



Article

Why Do Tree Ensemble Approximators Not Outperform the Recursive-Rule eXtraction Algorithm?

Soma Onishi *¹, Masahiro Nishimura¹, Ryota Fujimura¹ and Yoichi Hayashi *¹

Department Computer Science, Meiji University, Kawasaki 214-8571, Japan; nishimura_m@cs.meiji.ac.jp (M.N.); ryota.fujimura@cs.meiji.ac.jp (R.F.)

* Correspondence: soma.onishi@cs.meiji.ac.jp (S.O.); hayashiy@cs.meiji.ac.jp (Y.H.)

Abstract: Although machine learning models are widely used in critical domains, their complexity and poor interpretability remain problematic. Decision trees (DTs) and rule-based models are known for their interpretability, and numerous studies have investigated techniques for approximating tree ensembles using DTs or rule sets, even though these approximators often overlook interpretability. These methods generate three types of rule sets: DT based, unordered, and decision list based. However, very few metrics exist that can distinguish and compare these rule sets. Therefore, the present study proposes an interpretability metric to allow for comparisons of interpretability between different rule sets and investigates the interpretability of the rules generated by the tree ensemble approximators. We compare these rule sets with the *Recursive-Rule eXtraction algorithm (Re-RX) with J48graft* to offer insights into the interpretability gap. The results indicate that *Re-RX with J48graft* can handle categorical and numerical attributes separately, has simple rules, and achieves a high interpretability, even when the number of rules is large. *RuleCOSI+*, a state-of-the-art method, showed significantly lower results regarding interpretability, but had the smallest number of rules.

Keywords: interpretable machine learning; explainable artificial intelligence; rule extraction; rule-based model; decision list; decision tree; tree ensemble approximator



Citation: Onishi, S.; Nishimura, M.; Fujimura, R.; Hayashi, Y. Why Do Tree Ensemble Approximators Not Outperform the Recursive-Rule eXtraction Algorithm?. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 658–678. <https://doi.org/10.3390/make6010031>

Academic Editor: Ute Schmid

Received: 23 January 2024

Revised: 29 February 2024

Accepted: 14 March 2024

Published: 16 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Artificial intelligence (AI) has made great advances and AI algorithms are currently being applied to solve a wide variety of problems. However, this success has been driven by accepting their complexity and adopting “black box” AI models that lack transparency. On the other hand, eXplainable AI (XAI), which enhances the transparency of AI and facilitates its wider adoption in critical domains, has been attracting increasing attention [1–10].

Tree ensembles are often used for tabular data. Bagging [11] and random forests (RFs) [12] are known as independent ensembles, whereas gradient boosting machines (GBMs) [13] such as *XGBoost* [14], *LightGBM* [15], and *CatBoost* [16] are known as dependent ensembles. Tree ensembles are extensively utilized in academic and research contexts and applied in practical scenarios across a wide array of domains [17]. Recently, these models have been shown to be effective in many classification tasks. In fact, these models are used by most winners of Kaggle competitions (Kaggle is a platform for predictive modeling and analytics competitions in which statisticians and data miners compete to produce the best models for predicting and describing the datasets uploaded by companies and users). However, the structure of these algorithms is considered complex and very difficult to interpret. The effectiveness of ensemble trees generally improves as the number of trees increases, and in some cases, an ensemble can contain thousands of trees.

Rudin [18] pointed out the limitations of some approaches to explainable machine learning, suggesting that interpretable models should be used instead of black box models for making high stakes decisions. Recently, XAI has entered a new phase with the provisional agreement of the AI Act aimed at explaining AI [19]. This is important because black

box machine learning applications remain challenging in several domains, such as health care and finance. In the field of health care, for example, it is not sufficient for a medical diagnosis model to simply be accurate, it must also be transparent to health professionals who use the output to make decisions about a given patient [6,20,21]. Moreover, in the field of finance, recent regulations, such as the *General Data Protection Regulation* and the *Equal Credit Opportunity Act*, have increased the need for model interpretability to ensure that algorithmic decisions are understandable and consistent. Decision-making is critical and requires a rationale. In fact, the predictions of the XGBoost classifier in regard to business failures have been explained in a previous study [22]. These issues have been addressed by interpretable machine learning models, which are characterized as models that can be easily visualized or described in plain text for the end user [23]. On the other hand, some domains highly prioritize classification accuracy. Furthermore, deep learning is an option in domains where decision rationales are less important. Techniques that account for deep learning still rely on subjective methods, such as saliency maps, which have a limited explanatory capability for unstructured data, such as images. On the other hand, significant progress has been made in making structured data explainable with deep learning [24].

Decision trees (DTs), rule-based approaches, and knowledge graph-based approaches are widely used as examples of interpretable models [25–32]. Techniques for approximating tree ensembles with DTs or rule sets have also been investigated [33–39]. However, although tree ensemble approximators focus on reducing the number of rules and conditions, they do not consider the interpretability of the rules. For example, handling categorical and numerical attributes separately is known to increase interpretability [31].

There are three main types of rule sets generated by these methods: DT based, unordered based (the last rule in the rule set is *Else*), and decision list based. Figure 1 shows the concept of these three types of rule sets. However, previous studies have not provided a metric to distinguish and compare these different rule sets. Therefore, even if an existing metric such as the number of rules suggests a method's superiority in terms of interpretability, its practical interpretability may remain low. As existing studies have not properly assessed the superiority of the proposed method, a new metric is needed to fill this gap. Table 1 shows the existing interpretability metrics for rule sets.

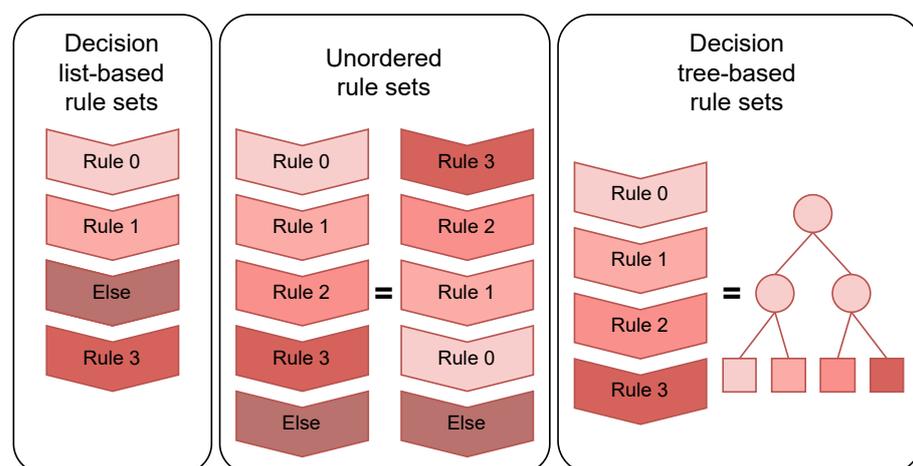


Figure 1. Concepts of the three types of rule sets. Decision list-based rule sets classify instances by sequentially referencing rules from top to bottom. Unordered rule sets classify instances by referencing rules in any order. Decision tree-based rule sets are variable according to decision trees and differ from the other two types of rule sets in that all instances are classified using only a single rule.

Table 1. Interpretability metrics for rule sets.

Metrics	Description
Number of rules	This metric is the total number of rules.
Number of conditions	This metric is the sum of the total or average number of conditions.
Complexity [40]	This metric relates the complexity, namely the number of conditions, to the number of classes.
Fraction uncover [41]	This metric measures the proportion of data points covered by at least one rule in the rule set.
Fraction overlap [41]	This metric calculates the degree to which rules in the rule set redundantly cover the same data points.
Uniq [39]	This metric quantifies the amount of non-redundant conditions contained in the rule set.
$expl_{avg}$ [42]	This metric is the weighted sum of the number of different attributes, based on the coverage of each rule.
Weighted sum of predictivity, stability, and simplicity [43]	This metric combines three critical aspects: predictivity, which assesses accuracy; stability, gauged through the Dice–Sorensen index comparing rule sets; and simplicity, measured by rule length sum, into a comprehensive weighted sum.

In the present study, we propose an interpretability metric, *Complexity of Rules with Empirical Probability (CREP)*, to allow for comparisons of interpretability between different rule sets. *CREP* enables a fair comparison of different types of rule sets, such as decision list- and DT-based rule sets. We also explore the interpretability of the rules generated by the tree ensemble approximators. Specifically, we present and compare not only objective metrics, but also rule sets generated by the tree ensemble approximators and the *Recursive-Rule eXtraction algorithm (Re-RX) with J48graft* [32].

