traditional RTs, with 95% confidence. We also draw some insights from the tree structure, explaining how the generalization error-based splitting criteria guide the tree to balance complexity with prediction accuracy, yielding robust models that perform well on different datasets.

The rest of this paper is organized as follows. Section 2 briefly reviews basic information on MSE and RTs, laying the foundations of our work. Section 3 demonstrates the defects of RT splitting criteria based on training error and proposes our enhancements. In Section 4, we present the framework of our proposed generalization error-based tree. Section 5 describes a series of numerical experiments conducted on various datasets to validate our proposed generalization error-based tree. Section 6 concludes this article.

## 2 | Preliminaries

In this section, we introduce the fundamental definitions of MSE and CART, which serve as the basis of our approaches.

### 2.1 | MSE of a Predictor

*General MSE*. In ML, MSE is commonly used to assess the quality of a predictor. We consider a random feature vector $X$ and a dependent random variable $Y$ with unknown underlying joint distribution function denoted by $F_{X,Y}$. Given a set of independent and identically distributed (i.i.d.) training samples denoted by $D = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$, an ML model denoted by $f(\theta, X; D)$ is developed to approximate $Y$ given $X$, where $\theta \in \Theta$ is the vector of parameters (e.g., in an RT, $\theta$ represents the chosen feature and the feature value at which to split). We note that $X$ and $Y$ are random, while $X_i$ and $Y_i$ are determined by the given training dataset $D$. Specifically, the general formula for the MSE given the training data $D$ can be written as follows:

$$\begin{aligned}\mathrm{MSE}^{\mathrm{general}}(\theta; D) \equiv\ & \mathbb{E}_{(X,Y)}\left[(f(\theta, X; D) - Y)^2\right]\\ =\ & \int (f(\theta, x; D) - y)^2 dF_{X,Y}(x, y)\end{aligned} \tag{1}$$

Note that the training set $D$ is fixed in Expression (1). We also denote by $\widetilde{D}$ a random sample of size $N$ drawn from the joint distribution $F_{X,Y}$. Then, the expected value of $\mathrm{MSE}^{\mathrm{general}}(\theta; \widetilde{D})$ on training data $\widetilde{D}$ yields the following general MSE, where the expectation refers to the distribution induced by the random variable $\widetilde{D}$:

$$\mathrm{MSE}^{\mathrm{general}}(\theta) \equiv \mathbb{E}_{\widetilde{D}}\left[\mathrm{MSE}^{\mathrm{general}}(\theta; \widetilde{D})\right] \tag{2}$$

*Training MSE*. However, because of the lack of knowledge of the joint distribution function $F_{X,Y}$ and the computational complexity of high-dimensional integrals, one may replace $F_{X,Y}$ in Equation (1) with the empirical distribution of the observed sample set $D$. A commonly adopted approach is to approximate Equation (1) by using the training MSE, which can be written as follows:

$$\mathrm{MSE}^{\mathrm{train}}(\theta; D) \equiv \frac{1}{N}\sum_{i=1}^{N}(f(\theta, X_i; D) - Y_i)^2 \tag{3}$$

Subsequently, the optimal solution of $\theta$ for an ML model is chosen as

$$\hat{\theta} \in \underset{\theta \in \Theta}{\arg\min}\ \mathrm{MSE}^{\mathrm{train}}(\theta; D) \tag{4}$$

*Out-of-sample MSE*. $\mathrm{MSE}^{\mathrm{train}}(\hat{\theta}; D)$ has been criticized as being optimistic and may cause overfitting when used as the metric for model training (Pekel 2020). The training data account for a portion of this optimism. Specifically, imagine a new dataset $D' = \{(X_1, Y_1'), \ldots, (X_N, Y_N')\}$, where the notation $Y_i'$ indicates that we observe a new response value at the training point $X_i$, $i = 1, 2, \ldots, N$. We can define the out-of-sample error on $D'$ of a predictor that minimizes the training error on $D$ as

$$\mathrm{MSE}^{\mathrm{out}\text{-}sample}(\hat{\theta}; D, D') \equiv \frac{1}{N}\sum_{i=1}^{N}(f(\hat{\theta}, X_i; D) - Y_i')^2 \tag{5}$$

Notably, $\text{MSE}^{\text{general}}(\hat{\theta}; D)$ is more general than $\text{MSE}^{\text{out-sample}}(\hat{\theta}; D, D')$ because the input vectors need not coincide with the training input vectors. Nevertheless, $\text{MSE}^{\text{train}}(\hat{\theta}; D)$ is still more optimistic than $\text{MSE}^{\text{out-sample}}(\hat{\theta}; D, D')$, and the nature of this optimism can be written as

$$\begin{aligned}\text{optimism} &\equiv \text{MSE}^{\text{out-sample}}(\hat{\theta}; D, D') - \text{MSE}^{\text{train}}(\hat{\theta}; D)\\ &= \frac{1}{N}\sum_{i=1}^{N}(f(\hat{\theta}, X_i; D) - Y_i')^2 - \frac{1}{N}\sum_{i=1}^{N}(f(\hat{\theta}, X_i; D) - Y_i)^2\end{aligned} \tag{6}$$

**Theorem 1.** *Taking the expectation of optimism, we can demonstrate that generally,*

$$\mathbb{E}_{\widetilde{Y}}[\text{optimism}] = \frac{2}{N}\sum_{i=1}^{N}\text{Cov}\left[f(\hat{\theta}, X_i; D), \widetilde{Y}_i\right] \tag{7}$$

*where* Cov *indicates the covariance. Here, the predictors, that is, $f(\hat{\theta}, X_i; D)$, in the training set are fixed, and the expectation is taken over $\widetilde{Y}$, that is, the vector of the training set outputs drawn from the same distribution of $F_{X,Y}$ given the inputs. Specifically, each $\widetilde{Y}_i$ follows the conditional distribution $F_{Y|X=X_i}$ for $i = 1, 2, \ldots, N$. The proof can be found in* Hastie et al. (2009).

The degree to which $\text{MSE}^{\text{train}}(\hat{\theta}; D)$ underestimates $\text{MSE}^{\text{general}}(\hat{\theta}; D)$ partially depends on the covariance of $Y_i$ and $f(\hat{\theta}, X_i; D)$, that is, how strongly $Y_i$ is related to its own forecast.

## 2.2 | Traditional RT

*RT Implementation*. The CART algorithm is a supervised learning approach that generates a tree structure and can be used for both classification and regression tasks (Hastie et al. 2009). In this paper, we are concerned with only regression tasks, and therefore, we discuss only the RT in the following sections.

Determining whether an RT is optimal is an NP-complete problem (Naumov 1991). Therefore, a greedy heuristic method is usually adopted to construct an RT in a depth-first manner. Suppose that we have a set of samples denoted by $D = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)\}$, where $X_i = (X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(J)})$ is the input vector of $J$ features of sample $i$, and $Y_i$ is the output of sample $i$, $i = 1, \ldots, N$. All samples are first input to the root node and subsequently split into child nodes based on the splitting criterion. Subsequently, the next node to split is found recursively, usually following a breadth-first search in which we explore all the nodes at the present depth level before moving on to the nodes at the next depth level. The iterative splitting process continues until no more nodes can be split according to the stopping criteria, that is, attaining the minimum number of samples required in a parent node, the maximum tree depth, or the same outputs for all samples in a node. The node that cannot be split further is referred to as a leaf node.

*Traditional RT Splitting Criterion*. For ease of description, we always take the root node as an example to illustrate the RT splitting criterion because other nodes have a similar structure. For RT model $f(\theta, X; D)$ with depth 2, $\theta = (j, s_j)$ is the splitting pair of the root node, where $j$ and $s_j$ are the index and value of the splitting feature, respectively. The objective of RT splitting is to make the outputs of samples within the same node as similar as possible, which can be achieved by

$$\min_{\substack{j\in\{1,\ldots,J\}\\ s_j\in S_j}}\left[\min_{C_1}\sum_{i\in\Gamma_1(j,s_j)}\left(C_1 - Y_i\right)^2 + \min_{C_2}\sum_{i\in\Gamma_2(j,s_j)}\left(C_2 - Y_i\right)^2\right], \tag{8}$$

where the set of potential splitting values of feature $j$ is $S_j$, which is usually $\mathbb{R}$ for a root node. The set of indices of the data contained in the parent node is $\Gamma = \{1, \ldots, N\}$ for the root node. The sets of indices of the data contained in the left and right child nodes are $\Gamma_1(j, s_j) = \left\{i \in \Gamma | X_i^{(j)} \leqslant s_j\right\}$ and $\Gamma_2(j, s_j) = \left\{i \in \Gamma | X_i^{(j)} > s_j\right\}$, respectively; $C_1$ and $C_2$ are the output of each child node. Notably, $j, s_j, C_1$, and $C_2$ are decision variables. Therefore, the predicted target of a sample with input vector $X_i$ is

$$f(\theta, X_i; D) = \begin{cases} C_1, & \text{if } i \in \Gamma_1(j, s_j)\\ C_2, & \text{if } i \in \Gamma_2(j, s_j)\end{cases} \tag{9}$$

In a traditional RT, for any choice of $j$ and $s_j$, the inner minimization problems are solved as $C_1 = \sum_{i\in\Gamma_1(j,s_j)} Y_i/|\Gamma_1(j, s_j)|$ and $C_2 = \sum_{i\in\Gamma_2(j,s_j)} Y_i/|\Gamma_2(j, s_j)|$. Therefore, Problem (8) is simplified as a problem of minimizing the least squares deviation of the samples contained in the child nodes (Hastie et al. 2009). Specifically, the mean of all samples in each of the two child nodes is first computed and denoted by $\overline{Y}(\Gamma_1(j, s_j)) = \sum_{i\in\Gamma_1(j,s_j)} Y_i/|\Gamma_1(j, s_j)|$ for the left child node and $\overline{Y}(\Gamma_2(j, s_j)) = \sum_{i\in\Gamma_2(j,s_j)} Y_i/|\Gamma_2(j, s_j)|$ for the right child node. Subsequently, the difference of each target value in each child node from the mean is calculated. The traditional splitting criterion is always written as follows:

**Definition 1.** Traditional splitting criterion

$$\begin{aligned}(j^*, s_{j^*}^*) \in \mathop{\arg\min}_{\substack{j\in\{1,\ldots,J\}\\ s_j\in S_j}}&\left[\sum_{i\in\Gamma_1(j,s_j)}\left(\overline{Y}(\Gamma_1(j, s_j)) - Y_i\right)^2\right.\\ &\left.+\sum_{i\in\Gamma_2(j,s_j)}\left(\overline{Y}(\Gamma_2(j, s_j)) - Y_i\right)^2\right]\end{aligned} \tag{10}$$

An RT based on this splitting criterion is typically called a least squares RT; we denote it as t-RT in this article.

## 3 | Revised Splitting Criteria

Following the definitions of MSE and traditional RT, this section first introduces the shortcomings of the traditional splitting criterion. The theoretical motivations for improving the RT splitting criterion are subsequently presented.

### 3.1 | Objective of RT Splitting

*General RT splitting criterion*. As introduced in Section 2.2, Formula (10) is one of the most widely used RT splitting criteria. However, its widespread use stems more from mathematical convenience than from considerations of actual loss in applications. As discussed in Section 2.2, the objective of splitting is to achieve the optimal binary partition by minimizing the MSE of each child node; however, only the training error is considered by the splitting criterion (10). Given a split $\theta = (j, s_j)$ and dataset $D$, the general MSE can be written as the sum of two parts relating to the two child nodes of the root node:

$$\mathrm{MSE}^{\mathrm{general}}(\theta; D) \equiv \int_{x^{(j)} \le s_j, y \in \mathbb{R}} (f(\theta, x; D) - y)^2 dF_{X,Y}(x, y) + \int_{x^{(j)} > s_j, y \in \mathbb{R}} (f(\theta, x; D) - y)^2 dF_{X,Y}(x, y) \tag{11}$$

Here, the training set $D$ is fixed, and $f(\theta, x; D)$ is a constant when $x^{(j)} \le s_j$ (or $x^{(j)} > s_j$). Recall that $\widetilde{D}$ is a random sample of size $N$ drawn from the joint distribution $F_{X,Y}$, we have the following general MSE by taking the expectation over $\widetilde{D}$:

$$\begin{aligned}\mathrm{MSE}^{\mathrm{general}}(\theta) &\equiv \mathbb{E}_{\tilde{D}}[\mathrm{MSE}^{\mathrm{general}}(\theta; \tilde{D})] \\ &= \underbrace{\mathbb{E}_{\tilde{D}}\left[\int_{x^{(j)} \le s_j, y \in \mathbb{R}} (f(\theta, x; \tilde{D}) - y)^2 dF_{X,Y}(x, y)\right]}_{\mathrm{MSE}^{\mathrm{general}}_{\text{left-part}}(\theta)} \\ &\quad + \underbrace{\mathbb{E}_{\tilde{D}}\left[\int_{x^{(j)} > s_j, y \in \mathbb{R}} (f(\theta, x; \tilde{D}) - y)^2 dF_{X,Y}(x, y)\right]}_{\mathrm{MSE}^{\mathrm{general}}_{\text{right-part}}(\theta)}\end{aligned} \tag{12}$$

**Theorem 2.** *For a fixed split, the general MSE comprises three parts. As an example,* $\mathrm{MSE}^{\mathrm{general}}_{\text{left-part}}(\theta)$ *can be decomposed into the following*:

$$\mathrm{MSE}^{\mathrm{general}}_{\text{left-part}}(\theta) = \int_{x^{(j)} \le s_j} \left\{ \mathbb{V}[f(\theta, x; \widetilde{D})] + [\mathrm{Bias}(f(\theta, x; \widetilde{D}))]^2 + \mathbb{V}[Y|X = x] \right\} dF_{X,\cdot}(x) \tag{13}$$

*where* $F_{X,\cdot}$ *is the marginal distribution of* $X$ *under the joint distribution* $F_{X,Y}$, $f(\theta, x; \widetilde{D})$ *is random because* $\widetilde{D}$ *is random,* $\mathrm{Bias}(f(\theta, x; \widetilde{D})) \equiv \mathbb{E}_{\widetilde{D}}[f(\theta, x; \widetilde{D})] - \mathbb{E}_Y[Y|X = x]$, *and* $\mathbb{V}[Y|X = x]$ *is the conditional variance of* $Y$ *given* $X = x$ *and depends only on the conditional distribution* $F_{Y|X}$ (James et al. 2013; Hastie et al. 2009).

