IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 May 2024, accepted 14 June 2024, date of publication 26 June 2024, date of current version 8 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3419228

RESEARCH ARTICLE

Assessing Decision Tree Stability: A Comprehensive Method for Generating a Stable Decision Tree

JEONGEON LEE, MIN KYU SIM, (Member, IEEE), AND JUNG-SIK HONG
Department of Data Science, Seoul National University of Science and Technology, Seoul 01811, South Korea

Corresponding author: Jung-Sik Hong (hong@seoultech.ac.kr)

This work was supported by the Basic Science Research Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2021R1A6A1A03039981 and Grant NRF-2020R1F1A107387912.

ABSTRACT **Objectives:** This paper proposes a novel stability metric for decision trees that does not rely on the elusive notion of tree similarity. Existing stability metrics have been constructed in a pairwise fashion to assess the tree similarity between two decision trees. However, quantifying the structural similarities between decision trees is inherently elusive. Conventional stability metrics are simply relying on partial information such as the number of nodes and the depth of the tree, which do not adequately capture structural similarities. **Methods:** We evaluate the stability based on the computational burden required to generate a stable tree. First, we generate a stable tree using the novel adaptive node-level stabilization method, which determines the most frequently selected predictor during the bootstrapping iterations of a decision tree branching process at each node. Second, the stability is measured based on the number of bootstraps required to achieve the stable tree. **Findings:** Using the proposed stability metric, we compare the stability of four popular decision tree splitting criteria: Gini index, entropy, gain ratio, and chi-square. In an empirical study across ten datasets, the gain ratio is the most stable splitting criterion among the four popular criteria. Additionally, a case study demonstrates that applying the proposed method to the classification and regression tree (CART) algorithm generates a more stable tree compared to the one produced by the original CART algorithm. **Novelty:** We propose a stability metric for decision trees without relying on measuring pairwise tree similarity. This paper provides a stability comparison of four popular decision tree splitting criteria, delivering practical insights into their reliability. The adaptive node-level stabilization method can be applied across various decision tree algorithms, enhancing tree stability and reliability in scenarios with updating data.

INDEX TERMS Decision trees, stability, splitting criteria.

I. INTRODUCTION

Recent advances in machine learning have not only improved model accuracy but also enhanced its utility as a decision-making tool [1], [2]. Highly interpretable models that demonstrate transparent decision-making processes are becoming increasingly important [3], [4], [5]. Decision trees provide transparent inferences with if-then rules. The superior interpretability of decision trees among machine learning classifiers is affirmed by their efficient response times to explanatory queries [6]. Owing to their interpretability, decision trees are widely used in various fields, including psychology [7], [8], finance [9], bioinformatics [10], smart farming [11], and predictive analytics such as customer churn prediction [12], [13].

Despite the strong interpretability of decision trees, their use as decision-making tools is often limited because of their inherent instability [14]. A slight change in the training dataset can result in a significantly different tree [15], [16]. This instability stems from the sensitivity of the branching processes of the decision tree to variations in data and exacerbated by the hierarchical structure of decision trees, in which branching decisions at higher levels have a cascading effect on those at lower levels [15]. The instability obstructs

The associate editor coordinating the review of this manuscript and approving it for publication was Leimin Wang.

effective knowledge discovery, particularly in online learning scenarios, where models are continually updated using new data [16], [17]. For instance, engineers using decision trees to identify manufacturing inefficiencies may find the interpretation challenging if the model undergoes significant changes following updates to the training dataset [16]. Difficulties caused by instability could potentially undermine the reliability of the model. Hence, examining the instability of decision trees has become a paramount concern from the perspective of the interpretability and reliability of a model. To assess instability, it is necessary to measure the stability of the decision trees.

The stability of a machine learning model can be defined from two principal perspectives: *semantic* and *syntactic* (or structural) [16], [18], [19]. Semantic stability evaluates the consistency of predictions for the same dataset across multiple models, without considering the model structure [16], [20]. Semantically identical trees with identical predictions may have different structures. By contrast, syntactic stability gauges the similarity between tree structures. Two syntactically identical trees possess the same structure as well as the same predictions [19]. In a context in which the decision-making process needs to be transparent and interpretable, syntactic stability postulates greater significance. Focusing on syntactic stability results in a model that offers both consistent decision making and predictable output.

However, assessing syntactic stability is challenging because of the elusive task of measuring structural similarity between multiple trees. Previous studies attempting to compare the structures of multiple trees have often relied on partial information, focused on specific aspects rather than the overall structure, or made assumptions about structural similarity, limiting their ability to comprehensively assess syntactic stability [21], [22], [23].

To the best of our knowledge, no previous study has adequately considered the entire tree structure under appropriate assumptions when measuring tree similarity, and by extension, syntactic stability. Our idea relies on an alternative approach to measuring syntactic stability, rather than tree similarity. We address this gap by defining a numerical measure of syntactic stability based on the number of iterations required to stabilize the tree structure.

We propose a measure for assessing the syntactic stability of decision trees, furthering the concept of ''node-level stabilization'' as proposed in the study of [22]. The study of [22] introduced an innovative method to stabilize decision trees by selecting the most frequent predictor at the node level. However, the study of [22] did not consider the number of bootstrap iterations required to achieve a statistically significant confidence level for the frequency ranking of the most frequent predictor. We introduce rank verification [24] for the most frequent predictor and repeat the bootstrap process until the statistical confidence level is satisfied to stabilize the tree. Our proposed stability measure is based on the idea that if the same most frequent predictor continuously emerges during the bootstrapping iterations at the node level, the tree is relatively stable. Specifically, we measure the stability during the predictor selection process by counting the number of iterations required to finalize the predictor at a bootstrapped node. If a predictor is selected with a small number of iterations, the branching process of the tree is regarded relatively stable. However, if a large number of iterations are required to determine the predictor, the branching process is regarded as relatively unstable.

Our empirical study using 10 datasets compares the splitting criteria of popular decision tree algorithms, including the Gini index (CART) [25], entropy (ID3) [26], gain ratio (C4.5) [27], and chi-square (CHAID) [28]. Splitting criteria are the methods utilized to determine the optimal split in the branching process of the decision tree, which influences the shaping of the tree structure. Prediction performance comparisons for four popular decision tree algorithms were discussed in [29], [30], to the best of our knowledge, no stability comparison studies have been conducted for these four algorithms. Although numerous studies [31], [32], [33] have proposed innovative splitting criteria to improve decision-tree performance, there has been limited discussion on the stability of these criteria. A stability evaluation provides splitting criteria that are relatively more effective for building interpretable models.

This paper makes the following main contributions:

- Proposes a novel stability measure to quantify the syntactic stability of decision tree algorithms without relying on pairwise tree similarity.
- Proposes a novel adaptive node-level stabilization process that generates a stable decision tree. The process improves upon the node stabilization process of (Dannegger, 2000) by incorporating rank verification to ensure statistical reliability and enhance computational efficiency while handling categorical predictors.
- Provides a comparison of the stability of popular decision tree splitting criteria. The proposed stability metric allows us to determine which splitting criterion is more reliable.

The remainder of this paper is organized as follows: In Section II, the related works of stability measures are introduced. In Section III, the proposed NLS process and the stability measures are described in detail. In Section IV, we assess the stability of the decision-tree splitting criteria using the UCI datasets. In Section V, we present a case study of the breast cancer Wisconsin dataset using the proposed adaptive NLS process. Finally, Section VI concludes the paper and outlines directions for future research.

II. LITERATURE REVIEW

A. DECISION TREES

A binary decision tree has a hierarchical structure consisting of a parent node and two child nodes. Each parent node features a split condition that specifies a predictor and the associated cutoff value. The split condition is determined to reduce the weighted sum of impurities in the two resulting

child nodes. The reduction in impurities achieved by splitting from the parent node to its child nodes is termed as *information gain,* and various splitting criteria are used to quantify the information gain. The final node or leaf node terminates the division and represents the predicted class of the response variable.