The contributions of this study are as follows:

1. **Introduction of a New Interpretability Metric:** We propose a metric designed to evaluate and compare the interpretability of different types of rule sets. This metric addresses the gap in quantitative assessments of the interpretability of rule sets.
2. **Comparative Analysis of Rule Sets:** We provide an exhaustive comparison of decision list- and DT-based rule sets generated by the tree ensemble approximator. Our analysis provides new insights into the strengths and weaknesses of each type of rule set. Furthermore, we explain why categorical and numerical attributes should be treated separately.
3. **Focus on the Interpretability of Categorical and Numerical Attributes:** We explain the necessity of separating categorical and numerical attributes. This focus addresses a significant gap in current research as many existing methods overlook the distinction between categorical and numerical attributes.

2. Related Work

The *Re-RX* algorithm [31] is a rule-based approach that can handle categorical and numerical attributes separately and extract rules recursively. By separating categorical and numerical attributes, *Re-RX* can generate rules that are intuitively easy to understand. *Re-RX with J48graft* [32] is the extended version of *Re-RX*. Numerous studies have conducted research on *Re-RX* [44–49]. *RuleFit* [33] is a method that employs a linear regression model with a DT-based model to utilize interactions. The rules generated by the ensemble tree are used as new features and fitted using Lasso linear regression. *inTrees* [34] extracts, measures, prunes, and selects rules from tree ensembles such as RFs and boosts trees to generate a simplified tree ensemble learner for interpretable predictions. *DefragTrees* [35] involves simplifying complex tree ensembles, such as RFs, to enhance interpretability by

formulating the simplification as a model selection problem and employing a Bayesian algorithm that optimizes the simplified model while preserving prediction performance. Initially introduced for independent tree ensembles by Sagi and Rokach [36], *Forest-based Trees (FBTs)* were later extended to dependent tree ensembles. Combined within both bagging (e.g., RFs) and boosting ensembles (e.g., GBMs), *FBTs* construct a singular DT from an ensemble of trees. *Rule COmbination and SIMplification (RuleCOSI+)* [39], a recent advance in the field, is a fast post-hoc explainability approach. In contrast to its precursor, *RuleCOSI*, which was limited to imbalanced data and Adaboost-based small trees according to Obregon et al. [38], *RuleCOSI+* was designed as an algorithm that extends the capabilities of *RuleCOSI* to function effectively in both bagging (e.g., RFs) and boosting (e.g., GBMs) ensembles. *DefragTrees*, *FBTs*, and *Re-RX* generate DT-based rule sets, *inTree* generates an unordered-based rule set, and *RuleCOSI+* generates a decision list-based rule set.

Given this background, in the present study, we aim to provide new insights into the reasons why DL-based and DL-inspired classifiers do not work well for categorical datasets mainly consisting of nominal attributes [50].

3. Materials and Methods

3.1. Datasets

We used 10 datasets from the University of California, Irvine, Machine Learning Repository [51] to compare each method. We provide the sources of these datasets in Appendix A. The details of the datasets are shown in Table 2. For each dataset, we split the data into training–test at a ratio of 8:2. Consistent splits were applied to all methods, with a unique seed-based split for each iteration. Each iteration means 10× in the 10 × 10-fold cross-validation (CV) scheme described in Section 4.3.

Table 2. Dataset properties. In this paper, # indicates quantity.

Dataset	#Instances	#Features	#Cate	#Cont	Major Class Ratio
heart	270	13	7	6	0.55
australian	690	14	6	8	0.555
mammographic	831	4	2	2	0.52
tic-tac-toe	958	9	9	0	0.65
german	1000	20	7	13	0.70
biodeg	1055	41	0	41	0.66
banknote	1372	4	0	4	0.55
bank-marketing	4521	16	9	7	0.89
spambase	4601	57	0	57	0.60
occupancy	8143	5	0	5	0.79

3.2. Baseline

We used scikit-learn’s *DT* and *J48graft* [52] as simple DT-based methods. *J48graft* is a grafted (pruned or unpruned) *C4.5* [53] DT. This *DT* generates a binary tree, while *J48graft*, capable of handling categorical attributes, generates an *m*-ary tree.

We used *FBTs* and *RuleCOSI+* for the tree ensemble approximator. Figures 2 and 3 show overviews of *FBTs* and *RuleCOSI+*, respectively. Both *FBTs* and *RuleCOSI+* were implemented using the official code provided by the authors (<https://github.com/sagyome/XGBoostTreeApproximator> (13 March 2024) <https://github.com/jobregon1212/rulecosi> (13 March 2024)).

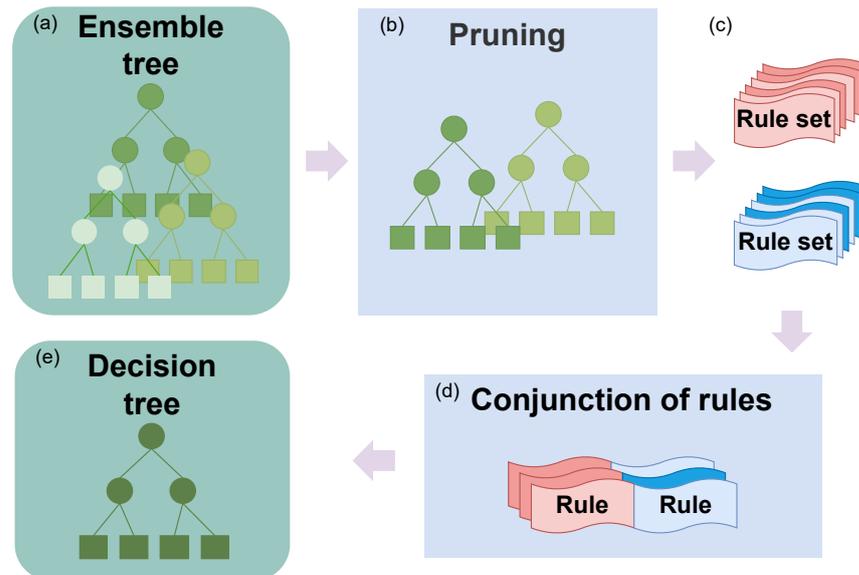


Figure 2. Overview of FBTs. (a) Fit the ensemble tree; (b) pruning: remove trees that do not improve the accuracy from the ensemble tree; (c) convert the tree to rules; (d) conjunction of rules: generate the conjunction set by gradually merging the conjunction sets of the base trees into a single set that represents the entire ensemble; (e) convert to a decision tree.

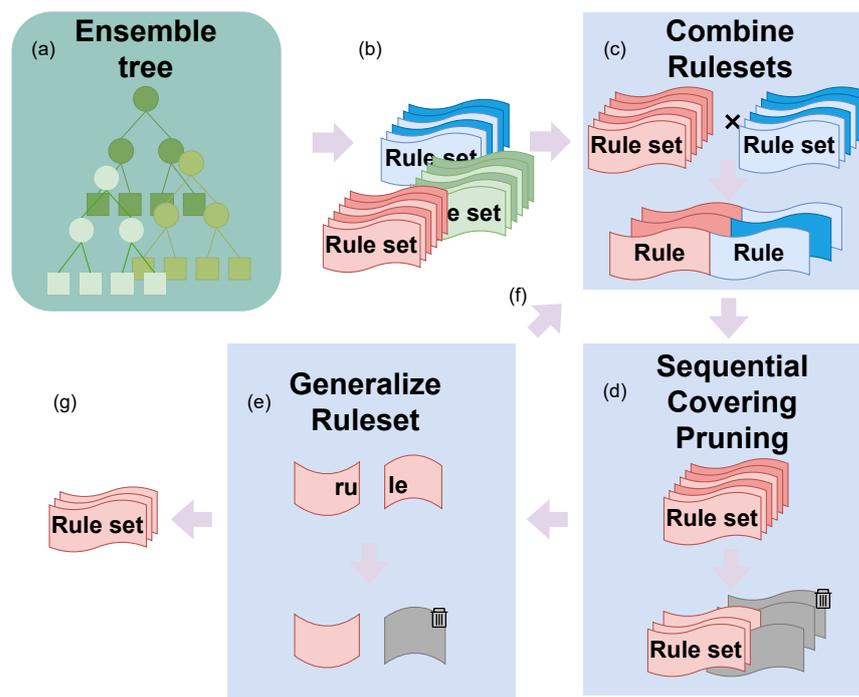


Figure 3. Overview of RuleCOSI+. (a) Fit the ensemble tree; (b) convert the tree to rule sets; (c) combine rule sets: greedily verify all combinations and determine the rules to adopt and create a new rule set; (d) sequential covering pruning: simplify the rule set; (e) generalize rule set: remove any unnecessary conditions; (f) repeat (c–e) using the rule set generated in (e) and the rule set obtained from the remaining ensemble tree (green rule set in the figure); (g) obtain the final rule set.