*Decomposition of the general splitting criterion*. In an RT, the mean of the outputs of the samples contained in each node is used as its predicted value, that is, $f(\theta, x; \widetilde{D})$. For example, the $f(\theta, x; \widetilde{D})$ of the left child node of the root node can be denoted by $\overline{Y}(X^{(j)} \le s_j; \widetilde{D}) = \frac{\sum_{i \in \widetilde{D}} (Y_i \cdot \mathbb{1}(X^{(j)} \le s_j))}{\sum_{i \in \widetilde{D}} \mathbb{1}(X^{(j)} \le s_j)}$, where $\mathbb{1}$(condition) is an indicating function that equals 1 if the condition is met. Here, $\overline{Y}(X^{(j)} \le s_j; \widetilde{D})$ is a random variable due to $\widetilde{D}$. Thus, we have

$$\begin{aligned}\mathrm{Bias}(f(\theta, x; \widetilde{D})) &= \mathrm{Bias}(\overline{Y}(X^{(j)} \le s_j; \widetilde{D})) \\ &= \mathbb{E}_{\widetilde{D}}[\overline{Y}(X^{(j)} \le s_j; \widetilde{D})] - \mathbb{E}[Y|X^{(j)} \le s_j] \\ &= \mathbb{E}[Y|X^{(j)} \le s_j] - \mathbb{E}[Y|X^{(j)} \le s_j] \\ &= 0\end{aligned}$$

The terms $\int_{x^{(j)} \le s_j} \mathbb{V}[Y|X = x] dF_{X,\cdot}(x)$ in $\mathrm{MSE}^{\mathrm{general}}_{\text{left-part}}(\theta)$ and $\int_{x^{(j)} > s_j} \mathbb{V}[Y|X = x] dF_{X,\cdot}(x)$ in $\mathrm{MSE}^{\mathrm{general}}_{\text{right-part}}(\theta)$ can sum to $\mathbb{E}_X[\mathbb{V}[Y|X]]$, which is an irreducible error, regardless of the splitting criterion. Therefore,

**Corollary 1.** *Formula (12) can be transformed into*

$$\mathrm{MSE}^{\mathrm{general}}(\theta) = \int_{x^{(j)} \le s_j} \mathbb{V}[f(\theta, x; \widetilde{D})] dF_{X,\cdot}(x) + \int_{x^{(j)} > s_j} \mathbb{V}[f(\theta, x; \widetilde{D})] dF_{X,\cdot}(x) + \text{constant} \tag{14}$$

*where the constant is equal to* $\mathbb{E}_X[\mathbb{V}[Y|X]]$ *and is independent of the splitting criterion.*

The term $\mathbb{V}[f(\theta, x; \widetilde{D})]$ equals $\mathbb{V}[\overline{Y}(X^{(j)} \le s_j; \widetilde{D})]$ for the left child node and $\mathbb{V}[\overline{Y}(X^{(j)} > s_j; \widetilde{D})]$ for the right child node, which are constants irrespective of $x$ in the domain $x^{(j)} \le s_j$ and $x^{(j)} > s_j$, respectively. Hence, we can further derive an equivalent formula for $\mathrm{MSE}^{\mathrm{general}}(\theta)$ as follows.

**Corollary 2.** *Given the splitting parameter* $\theta = (j, s_j)$, *we obtain*

$$\mathrm{MSE}^{\mathrm{general}}(\theta) = \mathbb{V}[\overline{Y}(X^{(j)} \le s_j; \widetilde{D})] \int_{x^{(j)} \le s_j} dF_{X,\cdot}(x) + \mathbb{V}[\overline{Y}(X^{(j)} > s_j; \widetilde{D})] \int_{x^{(j)} > s_j} dF_{X,\cdot}(x) + \text{constant} \tag{15}$$

*where the constant is equal to* $\mathbb{E}_X[\mathbb{V}[Y|X]]$ *and is independent of the splitting criterion.*

For an RT, we expect to obtain splits that minimize the generalization error in the child nodes, based on which the splitting criterion can be written as follows:

$$(j^*, s^*_{j^*}) \in \underset{\substack{j \in \{1, \dots, J\} \\ s_j \in S_j}}{\arg\min}\, \mathrm{MSE}^{\mathrm{general}}(j, s_j) \tag{16}$$

Nevertheless, the traditional splitting criterion (10) considers only the training error rather than the generalization error, which may result in prediction bias.

## 3.2 | Enhancements

In this section, we propose two solutions to generate superior splitting criteria. The first approach, called variance-estimated approximation, estimates the general MSE by determining the variance from the observed samples. Then, splitting criteria are generated through the minimization of this approximated generalization error. The second approach estimates the out-of-sample MSE through LOOCV. This LOOCV-informed splitting criterion is adopted to determine the splitting pair.

### 3.2.1 | Variance-Estimated Approximation

As our discussion in Corollary 2 reveals, the general MSE of samples in a child node is determined by the expected value of the variance of $\overline{Y}$ over the cumulative distribution function of $X$ in the node. While the joint distribution $F_{X,Y}$ is unknown, we try to estimate its value based on the observed samples. We take the child nodes of the root node as an example; assume that the potential split $(j, s_j)$ divides a dataset with $N$ samples into subset $\Gamma_1(j, s_j)$ in the left child node and subset $\Gamma_2(j, s_j)$ in the right child node. Denote $\overline{Y}(\Gamma_1(j, s_j))$ and $\overline{Y}(\Gamma_2(j, s_j))$ as the mean value of the samples' outputs in the left and right child node, respectively. We use the sample variance within the left node, denoted by $S^2_{\Gamma_1(j,s_j)} = \sum_{i\in\Gamma_1(j,s_j)} \left(Y_i - \overline{Y}(\Gamma_1(j, s_j))\right)^2 / (|\Gamma_1(j, s_j)| - 1)$, to estimate the population variance of the random variable $Y$ in the left child node, denoted by $\mathbb{V}[\overline{Y}(X^{(j)} \le s_j; \widetilde{D})]$, because the previous one is an unbiased estimator of the population variance in the left child node (Larsen and Marx 2005). Thus, the variance of the random variable $\overline{Y}(X^{(j)} \le s_j; \widetilde{D})$ in the left child node can be presented as

$$\mathbb{V}[\overline{Y}(X^{(j)} \le s_j; \widetilde{D})] \approx \frac{\mathbb{V}[Y(X^{(j)} \le s_j; \widetilde{D})]}{|\Gamma_1(j, s_j)|} \approx \frac{1}{|\Gamma_1(j, s_j)|} \cdot \frac{\sum_{i\in\Gamma_1(j,s_j)} \left(Y_i - \overline{Y}(\Gamma_1(j, s_j))\right)^2}{|\Gamma_1(j, s_j)| - 1} \tag{17}$$

where we take the approximation because we use the sample variance to estimate the (conditional) population variance. Moreover, because the probability that a data point is split into the left child node (i.e., $\int_{x^{(j)} \le s_j} dF_{X,\cdot}(x)$) is unknown, we approximate this probability by using the empirical value $|\Gamma_1(j, s_j)|/N$. Therefore, $\mathbb{V}[\overline{Y}(X^{(j)} \le s_j; \widetilde{D})] \int_{x^{(j)} \le s_j} dF_{X,\cdot}(x)$ can be approximated as

$$\begin{aligned} \mathrm{MSE}^{\text{approx-general}}_{\text{left-part}}(j, s_j) &\equiv \frac{1}{|\Gamma_1(j, s_j)|} \cdot \frac{\sum_{i\in\Gamma_1(j,s_j)} \left(Y_i - \overline{Y}(\Gamma_1(j, s_j))\right)^2}{|\Gamma_1(j, s_j)| - 1} \cdot \frac{|\Gamma_1(j, s_j)|}{N} \\ &= \frac{1}{N} \cdot \frac{\sum_{i\in\Gamma_1(j,s_j)} \left(Y_i - \overline{Y}(\Gamma_1(j, s_j))\right)^2}{|\Gamma_1(j, s_j)| - 1} \end{aligned} \tag{18}$$

According to this approximated generalization error, we propose the following variance-estimated splitting criterion, which approximates the splitting criterion (16):

**Definition 2.** Variance-estimated splitting criterion

$$\begin{aligned} (j^*, s^*_{j^*}) \in \underset{\substack{j\in\{1,\dots,J\} \\ s_j\in S_j}}{\arg\min} \Bigg[ &\frac{1}{|\Gamma_1(j, s_j)| - 1} \sum_{i\in\Gamma_1(j,s_j)} \left(\overline{Y}(\Gamma_1(j, s_j)) - Y_i\right)^2 \\ &+ \frac{1}{|\Gamma_2(j, s_j)| - 1} \sum_{i\in\Gamma_2(j,s_j)} \left(\overline{Y}(\Gamma_2(j, s_j)) - Y_i\right)^2 \Bigg] \end{aligned} \tag{19}$$

Notably, $1/N$ is omitted as it is a constant and will not influence the choice of splitting pair.

*Remark* 1. For the variance-estimated splitting criterion, we want to remark on the following points:

- *Enhancement analysis*. As demonstrated in Corollary 2, the general MSE of samples in a child node is determined by the expected value of the variance of $\overline{Y}$ over the cumulative distribution function of $X$ in the node. Our variance-estimated splitting criterion enhances the traditional approach by utilizing an unbiased estimation of $\mathbb{V}[Y(X^{(j)} \le s_j; \widetilde{D})]$ and $\mathbb{V}[Y(X^{(j)} > s_j; \widetilde{D})]$. In contrast, the traditional splitting criterion relies on a biased estimator. The proof is given in Appendix A. The main benefits of our method include: (1) Improved split quality: By accurately representing the variance, our splitting criterion ensures that splits are evaluated based on a more faithful representation of the data's variability, leading to more meaningful partitions. (2) Superiority on small sample sizes: When dealing with small sample sizes, a scenario common in practical applications such as high-dimensional regression, imbalanced datasets, or deep tree depth, the bias introduced by the traditional MSE can lead to overfitting or the selection of splits that do not generalize well to unseen data. By using the variance-estimated splitting criterion, our method mitigates these issues, resulting in more robust and generalizable trees.

- *Two approximations are involved here*: (1) The sample variance is used to estimate the conditional variance of $\overline{Y}$ (i.e., $\mathbb{V}[\overline{Y}(X^{(j)} \le s_j; \widetilde{D})]$ and $\mathbb{V}[\overline{Y}(X^{(j)} > s_j; \widetilde{D})]$); (2) The cumulative distributions $\int_{x^{(j)} \le s_j} dF_{X,\cdot}$ and $\int_{x^{(j)} > s_j} dF_{X,\cdot}$ are approximated from the number of training samples that fall into each child node divided by the number of samples in the parent node.

- In each split, the computational complexity of the variance-estimated splitting criterion is the same as that of the traditional splitting criterion, that is, $O(n)$, where $n$ is the number of samples in the node.

- If there is only one sample in a child node, then the MSE of this node is set to zero. For example, if we have $|\Gamma_1(j, s_j)| = 1$ for the left child node, then $\mathrm{MSE}^{\text{approx-general}}_{\text{left-part}} = 0$.