Popular splitting criteria of decision trees, such as the Gini index, entropy, and gain ratio, measure the information gain as the weighted sum of the impurities of the child nodes. The Gini index and entropy are biased toward selecting predictors with high cardinality during the branching process [34]. This bias can increase the generalization error [35]. The gain ratio mitigates bias by using a normalization term to measure the information gain [36]. Another popular splitting criterion is the chi-square test, which measures the statistical significance of the information gain. The chi-square test was proposed by CHAID [28]. Chi-square-based criteria are less biased toward selecting predictors with several unique values when determining split conditions at a branch [35].

B. SEMANTIC STABILITY

Semantic stability focuses on how similarly two trees make predictions. The study of [16] compared the extent to which the predictions of two trees agree on a validation dataset. The study of [37] measured the stability of predictions in an ensemble of trees using generalization error. The study of [38] assessed the degree to which individual validation data points are classified into the same class. These methods commonly define and measure the similarity between the predictions of two trees to estimate the semantic stability of decision trees. However, semantic stability measures have a limitation in that they do not account for the interpretability aspect of decision trees. By focusing solely on prediction consistency, they treat trees with different structures but identical predictions as equivalent.

However, semantic stability measures have a limitation in that they do not account for the interpretability aspect of decision trees. By focusing solely on prediction consistency, they treat trees with different structures but identical predictions as equivalent. This fails to capture the transparent decision-making structure of trees, which is composed of rules based on variables and cut-off values. The interpretability of decision trees relies on the explicit representation of the decision-making process, and semantic stability measures overlook this crucial aspect by only considering the final predictions.

C. SYNTACTIC (STRUCTURAL) STABILITY

In contrast, syntactic (structural) stability studies aim to measure the structural similarity between pairs of trees usually considering partial information. The study of [21] compares tree size and depth of trees. The study in [23] forcibly assumed that the structures of the two trees were the same and then compared the nodes in the same position. In our opinion, these studies have limitations in that they either rely only on partial structures from the trees rather than on the overall structure of the tree, or measure the similarity of tree structures under manufactured assumptions.

The study of [15] defined region similarity to measure syntactic stability. A region in a tree is a set of data instances in a leaf node of a decision tree. They represent a decision tree as a family of sets where an element of the family corresponds to a region. Region similarity is measured based on the extent to which regions, as elements of the families, match between two trees. The study of [19] further enhanced region similarity by considering the distribution of data instances within regions, capturing structural aspects more effectively. They performed pairwise comparisons not only between the regions, which are elements of the family, but also between the data instances belonging to those regions.

However, the region-based stability measurement may not fully address the interpretability aspect. Stability metrics based on pairwise region similarity have a limitation: they consider two trees with different structures but identical decision boundaries as equivalent. This makes the measure of dataset-dependent. Two trees with the same family of regions for one dataset may lead to very different for another dataset. On the other hand, two trees with perfectly identical structures always produce the same family of regions, even on the different datasets. The structure of the tree plays a crucial role in making decision trees interpretable for humans, and it is essential to take this into account when assessing stability from the perspective of interpretability.

The study of [39] proposes a new distance metric to quantify the structural differences and prediction similarities between decision trees. They measure the path distance between two trees by considering differences in predictors, cut-off values, and predictions corresponding to every path of two trees, where a path is defined as a set of split conditions from the root node to a leaf node. However, the metric does not consider the sequence of splits, thereby potentially overlooking the overall structural similarity of the trees. Furthermore, the approach suffers from high computational complexity due to the need for pairwise comparison of all paths between two trees, which can become computationally expensive as the number of leaf nodes increases.

Table 1 summarizes the stability types and descriptions of the related works. The common feature of the related works that measure semantic and syntactic stability is that they define measures to assess the degree of difference between two trees. These approaches compare trees in a pairwise fashion: semantic stability is assessed through prediction similarity, and syntactic stability through structural similarity, region, or path similarity. However, defining pairwise tree similarity itself is elusive, making it difficult to develop a comprehensive measurement method. There is currently no general definition for pairwise tree similarity measures [40]. Moreover, approaches comparing trees in a pairwise fashion means that, for n trees, the number of combinations to be calculated is $n(n-1)$. This implies the computational complexity that grows quadratically with the number of trees, making it inefficient when many trees are under the consideration.

TABLE 1. Stability metric comparison.

	Type	Description	Pairwise tree comparison
(Turney, 1995) [16]	Semantic	Measures the agreement between the predictions of two trees on a validation dataset.	Yes
(Breiman, 2001) [37]	Semantic	Measures the stability of predictions in an ensemble of trees using generalization error.	Yes
(Paul et al., 2012) [38]	Semantic	Measures the degree to which individual validation data points are consistently classified into the same class by different trees.	Yes
(Zimmermann, 2008) [21]	Syntactic	Compares the size and depth of two trees to assess their structural similarity.	Yes
(Briand et al., 2009) [23]	Syntactic	Assumes identical tree structures and compares the stability of variable selection and cut-points at corresponding nodes.	Yes
(Dwyer & Holte, 2007) [15]	Syntactic	Measures structural similarity based on the agreement between the regions (sets of instances corresponding to leaf nodes) of two trees.	Yes
(Wang et al., 2018) [19]	Syntactic	Compares the structural similarity of two trees by performing pairwise comparisons of data instances belonging to their regions.	Yes
(Bertsimas et al., 2023) [39]	Semantic and syntactic	Compares the structural similarity of two trees by measuring the distance between their paths, considering differences in split conditions and predictions.	Yes
This paper	Syntactic	Proposes a novel stability measure based on the number of bootstrap iterations required to generate a stable tree using adaptive node-level stabilization.	No

Therefore, we propose a novel syntactic stability measure that does not require defining pairwise tree similarity. Our approach measures stability based on the number of bootstrap iterations needed to construct a stable tree. The bootstrap iterations are adaptively determined during the process of generating a stable tree by introducing rank verification in the NLS process [22]. The more stable the tree generation process, the fewer bootstrap iterations are required.

D. NODE-LEVEL STABILIZATION (NLS)

The study in [22] proposed an NLS process for decision trees. Through experiments on simulated datasets featuring important predesignated predictors, [22] demonstrated that the instability of decision trees can result in the selection of unimportant predictors for splitting. To mitigate the problem of instability, the proposed NLS process in the study of [22] generates a stabilized decision tree model by collecting the predictors and their cut-off values from each bootstrap. In our opinion, this NLS process is an innovative attempt to enhance the stability of decision tree models. The NLS process is illustrated in Figure 1.

The NLS process follows the four steps below.

- Step 0: Consider node t containing the dataset D_t. Let $\mathbf{X} = (X_1, X_2, \ldots X_k)$ be the set of predictors.
- Step 1: Perform bootstrapping B times on D_t to create a set of bootstrapped samples $\{D_{t,b}, 1 \leq b \leq B\}$.
- Step 2: For each bootstrapped sample $D_{t,b}$, perform a branching process on the node. An optimal predictor $X_{t,b}$ and its splitting cut-off value $C(X_{t,b})$ are identified. The optimal predictor is identified by evaluating each predictor's ability to effectively split the data into homogeneous subsets, using splitting criteria.
- Step 3: After identifying the optimal predictors $\{X_{t,b}, 1 \leq b \leq B\}$ for each bootstrapping sample, we select the most frequent predictor X^{freq} among them. For all bootstrapped branches where X^{freq} is the optimal predictor, we identify the median of their cutoff values, denoted as $\overline{C}\left(X^{freq}\right)$. The median is more robust to outliers for numerical cut-off values.
- Step 4: Repeat Steps 0 – 3 for another node until all nodes are stabilized.

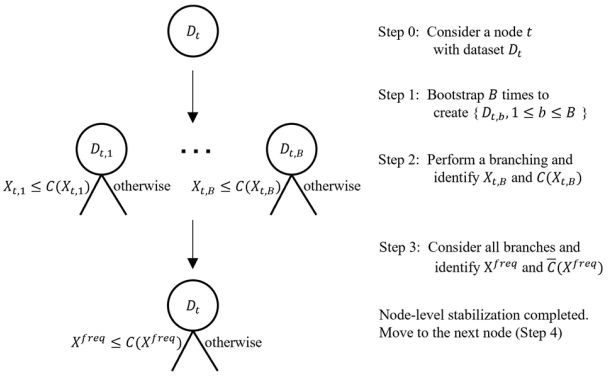

FIGURE 1. Node-level stabilization (NLS) process [22]. This process involves performing multiple bootstrap iterations to select the most frequent splitting among the splits of decision trees obtained from the bootstrapped samples. Following the described steps, this process identifies the most frequent predictor (X^{freq}) and the desirable cut-off value for the predictor ($\overline{C}\left(X^{freq}\right)$) at a node t.