As a rule-based method, we used *Re-RX with J48graft*, an overview of which is shown in Figure 4. In *Re-RX with J48graft*, a method derived from the Re-RX grafting family that incrementally adds rules to form a rule set is conditionally selected. If the accuracy does not significantly increase, it aims to improve interpretability. *Re-RX with J48graft* and its interpretability are positioned at one end of the spectrum, leading to a smaller

model. Thus, for large datasets or more complex ensemble models, *Re-RX with J48graft* does not consider either tree ensembles or growth strategies for increased tree depth. In learning a multilayer perceptron (MLP) in *Re-RX with J48graft*, we apply one-hot encoding (we used OneHotEncoder from scikit-learn [54]) to categorical attributes to enable efficient learning. With the application of the one-hot encoding to categorical attributes, we modified the pruning algorithm for the MLP. In the original pruning algorithm, the attributes are removed from \mathcal{D} when $w_{i,*} = 0$, where $w_{i,*}$ represents all weights in the first layer of the MLP connected to the i -th attribute. Let \mathcal{C} be the set of categorical attributes in the dataset \mathcal{D} , and $\bar{\mathcal{C}}_i$ be the set of one-hot encoded values for the i -th categorical attribute $c_i \in \mathcal{C}$. In this study, we modified the pruning algorithm as follows: $\forall j \in \{j = 0, \dots, |\bar{\mathcal{C}}_i| - 1\}$, if $w_{j,*} = 0$, then $\mathcal{D} \leftarrow \mathcal{D} \setminus \{c_i\}$.

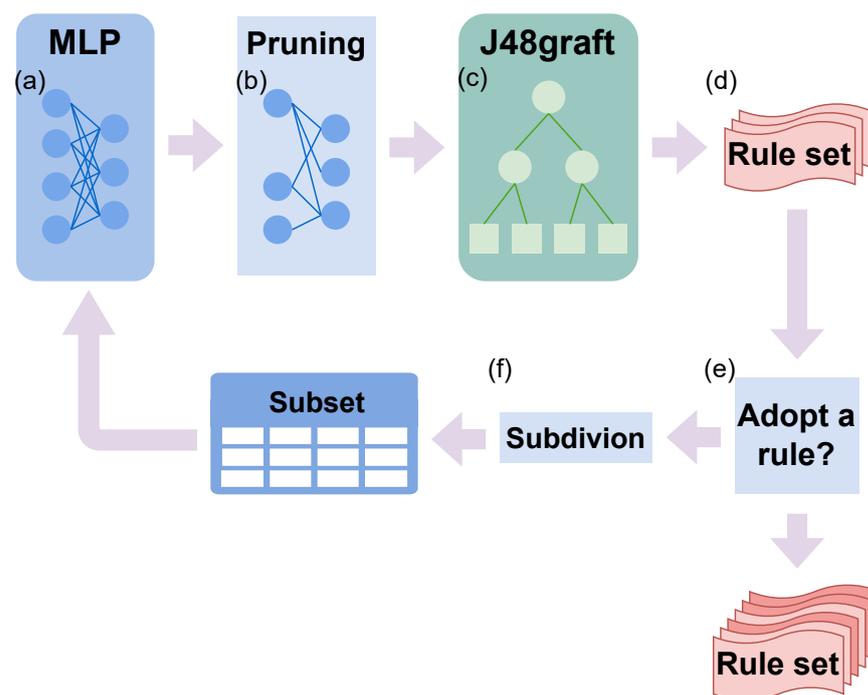


Figure 4. Overview of *Re-RX with J48graft*. (a) Fit a multilayer perceptron; (b) pruning: reduce the number of attributes; (c) fit *J48graft*; (d) convert the tree to a rule set; (e) adopt a rule: select a rule to adopt as is; (f) subdivision: recursive (a–e) for a rule that is not adopted using data and has attributes not included in the rule.

All fitted methods are converted to the RuleSet (<https://github.com/jobregon1212/rulecosi/blob/master/rulecosi/rules.py> (13 March 2024)) module implemented by Obregon and Jung [39].

4. Proposed Methodology

In this section, we present the methodology of experiments for comparing methods.

4.1. Data Preprocessing

We applied one-hot encoding to the categorical attributes because of the inability of *FBTs*, *RuleCOSI+*, and *DT* to handle categorical attributes. For example, if there is a categorical attribute $c \in \{x, y\}$, it is converted to the new attributes $c = "x" \in \{0, 1\}$ and $c = "y" \in \{0, 1\}$. For the numeric attributes, we applied standardization only to the training and prediction of the MLP in *Re-RX with J48graft*.

4.2. Interpretability Metrics

The metrics of interpretability, such as the *total number of rules* (N_{rules}) and *average number of conditions*, are often used. However, these metrics cannot distinguish between decision list-based, unordered, and DT-based rule sets.

We propose a new interpretability metric, *CREP*, to facilitate fair comparisons of interpretability between different rule sets. *CREP* quantifies the complexity of rules based on their empirical probability (coverage on the training data), and is defined as follows:

$$CREP = \sum_{r \in R} N_{cond_r} \cdot cov(r, \mathcal{D}) \quad (1)$$

where N_{cond_r} is the number of conditions in rule r , and $cov(r, \mathcal{D})$ is the coverage of rule r in training data \mathcal{D} . If the rule set is a decision list or an unordered rule set, the instances in the data refer to one or more rules in the rule set. Taking Figure 1 as an example, the actual rule for an instance classified into Rule 2 (r_2) in a decision list-based rule set is $\neg r_0 \wedge \neg r_1 \wedge r_2 \rightarrow y$. In the unordered rule set, the actual rule for an instance classified into *Else* is $\neg r_0 \wedge \neg r_1 \wedge \neg r_2 \wedge \neg r_3 \rightarrow y$. Therefore, if the rule set is a decision list, N_{cond_r} is accumulated from the top, and if the rule set is an unordered rule set, the N_{cond_r} of all rules is added together for the *Else* rule in the rule set. The operation of N_{cond_r} accumulation allows *CREP* to compare different rule sets fairly.

CREP represents the expected value of the number of conditions in the rules using the empirical probability obtained from the training data. In other words, if rules with a high likelihood of being referenced have fewer conditions, *CREP* decreases, and if they have more conditions, *CREP* increases. Conversely, rules with a low likelihood of being referenced may have many conditions, but their impact is minimal. Compared with N_{rules} and the *average number of conditions*, which evaluate the interpretability of the entire model, *CREP* can be considered a more practical metric.

CREP in Equation (1) treats all classes equally and therefore underestimates the interpretability of minority class rules when the dataset is class-imbalanced. This problem can be solved by calculating *CREP* for each class. We redefine as follows:

$$\text{micro-}CREP = \sum_{r \in R} N_{cond_r} \cdot cov(r, \mathcal{D}) \quad (2)$$

$$CREP_c = \sum_{r \in R_c} N_{cond_r} \cdot cov(r, \mathcal{D}) \quad (3)$$

$$\text{macro-}CREP = \frac{1}{|C|} \sum_{c \in C} CREP_c \quad (4)$$

where C is the set of all classes and R_c is the subset for each class in the rule set. *micro-}CREP* is useful for both unbalanced datasets and evaluating entire rule sets. In this paper, we refer to *micro-}CREP* as *CREP*.

4.3. Model Evaluation and Hyperparameter Optimization

We performed the experiment using a stratified 10×10 -fold CV (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html (13 March 2024)) scheme, which is a 10-fold CV repeated 10 times. In each CV-fold, we performed hyperparameter optimization using Optuna [55]. First, the hyperparameters of the base model, which is an *MLP* in *Re-RX with J48graft* and *XGBoost* in *FBTs* and *RuleCOSI+*, were optimized to maximize the classification performance. Then, the other hyperparameters were optimized using multi-objective optimization (https://optuna.readthedocs.io/en/stable/tutorial/20_recipes/002_multi_objective.html (13 March 2024)) to maximize both classification performance and interpretability simultaneously. For both *DT* and *J48graft*, the first step was skipped because these methods do not have a base model. We used the area under the receiver operating characteristics curve (AUC-ROC) [56] for the classification

performance metric, and the inverse of N_{rules} ($1/N_{rules}$) for the interpretability metric. See Appendix B for details on the hyperparameters in each method.