### 3.2.2 | Leave-One-Out Cross-Validation (LOOCV)

Another widely adopted ML method that partially mitigates the common underestimation of generalization errors is LOOCV, which is usually used to select the optimal *hyperparameters* in the RT model; the underlying principle is to minimize the out-of-sample error (defined in Formula (5)). In LOOCV, a dataset is divided into $N$ subsets, where $N$ is the total number of samples in the dataset. At each iteration, one sample is left out for use as the validation set, and the model is trained on the remaining $N - 1$ samples. This process is repeated $N$ times, with each sample being left out once as the validation set. The final evaluation metric (e.g., MSE) is computed by averaging the results of all $N$ iterations (James et al. 2013). We use LOOCV to select the optimal *parameters* (i.e., splitting criterion) by evaluating the out-of-sample MSE in each child node.

Given a training dataset $D$ of $N$ samples and a potential split $(j, s_j)$ of $D$ into $\Gamma_1(j, s_j)$ and $\Gamma_2(j, s_j)$, we apply LOOCV to estimate the out-of-sample MSE for the child node. Again, we take the left child node of the root node as an example.

With some abuse of notation, we denote the training MSE and out-of-sample MSE for the left child node as $\mathrm{MSE}^{\text{train}}(j, s_j; \Gamma_1)$ and $\mathrm{MSE}^{\text{out-sample}}(j, s_j; \Gamma_1, \Gamma_1')$, respectively, where $\Gamma_1'$ comprises the indices of a new dataset, in which we observe a new response value for each of the training points of $\Gamma_1$ (similar to the definition of $D$ and $D'$ in Section 2.1). Let $\overline{Y}_i^{(-i)}$ be the predicted value of leaving out each sample $(X_i, Y_i)$, $i \in \Gamma_1(j, s_j)$. Then, $\overline{Y}_i^{(-i)}$ can be computed from the expectation of the remaining samples as follows:

$$\overline{Y}_i^{(-i)} \equiv \frac{\sum_{k \in \Gamma_1(j,s_j), k \neq i} Y_k}{|\Gamma_1(j, s_j)| - 1} \tag{20}$$

Therefore, the LOOCV value of $\Gamma_1(j, s_j)$ can be presented as

$$\mathrm{LOOCV}(\Gamma_1(j, s_j)) \equiv \frac{1}{|\Gamma_1(j, s_j)|} \sum_{i \in \Gamma_1(j,s_j)} \left( \overline{Y}_i^{(-i)} - Y_i \right)^2 \tag{21}$$

Direct LOOCV is computationally inefficient. Fortunately, according to Stone (1974)., a "shortcut" identity can be used for LOOCV in the special case of least squares polynomial regression.

**Proposition 1.** *We can perform LOOCV with this simple shortcut in the left child node as follows*:

$$\mathrm{LOOCV}(\Gamma_1(j, s_j)) = \frac{1}{|\Gamma_1(j, s_j)|} \sum_{i \in \Gamma_1(j,s_j)} \left( \frac{\overline{Y}(\Gamma_1(j, s_j)) - Y_i}{1 - \gamma_1} \right)^2 \tag{22}$$

*where* $\gamma_1 = 1/|\Gamma_1(j, s_j)|$.

*Proof.* To streamline the notation, we write $\Gamma_1(j, s_j)$ as $\Gamma_1$ in the following proof.

$$\begin{aligned}
\mathrm{LOOCV}(\Gamma_1) &= \frac{1}{|\Gamma_1|} \sum_{i \in \Gamma_1} \left( \overline{Y}_i^{(-i)} - Y_i \right)^2 \\
&= \frac{1}{|\Gamma_1|} \sum_{i \in \Gamma_1} \left( \frac{\sum_{k \in \Gamma_1, k \neq i} Y_k}{|\Gamma_1| - 1} - Y_i \right)^2 \\
&= \frac{1}{|\Gamma_1|} \sum_{i \in \Gamma_1} \left( \frac{\sum_{k \in \Gamma_1} Y_k - Y_i}{|\Gamma_1| - 1} - Y_i \right)^2 \\
&= \frac{1}{|\Gamma_1|} \sum_{i \in \Gamma_1} \left( \frac{|\Gamma_1| \cdot \overline{Y}(\Gamma_1) - Y_i}{|\Gamma_1| - 1} - Y_i \right)^2 \\
&\qquad \left( \text{because } \overline{Y}(\Gamma_1) = \frac{\sum_{i \in \Gamma_1} Y_i}{|\Gamma_1|} \right) \\
&= \frac{1}{|\Gamma_1|} \sum_{i \in \Gamma_1} \left( \frac{|\Gamma_1| \cdot \overline{Y}(\Gamma_1) - Y_i - (|\Gamma_1| - 1) Y_i}{|\Gamma_1| - 1} \right)^2 \\
&= \frac{1}{|\Gamma_1|} \sum_{i \in \Gamma_1} \left( \frac{|\Gamma_1| \cdot \overline{Y}(\Gamma_1) - |\Gamma_1| \cdot Y_i}{|\Gamma_1| - 1} \right)^2 \\
&= \frac{1}{|\Gamma_1|} \sum_{i \in \Gamma_1} \left( \frac{\overline{Y}(\Gamma_1) - Y_i}{1 - \frac{1}{|\Gamma_1|}} \right)^2
\end{aligned}$$

Therefore, letting $\gamma_1 = 1/|\Gamma_1|$, we have

$$\mathrm{LOOCV}(\Gamma_1) = \frac{1}{|\Gamma_1|} \sum_{i \in \Gamma_1} \left( \frac{\overline{Y}(\Gamma_1) - Y_i}{1 - \gamma_1} \right)^2$$

□

Using the above LOOCV formula, we further prove that the result approximates the out-of-sample MSE, which largely mitigates the optimism of the training MSE.

**Proposition 2.** *The out-of-sample MSE can be approximated as* $\mathrm{LOOCV}(\Gamma_1(j, s_j))$; *that is,*

$$\mathrm{MSE}^{\text{out-}\mathit{sample}}(j, s_j; \Gamma_1, \Gamma_1') \approx \mathrm{LOOCV}(\Gamma_1(j, s_j)) \tag{23}$$

*Proof.* We denote $\Gamma_1(j, s_j)$ as $\Gamma_1$ for simplicity in this proof. Let $g(\gamma_1) = 1/(1 - \gamma_1)^2$; through Taylor expansion, we determine that

$$\begin{aligned}
g(\gamma_1) &= \frac{g(0)}{0!} + \frac{g'(0)}{1!}\gamma_1 + \frac{g''(0)}{2!}\gamma_1^2 + \frac{g'''(0)}{3!}\gamma_1^3 + \cdots \\
&= 1 + 2\gamma_1 + 3\gamma_1^2 + 4\gamma_1^3 + \cdots \\
&= 1 + 2\gamma_1 + o(\gamma_1) \\
&\approx 1 + 2\gamma_1
\end{aligned}$$

Therefore, we have $1/(1 - \gamma_1)^2 \approx 1 + 2\gamma_1$ when $|\Gamma_1|$ is large. Substituting it into (22), we obtain

$$\begin{aligned}
\mathrm{LOOCV}(\Gamma_1) &\approx \frac{1 + 2\gamma_1}{|\Gamma_1|} \sum_{i \in \Gamma_1} \left( \overline{Y}(\Gamma_1) - Y_i \right)^2 \\
&= \frac{1}{|\Gamma_1|} \sum_{i \in \Gamma_1} \left( \overline{Y}(\Gamma_1) - Y_i \right)^2 + \frac{2\gamma_1}{|\Gamma_1|} \sum_{i \in \Gamma_1} \left( \overline{Y}(\Gamma_1) - Y_i \right)^2 \\
&= \mathrm{MSE}^{\text{train}}(j, s_j; \Gamma_1) + \frac{2}{|\Gamma_1|^2} \sum_{i \in \Gamma_1} \left( \overline{Y}(\Gamma_1) - Y_i \right)^2 \\
&= \mathrm{MSE}^{\text{train}}(j, s_j; \Gamma_1) + \frac{2}{|\Gamma_1|} \sum_{i \in \Gamma_1} \frac{\left( \overline{Y}(\Gamma_1) - Y_i \right)^2}{|\Gamma_1|} \\
&\approx \mathrm{MSE}^{\text{train}}(j, s_j; \Gamma_1) + \frac{2}{|\Gamma_1|} \mathbb{V}[Y | X^{(j)} < s_j] \\
&= \mathrm{MSE}^{\text{train}}(j, s_j; \Gamma_1) + \frac{2}{|\Gamma_1|} \mathrm{Cov}[Y, Y | X^{(j)} < s_j] \\
&= \mathrm{MSE}^{\text{train}}(j, s_j; \Gamma_1) + \frac{2}{|\Gamma_1|} \sum_{i \in \Gamma_1} \frac{1}{|\Gamma_1|} \\
&\qquad \mathrm{Cov}[\widetilde{Y}_i, \widetilde{Y}_i] \quad \widetilde{Y}_i \text{ is a random variable given } X^{(j)} < s_j \\
&= \mathrm{MSE}^{\text{train}}(j, s_j; \Gamma_1) + \frac{2}{|\Gamma_1|} \sum_{i \in \Gamma_1} \mathrm{Cov}\left[ \widetilde{Y}_i, \frac{\widetilde{Y}_i}{|\Gamma_1|} \right] \\
&= \mathrm{MSE}^{\text{train}}(j, s_j; \Gamma_1) + \frac{2}{|\Gamma_1|} \\
&\qquad \sum_{i \in \Gamma_1} \left( \mathrm{Cov}\left[ \widetilde{Y}_i, \frac{\widetilde{Y}_i}{|\Gamma_1|} \right] + \sum_{j \in \Gamma_1, j \neq i} \mathrm{Cov}\left[ \widetilde{Y}_i, \frac{\widetilde{Y}_j}{|\Gamma_1|} \right] \right) \\
&\qquad \left( \because \widetilde{Y}_i \perp\!\!\!\perp \widetilde{Y}_j \;\therefore \mathrm{Cov}[\widetilde{Y}_i, \widetilde{Y}_j] = 0 \;\therefore \mathrm{Cov}\left[ \widetilde{Y}_i, \frac{\widetilde{Y}_j}{|\Gamma_1|} \right] = 0 \right) \\
&= \mathrm{MSE}^{\text{train}}(j, s_j; \Gamma_1) + \frac{2}{|\Gamma_1|} \sum_{i \in \Gamma_1} \mathrm{Cov}\left[ \widetilde{Y}_i, \frac{\sum_{i \in \Gamma_1} \widetilde{Y}_i}{|\Gamma_1|} \right]
\end{aligned}$$

$$
\begin{aligned}
&\approx \mathrm{MSE}^{\mathrm{train}}(j, s_j; \Gamma_1) + \frac{2}{|\Gamma_1|}\sum_{i\in\Gamma_1} \mathrm{Cov}\left[\widetilde{Y}_i, \overline{Y}(\Gamma_1)\right] \\
&= \mathrm{MSE}^{\mathrm{train}}(j, s_j; \Gamma_1) + \mathbb{E}[\text{optimism}] \\
&= \mathrm{MSE}^{\text{out-}sample}(j, s_j; \Gamma_1, \Gamma_1') \qquad \text{(by eq. (7)).}
\end{aligned}
$$

□

According to the above proof, we have $\mathrm{MSE}^{\text{out-sample}}(j, s_j; \Gamma_2, \Gamma_2') \approx \mathrm{LOOCV}(\Gamma_2(j, s_j))$ for the right child node. The above discussion reveals that the LOOCV values approach the out-of-sample error, partially correcting for the underestimation based on the training error. Thus, the leave-one-out splitting criterion can be presented as follows:

**Definition 3.** Leave-one-out splitting criterion

$$
\begin{aligned}
(j^*, s^*_{j^*}) \in \underset{\substack{j\in\{1,\dots,J\}\\ s_j\in S_j}}{\arg\min} &\left[ \frac{1}{|\Gamma_1(j, s_j)|}\sum_{i\in\Gamma_1(j,s_j)} \left(\frac{\overline{Y}(\Gamma_1(j, s_j)) - Y_i}{1-\gamma_1}\right)^2 \right. \\
&\left. + \frac{1}{|\Gamma_2(j, s_j)|}\sum_{i\in\Gamma_2(j,s_j)} \left(\frac{\overline{Y}(\Gamma_2(j, s_j)) - Y_i}{1-\gamma_2}\right)^2 \right]
\end{aligned}
\tag{24}
$$

where $\gamma_1 = 1/|\Gamma_1(j, s_j)|$ and $\gamma_2 = 1/|\Gamma_2(j, s_j)|$.