We identify two issues with this process. First, guidelines on the required number of bootstrapping iterations or determining B remain limited. The study in [22] unilaterally set the number of iterations $B = 100$ for each node. The choice of the number of iterations must be less arbitrary because too few iterations imply less statistical significance of the selected predictor, and too many iterations result in computational inefficiency. Our study adaptively determined the number of iterations based on continuous monitoring of statistical significance. Because the number of iterations is determined at a pre-specified level of statistical significance, the number of iterations itself may serve as a measure of stability. Second, the study of [22] does not specify a method for handling

categorical predictors. The proposed ANLS (Adaptive NLS) process incorporates a specific methodology to accommodate categorical predictors.

E. RANK VERIFICATION

The predictor selection process of the NLS process is analogous to the rank-verification problem in classical statistics. Suppose there are k players and n games are repeated. Among the players, if the winning count n_1 of one player significantly exceeds the winning count n_2 of the second-place player, the player with the win count n_1 can be declared as the *true winner*.

Similarly, consider the process of selecting a predictor at a node via bootstrapping. After n trials, suppose that predictor X_1 is selected n_1 times and predictor X_2 is selected n_2 times. For predictor X_1 to be the *true winner*, n_1 must be significantly larger than n_2, and the level of significance is determined by n, n_1, and n_2. Our study adopts the method in [24], which provides statistical significance for the winner using n, n_1, and n_2. Consequently, we obtain a stable decision tree, where n serves as a measure of syntactic stability.

In many real-world applications, researchers are interested in determining the rankings of multiple candidates. Consider a scenario with k candidates to be ranked based on n independent experiments. This scenario can be modeled using a multinomial distribution whose parameters are n and $p_1, p_2, \ldots, p_k$. Suppose $x_1, x_2, \ldots, x_k$ denotes the number of times that each candidate is chosen across n multinomial trials. The classic statistical problem of estimating $p_1, p_2, \ldots, p_k$ based on experimental outcomes is known as the *rank verification* problem.

Whereas typical rank verification problems seek to sort all k candidates in a specific order, our focus is solely on identifying the top predictor (winner) among all the candidate predictors. Assume that the number of wins for each candidate or the cell frequency of the sample is given by $\mathbf{x} = (x_1, x_2, \ldots, x_k)$. The cell frequency $\mathbf{x}$ is drawn from a multinomial distribution, $multinomial\,(n; p_1, p_2, \ldots, p_k)$, where $\sum_{i=1}^{k} x_i = n$, and $\sum_{i=1}^{k} p_i = 1$. Consider the order statistics of the cell frequency in descending order, i.e., $x_{(1)} \geq x_{(2)} \geq \ldots \geq x_{(k)}$. In other words, $x_{(1)} = \max_{1\leq i\leq k} x_{(i)}$,$x_{(2)}$ is the second largest among $x_{(i)}$, and so on. Naturally, let $X_{(i)}$ be the predictor corresponding to the i-th largest frequency $x_{(i)}$ and the probability of $X_{(i)}$ being selected is denoted as $p_{(i)}$. As our focus is on identifying the statistical significance of the top predictor, the related hypothesis is written as $p_{(1)} > p_{(2)}$.

The studies in [41] and [42] demonstrate the statistical significance of the winner in sparse scenarios, where the number of candidates is sufficiently large to allow the assumption of independence. However, in our study, the number of candidates is not sufficiently large to guarantee empirical independence. More relevant methods aligned with our purpose are available in [24]. In [24], the estimation process considered only the top two candidates (winner $X_{(1)}$ and runner-up $X_{(2)}$). The estimation process begins with the winner $X_{(1)}$ and runner-up $X_{(2)}$ from the order statistics. Specifically, if $p_{(1)} > 0.5$ is accepted under the distribution assumption of $bin(x_{(1)} + x_{(2)}; 0.5)$, then $X_{(1)}$ has sufficient statistical significance to be determined as the *true winner*. Compared with the multinomial setting [41], [42], this reduced perspective of the binomial approach allows for faster computation and stronger verification power [24].

We propose a novel adaptive node-level stabilization method that improves the statistical significance of the predictor selection. The node-level stabilization process proposed in [22] uses a preset number of bootstrapping iterations at each node to determine the predictor. By contrast, our study proposes running bootstrapping iterations until a statistically true winner is determined. In addition, whereas the study in [22] dealt with only numerical predictors, we improved the procedure to accommodate categorical predictors.

III. PROPOSED METHODS

This section proposes a novel adaptive NLS (ANLS) process and measures the syntactic stability of a decision trees.

A. PROPOSED ANLS PROCESS

Our proposed method involves the NLS process [22] with varying numbers of bootstraps and rank verifications for the top winner [24]. The central part of the new process is related to the aggregation of the bootstrapped trees. Consider the moment when Steps 0 through 2 of the original NLS process are completed, as described in Section II-D. Based on the rank verification notation, the highest cell frequency $x_{(1)}$ and second-highest cell frequency $x_{(2)}$ were obtained. Under the assumption of the null hypothesis that $x_{(1)} \sim bin\left(x_{(1)} + x_{(2)}; 0.5\right)$, a hypothesis test of the following must be conducted to validate the true winner. $H_0 : p_{(1)} \leq 0.5$ vs $H_1 : p_{(1)} > 0.5$. The Z-statistics for this test is given as $Z = (x_{(1)}/(x_{(1)} + x_{(2)}) - 0.5)/\sqrt{0.5(1-0.5)/(x_{(1)} + x_{(2)})}$. To illustrate this, consider a case with 80 bootstrap iterations performed thus far, where $x_{(1)} = 35$ and $x_{(2)} = 30$ are obtained. The computed Z-statistics is 0.62, which is insufficient to reject the null hypothesis $H_0 : p_{(1)} \leq 0.5$ as it is less than 1.645 at a confidence level of 90%. Given the current data, the statistical evidence is insufficient to declare $X_{(1)}$ the winner. In such cases, we proposed performing additional bootstrapping iterations.

In summary, the true winner is statistically validated if $H_0(p_{(1)} \leq 0.5)$ is rejected in favor of H_1 $(p_{(1)} > 0.5)$. If sufficient statistical significance is lacking, additional bootstrapping iterations is conducted. This adaptive design is computationally efficient because only the necessary number of bootstrapping iterations are performed.

In addition, it naturally identifies the number of bootstrap iterations required to achieve statistical significance. The required number of iterations indicates the tendency toward majority voting, which can be extended to reflect the stability tendency. This extension to stability is justified because a fewer number of required iterations imply greater consistency

in the optimal predictors, and thus, higher stability. The following subsection describes the concept of stability.

Pseudocode outlining the proposed method is presented at the end of this subsection. In the pseudocode, the while loop in lines 4-10 performs adaptive bootstrapping for node t. On line 9, the Z-statistics is calculated to verify whether the optimal predictor $X_{(1)}$ is the true winner. The loop iterates until the Z-statistics is greater than Z_α. When the white loop is concluded, the number of bootstrap iterations required for statistical significance is determined. In line 11, the term $C^m(X_{(1)})$ denotes the most frequent cutoff value for the predictor $X_{(1)}$ across the bootstrap iterations. This section accommodates categorical predictors by assigning the most frequent observed cutoff values, unlike the study in [22].