In the case of multi-objective optimization, the optimal hyperparameters are provided on the Pareto front. From the Pareto front, we selected the hyperparameters that maximize the following equation:

$$k = -\log_2 N_{rules} + \alpha \cdot AUC \quad (5)$$

where α is a parameter that controls the trade-off between classification performance and interpretability. This equation indicates that AUC increases by $1/\alpha$ and N_{rules} decreases by half, which are equivalent. A higher α prioritizes classification performance, whereas a lower α prioritizes interpretability. In this experiment, we set $\alpha = 0.25$. In other words, the AUC value increasing by four points and N_{rules} decreasing by half are equivalent. We excluded the Pareto solution with $N_{rules} = 1$. When $N_{rules} = 1$, the rule set R classifies all instances into the same label, which is a meaningless rule set.

4.4. Summary of Evaluation Schemes for Each CV-Fold

Each CV-fold evaluation scheme of our proposed method is shown in Algorithm 1. First, we preprocess the datasets \mathcal{D}_{train} , \mathcal{D}_{val} , and \mathcal{D}_{test} . Next, using the training dataset \mathcal{D}'_{train} , we conduct hyperparameter optimization on the validation dataset \mathcal{D}'_{val} . From the Pareto front, we select the optimal hyperparameters using Equation (5). Subsequently, we fit the model to the dataset \mathcal{D}'_{train} with the selected hyperparameters and evaluate it on dataset \mathcal{D}'_{test} . Finally, we return the score set S for the rule set R_{best} . By using this scheme, we are able to maximize the performance of each method.

Algorithm 1 Evaluation scheme for each CV-fold

Require: Training dataset \mathcal{D}_{train} , Validation dataset \mathcal{D}_{val} , Test dataset \mathcal{D}_{test} , Method M

Ensure: The score set S for rule set R_{best}

- 1: $\mathcal{D}'_{train}, \mathcal{D}'_{val}, \mathcal{D}'_{test} \leftarrow \text{Preprocessing}(\mathcal{D}_{train}, \mathcal{D}_{val}, \mathcal{D}_{test})$
 - 2: $\text{ParetoFront} \leftarrow \text{OptimizeHyperparametersWithOptuna}(\mathcal{D}'_{train}, \mathcal{D}'_{val}, M)$
 - 3: $S_N, S_A \leftarrow \text{ParetoFront}$ ▷ Get the sets of N_{rules} and AUC from the Pareto front
 - 4: $k \leftarrow \underset{k \in \{1, \dots, |S_N|\}}{\text{argmax}} (-\log_2 S_N[k] + \alpha \cdot S_A[k])$
 - 5: $R_{best} \leftarrow \text{Fit}(\mathcal{D}'_{train}, \mathcal{D}'_{val}, M, \text{ParetoFront}[k])$
 - 6: $S \leftarrow \text{Evaluate}(\mathcal{D}'_{test}, R_{best})$
 - 7: **return** S
-

5. Results

In this section, we present the experimental results and their analyses.

5.1. Classification Results

The classification results are presented in Table 3. *RuleCOSI+* outperformed the other methods in many datasets. *DT* was inferior to *RuleCOSI+* but superior to the other baselines. *FBTs* and *J48graft* performed better than *Re-RX with J48graft*, but tended to generate more rules and less interpretability than the other baselines, as discussed in the next subsection. *Re-RX with J48graft* was inferior to the other baselines on average, but showed competitive results against *RuleCOSI+* in some datasets.

Table 3. Results for classification performance ($\mu \pm \sigma$). All metrics are reported as *AUC*. In this paper, for each dataset, the top results are in **bold** (here, “top” means that the gap between this result and the result with the best score is not statistically significant at a level of 0.05 (Welch’s *t*-test [57])). For each dataset, ranks are calculated by sorting the average of the reported scores, and the “rank” row reports the average rank across all datasets.

Dataset	<i>FBTs</i>	<i>RuleCOSI+</i>	<i>Re-RX with J48graft</i>	<i>J48graft</i>	<i>DT</i>
heart	72.60 ± 7.90	73.88 ± 6.20	69.99 ± 7.31	71.51 ± 5.61	72.37 ± 5.59
australian	86.21 ± 4.93	86.01 ± 2.81	84.91 ± 8.30	86.70 ± 2.14	86.73 ± 2.19
mammographic	76.96 ± 4.60	79.27 ± 2.81	76.76 ± 4.74	73.11 ± 4.98	77.84 ± 3.36
tic-tac-toe	77.35 ± 11.25	76.79 ± 3.92	65.76 ± 4.37	66.57 ± 2.86	71.26 ± 7.70
german	55.65 ± 4.94	64.48 ± 4.62	63.48 ± 5.31	62.65 ± 4.55	64.23 ± 4.56
biodeg	75.11 ± 3.77	75.08 ± 3.92	73.55 ± 4.51	77.28 ± 4.07	73.09 ± 3.06
banknote	91.74 ± 4.40	92.35 ± 2.45	89.31 ± 5.99	87.75 ± 4.83	88.21 ± 3.34
bank-marketing	66.67 ± 6.51	71.35 ± 1.96	58.31 ± 3.87	62.75 ± 2.93	63.57 ± 6.69
spambase	79.40 ± 3.53	85.15 ± 1.55	82.91 ± 4.24	87.06 ± 1.72	82.57 ± 3.52
occupancy	98.72 ± 0.48	97.84 ± 0.60	98.90 ± 0.34	98.98 ± 0.22	99.00 ± 0.22
ranking	2.9	2.1	4.0	3.2	2.8

5.2. Interpretability Results

The interpretability results and the number of rules in N_{rules} and *CREP* are presented in Tables 4 and 5. *RuleCOSI+* outperformed the other methods for many datasets in N_{rules} . In particular, the variance was considerably smaller than that of the other methods, resulting in stable rule set generation. On the other hand, *CREP* was larger than other methods. Because *RuleCOSI+* generated a decision list-based rule set, *CREP* tended to be large. This is discussed in detail in Sections 5.4 and 6.4. *DT* obtained superior N_{rules} and *CREP* for many datasets. *FBTs* and *J48graft* produced very large N_{rules} for some datasets. *FBTs* had higher N_{rules} variance in the *tic-tac-toe*, *german*, *biodeg*, and *bank-marketing* datasets, indicating unstable rule set generation. Furthermore, *CREP* was larger for *FBTs*, even though it is a tree-based method. Although *Re-RX with J48graft* resulted in N_{rules} being slightly higher than the other methods, except *FBTs*, on average, *CREP* was much lower than the other methods. *Re-RX with J48graft* and *J48graft* tended to have a large N_{rules} because they both handle categorical attributes.

5.3. Summary of Comparative Experiments

Table 6 shows a summary of all the classification and interpretability results presented in the previous subsections. *RuleCOSI+* had the highest scores for *AUC* and N_{rules} , a result that overwhelmed the other methods when N_{rules} was emphasized as an indicator of interpretability. Although *DT* and *Re-RX with J48graft* were inferior to *RuleCOSI+* in terms of classification performance, they outperformed the other methods in *CREP*. In other words, *DT* and *Re-RX with J48graft* are appropriate when the interpretability and classification frequency of the rules by which instances are classified are important. *FBTs* and *J48graft* were not significantly better than the other methods in any of the metrics, and were relatively unsuitable when interpretability was more important.

Table 4. Results for the number of rules.

Dataset	FBTs	RuleCOSI+	Re-RX with J48graft	J48graft	DT
heart	13.95 ± 28.59	3.80 ± 2.20	4.51 ± 2.82	6.26 ± 5.40	3.56 ± 2.86
australian	3.05 ± 6.09	2.19 ± 0.61	2.00 ± 0.37	2.17 ± 0.76	2.08 ± 0.58
mammographic	4.23 ± 4.51	2.15 ± 0.50	4.79 ± 2.00	4.38 ± 3.25	2.23 ± 1.08
tic-tac-toe	71.93 ± 136.98	3.39 ± 1.00	6.40 ± 10.96	4.70 ± 6.16	6.15 ± 7.11
german	37.14 ± 139.94	3.24 ± 1.31	10.59 ± 10.14	9.97 ± 7.83	3.76 ± 1.98
biodeg	69.41 ± 142.78	3.83 ± 3.28	12.54 ± 11.63	47.40 ± 38.84	2.92 ± 2.06
banknote	11.55 ± 16.74	4.06 ± 1.04	6.42 ± 3.89	8.06 ± 6.88	3.91 ± 2.25
bank-marketing	27.43 ± 81.14	2.73 ± 0.56	5.95 ± 2.03	7.89 ± 5.97	3.82 ± 1.85
spambase	3.95 ± 4.36	2.25 ± 0.79	13.69 ± 11.08	101.30 ± 51.57	3.72 ± 1.64
occupancy	2.54 ± 2.51	2.02 ± 0.14	2.07 ± 0.26	6.00 ± 0.00	2.04 ± 0.20
ranking	4.5	1.6	3.3	3.8	1.8

Table 5. Results for CREP.