*Remark* 2. For the LOOCV-based splitting criterion, we want to remark on the following points:

- *Enhancement analysis*. As demonstrated in Proposition 2, the LOOCV-based splitting criterion can approximate the out-of-sample error, which is less optimistic than the training error by containing an additional term $\mathbb{E}$[optimism]. Consequently, the LOOCV-based splitting criterion provides a more accurate representation of the generalization error than the traditional splitting criterion, yielding predictions that are more likely to generalize well to unseen data.
- *The LOOCV-based splitting criterion approximates the general MSE-based splitting criterion in two steps*: (1) LOOCV is used to approximate the out-of-sample MSE; (2) the sum of out-of-sample MSE instead of the precise general MSE is utilized as the splitting criterion.
- Given the expression based on "short cut" in Proposition 1, in each split, the computational complexity of the LOOCV-based splitting criterion is the same as that of the traditional splitting criterion, that is, $O(n)$ where $n$ is the number of samples in the node.
- If there is only one sample in a child node, then the MSE of this node is set to 0. For example, if we have $|\Gamma_1(j, s_j)| = 1$ for the left child node, then $\mathrm{LOOCV}(\Gamma_1)$ is 0.

## 4 | Generalization Error-Based Tree Models

This section introduces tree models constructed with different splitting criteria based on the enhancements presented in Section 3.2. Subsequently, a synthetic example is presented to demonstrate how the proposed splitting criteria influence RT construction.

**ALGORITHM 1** | Pseudocode for the RT algorithm.

```
Input: Training dataset $D$ and stopping rules
Output: RT model
Step 1: Input all training samples in $D$ to the root node
Step 2: Find the optimal splitting pair for the current node based on the splitting criterion (i.e., (10), (19), or (24))
Step 3: Divide the samples in the current node into two child nodes (i.e., subsets based on the optimal splitting pair)
Step 4: If a stopping criterion is reached, advance to Step 5. Otherwise, recursively apply Steps 2 and 3 to each child node until no nodes can be further split
Step 5: Assign the average value of the sample outputs of each leaf node as the predicted value
Step 6: Return RT model.
```

### 4.1 | Tree Models

This section provides the training process and the summaries of the RT models incorporating different splitting criteria.

### 4.1.1 | Regression Tree

As introduced in Section 2.2, one of the most commonly used and intuitive tree-based models is the decision tree (in our case, which is a regression problem, the RT). Based on different splitting criteria, our RT model can be constructed following the pseudocode presented in Algorithm 1.

### 4.1.2 | Summary of Models

To simplify the notation of our RT models based on different generalization error-based splitting criteria, we denote them as g-RT-$k$, where $g$ indicates "generalization error" and $k$ is the index of the splitting criterion. We summarize the critical formula for four splitting criteria and the notation of the respective RT models in Table 1, where an F-test-based RT model is also included. The F-test-based RT model uses the F-statistic to measure the reduction in variance and selects the split that mostly decreases prediction error as indicated by statistically significant F-test results (Murtaugh 1998; Gkioulekas and Papageorgiou 2021).

### 4.2 | Synthetic Example in an RT

To better illustrate the aforementioned splitting criteria, we compare them with the traditional splitting criterion in a synthetic example below.

**Example 1.** We consider five training samples, each with two input features denoted by $(x_1, x_2)$ and one output denoted by $y$. The samples' input feature vectors are (6, 6), (8, 5), (4, 9), (10, 10), TABLE 1 | Summary of tree-based models with different splitting criteria.

| Splitting criterion | Formula | RT model | Property |
|---|---|---|---|
| Traditional (benchmark) | $\min_{\substack{j\in\{1,\ldots,J\} \\ s_j\in S_j}} \left[\sum_{i\in\Gamma_1}\left(\overline{Y}(\Gamma_1)-Y_i\right)^2+\sum_{i\in\Gamma_2}\left(\overline{Y}(\Gamma_2)-Y_i\right)^2\right]$ (10) | t-RT | training MSE |
| F-test-based (benchmark) | $\max_{\substack{j\in\{1,\ldots,J\} \\ s_j\in S_j}} \frac{\left(\overline{Y}(\Gamma_1)-\overline{Y}(\Gamma_2)\right)^2}{S_p^2}$, where $S_p^2=\frac{(\lvert\Gamma_1\rvert-1)S^2_{\Gamma_1(j,s_j)}+(\lvert\Gamma_2\rvert-1)S^2_{\Gamma_2(j,s_j)}}{\lvert\Gamma_1\rvert+\lvert\Gamma_2\rvert-1}$ [a] | Ftest-RT | F-Statistic |
| Variance-estimated | $\min_{\substack{j\in\{1,\ldots,J\} \\ s_j\in S_j}} \left[\frac{1}{\lvert\Gamma_1\rvert-1}\sum_{i\in\Gamma_1}(\overline{Y}(\Gamma_1)-Y_i)^2+\frac{1}{\lvert\Gamma_2\rvert-1}\sum_{i\in\Gamma_2}(\overline{Y}(\Gamma_2)-Y_i)^2\right]$ (19) | g-RT-1 | approximated general MSE |
| Leave-one-out cross-validation | $\min_{\substack{j\in\{1,\ldots,J\} \\ s_j\in S_j}} \left[\frac{1}{\lvert\Gamma_1\rvert}\sum_{i\in\Gamma_1}\left(\frac{\overline{Y}(\Gamma_1)-Y_i}{1-1/\lvert\Gamma_1\rvert}\right)^2+\frac{1}{\lvert\Gamma_2\rvert}\sum_{i\in\Gamma_2}\left(\frac{\overline{Y}(\Gamma_2)-Y_i}{1-1/\lvert\Gamma_2\rvert}\right)^2\right]$ (24) | g-RT-2 | out-of-sample MSE |

[a] $S^2_{\Gamma_1(j,s_j)}=\sum_{i\in\Gamma_1(j,s_j)}\left(Y_i-\overline{Y}(\Gamma_1(j,s_j))\right)^2/(\lvert\Gamma_1(j,s_j)\rvert-1)$ and $S^2_{\Gamma_2(j,s_j)}=\sum_{i\in\Gamma_2(j,s_j)}\left(Y_i-\overline{Y}(\Gamma_2(j,s_j))\right)^2/(\lvert\Gamma_2(j,s_j)\rvert-1)$.

TABLE 2 | Testing data and predicted performance of RTs with different splitting criteria.

| Splitting feature | Splitting value[a] | Left child node | Right child node | Errors[b] | | |
|---|---|---|---|---|---|---|
| | | | | t-RT | g-RT-1 | g-RT-2 |
| $x_1$ | 3.5 | sample 5 | sample 1, 2, 3, and 4 | 38.75 | **12.92** | 161.22 |
| $x_1$ | 5 | sample 3 and 5 | sample 1, 2, and 4 | 35.17 | 17.83 | **27.00** |
| $x_1$ | 7 | sample 1, 3, and 5 | sample 2 and 4 | **34.00**[c] | 33.00 | 65.50 |
| $x_1$ | 9 | sample 1, 2, 3, and 5 | sample 4 | 38.75 | **12.92** | 161.22 |
| $x_2$ | 5.5 | sample 2 and 5 | sample 1, 3, and 4 | **34.00** | 33.00 | 65.5 |
| $x_2$ | 7.5 | sample 1, 2, and 5 | sample 3 and 4 | 35.17 | 17.83 | **27.00** |
| $x_2$ | 9.5 | sample 1, 2, 3, and 5 | sample 4 | 38.75 | **12.92** | 161.22 |

[a] We adhere to the scikit-learn implementation of CART, which treats the midpoints between a feature's consecutive unique values—rather than the unique values themselves—as possible splitting thresholds (Buitinck et al. 2013).

[b] The errors of each splitting pair are calculated by the respective splitting criterion formula.

[c] When multiple splitting pairs have the same minimum error, we randomly choose one from them. For simplicity, in this illustrative example, we choose the first feature with the minimum error of these potential splits.

and $(3, 5)$, and their corresponding targets are 14, 20, 13, 12, and 12, respectively. These samples are indexed from 1 to 5 sequentially. The potential splitting points and errors calculated using each splitting criterion on this training set are presented in Table 2.

Therefore, for the root node, the optimal splitting criterion based on t-RT is $(x_1, 7)$, while those based on g-RT-1 and g-RT-2 are $(x_1, 3.5)$ and $(x_1, 5)$, respectively. Following a similar procedure, we can develop corresponding RTs with a maximum depth of 3; as shown in Figure 1, the revised splitting criteria yield considerably different tree structures in this synthetic example. Additionally, it is clear that the g-RT-2 model generates a significantly wider range of errors, ranging from 27.00 to 161.22, compared to the t-RT (from 34.00 to 38.75) and g-RT-1 (from 12.92 to 33.00) models. This increased diversity can be attributed to the larger coefficients used in the expression of the splitting criterion in the g-RT-2 model.

## 5 | Numerical Experiments

This section compares the prediction performance of our proposed generalization error-based RT models with that of two benchmark models (i.e., t-RT and Ftest-RT). Specifically, the computational efficiency, prediction accuracy, model stability, and tree structures are analyzed sequentially. We further enhance our model comparison by evaluating with fixed hyperparameters in Section 5.3 and on non-independent and identically distributed (non-i.i.d.) datasets in Section 5.4, respectively.

### 5.1 | Experimental Settings

To evaluate the performance of our generalization error-based models, this section adopts 12 classic datasets sourced from the UC Irvine Machine Learning Repository (Kelly et al. 2023), which is an online platform that provides a collection of valuable datasets for ML research and experimentation. We adopt the 12 most frequently downloaded datasets in the regression domain and label them 1 to 12. Detailed information on each dataset is presented at https://github.com/ShadowY1998/Data-sets. Each dataset is randomly divided into a training set comprising 70% of the samples and a test set comprising the other 30%.

Two main hyperparameters are considered in RT models: *Minimum sample split* and *maximum tree depth*. Specific descriptions and relative tuning values are presented in Table 3. Notably, since the minimum number of samples contained in a leaf

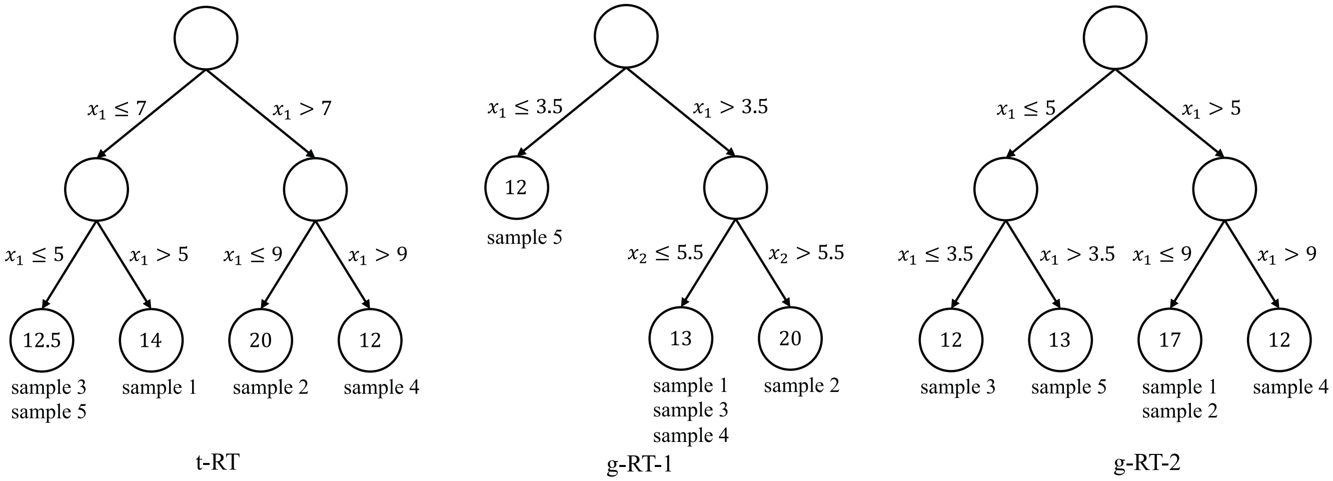

**FIGURE 1** | The depth-3 trees based on different splitting criteria.

**TABLE 3** | Descriptions and tuning methods of hyperparameters in RT models.

| Hyperparameters | Description | Tuning values[a] |
|---|---|---|
| *Minimum-sample-split* | It determines the minimum number of samples required to split a node in an RT model and is used as the stopping criterion for further partitioning of nodes during the tree construction process. | {2, 4, 6} |
| *Maximum-tree-depth* | It limits the maximum depth of a tree and helps reduce overfitting by restricting the complexity and size of the resulting trees. | {10, 15, 20} |

[a] The tuning values show the possible values that the hyperparameters are chosen from.

node can be controlled by limiting the minimum number of samples required to split a node, the former one is not considered here. We train the model with different combinations of hyperparameters and record the five-fold cross-validated MSE, based on which the optimal hyperparameter combination can be determined.

All experiments are conducted on a computer using the Mac operating system, equipped with an Apple M2 Pro processor and 16GB of RAM. The models are implemented using Python 3.11.