Pseudocode: Adaptive NLS process

0: D_t: dataset at node t; **X**: predictors; Z_α: threshold for confidence level α
1: **FOR** every node t containing D_t **DO**
2: $b = 0$ // iteration counter
3: $z = -\infty$ // Z-statistics
4: **WHILE** $z \leq Z_\alpha$ **DO**
5: $b = b + 1$
6: Generate a bootstrap sample $D_{t,b}$
7: Find the optimal split for $D_{t,b}$, with the optimal predictor $X_{t,b}$ and its splitting cut-off value $C(X_{t,b})$
8: Collect and sort frequencies of predictors in the optimal splits, denoted as $x_{(1)}, x_{(2)} \ldots x_{(k)}$
9: Calculate Z-statistic value: $z = \frac{\frac{x_{(1)}}{x_{(1)}+x_{(2)}} - 0.5}{\sqrt{\frac{0.5(1-0.5)}{x_{(1)}+x_{(2)}}}}$
10: **END**
11: Determine the stabilized split at node t:
IF $X_{(1)}$ is continuous, **THEN** the split is

$$X_{(1)} \leq \overline{C}\left(X_{(1)}\right)$$

ELSE the split is $X_{(1)} \in C^m(X_{(1)})$
END
12: **END**

B. STABILITY MEASURE

In Section III-A, we propose a novel adaptive NLS process designed to calculate the precise number of bootstrap iterations required to achieve statistical significance. This required number of bootstrap iterations offers an assessment of the syntactic stability at each node. Namely, fewer bootstrap iterations mean that the decision tree's branching process exhibits higher syntactic stability, as the choice of the optimal predictor remains relatively invariant across multiple bootstrap iterations. In contrast, a higher number of bootstrap iterations indicates more frequent variability of optimal predictors within multiple branches, indicating lower syntactic stability. Thus, the adaptive NLS process provides an evaluation of the syntactic stability of individual nodes.

Then, it remains to extend the measure from a single node to encompass the entire decision tree. To address this issue, we propose a stability measure for the entire decision tree as a *weighted summation of the required bootstraps* (WSRB) as follows:

$$WSRB = \sum_{\forall t} w_t B_t,$$

where B_t is the number of *required bootstraps (RB)* for node t and w_t is the weight assigned to the node. The weight assigned to the node can be set as $w_t = 1/\delta_t$, where δ_t denotes the depth level of node t. The rationale behind this weighting scheme is based on the hierarchical properties of a decision tree, where its upper nodes (those at lower depth levels) have more influence than the lower nodes (those at higher depth levels). In other words, the stability of the upper nodes is more important than that of the lower nodes. The proposed WSRB serves as a direct measure of the syntactic stability of the decision trees. A lower WSRB value implies greater syntactic stability in the decision tree. This novel metric eliminates the need for ambiguous tasks such as measuring the similarity between multiple trees to assess syntactic stability.

C. COMPUTATION COMPLEXITY

The ANLS process involves repeating the branching process of a decision tree, but, unless in the worst-case scenario, it is comparable to a single decision tree in terms of computational complexity. With a sample size of n and m predictors, the computational complexity of a decision tree is $O(mnlog(n))$. The computational complexity of the ANLS process varies depending on the number of bootstrap samples, B_t, required to determine a stable split in a node. The value of B_t depends on the characteristics and distribution of the data, which are not necessarily correlated directly with m and n. Thus, unless B_t approaches or exceeds n in the worst-case scenario, the computational complexity of the ANLS process is similar to that of a decision tree.

IV. EXPERIMENTAL RESULTS

This section presents an empirical study of the proposed WSRB measure for assessing the stability of decision trees. We offer a comparative stability analysis focusing on the widely used decision tree splitting criteria: the Gini index, entropy, gain ratio, and chi-square. In addition to the WSRB measure for stability comparison, we also compare the computational time required to produce a stable tree using the ANLS process. In addition, we explored how specific structural characteristics such as node depth and sample size at each node influence the stability of the decision tree.

A. DATASETS

The experimental dataset was sourced from the UCI machine learning repository [43], which is a renowned resource commonly utilized in machine learning studies. The datasets included six binary classification problems and four multiple-classification problems. Table 2 outlines the characteristics of the ten datasets used, providing details on the sample size, numerical and categorical predictors, and the number of classes in the response variables.

TABLE 2. Datasets from the UCI machine learning repository.

Dataset	Sample size	Predictors type		# of classes
		Categorical	Numerical	
BC-Wisconsin	699	-	9	2
Car	1,728	6	-	4
Ionosphere	351	-	23	2
Letter	20,000	-	16	2
Pima-Indians-diabetes	768	-	8	2
Shuttle	56,794	-	9	7
Sonar	208	-	59	2
Spambase	4,601	-	57	2
Splice-jxn	3,190	60	-	3
Yeast	1,484	1	8	9

B. EXPERIMENTAL SETTINGS

We employed the following parameter settings throughout the proposed ANLS process:

- Statistical confidence level α: 90%
- Minimum number of bootstrap iterations: 10
- Minimum sample size of a leaf node: 1%

The statistical confidence level α is set at 90%, corresponding to a threshold value Z_α of 1.645. This threshold serves as the criterion for statistically determining the most frequently selected optimal predictor. We set a minimum of 10 bootstrap iterations to commence the ANLS process, which helps mitigate potential statistical errors that may arise from a smaller number of iterations. During the generation of the decision tree in the ANLS process, the pre-pruning condition is set such that the minimum sample size in a leaf node is less than 1% of the training dataset. The ANLS process is implemented using binary splits in the branching process, the same as the CART algorithm. Our experiment does not have a specific maximum number of iterations defined. The winner will eventually be determined as the number of iterations increases and more evidence accumulates. However, users can set a maximum number of iterations based on computational resource limitations or other practical constraints. The experiments were conducted in the following environment:

- CPU: Intel Core i7-11700K 3.60GHz
- RAM: 32GB DDR4
- Software version: Python 3.9.13

C. COMPARISON OF STABILITY FOR POPULAR SPLITTING CRITERIA

This subsection compares the stability of the four splitting criteria of popular decision tree algorithms: Gini index (CART), entropy (ID3), gain ratio (C4.5), and chi-square (CHAID). For our experiments, we generated trees for ten datasets by performing a branching process with four splitting criteria in the ANLS process. A total of 40 trees (4 criteria $\times$ 10 datasets) were generated, from which the WSRB values were obtained.

Table 3 presents values of WSRB for each splitting criterion across 10 datasets, alongside the comparative ranking of WSRB for each splitting criterion within the same dataset. Among the four splitting criteria, gain ratio exhibits the lowest WSRB with the average of 1,156. Out of the 10 datasets, gain ratio has the lowest WSRB in seven datasets, indicating that gain ratio is the most stable algorithm among those considered. The stability order of gain ratio – entropy – chi-square – Gini index is established based on the average WSRB.

TABLE 3. WSRB and its ranking in a same dataset for the different splitting criteria.

Dataset	Gain ratio (rank)	Entropy (rank)	Chi-square (rank)	Gini index (rank)
BC-Wisconsin	430 (4)	414 (3)	**266 (1)**	278 (2)
Car	**85 (1)**	164 (3)	145 (2)	216 (4)
Ionosphere	**386 (1)**	673 (2)	3,454 (3)	15,938 (4)
Letter	**36 (1)**	636 (2)	1,100 (4)	961 (3)
Pima-Indians-diabetes	**1,021 (1)**	1,722 (2)	3,624 (4)	2,560 (3)
Shuttle	**42 (1)**	185 (3)	299 (4)	55 (2)
Sonar	**248 (1)**	7,687 (3)	2,504 (2)	13,620 (4)
Spambase	1,418 (3)	**1,240 (1)**	3,034 (4)	1,400 (2)
Splice-jxn	7,642 (4)	**4,286 (1)**	6,239 (3)	5,096 (2)
Yeast	**247 (1)**	3,126 (4)	659 (2)	1,452 (3)
Average	**1,156 (1)**	2,013 (2)	2,132 (3)	4,158 (4)
Avg. Rank	**1.8**	2.4	2.9	2.9
# of wins	**7**	2	1	0

Gain ratio consistently shows the lowest WSRB, indicating high stability across various datasets. Entropy and chi-square exhibit similar average WSRBs (2,013 and 2,132, respectively). The Gini index shows the highest WSRB, indicating lower stability among the splitting criteria.