Dataset	FBTs	RuleCOSI+	Re-RX with J48graft	J48graft	DT
heart	2.68 ± 1.59	5.01 ± 3.00	1.25 ± 0.48	1.47 ± 0.58	1.53 ± 0.89
australian	1.13 ± 0.72	2.62 ± 1.22	0.97 ± 0.25	1.05 ± 0.21	1.03 ± 0.23
mammographic	1.70 ± 0.92	2.18 ± 0.80	1.03 ± 0.14	1.27 ± 0.24	1.10 ± 0.37
tic-tac-toe	3.94 ± 2.84	5.62 ± 1.44	1.16 ± 0.56	1.11 ± 0.36	1.83 ± 1.32
german	3.27 ± 1.88	4.42 ± 2.50	1.43 ± 0.48	1.63 ± 0.37	1.86 ± 0.56
biodeg	3.83 ± 2.66	4.44 ± 4.01	3.15 ± 1.08	6.02 ± 2.67	1.38 ± 0.75
banknote	2.88 ± 1.35	3.90 ± 0.93	2.13 ± 0.64	2.33 ± 0.55	1.82 ± 0.75
bank-marketing	3.01 ± 2.03	3.18 ± 0.58	1.29 ± 0.44	1.65 ± 0.28	2.19 ± 0.90
spambase	1.56 ± 0.94	3.42 ± 1.03	3.29 ± 1.24	9.06 ± 1.20	1.95 ± 0.76
occupancy	1.17 ± 0.48	1.81 ± 0.40	1.02 ± 0.06	1.89 ± 0.01	1.03 ± 0.15
ranking	3.5	4.7	1.5	3.1	2.2

Table 6. Summary of classification and interpretability performance ($\mu \pm \sigma$) across all datasets. For each metric, the top results are in **bold** (here, “top” means that the gap between this result and the result with the best score is not statistically significant at a level of 0.05 (Welch’s *t*-test [57])).

Method	AUC	N_{rules}	CREP
FBTs	78.04 ± 13.05	24.52 ± 85.43	2.52 ± 1.99
RuleCOSI+	80.22 ± 10.20	2.97 ± 1.63	3.66 ± 2.28
Re-RX with J48graft	76.39 ± 13.14	6.90 ± 8.14	1.67 ± 1.06
J48graft	77.44 ± 12.23	19.81 ± 36.50	2.75 ± 2.70
DT	77.89 ± 11.60	3.42 ± 3.05	1.57 ± 0.85

5.4. Two Examples

We present two examples of rules actually generated in the *german* and *bank-marketing* datasets and compare *DT*, *Re-RX with J48graft*, and *RuleCOSI+*. We excluded *FBTs* and *J48graft* from the comparison in this section because of the relatively large numbers of rules and the difficulty of analyzing the rules. For each method, rules with N_{rules} matching the median were adopted.

In *RuleCOSI+* and *DT*, categorical attributes were renamed into columns by one-hot encoding. For example, the one-hot encoding for element a of attribute x generates the attribute $x = "a"$. In other words, a rule such as $x = "a" > 0.5$ is equivalent to $x = a$, and a rule such as $x = "a" \leq 0.5$ is equivalent to $x \neq a$. To maintain consistency in notation, we converted all one-hot encoded categorical attributes in the rules to the format $x = a$ or $x \neq a$.

5.4.1. bank-marketing

Table 7 shows an example of the rule set generated by each method. The rule set generated by *RuleCOSI+* consists of complex rules involving both numerical and categorical attributes. Furthermore, the rule for class 1 had extremely low interpretability because it was expressed as follows:

$$r_{\text{class 1}} := \neg r_0 \wedge \neg r_1 \rightarrow [1] \tag{6}$$

By contrast, the rule set generated by *Re-RX with J48graft* consists exclusively of rules based on categorical attributes, resulting in a relatively high interpretability. Furthermore, it is simplified by post-processing, as shown in Table 8. The rule set generated by *DT*, while containing numerical attributes, was composed of simple rules.

Table 7. Rule sets generated from the bank-marketing dataset.

<i>RuleCOSI+</i>		Coverage
r_1	$(V16 \neq \text{success}) \wedge (V12 \leq 351.5) \wedge (V11 \neq \text{oct}) \rightarrow [0]$	0.744
r_2	$(V16 \neq \text{success}) \wedge (V12 \leq 645.5) \wedge (V1 \leq 70.5) \rightarrow [0]$	0.148
r_3	$\rightarrow [1]$	0.109
<i>Re-RX with J48graft</i>		Coverage
r_1	$(V16 = \text{unknown}) \rightarrow [0]$	0.820
r_2	$(V16 = \text{failure}) \rightarrow [0]$	0.108
r_5	$(V16 = \text{other}) \rightarrow [0]$	0.044
r_3	$(V16 = \text{success}) \wedge (V5 = \text{yes}) \rightarrow [0]$	0.0
r_4	$(V16 = \text{success}) \wedge (V5 \neq \text{yes}) \rightarrow [1]$	0.029
<i>DT</i>		Coverage
r_1	$(V12 \leq 631.5) \wedge (V16 \neq \text{success}) \rightarrow [0]$	0.891
r_2	$(V12 \leq 631.5) \wedge (V16 = \text{success}) \rightarrow [1]$	0.025
r_3	$(V12 > 631.5) \rightarrow [1]$	0.084

Table 8. Post-processed rule set generated from the bank-marketing dataset in *Re-RX with J48graft*.

<i>Re-RX with J48graft</i>		Coverage
r_1	$(V16 \neq \text{success}) \rightarrow [0]$	0.971
r_3	$(V16 = \text{success}) \wedge (V5 = \text{yes}) \rightarrow [0]$	0.0
r_4	$(V16 = \text{success}) \wedge (V5 \neq \text{yes}) \rightarrow [1]$	0.029

5.4.2. german

Table 9 shows an example of the rule set generated by each method. As in Section 5.4.1, the rule set generated by *RuleCOSI+* contained a complex mixture of numerical and categorical attributes, and the rule for class 1 was Equation (6), which had extremely low interpretability. On the other hand, the rule set generated by *Re-RX with J48graft* was relatively highly interpretable because the rules were composed of categorical attributes, except for r_8 and r_9 . For r_8 and r_9 , *Re-RX with J48graft* performed subdivision and added the numerical

attribute *duration* to improve the accuracy of the rule for $checking_status = 0 \leq X < 200$. Also, as shown in Table 10, the rule set can be simpler, as in Section 5.4.1. The rule set generated by *DT*, while containing numerical attributes, was composed of simple rules.

Table 9. Rule sets generated from the german dataset.

	RuleCOSI+	Coverage
r_1	$(age > 19.5) \wedge (checking_status = no\ checking) \wedge (other_payment_plans = none) \rightarrow [0]$	0.329
r_2	$(duration \leq 26.5) \wedge (checking_status \neq <0) \wedge (credit_amount \leq 10841.5) \rightarrow [0]$	0.283
r_3	$\rightarrow [1]$	0.388
	Re-RX with J48graft	Coverage
r_1	$(checking_status = no\ checking) \rightarrow [0]$	0.394
r_2	$(checking_status = \geq 200) \rightarrow [0]$	0.063
r_3	$(checking_status = <0) \wedge (credit_history = existing\ paid) \rightarrow [1]$	0.16
r_4	$(checking_status = <0) \wedge (credit_history = critical/other\ existing\ credit) \rightarrow [0]$	0.067
r_5	$(checking_status = <0) \wedge (credit_history = no\ credits/all\ paid) \rightarrow [1]$	0.013
r_6	$(checking_status = <0) \wedge (credit_history = all\ paid) \rightarrow [1]$	0.022
r_7	$(checking_status = <0) \wedge (credit_history = delayed\ previously) \rightarrow [1]$	0.012
r_8	$(checking_status = 0 \leq X < 200) \wedge (duration \leq 26) \rightarrow [0]$	0.19
r_9	$(checking_status = 0 \leq X < 200) \wedge (duration > 26) \rightarrow [1]$	0.079
	DT	Coverage
r_1	$(checking_status = no\ checking) \rightarrow [0]$	0.394
r_2	$(checking_status \neq no\ checking) \wedge (duration \leq 19) \rightarrow [0]$	0.325
r_3	$(checking_status \neq no\ checking) \wedge (duration > 19) \rightarrow [1]$	0.281

Table 10. Post-processed rule set generated from the german dataset in *Re-RX with J48graft*.