## 5.2 | Numerical Results

The detailed experimental results of the RT models on all datasets are presented in Table 4. Notably, the column "Coeff" presents the correlation coefficient between the actual targets and the predicted targets, which is a standardized metric taking values between zero and one and is used to assess prediction accuracy. Specifically, the larger the correlation coefficient, the more accurate the prediction. The column "Hyperparameters" presents the optimal hyperparameter combination for each model, that is, the combination that achieves the minimum MSE on the training set in five-fold cross-validation. The column "Best model" presents the model with the maximum correlation coefficient on the test set. If two models have the same correlation coefficient on the test set, we compare the MSE on the test set and select the model with the smaller MSE as the best model.

### 5.2.1 | Computational Efficiency

The computational efficiency of ML models can be gauged by the duration of model training, which is presented in the column "Time (s)" in Table 4, where the time used in the validation stage is also included. The results indicate similar computing times for the four models, revealing that employing our proposed generalization error-based splitting criteria does not impose a greater computational burden during model training—and even shortens the training time in some cases. While the result is straightforward for g-RT-1, as the variance-estimated splitting criterion only revises the denominator of the traditional splitting criterion, it may seem counterintuitive for g-RT-2. Typically, employing leave-one-out in the traditional cross-validation stage would increase computational time due to the iterative calculation of the mean value on all but one sample. However, given the "shortcut" identity outlined in Proposition 1, we offer a computationally efficient equivalent for the variance-estimated splitting criterion, which requires calculating the mean value only once.

### 5.2.2 | Prediction Accuracy

The prediction accuracy of these models is evaluated based on their MSE and correlation coefficients, which are indicated by "MSE" and "Coeff," respectively, in Table 4. The results suggest that all models yield higher performance on the training sets than on the test set. The test results display divergence across datasets. Specifically, g-RT-2 performs optimally on nine datasets and t-RT

**TABLE 4** | Experiment results of different RT models.

| Dataset | Model | Time (s) | Training set | | Testing set | | Hyperparameters[a] | Best model |
|---|---|---|---|---|---|---|---|---|
| | | | MSE | Coeff | MSE | Coeff | | |
| 1 | t-RT | 160.71 | 0.36 | 0.757 | 0.68 | 0.428 | [10,4] | g-RT-1 |
| | Ftest-RT | 215.19 | 0.84 | 0.088 | 0.66 | 0.069 | [10,6] | |
| | g-RT-1 | 80.88 | 0.64 | 0.499 | 0.56 | 0.439 | [20,2] | |
| | g-RT-2 | 93.79 | 0.37 | 0.750 | 0.70 | 0.411 | [10,6] | |
| 2 | t-RT | 2.03 | 0.02 | 0.937 | 0.04 | 0.761 | [10,6] | g-RT-2 |
| | Ftest-RT | 3.29 | 0.04 | 0.878 | 0.11 | 0.486 | [15,2] | |
| | g-RT-1 | 2.11 | 0.03 | 0.906 | 0.04 | 0.836 | [20,6] | |
| | g-RT-2 | 0.85 | 0.02 | 0.936 | 0.03 | 0.848 | [20,6] | |
| 3 | t-RT | 7.68 | 2.11 | 0.995 | 6.89 | 0.990 | [20,4] | g-RT-2 |
| | Ftest-RT | 3.72 | 161.62 | 0.512 | 183.50 | 0.547 | [10,2] | |
| | g-RT-1 | 7.75 | 12.11 | 0.972 | 121.10 | 0.723 | [15,2] | |
| | g-RT-2 | 3.64 | 0.56 | 0.999 | 4.21 | 0.992 | [20,2] | |
| 4 | t-RT | 124.40 | 2.71 | 0.871 | 6.33 | 0.589 | [10,6] | g-RT-2 |
| | Ftest-RT | 115.10 | 7.21 | 0.598 | 5.96 | 0.556 | [20,6] | |
| | g-RT-1 | 72.60 | 7.39 | 0.585 | 5.78 | 0.569 | [20,6] | |
| | g-RT-2 | 73.56 | 2.69 | 0.872 | 5.91 | 0.628 | [10,2] | |
| 5 | t-RT | 25.25 | 0.79 | 0.991 | 36.72 | 0.638 | [15,2] | g-RT-2 |
| | Ftest-RT | 22.42 | 17.70 | 0.782 | 41.14 | 0.469 | [15,2] | |
| | g-RT-1 | 24.30 | 2.21 | 0.975 | 30.69 | 0.702 | [20,4] | |
| | g-RT-2 | 14.09 | 2.66 | 0.970 | 33.26 | 0.704 | [20,2] | |
| 6 | t-RT | 12.91 | 0.72 | 0.991 | 42.08 | 0.711 | [15,4] | g-RT-2 |
| | Ftest-RT | 19.52 | 15.49 | 0.780 | 58.05 | 0.538 | [10,2] | |
| | g-RT-1 | 13.26 | 4.07 | 0.947 | 56.29 | 0.749 | [10,2] | |
| | g-RT-2 | 5.89 | 2.14 | 0.973 | 42.66 | 0.783 | [10,6] | |
| 7 | t-RT | 106.49 | 9.49 | 0.983 | 18.67 | 0.969 | [10,6] | t-RT |
| | Ftest-RT | 124.26 | 189.27 | 0.582 | 223.14 | 0.500 | [15,6] | |
| | g-RT-1 | 75.64 | 17.50 | 0.969 | 20.88 | 0.964 | [15,4] | |
| | g-RT-2 | 56.76 | 9.92 | 0.983 | 18.05 | 0.968 | [10,6] | |
| 8 | t-RT | 4.16 | 27.14 | 0.997 | 421.26 | 0.981 | [20,2] | g-RT-2 |
| | Ftest-RT | 6.47 | 42.33 | 0.996 | 193.10 | 0.956 | [20,4] | |
| | g-RT-1 | 4.37 | 15.96 | 0.998 | 163.53 | 0.975 | [15,2] | |
| | g-RT-2 | 1.81 | 30.65 | 0.997 | 328.84 | 0.987 | [10,2] | |
| 9 | t-RT | 208.61 | 14.44 | 0.785 | 29.86 | 0.526 | [10,6] | g-RT-2 |
| | Ftest-RT | 236.44 | 36.15 | 0.202 | 37.83 | 0.133 | [15,4] | |
| | g-RT-1 | 103.62 | 34.19 | 0.305 | 36.49 | 0.223 | [20,6] | |
| | g-RT-2 | 124.24 | 13.94 | 0.794 | 29.58 | 0.535 | [10,6] | |
| 10 | t-RT | 20.46 | 0.32 | 0.923 | 1.29 | 0.695 | [15,6] | g-RT-2 |
| | Ftest-RT | 24.16 | 1.60 | 0.516 | 1.85 | 0.452 | [15,4] | |
| | g-RT-1 | 17.61 | 0.71 | 0.822 | 1.40 | 0.652 | [15,6] | |
| | g-RT-2 | 10.34 | 0.39 | 0.906 | 1.26 | 0.697 | [15,6] | |
| 11 | t-RT | 7.62 | 822.78 | 0.976 | 2515.46 | 0.965 | [15,2] | t-RT |
| | Ftest-RT | 10.74 | 4022.34 | 0.879 | 28145.06 | 0.513 | [15,4] | |
| | g-RT-1 | 6.19 | 2.82 | 1.000 | 55274.27 | 0.668 | [20,2] | |
| | g-RT-2 | 3.29 | 849.91 | 0.976 | 2767.98 | 0.965 | [10,2] | |
| 12 | t-RT | 11.26 | 11.20 | 0.970 | 69.24 | 0.781 | [15,4] | g-RT-2 |
| | Ftest-RT | 16.30 | 130.50 | 0.563 | 137.08 | 0.454 | [15,4] | |
| | g-RT-1 | 10.93 | 46.64 | 0.869 | 63.32 | 0.803 | [10,6] | |
| | g-RT-2 | 5.28 | 18.20 | 0.951 | 56.62 | 0.826 | [10,2] | |

[a] The tuple $[a, b]$ is the best hyperparameter combination, where $a$ is the *maximum-tree-depth* and $b$ is the *minimum-sample-split*.

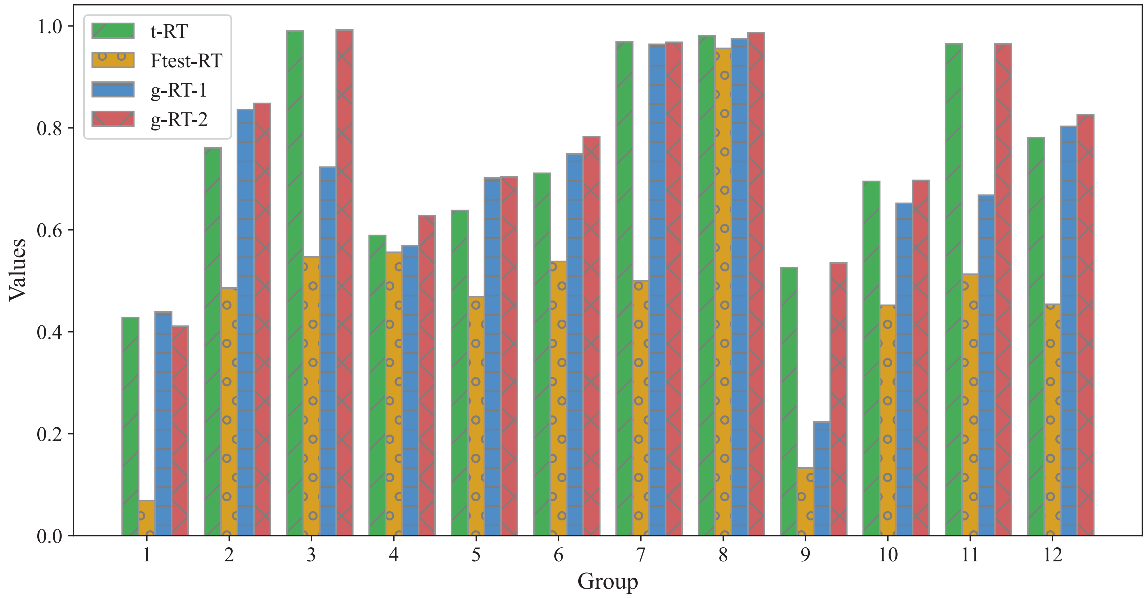

**FIGURE 2** | The testing correlation coefficients of different RT models on each dataset.

**TABLE 5** | The Wilcoxon signed-rank test of the correlation coefficients between RT models.

| Tested models | Sum of positive ranks | Sum of negative ranks | $W$ value | $p$-value |
|---|---|---|---|---|
| g-RT-1 and t-RT | 33 | 58 | 33 | 0.414 |
| g-RT-2 and t-RT | 71 | 7 | 7 | 0.009* |

*Note:* ∗ indicates that the null hypothesis can be rejected with a confidence level of 95%.

performs optimally on two datasets; g-RT-1 performs optimally on only dataset 1. In contrast, Ftest-RT consistently shows significantly worse performance than the other three models on both MSE and correlation coefficients, which may be attributed to its potential sensitivity to outliers and the non-normality of the data, as the F-test assumes homoscedasticity and normally distributed errors (James et al. 2013). In Figure 2, we present a bar plot illustrating the coefficients of all models on the test sets. g-RT-2 outperforms t-RT on average over the 12 datasets, but the performance of g-RT-1 is unstable, being much lower on datasets 9 and 11. The results underscore the effectiveness of our proposed enhanced splitting criteria, which achieve better approximations of the generalization error. Besides, the consistently superior prediction accuracy of g-RT-2 across various datasets can be attributed to its ability to generate a more diverse range of errors, as demonstrated in Example 1. This diversity aids in distinguishing more suitable split points, thereby enhancing model performance. The Ftest-RT exhibits markedly inferior performance, showing lower correlation coefficients across all twelve datasets. Since the prediction results of the Ftest-RT model are notably worse than those of the other three models, and its splitting principle is quite different from the other models based on generalization error, we do not conduct further significance tests or tree structure explorations on it.

We conduct the Wilcoxon Signed-Rank test to detect significant differences between the correlation coefficients of the t-RT model and the generalization error-based RT models. The null hypothesis is that the difference between the median of the correlation coefficients of the two models is zero. We use test correlation coefficients rather than MSE to compare prediction accuracy because they are standardized between zero and one, while the test MSE varies greatly across the dataset, making it meaningless for computing statistics such as median values. For simplicity, we refer to the test correlation coefficients simply as "coefficients" in the following illustration. We use the two-tailed Wilcoxon Signed-Rank test with $\alpha = 0.05$ and the critical value $W_{0.05}(12) = 13$, where 0.05 represents the significance level and 12 represents the number of samples. The coefficients of each model on the 12 datasets can be used to construct the $W$ value, as shown in Table 5. The columns "Sum of positive ranks" and "Sum of negative ranks" refer to the sums of the ranks for all differences where the coefficient of the first model is greater than and less than that of the second model, respectively. The $W$ value represents the smaller value of the two sums, which, if less than the critical value, indicates that the observed differences between the paired samples are statistically significant. The column "$p$-value" is the probability of observing a value for the statistic more extreme than, or as extreme as, the actual value of the statistic.