Figure 2 presents a visual depiction of the results using a box plot, reaffirming the stability ranking of the considered algorithms. The gain ratio had the lowest average WSRB and its variation was also the smallest. Although entropy and chi-square exhibited similar average WSRBs, entropy had a significantly lower variation than chi-square. Overall, the

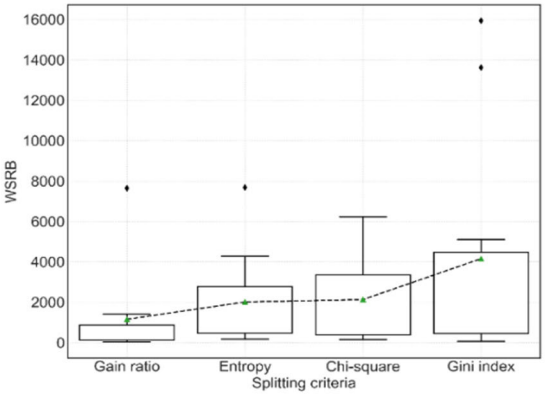

FIGURE 2. Comparison of WSRB for each splitting criterion. The green triangular marker represents the average of WSRB for 10 datasets.

experimental results indicate that the gain ratio is the most stable criterion among the four popular options evaluated.

D. COMPARISON OF COMPUTATIONAL TIME

In this subsection, we compare the computational times required to obtain a stable decision tree using the ANLS process for the four splitting criteria.

TABLE 4. Computational time (hours) and its ranking in a same dataset for the different splitting criteria.

Dataset	Gain ratio (rank)	Entropy (rank)	Chi-square (rank)	Gini index (rank)
BC-Wisconsin	**0.28 (1)**	1.01 (3)	0.55 (2)	1.05 (4)
Car	1.14 (2)	**1.00 (1)**	1.21 (4)	1.18 (3)
Ionosphere	**2.00 (1)**	4.06 (3)	6.51 (4)	5.11 (2)
Letter	**3.11 (1)**	38.45 (4)	33.45 (2)	37.38 (3)
Pima-Indians-diabetes	**10.07 (1)**	13.29 (4)	12.51 (3)	11.48 (2)
Shuttle	10.28 (3)	16.14 (4)	**6.38 (1)**	7.42 (2)
Sonar	**5.36 (1)**	12.35 (3)	12.29 (2)	15.12 (4)
Spambase	114.59 (4)	95.11 (3)	88.29 (2)	**78.13 (1)**
Splice-jxn	68.04 (3)	60.43 (2)	71.30 (4)	**52.51 (1)**
Yeast	**2.13 (1)**	81.45 (4)	7.18 (2)	11.15 (3)
Average	**21.70 (1)**	32.32 (4)	23.97 (3)	22.05 (2)
Avg. Rank	**1.8**	3.1	2.6	2.5
# of wins	**6**	1	1	2

Table 4 presents the computational time for the splitting criteria across the ten datasets. The gain ratio exhibited the shortest average computational time compared to the other three criteria, indicating that the gain ratio can efficiently produce a stable tree using the ANLS process. Notably, the gain ratio exhibited the highest computational efficiency in this computational time comparison and was also the most stable in the previous WSRB evaluation.

The order from shortest to longest average computational time was the gain ratio – Gini index – chi-square – entropy. Among the four splitting criteria, entropy had the longest average computational time. This result does not match the fact that the stability performance of entropy is the second-best, owing to the need for more bootstrapping iterations, which are often necessary for the upper nodes. Bootstrapping in the upper nodes, which address larger subsets of the training dataset, inherently requires more computational time. Consequently, the computation time and stability results may not always coincide.

E. STRUCTURAL CHARACTERISTICS ASSOCIATED WITH STABILITY

In subsection IV-C, we examine how the stability of the decision tree model is considerably affected by the splitting criteria. Apart from the splitting criteria, this subsection explores how the structural characteristics in the branching process, depth level of the node, and sample size in the node affect stability. We explore the influence of these two structural characteristics using required bootstraps (RB, as defined in Section III-B), which measures the stability of the branching process at a node. We observed the RB for 1,220 nodes across the 40 trees generated in the experiments.

The RB distribution for each node is shown in Figure 3. In Figure 3, the RB is heavily right-skewed, with a median much lower than the average value. The average value of RB was 364, suggesting that an average of 364 bootstrap iterations was required to determine the most frequent predictor for a single node. The median RB was 23, indicating that 23 or fewer bootstrap iterations were required for half of the nodes. Most nodes (75%) have an RB of 76 or less, whereas 25% have an RB greater than 76. The maximum RB was 47,587, which is extremely high. In this extreme case, it is difficult to determine the most frequent predictor because multiple predictors fiercely compete to become optimal predictors during the bootstrap iteration.

The results concerning the two structural characteristics and their relationship with RB are presented in Figures 4 and 5. A box plot of the RB for each node depth level is shown in Figure 4. At depths ranging from 2 to 7, the median RB tends to increase, suggesting that nodes at higher depth levels tend to be less stable because they have undergone more branching processes, and thus have a smaller sample size. Figure 5 shows the relationship between the sample size and stability of a node. The larger the sample size of a node, the more stable the selection of an optimal predictor is. In the four sample size bins – [1, 100), [100, 1000), [1000, 3000), and [3000, ∞) – both the median and variation of RB decrease as the sample size increases, implying that stability tends to improve with larger sample sizes.

The RB results related to the node depth level and sample size suggest that stability decreases during the branching process of a decision tree when the sample size for a node decreases, consistent with the general understanding that a larger sample size improves the stability of machine learning algorithms [16]. Analysis of these two structural characteristics provides empirical evidence supporting the validity of RB as a measure of stability.

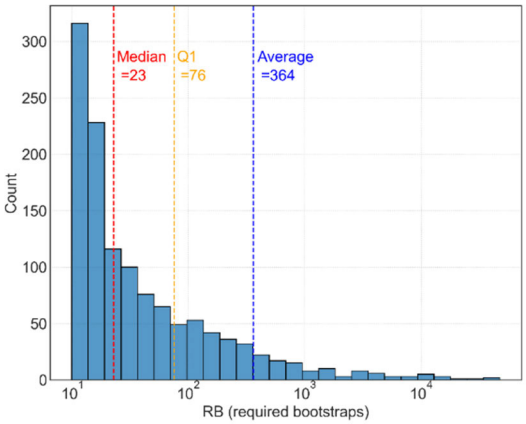

FIGURE 3. Distribution of RB for the whole 1,220 nodes.

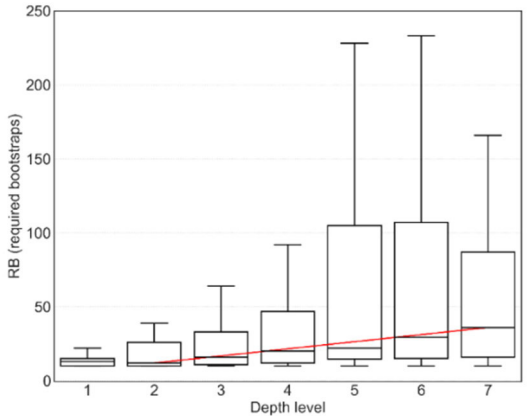

FIGURE 4. Distribution of RB for the nodes at each depth level.

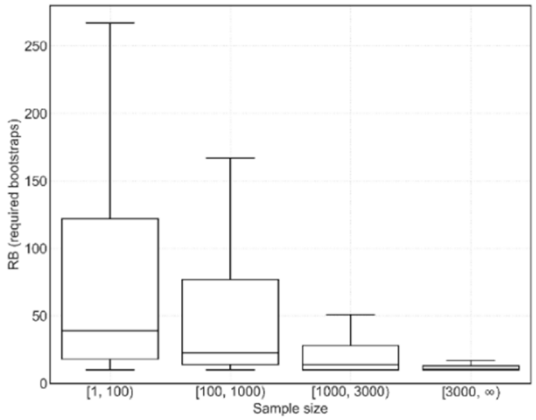

FIGURE 5. Distribution of RB across different nodes' sample size bins.

Additionally, the observation that RB decreases with increasing sample size implies that the ANLS process achieves scalability comparable to decision trees when applied to large datasets.

F. SENSITIVITY ANALYSIS

We examine the key parameters of WSRB: confidence level α and the minimum number of bootstrap iterations. The sensitivity of WSRB to these parameters is tested on two datasets, breast cancer-Wisconsin and car. The splitting criteria used is entropy, and the maximum tree depth is set to 5.