	Re-RX with J48graft	Coverage
r_1	$(checking_status = no\ checking) \rightarrow [0]$	0.394
r_2	$(checking_status = \geq 200) \rightarrow [0]$	0.063
r_3	$(checking_status = <0) \wedge (credit_history = critical/other\ existing\ credit) \rightarrow [0]$	0.067
r_4	$(checking_status = <0) \wedge (credit_history \neq critical/other\ existing\ credit) \rightarrow [1]$	0.207
r_5	$(checking_status = 0 \leq X < 200) \wedge (duration \leq 26) \rightarrow [0]$	0.19
r_6	$(checking_status = 0 \leq X < 200) \wedge (duration > 26) \rightarrow [1]$	0.079

6. Discussion

6.1. Why Should We Avoid a Mixture of Categorical and Numerical Attributes?

Many methods that have been proposed to enhance interpretability cannot handle categorical and numerical attributes separately. If we could adequately express a rule using only categorical attributes, the use of numerical attributes would reduce interpretability. The conditions for categorical attributes are easy to understand intuitively because they are categorized into a finite group. On the other hand, the conditions for numerical attributes are difficult to understand intuitively because there are an infinite number of thresholds, so the division is not deterministic. Furthermore, it is not common for the division to be performed convincingly. For example, in the german dataset, there are only four conditions for the attribute $checking_status \in \{no\ checking, <0, 0 \leq X < 200, \geq 200\}$, whereas the conditions for the attribute $credit_amount \in \mathbb{N}$ are infinite. In the division of the condition $credit_amount \leq 10,841.5$ in r_2 in *RuleCOSI+* in Table 9, it is difficult to understand why the value 10,841.5 was chosen. Furthermore, rules that contain many

numerical attribute conditions make it more difficult to understand how each condition relates to the other. Setting thresholds for numerical attributes is infinite, and an intuitive understanding of how such thresholds affect the other attributes and the overall rules is difficult. Combining the conditions of multiple numerical attributes exponentially increases the complexity of the rule. By contrast, the conditions for categorical attributes are clustered in a finite group, which makes their relationships and effects easier to understand. Therefore, to realize high interpretability, mixing categorical and numerical attributes should be avoided.

6.2. Optimal Selection of the Pareto Solutions

In this study, we selected the Pareto optimal solution from the Pareto front obtained by multi-objective optimization with Optuna using Equation (5). However, the Pareto optimal solution selected by Equation (5) is not always the optimal solution sought by the user. As shown in Figure 5, we observed that the Pareto front depends significantly on the dataset, method, and data splitting. In other words, to select the optimal Pareto solution in real-world applications, it is desirable to verify the Pareto front individually.

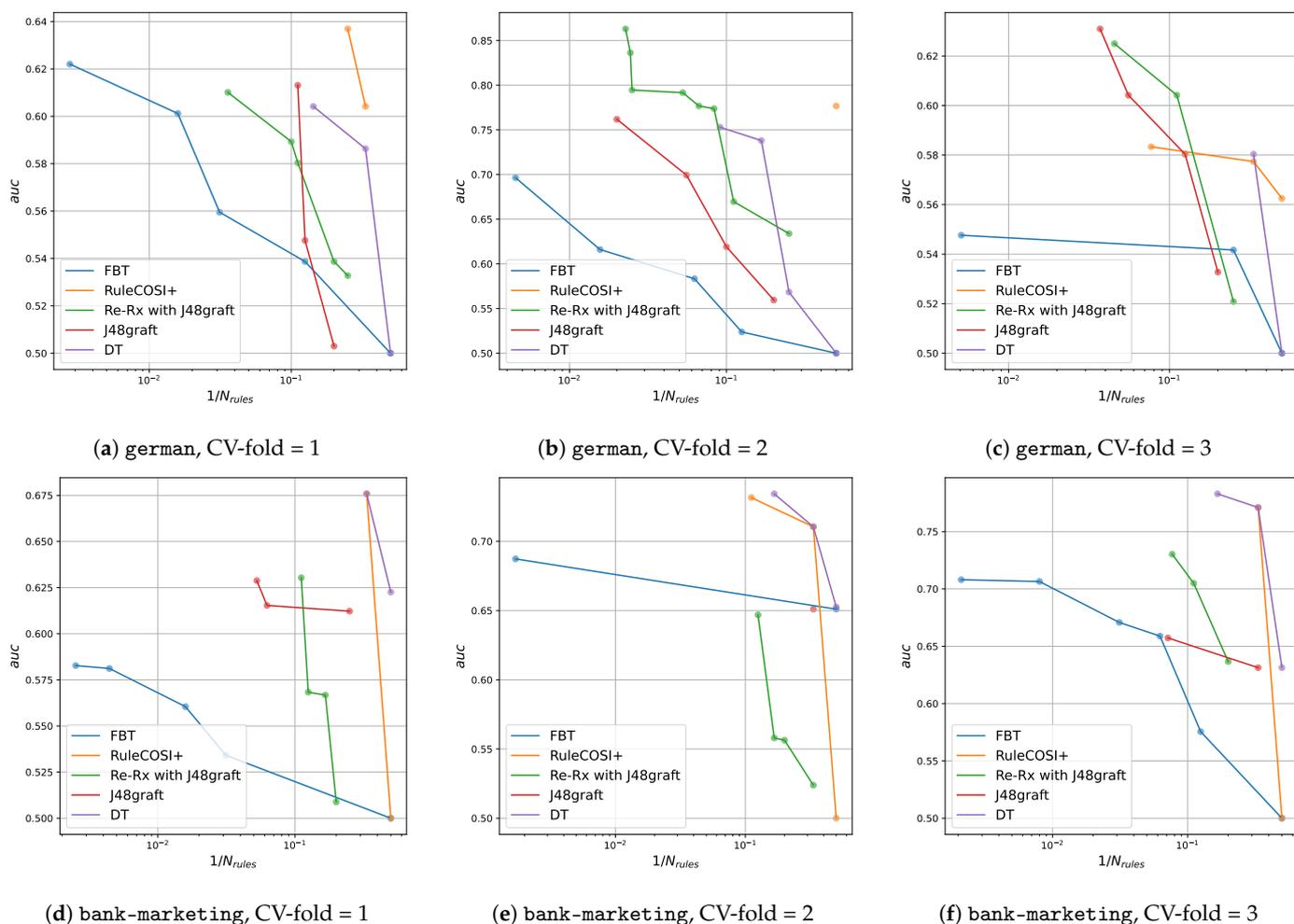


Figure 5. The Pareto fronts for each method obtained by multi-objective optimization using Optuna with seed = 1 and CV-fold = 1, 2, 3 in the german (a–c) and bank-marketing (d–f) datasets.

6.3. Decision Lists vs. Decision Trees

DTs delineate distinct, nonoverlapping regions within the training data, which affects the depth of the tree when representing complex regions. Conversely, decision lists are superior to DTs in that they allow overlapped regions in the feature space to be represented

by different rules, thereby generating more concise rule descriptions for decision boundaries for the classification problem.

On the other hand, when interpreting rules corresponding to an instance, DTs only require tracing the node corresponding to the instance, whereas decision lists require concatenating rules until the instance is classified, which makes the rules more complex. In other words, even if the apparent size of the decision list is small, such as N_{rule} and the *average number of conditions*, it is actually a very complex rule set with less interpretability than DTs. Therefore, if interpretability is important, DTs or DT-based rule sets are the better choice.

6.4. CREP as a Metric of Interpretability

CREP measures the complexity of a rule set based on empirical probabilities and is an intuitive metric of the interpretability of the rule set. CREP can be used to compare the interpretability of both decision list- and DT-based rule sets because it is calculated separately for these models. *RuleCOSI+*, which is a decision list-based rule set that was concluded to have low interpretability in Sections 5.4.1 and 5.4.2, has a high CREP value compared with the other methods in Table 5, showing that the examples and the indicator are consistent. From the above, we can conclude that CREP is an appropriate metric for evaluating the interpretability of rule sets.

6.5. Limitations

In this study, all datasets were used for the evaluation of binary classification. We found the Pareto optimal solution from the Pareto front obtained by multi-objective optimization using Equation (5), but this equation is specialized for binary classification. In multi-class classification, effective results were not obtained using Equation (5). For example, a solution in which there are no rules corresponding to a certain class was sometimes selected as the best solution. To solve this issue, it will be necessary to devise an equation which is specialized for multi-class classification.

Furthermore, it is important to recognize that the size of the datasets used in this study was small to medium. The computational complexity of *RuleCOSI+* and *FBTs* increases exponentially with the size of the data and the number of attributes. This is primarily due to the greedy algorithms employed in these methods. It was not practical to conduct experiments on large datasets within a reasonable time frame; therefore, while comparisons between tree ensembles and *Re-RX with J48graft* were possible, further validation is needed to provide conclusive evidence for the effectiveness of CREP on broader datasets. Such an attempt would improve our understanding of not only the metric's applicability, but also its generalizability across different domains and data types beyond those categorically addressed in this study.