As shown in Table 5, the $W$ value and the $p$-value for g-RT-2 versus t-RT are 7 and 0.009, respectively, which fall in the rejection domain (i.e., g-RT-2 significantly outperforms t-RT at the 95% confidence level). However, the $W$ value of g-RT-1 versus t-RT is 33 with a $p$-value of 0.414, which is larger than 0.05. Therefore, there is no difference between the coefficients of g-RT-1 and t-RT at a significance level of less than 5%.

### 5.2.3 | Model Robustness

The robustness of a model, defined as its ability to maintain consistent performance in different situations, is another important consideration. Our model robustness evaluation is twofold. First,

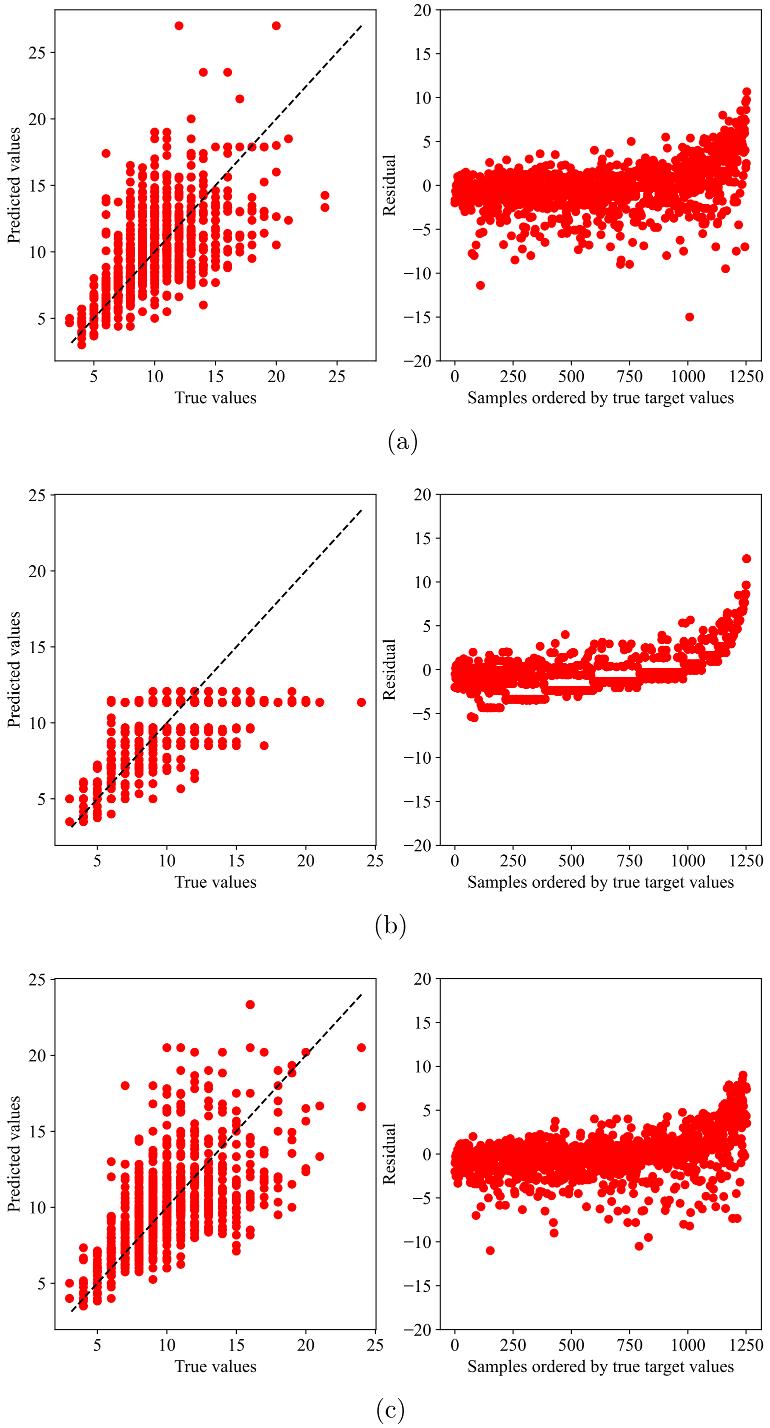

**FIGURE 3** | Residuals of RT models within dataset 4. (a) t-RT, (b) g-RT-1, and (c) g-RT-2.

we compare the prediction accuracy of the models on all points within one dataset. Second, we compare the performance of the models on various datasets.

*Robustness on one dataset*. We select dataset 4 as an example because the RT models exhibit diverse prediction accuracy on this dataset yet follow the same average overall performance trend (i.e., g-RT-2 performing optimally, followed by t-RT and g-RT-1). For each data point in the test set, we illustrate the relationship between the true value and its prediction in Figure 3. In the left panel of each figure, the true values are plotted against the predicted values. The right panel presents the residuals calculated as the true values minus the predicted values and orders them according to their true values for better visualization.

The left scatter plot of g-RT-2 shows a close relationship between the actual and predicted values, with the data points closely distributed around the line with slope one that passes through the origin. In contrast, the left scatter plot of t-RT has a wide distribution of data points, indicating that the relationship between the predicted and actual values is weaker than that of g-RT-2. Notably, the left scatter plot of g-RT-1 exhibits data points tightly distributed around a line with a slope less than 1, which may indicate that g-RT-1 systematically underestimates its predictions. The right scatter plot reveals that the predictions of all three models produce larger residuals when the true target values increase; however, g-RT-1 and g-RT-2 exhibit few outliers. However, g-RT-1 consistently underestimates the targets as the actual outputs increase. In summary, g-RT-2 emerges as the most robust model, followed by g-RT-1 and t-RT. Notably, g-RT-1 shows a tendency toward biased predictions, especially as the target values increase.

*Robustness across datasets*. We present a box plot of the coefficients of the RT models in Figure 4 for comparing their performance on different datasets.

According to the figure, g-RF-2 exhibits optimal performance, displaying the largest mean (cross mark) and median (midline) coefficients among the three models, followed by t-RT and g-RT-1. Model g-RT-2 has a slightly shorter box length than t-RT, indicating more stable performance across datasets; while g-RT-1 has the shortest box length but one outlier, implying more consistent performance on different datasets but occasionally unexpectedly poor outcomes. Conclusively, g-RF-2 exhibits the optimal average performance and is the most reliable model, producing consistently satisfactory results on various datasets.

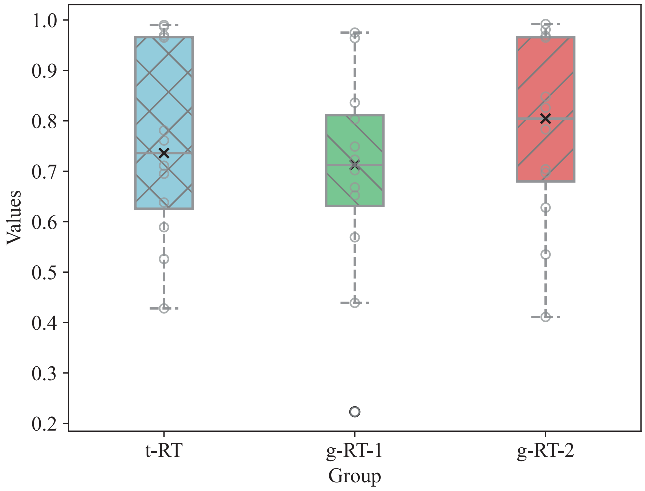

**FIGURE 4** | Box plot of the testing correlation coefficients of the RT models on different datasets.

### 5.2.4 | Tree Structure

In this section, we analyze the tree structure, including the maximum tree depth and the number of leaf nodes, to better understand how the three models perform regression. The specific tree structure information is provided in Table 6.

Model t-RT has the lowest average tree depth but the most leaf nodes. Low depth indicates that there are fewer decision points before reaching a leaf node, and thus the decision paths are simpler. More leaf nodes mean that there are more terminal regions for the target, which could indicate that the model is trying to capture more detail and variance in the data, and thus might be a sign of overfitting. The g-RT-1 algorithm exhibits the smallest average number of leaf nodes alongside the largest maximum tree depth among the models compared. This pattern suggests that the underlying tree structure of g-RT-1 possesses a few dominant branching paths characterized by numerous splits. This imbalance indicates that while certain branches are extensively developed, potentially capturing complex patterns in specific areas of the data, other branches are considerably underdeveloped, resulting in a lack of generalization across different segments of the dataset. Such a skewed tree structure may impede the model's overall performance on diverse datasets such as datasets 3, 9, and 11. Therefore, although the g-RT-1 may provide better approximations to the generalization error in each split, its tendency to focus on certain data features likely contributes to its suboptimal performance on datasets that require a more balanced

**TABLE 6** | The maximum tree depth and the number of leaf nodes of three RT models.

| | Maximum tree depth | | | Number of leaf nodes | | |
|---|---|---|---|---|---|---|
| **Dataset** | **t-RT** | **g-RT-1** | **g-RT-2** | **t-RT** | **g-RT-1** | **g-RT-2** |
| 1 | 10 | 15 | 10 | 158 | 66 | 174 |
| 2 | 10 | 15 | 15 | 21 | 28 | 26 |
| 3 | 10 | 10 | 15 | 111 | 55 | 120 |
| 4 | 10 | 20 | 10 | 376 | 134 | 386 |
| 5 | 15 | 20 | 15 | 533 | 487 | 323 |
| 6 | 20 | 15 | 15 | 73 | 102 | 96 |
| 7 | 10 | 20 | 10 | 507 | 812 | 443 |
| 8 | 15 | 10 | 10 | 68 | 69 | 67 |
| 9 | 10 | 20 | 10 | 373 | 136 | 341 |
| 10 | 10 | 10 | 15 | 111 | 62 | 151 |
| 11 | 20 | 15 | 15 | 92 | 36 | 83 |
| 12 | 15 | 10 | 20 | 114 | 38 | 99 |
| Average | 12.92 | 15.00 | 13.33 | 211.42 | 168.75 | 192.42 |

exploration of feature space. The tree structure of g-RT-2 stands out as having an appropriate depth and number of leaf nodes, indicating that the model partitions the input space into a reasonable number of regions, each representing a different output of the response variable; this structure may explain why g-RT-2 outperforms the other two models.

*Summary*. Our experiments and comparative analysis of the four RT models on 12 datasets indicate the marked superiority of our proposed g-RT-2 model. This model not only outperforms t-RT, as evidenced by a Wilcoxon Signed-Rank test in the rejection domain, indicating its significant superiority at the 95% confidence level, but also demonstrates superior robustness and consistent performance. In contrast, g-RT-1, despite having a performance generally similar to that of t-RT, exhibits notable instability, with especially poor results on datasets 9 and 11. Our tree structure analysis reveals that t-RT has the smallest average tree depth and most leaf nodes, indicating potential overfitting; g-RT-1 has the fewest leaf nodes on average yet the largest maximum tree depth, suggesting high model complexity. The structural attributes of g-RT-2, with its appropriate tree depth and balanced number of leaf nodes, are likely to contribute to its effective partitioning of the input space, facilitating its superior performance across datasets and demonstrating the effectiveness of our LOOCV-informed splitting criterion.

## 5.3 | Model Comparison Without Hyperparameter Tuning

This section conducts two sets of experiments to ensure a fair comparison of the splitting criteria and mitigate the potential influence of varying hyperparameters and recursive partitioning.

### 5.3.1 | Model Evaluation With Fixed Hyperparameters

To mitigate the influence of hyperparameter selection, we test all models with fixed hyperparameters as we set the maximum tree depth as well as the minimum number of samples to split as 50 and 2, respectively, which are consistent with the default settings in the commonly used ML library scikit-learn (Buitinck et al. 2013). Other experimental settings align with those specified in Section 5.1. The experiment results are presented in Table 7, where the column definitions are the same as Table 4.

Table 7 shows a minor variation of computational time between different models, with g-RT-2 generally requiring less time compared to t-RT and g-RT-1. For example, in Dataset 1, g-RT-2 completes the computations within approximately 20 s, while t-RT requires 42.96 s and g-RT-1 takes even longer (56.25 s). However, the computational time is more influenced by the dataset's total number of samples and the number of features. Moreover, the computational times for all datasets are generally shorter than those reported in Table 4, as the hyperparameters are fixed and no additional time is needed for hyperparameter tuning.