Figure 6 presents WSRB in relation to the confidence level on a log scale. For both datasets, as the confidence level increases beyond 90%, WSRB shows a sharp increase. This indicates that WSRB is highly sensitive when the confidence level exceeds 90%.

Figure 7 presents the sensitivity to the minimum number of bootstrap iterations. For both datasets, there is a clear trend of increasing WSRB as the minimum bootstrap iterations increase. Due to the randomness inherent in the bootstrap process and the instability of decision tress, there is an outlier for the breast cancer dataset at 40 minimum bootstrap iterations, where WSRB spikes.

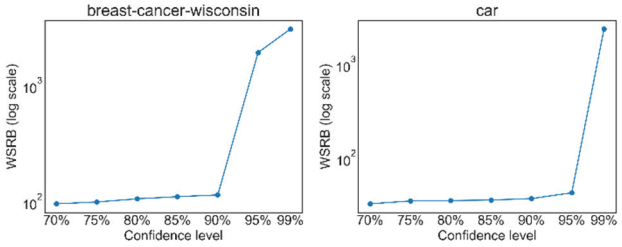

FIGURE 6. WSRB sensitivity to confidence level for two datasets. WSRB is displayed on a log scale.

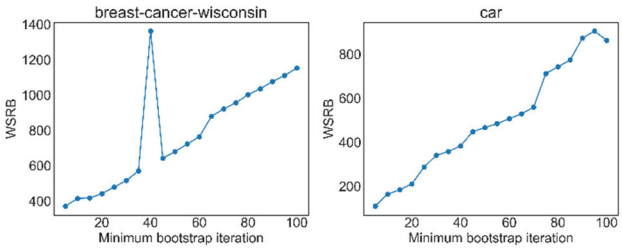

FIGURE 7. WSRB sensitivity to the minimum number of bootstrap iterations for two datasets.

TABLE 5. Rule sets generated by CART and ANLS process, considering the update scenario.

	70% dataset	100% dataset
CART	CART-70	CART-100
ANLS	ANLS-70	ANLS-100

The confidence level and the minimum number of bootstrap iterations both significantly impact WSRB. As the confidence level increases, WSRB also increases. Similarly, WSRB rises with higher minimum bootstrap iterations. Regarding statistical significance and computational complexity, these parameters can be adjusted. Generally, the confidence level is set above 90%, but it can be lowered to reduce the computational burden.

V. CASE STUDY

In Section IV, we measure and compare the stability of popular decision tree splitting criteria with our proposed stability measure. This section presents a case study that demonstrates the practical usefulness of the proposed ANLS process.

From an interpretability perspective, the practical usefulness of a model encompasses various aspects including the algorithm's reliability, transparency, causality, transferability, and fairness [2], [44]. Among these, reliability is directly associated with the stability of the decision tree [2]. A decision tree is reliable when the suggested rules do not change significantly when the input data are updated.

We analyzed how the rules indicated by the decision tree change in response to data updates. We compared the rules generated by CART with those generated by the ANLS process when the data were updated. For this analysis, we examined an *update scenario* with two distinct data stages: *current data*, represented by a random 70% sample

TABLE 6. The rules generated by CART and ANLS process for the update scenario. CART-70 and CART-100 differ significantly in their root node conditions, shifting from UCSz to UCSh, while ANLS-70 and ANLS-100 consistently select UCSz.

Generated rule
CART-70 (*The rules generated by CART for the random 70% sample of dataset*)
`IF UCSz <= 2.5 AND BN <= 3.5 AND SECS <= 2.5 THEN y = 0`
`IF UCSz <= 2.5 AND BN <= 3.5 AND SECS > 2.5 THEN y = 0`
`IF UCSz <= 2.5 AND BN > 3.5 THEN y = 0`
`IF UCSz > 2.5 AND UCSz <= 4.5 AND BN <= 2.5 THEN y = 0`
`IF UCSz > 2.5 AND UCSz <= 4.5 AND BN > 2.5 AND CT <= 6.5 THEN y = 1`
`IF UCSz > 2.5 AND UCSz <= 4.5 AND BN > 2.5 AND CT > 6.5 THEN y = 1`
`IF UCSz > 2.5 AND UCSz > 4.5 AND BC <= 4.5 AND M <= 1.5 THEN y = 1`
`IF UCSz > 2.5 AND UCSz > 4.5 AND BC <= 4.5 AND M > 1.5 THEN y = 1`
`IF UCSz > 2.5 AND UCSz > 4.5 AND BC > 4.5 THEN y = 1`
CART-100 (*The rules generated by CART for the 100% sample of dataset*)
`IF UCSh <= 2.5 AND BN <= 1.5 THEN y = 0`
`IF UCSh <= 2.5 AND BN > 1.5 AND BC <= 2.5 THEN y = 0`
`IF UCSh <= 2.5 AND BN > 1.5 AND BC > 2.5 THEN y = 0`
`IF UCSh > 2.5 AND BN <= 2.5 AND UCSz <= 3.5 THEN y = 0`
`IF UCSh > 2.5 AND BN <= 2.5 AND UCSz > 3.5 THEN y = 1`
`IF UCSh > 2.5 AND BN > 2.5 AND MA <= 5.5 AND CT <= 6.5 AND BC <= 4.5 THEN y = 1`
`IF UCSh > 2.5 AND BN > 2.5 AND MA <= 5.5 AND CT <= 6.5 AND BC > 4.5 THEN y = 1`
`IF UCSh > 2.5 AND BN > 2.5 AND MA <= 5.5 AND CT > 6.5 AND NN <= 6.5 THEN y = 1`
`IF UCSh > 2.5 AND BN > 2.5 AND MA <= 5.5 AND CT > 6.5 AND NN > 6.5 THEN y = 1`
`IF UCSh > 2.5 AND BN > 2.5 AND MA > 5.5 THEN y = 1`
ANLS-70 (*The rules generated by ANLS for the random 70% sample of dataset*)
`IF UCSz > 2.2 AND UCSz > 3.8 AND UCSz > 4.0 AND MA > 2.0 THEN y = 1`
`IF UCSz > 2.2 AND UCSz > 3.8 AND UCSz > 4.0 AND MA <= 2.0 THEN y = 1`
`IF UCSz > 2.2 AND UCSz > 3.8 AND UCSz <= 4.0 THEN y = 1`
`IF UCSz > 2.2 AND UCSz <= 3.8 AND BN > 2.1 THEN y = 1`
`IF UCSz > 2.2 AND UCSz <= 3.8 AND BN <= 2.1 THEN y = 0`
`IF UCSz <= 2.2 AND BN > 2.4 THEN y = 0`
`IF UCSz <= 2.2 AND BN <= 2.4 AND SECS > 2.0 THEN y = 0`
`IF UCSz <= 2.2 AND BN <= 2.4 AND SECS <= 2.0 THEN y = 0`
ANLS-100 (*The rules generated by ANLS for the 100% sample of dataset*)
`IF UCSz > 2.3 AND UCSz > 3.8 AND UCSz > 4.0 AND BC > 3.8 AND NN > 2.0 THEN y = 1`
`IF UCSz > 2.3 AND UCSz > 3.8 AND UCSz > 4.0 AND BC > 3.8 AND NN <= 2.0 THEN y = 1`
`IF UCSz > 2.3 AND UCSz > 3.8 AND UCSz > 4.0 AND BC <= 3.8 THEN y = 1`
`IF UCSz > 2.3 AND UCSz > 3.8 AND UCSz <= 4.0 THEN y = 1`
`IF UCSz > 2.3 AND UCSz <= 3.8 AND BN > 2.2 THEN y = 1`
`IF UCSz > 2.3 AND UCSz <= 3.8 AND BN <= 2.2 THEN y = 0`
`IF UCSz <= 2.3 AND BN > 2.6 THEN y = 0`
`IF UCSz <= 2.3 AND BN <= 2.6 AND SECS > 2.0 THEN y = 0`
`IF UCSz <= 2.3 AND BN <= 2.6 AND SECS <= 2.0 THEN y = 0`

of the dataset, and *updated data*, encompassing 100% of the dataset. Each data stage was trained using two decision-tree algorithms: CART and ANLS. Therefore, we had four rule sets: CART-70, CART-100, ANLS-70, and ANLS-100, as presented in Table 5.