7. Conclusions

In the present study, we compared the tree ensemble approximator with *Re-RX with J48graft* and showed the importance of handling categorical and numerical attributes separately. *RuleCOSI+* obtained a high interpretability on the measure with a small number of rules. However, the rules that are actually used to classify instances are complex and have quite a low interpretability. On the other hand, *Re-RX with J48graft* obtained a low interpretability on the measure with a large number of rules. However, it can handle categorical and numerical attributes separately, has simple rules, and achieves high interpretability, even when the number of rules is large. We newly proposed CREP as a metric for interpretability, which is based on the empirical probability of the rules and measures their complexity. CREP can be used for a fair comparison of decision list- and DT-based rule sets. Furthermore, by using macro-CREP in Equation (4), interpretability can be evaluated appropriately, even for class-imbalanced datasets. Few studies have considered handling categorical and numerical attributes separately, and we believe that this is an important issue for future work. Existing tree ensembles do not distinguish

between categorical and numerical attributes because they prioritize accuracy. Therefore, they cannot be distinguished by tree ensemble approximators such as *RuleCOSI+* and *FBTs*. In the future, we plan to develop a tree ensemble that distinguishes categorical from numeric attributes and serves as an approximator with even better interpretability.

Author Contributions: Conceptualization, S.O. and Y.H.; methodology, S.O.; software, S.O., M.N. and R.F.; validation, S.O.; formal analysis, S.O.; investigation, S.O.; resources, S.O.; data curation, S.O., M.N. and R.F.; writing—original draft preparation, S.O.; writing—review and editing, S.O., M.N., R.F. and Y.H.; visualization, S.O.; supervision, Y.H.; project administration, S.O.; funding acquisition, Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data were presented in the main text. The source code is available at <https://github.com/somaonishi/InterpretableML-Comparisons> (13 March 2024).

Acknowledgments: We sincerely thank the reviewer and editor for supporting our research in all aspects based on their expertise and for their help in preparing the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Dataset Sources

The sources of the datasets used in the experiments are listed in Table A1. We used the OpenML-Python package [58] to obtain the datasets. Only the occupancy data were directly obtained from the UCI repository. We utilized the `datatraining.txt` file for our experiments.

Table A1. Dataset OpenML id and UCI link.

Dataset	ID	URL
heart	53	https://archive.ics.uci.edu/dataset/145/statlog+heart (13 March 2024)
australian	40981	https://archive.ics.uci.edu/dataset/143/statlog+australian+credit+approval (13 March 2024)
mammographic	45557	https://archive.ics.uci.edu/dataset/161/mammographic+mass (13 March 2024)
tic-tac-toe	50	https://archive.ics.uci.edu/dataset/101/tic+tac+toe+endgame (13 March 2024)
german	44096	https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data (13 March 2024)
biodeg	1494	https://archive.ics.uci.edu/dataset/254/qsar+biodegradation (13 March 2024)
banknote	1462	https://archive.ics.uci.edu/dataset/267/banknote+authentication (13 March 2024)
bank-marketing	1558	https://archive.ics.uci.edu/dataset/222/bank+marketing (13 March 2024)
spambase	44	https://archive.ics.uci.edu/dataset/94/spambase (13 March 2024)
occupancy	-	https://archive.ics.uci.edu/dataset/357/occupancy+detection (13 March 2024)

Appendix B. Implementation Details and Hyperparameters

In this section, we provide the implementation details and hyperparameters for each method. See our repository (<https://github.com/somaonishi/InterpretableML-Comparisons> (13 March 2024)) for more details.

Appendix B.1. XGBoost

Implementation. We used *XGBoost* (https://xgboost.readthedocs.io/en/stable/python/python_api.html#xgboost.XGBClassifier (13 March 2024)) for the base tree ensemble for *FBTs* and *RuleCOSI+*. We fixed and did not tune the following hyperparameters:

- $early_stopping_rounds = 10$
- $n_estimators = 250$

In Table A2, we provide the hyperparameter space.

Table A2. XGBoost hyperparameter space.

Parameter	Space
<i>max_depth</i>	UniformInt (1, 10)
η	LogUniform (1×10^{-4} , 1.0)
# Iterations	50

Appendix B.2. FBTs

Implementation. We used the official implementation of FBTs (<https://github.com/sagoume/XGBoostTreeApproximator> (13 March 2024)). We fixed and did not tune the following hyperparameters:

- *min_forest_size* = 10
- *max_number_of_conjunctions* = 1000

In Table A3, we provide the hyperparameter space.

Table A3. FBTs hyperparameter space.

Parameter	Space
<i>max_depth</i>	UniformInt (1, 10)
<i>pruning_method</i>	{auc, None}
# Iterations	50

Appendix B.3. RuleCOSI+

Implementation. We used the official implementation of RuleCOSI+ (<https://github.com/jobregon1212/rulecosi> (13 March 2024)). In Table A4, we provide the hyperparameter space.

Table A4. RuleCOSI+ hyperparameter space.

Parameter	Space
<i>conf_threshold</i>	Uniform (0.0, 0.95)
<i>cov_threshold</i>	Uniform (0.0, 0.5)
<i>c</i>	Uniform (0.1, 0.5)
# Iterations	50

Appendix B.4. Re-RX with J48graft

Implementation. We used the repository (<https://github.com/somaonishi/rerx> (13 March 2024)) that we implemented for *Re-RX with J48graft*. We used *batch_size* = $2^{\lfloor \log(d)+0.5 \rfloor}$, where *d* is the amount of training data. In addition, we fixed and did not tune the following hyperparameters in the MLP:

- *epochs* = 200
- *early_stopping* = 10
- *optimizer* = AdamW [59]

In Table A5, we provide the hyperparameter space of the MLP. We searched for the optimal parameters of the MLP and then the other parameters of *Re-RX with J48graft*. In Table A6, we provide the hyperparameter space for the other parameters of *Re-RX with J48graft*.

Table A5. MLP hyperparameter space.

Parameter	Space
<i>dim</i>	UniformInt (1, 5)
<i>learning_rate</i>	LogUniform (5×10^{-3} , 0.1)
<i>weight_decay</i>	LogUniform (1×10^{-6} , 1×10^{-2})
# Iterations	50

Table A6. Re-RX with J48graft hyperparameter space.

Parameter	Space
<i>j48graft.min_instance</i>	{2, 4, 8, ..., 128}
<i>j48graft.pruning_threshold</i>	Uniform (0.1, 0.5)
<i>pruning_lambda</i>	LogUniform (0.001, 0.25)
δ_1	Uniform (0.05, 0.4)
δ_2	Uniform (0.05, 0.4)
# Iterations	50

Appendix B.5. DT

Implementation. We used scikit-learn's DT (<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html> (13 March 2024)). In Table A7, we provide the hyperparameter space. We used the default hyperparameters of scikit-learn for the other parameters.

Table A7. DT hyperparameter space.

Parameter	Space
<i>max_depth</i>	UniformInt (1, 10)
<i>min_samples_split</i>	Uniform (0.0, 0.5)
<i>min_samples_leaf</i>	Uniform (0.0, 0.5)
# Iterations	100

Appendix B.6. J48graft

Implementation. We used J48graft as implemented in the rerx repository (<https://github.com/somaonishi/rerx/blob/main/rerx/tree/tree.py> (13 March 2024)). Table A8 shows the hyperparameter space.

Table A8. J48graft hyperparameter space.

Parameter	Space
<i>min_instance</i>	{2, 4, 8, ..., 128}
<i>pruning_threshold</i>	Uniform (0.1, 0.5)
# Iterations	100

Appendix C. Results for Other Metrics

We show the results for the other metrics in Tables A9–A12.

Table A9. Results for the average number of conditions.

Dataset	FBTs	RuleCOSI+	Re-RX with J48graft	J48graft	DT
heart	2.68 ± 1.59	1.76 ± 0.63	1.43 ± 0.65	1.96 ± 1.00	1.53 ± 0.89
australian	1.13 ± 0.73	1.25 ± 0.51	0.98 ± 0.29	1.10 ± 0.42	1.03 ± 0.23
mammographic	1.70 ± 0.92	1.05 ± 0.30	1.12 ± 0.26	1.28 ± 0.29	1.11 ± 0.39
tic-tac-toe	3.95 ± 2.85	1.95 ± 0.25	1.27 ± 0.74	1.20 ± 0.63	1.90 ± 1.44
german	3.27 ± 1.87	1.71 ± 0.66	1.85 ± 0.93	2.01 ± 0.69	1.90 ± 0.53
biodeg	3.82 ± 2.64	1.55 ± 0.63	5.10 ± 2.82	10.83 ± 2.64	1.37 ± 0.70
banknote	2.87 ± 1.34	1.54 ± 0.20	2.72 ± 1.03	2.91 ± 0.96	1.81 ± 0.73
bank-marketing	3.00 ± 2.01	1.57 ± 0.30	1.72 ± 0.65	3.02 ± 1.34	1.92 ± 0.70
spambase	1.56 ± 0.94	1.64 ± 0.34	4.30 ± 1.96	15.72 ± 1.78	1.89 ± 0.70
occupancy	1.18 ± 0.52	0.90 ± 0.20	1.05 ± 0.17	3.33 ± 0.00	1.03 ± 0.13
ranking	4.0	2.1	2.5	4.1	2.3

Table A10. Results for precision.