In terms of prediction accuracy, evaluated through MSE and correlation coefficients, the performances of all three models on the testing set are inferior to those in Table 4, where models are trained after hyperparameter tuning. In the testing phase, our proposed models, namely g-RT-1 and g-RT-2, consistently outperform t-RT. Specifically, g-RT-2 achieves the highest correlation coefficients on nine datasets, and g-RT-1 outperforms other models in two datasets, while t-RT achieves a better testing correlation coefficient and MSE on only one dataset. Overall, the results highlight the superior prediction accuracy of our generalization error-based RT models under the fixed hyperparameters. We note that the performance of g-RT-1 on dataset 3 is significantly inferior to that of the other two models. The abnormal results of g-RT-1 on dataset 3 can be attributed to the characteristics of the dataset. Specifically, dataset 3 contains only 309 samples, making it a relatively small dataset. Additionally, the target variable in this dataset has a wide range, varying from 0.01 to 62.42. This combination of limited data and large variability in the target can amplify the sensitivity of regression tree models to small structural changes in the tree.

### 5.3.2 | Model Evaluation With Limited Splits

To diminish the effects of greedy splitting and the recursive partitioning process, we also compare our RT models with the t-RT by limiting the number of splits to three. Specifically, we set the maximum tree depth to three while keeping all other experimental settings consistent with those outlined in Section 5.1. The experimental results of computational time and prediction accuracy for various RT models across multiple datasets are shown in Table 8.

The results demonstrate that when the number of splits is limited, the performances of all RT models are generally similar across most datasets. For example, models such as t-RT, g-RT-1, and g-RT-2 often achieve nearly identical MSE and correlation coefficients, as seen in datasets 1, 2, 4, 5, 7, 8, and 9. However, the t-RT and g-RT-1 occasionally show significantly worse performance than g-RT-2, as reflected in datasets 3, 6, 10, 11, and 12, where their testing MSE and correlation coefficients are noticeably inferior compared to g-RT-2.

Despite the similarity, our proposed models, particularly g-RT-2, still stand out by providing better prediction accuracy. For instance, in dataset 6, g-RT-2 achieves the highest correlation coefficient (0.872), followed by g-RT-1 and then t-RT. Similarly, in datasets 10, 11, and 12, g-RT-2 achieves improvements in testing metrics, that is, a higher correlation coefficient and a lower MSE, compared to the t-RT. These results highlight the effectiveness of the generalization error-based RT models under the settings of limited splits, indicating the enhancement of our proposed splitting method in a more direct manner.

## 5.4 | Model Comparison on Non-I.I.D. Datasets

In this article, the assumption that training samples are i.i.d. is central to the validity of key findings. To illustrate the robustness of their method under relaxed i.i.d. conditions, this section tests and compares the regression tree models based on our proposed splitting criterion using two classic non-i.i.d. datasets, that is, the Yahoo finance dataset and the European climate assessment & dataset (ECA&D).

**TABLE 7** | Experiment results of different RT models with the same hyperparameter.

| Dataset | Model | Time (s) | Training set | | Testing set | | Best model |
|---|---|---|---|---|---|---|---|
| | | | MSE | Coeff | MSE | Coeff | |
| 1 | t-RT | 42.96 | 0.02 | 0.987 | 1.04 | 0.373 | g-RT-2 |
| | g-RT-1 | 56.25 | 0.18 | 0.888 | 0.91 | 0.337 | |
| | g-RT-2 | 19.49 | 0.06 | 0.963 | 0.96 | 0.399 | |
| 2 | t-RT | 0.41 | 0.01 | 0.982 | 0.04 | 0.793 | g-RT-2 |
| | g-RT-1 | 0.42 | 0.00 | 0.989 | 0.05 | 0.793 | |
| | g-RT-2 | 0.16 | 0.01 | 0.974 | 0.04 | 0.820 | |
| 3 | t-RT | 1.50 | 0.58 | 0.999 | 9.09 | 0.985 | g-RT-2 |
| | g-RT-1 | 1.49 | 31.98 | 0.924 | 186.98 | 0.529 | |
| | g-RT-2 | 0.65 | 0.18 | 1.000 | 2.68 | 0.995 | |
| 4 | t-RT | 29.25 | 0.32 | 0.986 | 9.10 | 0.518 | g-RT-1 |
| | g-RT-1 | 43.76 | 0.62 | 0.972 | 8.45 | 0.545 | |
| | g-RT-2 | 13.98 | 0.57 | 0.974 | 8.36 | 0.534 | |
| 5 | t-RT | 4.64 | 1.14 | 0.987 | 28.90 | 0.701 | t-RT |
| | g-RT-1 | 4.82 | 3.11 | 0.965 | 33.11 | 0.632 | |
| | g-RT-2 | 2.40 | 3.03 | 0.966 | 29.37 | 0.671 | |
| 6 | t-RT | 2.49 | 0.53 | 0.993 | 41.41 | 0.694 | g-RT-2 |
| | g-RT-1 | 2.93 | 0.26 | 0.997 | 53.17 | 0.675 | |
| | g-RT-2 | 1.06 | 0.93 | 0.988 | 42.64 | 0.752 | |
| 7 | t-RT | 25.78 | 1.08 | 0.998 | 21.96 | 0.962 | g-RT-2 |
| | g-RT-1 | 30.67 | 0.87 | 0.998 | 23.02 | 0.961 | |
| | g-RT-2 | 11.60 | 2.43 | 0.996 | 20.78 | 0.965 | |
| 8 | t-RT | 0.84 | 23.88 | 0.998 | 176.68 | 0.975 | g-RT-1 |
| | g-RT-1 | 0.87 | 34.66 | 0.996 | 320.99 | 0.982 | |
| | g-RT-2 | 0.34 | 21.67 | 0.998 | 324.02 | 0.982 | |
| 9 | t-RT | 53.52 | 0.95 | 0.987 | 35.11 | 0.540 | g-RT-2 |
| | g-RT-1 | 74.15 | 3.29 | 0.955 | 37.59 | 0.490 | |
| | g-RT-2 | 26.91 | 1.20 | 0.984 | 33.30 | 0.561 | |
| 10 | t-RT | 4.24 | 0.07 | 0.983 | 1.76 | 0.626 | g-RT-2 |
| | g-RT-1 | 5.18 | 0.03 | 0.994 | 1.67 | 0.637 | |
| | g-RT-2 | 1.85 | 0.16 | 0.964 | 1.40 | 0.714 | |
| 11 | t-RT | 1.41 | 5.14 | 1.000 | 19927.54 | 0.816 | g-RT-2 |
| | g-RT-1 | 1.78 | 5.37 | 1.000 | 5111.47 | 0.950 | |
| | g-RT-2 | 0.60 | 879.68 | 0.975 | 1831.59 | 0.974 | |
| 12 | t-RT | 2.33 | 4.14 | 0.989 | 118.08 | 0.689 | g-RT-2 |
| | g-RT-1 | 2.69 | 1.28 | 0.997 | 53.91 | 0.842 | |
| | g-RT-2 | 0.94 | 13.86 | 0.963 | 47.19 | 0.849 | |

The Yahoo finance dataset comprises financial time-series data recorded from April 1, 2018, to March 31, 2023, containing 1257 rows and 6 columns (Kaggle 2023). It includes five key features: Opening price, closing price, highest price, lowest price, and trading volume. The target variable is the asset's closing price on a given date. As the observations are sequential, with each data point influenced by its predecessors, the dataset is inherently non-i.i.d. The ECA&D dataset consists of 3654 daily climate-related observations collected from meteorological stations across Europe between 2000 and 2010 (Klein Tank et al. 2002). It includes 11 climate-related feature variables, such as precipitation, cloud cover, humidity, wind speed, and others, with the target variable being the mean temperature. Because the daily climate observations exhibit temporal dependencies and spatial correlations, they are also non-i.i.d.

**TABLE 8** | Experiment results of different RT models with small tree depth.

| | | | Training set | | Testing set | | |
|---|---|---|---|---|---|---|---|
| **Dataset** | **Model** | **Time (s)** | **MSE** | **Coeff** | **MSE** | **Coeff** | **Best model** |
| 1 | t-RT | 0.95 | 0.58 | 0.530 | 0.53 | 0.525 | — |
| | g-RT-1 | 0.99 | 0.58 | 0.530 | 0.53 | 0.525 | |
| | g-RT-2 | 4.65 | 0.58 | 0.530 | 0.53 | 0.525 | |
| 2 | t-RT | 0.01 | 0.03 | 0.930 | 0.07 | 0.645 | — |
| | g-RT-1 | 0.01 | 0.03 | 0.930 | 0.07 | 0.645 | |
| | g-RT-2 | 0.07 | 0.03 | 0.924 | 0.07 | 0.645 | |
| 3 | t-RT | 0.17 | 10.65 | 0.975 | 19.26 | 0.962 | g-RT-2 |
| | g-RT-1 | 0.16 | 42.47 | 0.898 | 143.06 | 0.666 | |
| | g-RT-2 | 0.12 | 1.92 | 0.996 | 3.95 | 0.992 | |
| 4 | t-RT | 1.73 | 5.87 | 0.652 | 6.19 | 0.657 | — |
| | g-RT-1 | 1.73 | 5.87 | 0.652 | 6.19 | 0.657 | |
| | g-RT-2 | 2.89 | 5.87 | 0.652 | 6.19 | 0.657 | |
| 5 | t-RT | 0.05 | 24.66 | 0.711 | 24.70 | 0.647 | — |
| | g-RT-1 | 0.05 | 24.66 | 0.711 | 24.70 | 0.647 | |
| | g-RT-2 | 4.60 | 24.66 | 0.711 | 24.70 | 0.647 | |
| 6 | t-RT | 0.09 | 9.48 | 0.919 | 17.24 | 0.847 | g-RT-2 |
| | g-RT-1 | 0.09 | 9.50 | 0.919 | 16.86 | 0.851 | |
| | g-RT-2 | 2.86 | 9.51 | 0.919 | 14.47 | 0.872 | |
| 7 | t-RT | 2.41 | 25.89 | 0.954 | 28.30 | 0.950 | — |
| | g-RT-1 | 2.40 | 25.89 | 0.954 | 28.30 | 0.950 | |
| | g-RT-2 | 2.62 | 25.89 | 0.954 | 28.30 | 0.950 | |
| 8 | t-RT | 0.02 | 471.71 | 0.944 | 507.79 | 0.932 | — |
| | g-RT-1 | 0.02 | 471.71 | 0.944 | 507.79 | 0.932 | |
| | g-RT-2 | 0.26 | 471.71 | 0.944 | 507.79 | 0.932 | |
| 9 | t-RT | 9.55 | 28.86 | 0.492 | 29.08 | 0.476 | — |
| | g-RT-1 | 9.55 | 28.86 | 0.492 | 29.08 | 0.476 | |
| | g-RT-2 | 20.07 | 28.86 | 0.492 | 29.08 | 0.476 | |
| 10 | t-RT | 0.28 | 0.96 | 0.739 | 1.25 | 0.645 | g-RT-2 |
| | g-RT-1 | 0.27 | 0.96 | 0.739 | 1.25 | 0.645 | |
| | g-RT-2 | 18.15 | 0.99 | 0.731 | 1.19 | 0.664 | |
| 11 | t-RT | 0.04 | 586.69 | 0.989 | 6042.49 | 0.895 | g-RT-2 |
| | g-RT-1 | 0.04 | 586.69 | 0.989 | 6042.49 | 0.895 | |
| | g-RT-2 | 1.21 | 586.69 | 0.989 | 2705.65 | 0.931 | |
| 12 | t-RT | 0.13 | 46.67 | 0.852 | 92.52 | 0.761 | g-RT-2 |
| | g-RT-1 | 0.13 | 46.67 | 0.852 | 92.52 | 0.761 | |
| | g-RT-2 | 4.67 | 46.62 | 0.852 | 92.05 | 0.763 | |

We test the three RT models on the two datasets with other experimental settings aligned with those specified in Section 5.1. The detailed results are given in Table 9.

For both datasets, g-RT-2 is identified as the best model, achieving the highest correlation coefficients on the testing set, outperforming the other models. However, g-RT-2 requires more computational time (117.68 s) compared to t-RT and g-RT-1 on the weather prediction dataset, both of which are faster but less accurate. For the Yahoo finance dataset, g-RT-2 stands out by achieving the lowest testing MSE at 12805.82 and the highest correlation coefficients at 0.994, followed by g-RT-1 and then t-RT. For the Yahoo finance dataset, t-RT and g-RT-1 demonstrate similar computational times, while g-RT-2 requires less time, which may be due to its smaller tree depth.

**TABLE 9** | Experiment results of different RT models on non-i.i.d. datasets.

| Dataset | Model | Time (s) | Training set | | Testing set | | Hyperparameters[a] | Best model |
|---|---|---|---|---|---|---|---|---|
| | | | MSE | Coeff | MSE | Coeff | | |
| Weather prediction | t-RT | 67.41 | 12.66 | 0.898 | 31.33 | 0.727 | [10,6] | g-RT-2 |
| | g-RT-1 | 56.31 | 33.27 | 0.702 | 30.77 | 0.699 | [20,2] | |
| | g-RT-2 | 177.68 | 12.99 | 0.895 | 29.30 | 0.740 | [10,6] | |
| Yahoo finance | t-RT | 47.05 | 864.89 | 1.000 | 18338.20 | 0.991 | [20,2] | g-RT-2 |
| | g-RT-1 | 50.18 | 837.29 | 1.000 | 18117.51 | 0.992 | [20,4] | |
| | g-RT-2 | 21.90 | 1102.71 | 1.000 | 12805.82 | 0.994 | [15,2] | |

[a]The tuple $[a, b]$ is the best hyperparameter combination, where $a$ is the *maximum-tree-depth* and $b$ is the *minimum-sample-split*.