The real-world data used for the analysis was the breast cancer-Wisconsin (BC-Wisconsin) dataset from the UCI repository [43]. This dataset includes nine predictors along with class labels for cancer and normal tissues. These nine predictors were bland chromatin (`BC`), bare nuclei (`BN`), clump thickness (`CT`), marginal adhesion (`MA`), mitoses (`M`), normal nucleoli (`NN`), single epithelial cell size (`SECS`), uniformity of cell shape (`UCSh`), and uniformity of cell size (`UCSz`). This dataset is widely used in comparative studies of classification algorithms [45], as well as in research focusing on the medical domain [46], [47].

Table 6 presents the four rule sets listed in Table 5. To simplify the analysis, a pre-pruning technique was employed, wherein splitting was stopped when the number of samples in a node fell below 3% of the total number of samples.

Between CART-70 and CART-100, a notable difference in root node conditions was observed: `UCSz` was selected in CART-70, whereas `UCSh` was selected in CART-100, indicating a fundamental divergence in their initial split condition. However, ANLS-70 and ANLS-100 consistently selected `UCSz` as the root node condition. The first five rules from ANLS-70 and the first six rules from ANLS-100 had matching predictors up to the third condition. The remaining rules, three from ANLS-70 and four from ANLS-100, consisted entirely of identical predictors. The root node's cutoff value also showed only a 5% difference: 2.2 in ANLS-70 and 2.3 in ANLS-100.

For a more detailed quantitative comparison, we employed a set-based representation to analyze the rule sets (CART-70 vs. CART-100) and (ANLS-70 vs. ANLS-100). The set-based representation of a rule set consists of pairs of

TABLE 7. Condition order and predictor of rule sets generated by CART and ANLS process.

	Order of condition	Predictor 70% dataset	Predictor 100% dataset
CART	1	UCSz	UCSh
	2	BN, UCSz	BN
	3	BC, BN, SECS	BC, MA, UCSz
	4	CT, M	CT
	5	-	BC, NN
ANLS	1	UCSz	UCSz
	2	BN, UCSz	BN, UCSz
	3	BN, SECS, UCSz	BN, SECS, UCSz
	4	MA	BC
	5	-	NN

condition orders and their corresponding predictors. The order of conditions and corresponding predictors within each rule for the four rule sets are presented in Table 7. This set-based representation summarizes the list-based representations in Table 6.

To evaluate how consistently CART and ANLS maintain their rules when the dataset is updated, we calculated the match rate as the intersection over the union of elements in the set-based representations of the two rule sets. For CART-70 and CART-100, only three elements were shared: {(2, BN), (3, BC), and (4, CT)}, accounting for 23% (=3/13) of the total. By contrast, the sets ANLS-70 and ANLS-100 share six elements: {(1, UCSz), (2, BN), (2, UCSz), (3, BN), (3, SECS), and (3, UCSz)}, accounting for 67% (=6/9) of the total, indicating that the match rate for ANLS is nearly three times higher at 67% compared to 23% for CART. These differences highlight a significant change in the CART rules when the data were updated from 70% to 100%, whereas the ANLS process exhibited a significantly smaller change in its rules.

The remarkable stability of the ANLS process, even with data updates of 30%, indicates that it reliably extracts knowledge from real-world problems. This stability makes the ANLS process useful for applications in which data updates are common.

VI. CONCLUSION

Decision trees are known for their interpretability, but their instability limits their reliability. Evaluating syntactic stability is crucial for improving decision trees, but the natural ambiguity in comparing pairwise tree similarity has hindered the development of a general stability measure [40].

We propose a novel stability measure, WSRB, derived from our proposed ANLS process, which uses rank verification [24] to determine the most frequent predictor at each node during bootstrapping. The WSRB measure evaluates the syntactic stability of decision trees without relying on pairwise tree similarity.

Our empirical study compares the stability of four widely used splitting criteria and reveals that the gain ratio exhibits the highest stability across ten datasets. The case study using the breast cancer-Wisconsin dataset demonstrates that the tree generated by the ANLS process produces more stable rules compared to the one by the CART, significantly improving reliability.

The proposed stability measure and ANLS process contribute to the development of more reliable decision trees and can be used for stability assessments in various data analysis processes involving decision trees.

This study suggests four primary directions for future research. First, our comparison of the stability of splitting criteria lays the groundwork for theoretical studies focusing on the stability of various decision tree splitting criteria. For example, the gain ratio is the addition of a normalization term to the information gain of entropy, and the relationship between this difference and stability can be studied further. Second, future research is needed to develop frameworks for quantitative comparison of different stability measures, particularly because the absence of ground truth on decision tree stability makes it difficult to compare these measures fairly. Third, the proposed stability measure can be extended to evaluate the stability of rule-induction algorithms [48], [49] that employ a separate-and-conquer strategy akin to the decision-tree induction strategy. Finally, the ANLS process in this study can be included to represents the decision trees discussed in [50]. Methods for extracting representative decision trees or rules from tree ensemble algorithms have been proposed [40], [50], [51]. Considering the ability of the ANLS process to generate a single interpretable tree from diverse branching possibilities, future research could explore the application of this method to discover representative decision trees from tree ensemble algorithms.

REFERENCES

[1] H. Bastani, O. Bastani, and W. P. Sinchaisri, "Improving human sequential decision-making with reinforcement learning," 2021, *arXiv:2108.08454*.

[2] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, Jul. 2019.

[3] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, May 2019.

[4] V. Belle and I. Papantonis, "Principles and practice of explainable machine learning," *Frontiers Big Data*, vol. 4, p. 39, Jul. 2021.

[5] M. Wan, D. Zha, N. Liu, and N. Zou, "In-processing modeling techniques for machine learning fairness: A survey," *ACM Trans. Knowl. Discovery Data*, vol. 17, no. 3, pp. 1–27, Apr. 2023.

[6] G. Audemard, S. Bellart, L. Bounia, F. Koriche, J.-M. Lagniez, and P. Marquis, "On the computational intelligibility of Boolean classifiers," 2021, *arXiv:2104.06172*.

[7] V. Kaushal and M. Patwardhan, "Emerging trends in personality identification using online social networks—A literature survey," *ACM Trans. Knowl. Discovery Data*, vol. 12, no. 2, pp. 1–30, Apr. 2018.

[8] T. Yarkoni and J. Westfall, "Choosing prediction over explanation in psychology: Lessons from machine learning," *Perspect. Psychol. Sci.*, vol. 12, no. 6, pp. 1100–1122, Nov. 2017.

[9] L. Yao, "Financial accounting intelligence management of Internet of Things enterprises based on data mining algorithm," *J. Intell. Fuzzy Syst.*, vol. 37, no. 5, pp. 5915–5923, Nov. 2019.

[10] S. J. Darnell, D. Page, and J. C. Mitchell, "An automated decision-tree approach to predicting protein interaction hot spots," *Proteins, Struct., Function, Bioinf.*, vol. 68, no. 4, pp. 813–823, Sep. 2007.

[11] Y. Akkem, S. K. Biswas, and A. Varanasi, "Smart farming using artificial intelligence: A review," *Eng. Appl. Artif. Intell.*, vol. 120, Apr. 2023, Art. no. 105899.

[12] G. Nie, W. Rowe, L. Zhang, Y. Tian, and Y. Shi, "Credit card churn forecasting by logistic regression and decision tree," *Expert Syst. Appl.*, vol. 38, no. 12, pp. 15273–15285, Nov. 2011.

[13] E. Shaaban, Y. Helmy, A. Khedr, and M. Nasr, "A proposed churn prediction model," *Int. J. Eng. Res. Appl.*, vol. 2, no. 4, pp. 693–697, 2012.
[14] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
[15] K. Dwyer and R. Holte, "Decision tree instability and active learning," in *Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discovery Databases (ECML-PKDD)*, 2007, pp. 128–139.
[16] P. Turney, "Bias and the quantification of stability," *Mach. Learn.*, vol. 20, no. 1, pp. 23–33, 1995.
[17] J. Duarte, J. Gama, and A. Bifet, "Adaptive model rules from high-speed data streams," *ACM Trans. Knowl. Discovery from Data*, vol. 10, no. 3, pp. 1–22, Feb. 2016.
[18] M. Last, O. Maimon, and E. Minkov, "Improving stability of decision trees," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 16, no. 2, pp. 145–159, Mar. 2002.
[19] L. Wang, Q. Li, Y. Yu, and J. Liu, "Region compatibility based stability assessment for decision trees," *Expert Syst. Appl.*, vol. 105, pp. 112–128, Sep. 2018.
[20] P. Domingos, "Knowledge discovery via multiple models," *Intell. Data Anal.*, vol. 2, no. 3, pp. 187–202, Jul. 1998.
[21] A. Zimmermann, "Ensemble-trees: Leveraging ensemble power inside decision trees," in *Proc. Int. Conf. Discovery Sci.* Berlin, Germany: Springer, 2008, pp. 76–87.
[22] F. Dannegger, "Tree stability diagnostics and some remedies for instability," *Statist. Med.*, vol. 19, no. 4, pp. 475–491, Feb. 2000.
[23] B. Briand, G. R. Ducharme, V. Parache, and C. Mercat-Rommens, "A similarity measure to assess the stability of classification trees," *Comput. Statist. Data Anal.*, vol. 53, no. 4, pp. 1208–1217, Feb. 2009.
[24] K. Hung and W. Fithian, "Rank verification for exponential families," *Ann. Statist.*, vol. 47, no. 2, pp. 758–782, Apr. 2019.
[25] L. Breiman, *Classification and Regression Trees*, 1st ed. New York, NY, USA: Routledge, 1984.
[26] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.
[27] J. R. Quinlan, *C4. 5: Programs for Machine Learning*. Amsterdam, The Netherlands: Elsevier, 2014.
[28] G. V. Kass, "An exploratory technique for investigating large quantities of categorical data," *J. Roy. Stat. Soc., Ser. C Appl. Statist.*, vol. 29, no. 2, pp. 119–127, Jun. 1980.
[29] M. Ture, F. Tokatli, and I. Kurt, "Using Kaplan–Meier analysis together with decision tree methods (C&RT, CHAID, QUEST, C4.5 and ID3) in determining recurrence-free survival of breast cancer patients," *Exp. Syst. Appl.*, vol. 36, no. 2, pp. 2017–2026, Mar. 2009.
[30] F. M. J. M. Shamrat, R. Ranjan, K. M. Hasib, A. Yadav, and A. H. Siddique, "Performance evaluation among ID3, C4. 5, and cart decision tree algorithm," in *Proc. Pervasive Comput. Social Netw.* Singapore: Springer, 2022, pp. 127–142.
[31] S. Hwang, H. G. Yeo, and J.-S. Hong, "A new splitting criterion for better interpretable trees," *IEEE Access*, vol. 8, pp. 62762–62774, 2020.
[32] B. Chandra, R. Kothari, and P. Paul, "A new node splitting measure for decision tree construction," *Pattern Recognit.*, vol. 43, no. 8, pp. 2725–2731, Aug. 2010.
[33] M. Jaworski, P. Duda, and L. Rutkowski, "New splitting criteria for decision trees in stationary data streams," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2516–2529, Jun. 2018.
[34] C. Strobl, A.-L. Boulesteix, and T. Augustin, "Unbiased split selection for classification trees based on the Gini index," *Comput. Statist. Data Anal.*, vol. 52, no. 1, pp. 483–501, Sep. 2007.
[35] E. Harris Jr., "Information gain versus gain ratio: A study of split method biases," in *Proc. ISAIM*, Fort Lauderdale, FL, USA, Jan. 2002.
[36] A. P. White and W. Z. Liu, "Bias in information-based measures in decision tree induction," *Mach. Learn.*, vol. 15, no. 3, pp. 321–329, Jun. 1994.
[37] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[38] J. Paul, M. Verleysen, and P. Dupont, "The stability of feature selection and class prediction from ensemble tree classifiers," in *Proc. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn.*, 2012, pp. 263–268.
[39] D. Bertsimas and V. Digalakis Jr., "Improving stability in decision tree models," 2023, *arXiv:2305.17299*.
[40] C. Bénard, G. Biau, S. Da Veiga, and E. Scornet, "SIRUS: Stable and interpretable RUle set for classification," *Electron. J. Statist.*, vol. 15, no. 1, pp. 427–505, Jan. 2021.
[41] S. S. Gupta and S. Panchapakesan, "On multiple decision (subset selection) procedures," *J. Math. Phys. Sci.*, vol. 6, no. 1, pp. 1–71, 1972.
[42] S. Gutmann and Z. Maymin, "Is the selected population the best?" *Ann. Statist.*, vol. 15, no. 1, pp. 456–461, Mar. 1987.
[43] A. Asuncion and D. Newman, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, Tech. Rep., 2007. [Online]. Available: http://archive.ics.uci.edu/ml
[44] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *J. Artif. Intell. Res.*, vol. 70, pp. 245–317, Jan. 2021.
[45] W. Zhu, H. Peng, C. Leng, C. Deng, and Z. Wu, "Surrogate-assisted firefly algorithm for breast cancer detection," *J. Intell. Fuzzy Syst.*, vol. 40, no. 5, pp. 8915–8926, Apr. 2021.
[46] S. Wang, Y. Wang, D. Wang, Y. Yin, Y. Wang, and Y. Jin, "An improved random forest-based rule extraction method for breast cancer diagnosis," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105941.
[47] M. M. Ghiasi and S. Zendehboudi, "Application of decision tree-based ensemble learning in the classification of breast cancer," *Comput. Biol. Med.*, vol. 128, Jan. 2021, Art. no. 104089.
[48] H. Liu and M. Cocea, "Induction of classification rules by gini-index based rule generation," *Inf. Sci.*, vols. 436–437, pp. 227–246, Apr. 2018.
[49] J. Cendrowska, "PRISM: An algorithm for inducing modular rules," *Int. J. Man-Machine Stud.*, vol. 27, no. 4, pp. 349–370, Oct. 1987.
[50] L. Breiman and N. Shang, "Born again trees," Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. 1:2, 1996.
[51] A. I. Weinberg and M. Last, "Selecting a representative decision tree from an ensemble of decision-tree models for fast big data classification," *J. Big Data*, vol. 6, no. 1, pp. 1–17, Dec. 2019.

JEONGEON LEE received the B.S. degree in computer science and engineering from Incheon National University, Incheon, South Korea, in 2021, and the M.S. degree from the Department of Data Science, Seoul National University of Science and Technology (SeoulTech), Seoul, South Korea, in 2023, where he is currently pursuing the Ph.D. degree in data science. His research interests include explainable data mining, decision tree, and stochastic processes application.

MIN KYU SIM (Member, IEEE) received the Ph.D. degree in industrial engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2014. From 2015 to 2017, he was a Portfolio Manager and a Quantitative Researcher with asset management firms. From 2018 to 2019, he was a Research Professor with the Smart Energy Research Center, Kyung Hee University, South Korea. Since September 2019, he has been an Associate Professor with the Department of Data Science and the Department of Industrial Engineering, Seoul National University of Science and Technology (SeoulTech), Seoul, South Korea. His research interests include stochastic processes application, reinforcement learning-based optimal control, and quantitative finance.

JUNG-SIK HONG received the B.S., M.S., and Ph.D. degrees in industrial engineering from Seoul National University, Seoul, South Korea. He is currently a Professor with the Department of Data Science and the Department of Industrial Engineering, Seoul National University of Science and Technology (SeoulTech), Seoul. He has authored more than 90 peer-reviewed publications focused on decision tree analysis, forecasting with big data, the development and parameter estimation of diffusion models, and performance analysis of communication networks. His research interests include mathematical modeling of innovation diffusion, data mining, and performance analysis of computer networks. He is a member of Korea Institute of Industrial Engineers and Korean Operations Research Society and a member of the Editorial Board of Korea Institute of Industrial Engineers.