Overall, the performance of g-RT-2 across these two datasets is the best, slightly outperforming others on the Yahoo finance dataset and excelling on the weather prediction dataset. These results highlight the robustness of the proposed RT models on non-i.i.d. datasets. In fact, while much real-world data is not strictly i.i.d., the i.i.d. assumption often serves as a reasonable approximation, particularly for large datasets where individual dependencies or non-homogeneous distributions can be treated as noise.

## 6 | Conclusion

RT models are favored in prediction tasks for their interpretability and ability to handle diverse data without stringent assumptions. Despite these advantages, we believe that traditional RT models are susceptible to overfitting because they use a splitting criterion based solely on minimizing training errors. To overcome this ignorance of generalization errors in the determination of the splitting criterion, this article presents an accurate formula for the generalization error for each split. While the actual generalization error is intractable, we mathematically derive two approximations. One approximation estimates the variance of the predicted target for each child node, and the other uses LOOCV to approximate the out-of-sample error. Both methods can be implemented efficiently in model construction. We conduct extensive numerical experiments on 12 widely used ML datasets to test our proposed RT models and compare them with a traditional RT model. The results demonstrate the superiority of our proposed LOOCV-informed splitting criterion over the traditional splitting criterion in terms of prediction accuracy and model stability. We further analyze the tree structures resulting from different splitting criteria, indicating more reasonable splits based on the LOOCV-informed splitting criterion.

The primary limitations and edge cases of this research lie in the following two aspects. Firstly, while the proposed splitting criteria are shown to provide closer approximations to the generalization error compared to traditional in-sample MSE at each split, and experimental data indicate enhanced predictive accuracy, particularly with g-RT-2, across the entire tree in widely used ML datasets, a more comprehensive theoretical analysis of these enhancements throughout the whole tree can still be explored in the future. Secondly, our splitting criteria are more complex and less interpretable than in-sample MSE. While our splitting criteria are advantageous in scenarios with limited data availability, for large datasets, all splitting methods are likely to probabilistically converge to the true conditional mean, which may diminish the performance improvements of our model compared to the traditional approach.

### Data Availability Statement

The data supporting the findings of this study are openly available in the Data-sets repository at https://github.com/ShadowY1998/Data-sets.

### References

Aouad, A., A. N. Elmachtoub, K. J. Ferreira, and R. McNellis. 2023. "Market Segmentation Trees." *Manufacturing & Service Operations Management* 25, no. 2: 648–667.

Bandi, H., and D. Bertsimas. 2021. "Optimizing Influenza Vaccine Composition: A Machine Learning Approach." *Naval Research Logistics* 68, no. 7: 857–870.

Baron, S., and D. Phillips. 1994. "Attitude Survey Data Reduction Using CHAID: An Example in Shopping Centre Market Research." *Journal of Marketing Management* 10, no. 1–3: 75–88.

Bengio, Y., A. Lodi, and A. Prouvost. 2021. "Machine Learning for Combinatorial Optimization: A Methodological Tour D'horizon." *European Journal of Operational Research* 290, no. 2: 405–421.

Bertsimas, D., J. Pauphilet, J. Stevens, and M. Tandon. 2022. "Predicting Inpatient Flow at a Major Hospital Using Interpretable Analytics." *Manufacturing & Service Operations Management* 24, no. 6: 2809–2824.

Biggs, D., B. De Ville, and E. Suen. 1991. "A Method of Choosing Multiway Partitions for Classification and Decision Trees." *Journal of Applied Statistics* 18, no. 1: 49–62.

Breiman, L. 2017. *Classification and Regression Trees*. Routledge.

Buitinck, L., G. Louppe, M. Blondel, et al. 2013. "API Design for Machine Learning Software: Experiences From the Scikit-Learn Project." In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108–122. Springer.

Che, D., Q. Liu, K. Rasheed, and X. Tao. 2011. "Decision Tree and Ensemble Learning Algorithms With Their Applications in Bioinformatics." In *Software Tools and Algorithms for Biological Systems*, Vol. 696, 191–199. Springer.

Chou, Y.-C., H. H.-C. Chuang, P. Chou, and R. Oliva, 2023. "Supervised Machine Learning for Theory Building and Testing: Opportunities in Operations Management." *Journal of Operations Management* 69, no. 4: 643–675.

Fisher, R. A., R. A. Fisher, S. Genetiker, et al. 1966. *The Design of Experiments*, Vol. 21. Springer.

Fonarow, G. C., K. F. Adams, W. T. Abraham, et al. 2005. "Risk Stratification for in-Hospital Mortality in Acutely Decompensated Heart Failure: Classification and Regression Tree Analysis." *JAMA* 293, no. 5: 572–580.

Gkioulekas, I., and L. G. Papageorgiou. 2021. "Tree Regression Models Using Statistical Testing and Mixed Integer Programming." *Computers & Industrial Engineering* 153: 107059.

Hastie, T., J. Friedman, and R. Tibshirani. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Vol. 2. Springer.

Hoerl, A. E., and R. W. Kennard. 1970. "Ridge Regression: Biased Estimation for Nonorthogonal Problems." *Technometrics* 12, no. 1: 55–67.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 112. Springer.

Kaggle. 2023. "Yahoo Finance Dataset." https://www.kaggle.com/datasets/suruchiarora/yahoo-finance-dataset-2018-2023.

Kao, H.-P., and K. Tang. 2014. "Cost-Sensitive Decision Tree Induction With Label-Dependent Late Constraints." *INFORMS Journal on Computing* 26, no. 2: 238–252.

Karim, M., and R. M. Rahman. 2013. "Decision Tree and Naive Bayes Algorithm for Classification and Generation of Actionable Knowledge for Direct Marketing." *Journal of Software Engineering and Applications* 6, no. 4: 196–206.

Kass, G. V. 1980. "An Exploratory Technique for Investigating Large Quantities of Categorical Data." *Journal of the Royal Statistical Society: Series C: Applied Statistics* 29, no. 2: 119–127.

Kelly, M., R. Longjohn, and K. Nottingham. 2023. "The UCI Machine Learning Repository." http://archive.ics.uci.edu/.

Klein Tank, A. M., J. Wijngaard, G. Können, et al. 2002. "Daily Dataset of 20th-Century Surface Air Temperature and Precipitation Series for the European Climate Assessment." *International Journal of Climatology: A Journal of the Royal Meteorological Society* 22, no. 12: 1441–1453.

Kobayashi, D., O. Takahashi, H. Arioka, S. Koga, and T. Fukui. 2013. "A Prediction Rule for the Development of Delirium Among Patients in Medical Wards: Chi-Square Automatic Interaction Detector (Chaid) Decision Tree Analysis Model." *American Journal of Geriatric Psychiatry* 21, no. 10: 957–962.

Kuhn, R., and R. De Mori. 2002. "The Application of Semantic Classification Trees to Natural Language Understanding." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17, no. 5: 449–460.

Kumar, A., and A. Kaur. 2023. "Predicting Complaint Voicing or Exit Amidst Indian Consumers: A CHAID Analysis." *Journal of Advances in Management Research* 20, no. 1: 55–78.

Larsen, R. J., and M. L. Marx. 2005. *An Introduction to Mathematical Statistics*. Prentice Hall.

Liu, Y. 2022. "Bsnsing: A Decision Tree Induction Method Based on Recursive Optimal Boolean Rule Composition." *INFORMS Journal on Computing* 34, no. 6: 2908–2929.

Loh, W.-Y. 2011. "Classification and Regression Trees." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1, no. 1: 14–23.

MathWorks. 2023. "Growing Decision Trees." https://ww2.mathworks.cn/help/stats/growing-decision-trees.html.

Miller, B., M. Fridline, P.-Y. Liu, and D. Marino. 2014. "Use of Chaid Decision Trees to Formulate Pathways for the Early Detection of Metabolic Syndrome in Young Adults." *Computational and Mathematical Methods in Medicine* 2014, no. 1: 242717.

Mišić, V. V., and G. Perakis. 2020. "Data Analytics in Operations Management: A Review." *Manufacturing & Service Operations Management* 22, no. 1: 158–169.

Murtaugh, P. A. 1998. "Methods of Variable Selection in Regression Modeling." *Communications in Statistics: Simulation and Computation* 27, no. 3: 711–734.

Naumov, G. 1991. "NP-Completeness of Problems of Construction of Optimal Decision Trees." *Soviet Physics – Doklady* 36: 270.

Pekel, E. 2020. "Estimation of Soil Moisture Using Decision Tree Regression." *Theoretical and Applied Climatology* 139, no. 3–4: 1111–1119.

Quinlan, J. R. 1986. "Induction of Decision Trees." *Machine Learning* 1: 81–106.

Salari, N., S. Liu, and Z.-J. M. Shen. 2022. "Real-Time Delivery Time Forecasting and Promising in Online Retailing: When Will Your Package Arrive?" *Manufacturing & Service Operations Management* 24, no. 3: 1421–1436.

Scikit-learn. 2025. "Decision Trees." https://scikit-learn.org/stable/modules/tree.html.

Shannon, C. E. 1948. "A Mathematical Theory of Communication." *Bell System Technical Journal* 27, no. 3: 379–423.

Stone, M. 1974. "Cross-Validatory Choice and Assessment of Statistical Predictions." *Journal of the Royal Statistical Society, Series B: Statistical Methodology* 36, no. 2: 111–133.

Sun, L., G. Lyu, Y. Yu, and C.-P. Teo. 2020. "Fulfillment by Amazon versus Fulfillment by Seller: An Interpretable Risk-Adjusted Fulfillment Model." *Naval Research Logistics* 67, no. 8: 627–645.

Tibshirani, R. 1996. "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society, Series B: Statistical Methodology* 58, no. 1: 267–288.

Wu, X., V. Kumar, J. Ross Quinlan, et al. 2008. "Top 10 Algorithms in Data Mining." *Knowledge and Information Systems* 14: 1–37.

Yilgör, A. G., Ü. Doğrul, and T. Orekici. 2011. "A New Approach to Predict Financial Failure: Classification and Regression Trees (Cart)." *Journal of Modern Accounting and Auditing* 7, no. 4: 329–339.

Ying, X. 2019. "An Overview of Overfitting and Its Solutions." In *Journal of Physics: Conference Series*, vol. 1168, 22022. IOP Publishing.

## Appendix A

### Enhancement Analysis of Variance-Estimated Splitting Criterion

We take the left child node of the root node as an example. For notation convenience, we let $M = |\Gamma_1(j, s_j)|$. Then, let $Y_1, Y_2, \ldots, Y_M$ be $M$ random samples of $Y$ given $X^{(j)} \le s_j$ and $\overline{Y} = \frac{\sum_{i=1}^{M} Y_i}{M}$. We denote $\mathbb{E}[Y(X^{(j)} \le s_j; \widetilde{D})] = \mu$ and $\mathbb{V}[Y(X^{(j)} \le s_j; \widetilde{D})] = \sigma^2$. Then, we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{M}\left(Y_i - \overline{Y}\right)^2\right] &= \mathbb{E}\left[\sum_{i=1}^{M} Y_i^2 - \frac{1}{M}\left(\sum_{i=1}^{M} Y_i\right)^2\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{M} Y_i^2 - M\overline{Y}^2\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{M} Y_i^2\right] - M\mathbb{E}\left[\overline{Y}^2\right] \\
&= \sum_{i=1}^{M} \mathbb{E}\left[Y_i^2\right] - M\mathbb{E}\left[\overline{Y}^2\right] \\
&= \sum_{i=1}^{M} (\sigma^2 + \mu^2) - M\left(\frac{\sigma^2}{M} + \mu^2\right) \\
&\left(\sigma^2 = \mathbb{E}\left[Y_i^2\right] - \mu^2, \frac{\sigma^2}{M} = \mathbb{E}\left[\overline{Y}^2\right] - \mu^2\right) \\
&= M(\sigma^2 + \mu^2) - M\left(\frac{\sigma^2}{M} + \mu^2\right)
\end{aligned}$$

$$= (M-1)\sigma^2$$

Therefore, we have $\mathbb{E}\left[\frac{\sum_{i=1}^{M}\left(Y_i-\overline{Y}\right)^2}{M-1}\right] = \sigma^2$, that is, it is an unbiased estimator to $\mathbb{V}[Y(X^{(j)} \leq s_j; \widetilde{D})]$. In contrast, the traditional splitting criterion yields $\frac{M-1}{M}\sigma^2$, which is a biased estimator. Similarly, we can prove it for the right child node of the root node.