Dataset	FBTs	RuleCOSI+	Re-RX with J48graft	J48graft	DT
heart	71.26 ± 11.18	70.09 ± 11.06	68.07 ± 16.82	69.04 ± 7.27	70.97 ± 8.66
australian	78.78 ± 5.99	78.28 ± 5.10	75.81 ± 17.96	79.91 ± 4.63	79.55 ± 4.17
mammographic	75.10 ± 4.02	75.22 ± 4.22	72.66 ± 8.34	71.21 ± 4.20	75.03 ± 4.64
tic-tac-toe	70.55 ± 15.35	56.39 ± 5.07	54.12 ± 11.70	57.44 ± 6.02	61.32 ± 9.12
german	49.86 ± 19.72	47.61 ± 6.93	49.79 ± 10.35	54.68 ± 7.31	53.08 ± 7.49
biodeg	62.36 ± 7.88	61.56 ± 7.79	72.00 ± 9.73	71.10 ± 8.89	58.79 ± 5.57
banknote	88.72 ± 6.51	90.79 ± 3.75	87.17 ± 11.11	82.96 ± 6.90	85.99 ± 5.84
bank-marketing	42.10 ± 13.51	50.03 ± 5.13	58.60 ± 8.95	54.12 ± 9.86	49.12 ± 17.98
spambase	78.06 ± 8.65	78.48 ± 5.70	80.17 ± 9.17	86.04 ± 3.62	76.75 ± 7.50
occupancy	93.58 ± 3.04	87.20 ± 3.88	94.35 ± 2.08	95.14 ± 0.97	94.95 ± 1.30
ranking	2.8	3.3	3.3	2.5	3.1

Table A11. Results for recall.

Dataset	FBTs	RuleCOSI+	Re-RX with J48graft	J48graft	DT
heart	68.62 ± 13.44	77.46 ± 13.27	62.42 ± 17.81	68.29 ± 9.37	68.58 ± 11.26
australian	93.08 ± 3.35	93.11 ± 6.68	87.82 ± 20.65	92.20 ± 4.65	92.66 ± 3.78
mammographic	79.14 ± 12.84	85.77 ± 6.67	81.64 ± 11.28	74.67 ± 9.63	82.02 ± 8.61
tic-tac-toe	71.13 ± 14.72	92.61 ± 12.00	56.58 ± 12.34	55.72 ± 5.57	64.12 ± 12.08
german	19.85 ± 16.90	55.42 ± 16.79	49.28 ± 21.15	40.32 ± 12.32	46.22 ± 8.96
biodeg	74.35 ± 8.95	75.24 ± 8.98	60.52 ± 11.76	69.94 ± 6.14	72.76 ± 7.57
banknote	93.43 ± 6.25	92.35 ± 4.25	88.70 ± 10.97	90.86 ± 6.19	88.30 ± 4.56
bank-marketing	42.04 ± 18.36	49.25 ± 4.51	18.34 ± 8.62	28.92 ± 6.52	30.80 ± 15.45
spambase	73.48 ± 10.28	86.26 ± 3.62	80.45 ± 11.49	83.01 ± 2.99	82.07 ± 5.03
occupancy	99.30 ± 0.50	99.70 ± 0.31	99.42 ± 0.34	99.33 ± 0.32	99.42 ± 0.29
ranking	2.9	1.1	3.9	3.8	3.0

Table A12. Results for F1-score.

Dataset	FBTs	RuleCOSI+	Re-RX with J48graft	J48graft	DT
heart	68.99 ± 10.30	72.00 ± 7.25	63.98 ± 14.88	68.22 ± 6.31	68.85 ± 7.19
australian	85.14 ± 3.88	84.73 ± 3.07	81.19 ± 18.77	85.42 ± 2.24	85.47 ± 2.27
mammographic	76.35 ± 6.78	79.86 ± 2.82	76.67 ± 8.84	72.64 ± 6.04	77.99 ± 3.89
tic-tac-toe	70.61 ± 14.44	69.54 ± 4.17	54.77 ± 10.63	56.26 ± 4.15	62.52 ± 9.87
german	24.99 ± 15.37	49.91 ± 8.21	46.76 ± 12.02	45.07 ± 8.52	49.02 ± 7.25
biodeg	67.10 ± 4.71	67.04 ± 4.75	64.45 ± 6.84	70.04 ± 5.34	64.59 ± 3.80
banknote	90.79 ± 4.73	91.46 ± 2.69	87.70 ± 10.10	86.54 ± 5.22	86.95 ± 3.60
bank-marketing	38.39 ± 11.63	49.40 ± 3.40	26.87 ± 9.16	36.94 ± 6.35	35.92 ± 15.36
spambase	74.72 ± 4.99	81.93 ± 1.91	79.20 ± 5.49	84.42 ± 2.15	79.00 ± 4.15
occupancy	96.33 ± 1.60	92.98 ± 2.12	96.81 ± 1.11	97.19 ± 0.53	97.13 ± 0.68
ranking	3.0	2.1	4.0	3.0	2.9

References

- Saeed, W.; Omlin, C. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowl. Based Syst.* **2023**, *263*, 110273. [CrossRef]
- Barredo Arrieta, A.; Díaz-Rodríguez, N.; Del Ser, J.; Benetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
- Adadi, A.; Berrada, M. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* **2018**, *6*, 52138–52160. [CrossRef]
- Zhang, Y.; Tiño, P.; Leonardis, A.; Tang, K. A Survey on Neural Network Interpretability. *IEEE Trans. Emerg. Top Comput. Intell.* **2021**, *5*, 726–742. [CrossRef]
- Demajo, L.M.; Vella, V.; Dingli, A. Explainable AI for Interpretable Credit Scoring. In *Computer Science & Information Technology (CS & IT)*; AIRCC Publishing Corporation: Chennai, India, 2020. [CrossRef]
- Petch, J.; Di, S.; Nelson, W. Opening the Black Box: The Promise and Limitations of Explainable Machine Learning in Cardiology. *Can. J. Cardiol.* **2022**, *38*, 204–213. [CrossRef]
- Weber, L.; Lapuschkin, S.; Binder, A.; Samek, W. Beyond explaining: Opportunities and challenges of XAI-based model improvement. *Inf. Fusion* **2023**, *92*, 154–176. [CrossRef]
- Vilone, G.; Longo, L. Classification of Explainable Artificial Intelligence Methods through Their Output Formats. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 615–661. [CrossRef]
- Cabitzza, F.; Campagner, A.; Malgieri, G.; Natali, C.; Schneeberger, D.; Stoeger, K.; Holzinger, A. Quod erat demonstrandum?—Towards a typology of the concept of explanation for the design of explainable AI. *Expert Syst. Appl.* **2023**, *213*, 118888. [CrossRef]
- Deck, L.; Schoeffer, J.; De-Arteaga, M.; Kühl, N. A Critical Survey on Fairness Benefits of XAI. *arXiv* **2023**. [CrossRef]
- Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
- Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
- Mason, L.; Baxter, J.; Bartlett, P.; Frean, M. Boosting Algorithms as Gradient Descent. In *Advances in Neural Information Processing Systems*; Solla, S., Leen, T., Müller, K., Eds.; MIT Press: Cambridge, MA, USA, 1999; Volume 12.
- Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the KDD '16: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [CrossRef]
- Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 4–9 December 2017; NIPS'17; pp. 3149–3157.
- Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased boosting with categorical features. *arXiv* **2019**. [CrossRef]
- Sagi, O.; Rokach, L. Ensemble learning: A survey. *WIREs Data Min. Knowl. Discov.* **2018**, *8*, e1249, [CrossRef]
- Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [CrossRef]
- Longo, L.; Brcic, M.; Cabitzza, F.; Choi, J.; Confalonieri, R.; Ser, J.D.; Guidotti, R.; Hayashi, Y.; Herrera, F.; Holzinger, A.; et al. Explainable Artificial Intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Inf. Fusion* **2024**, *106*, 102301. [CrossRef]

20. Zihni, E.; Madai, V.I.; Livne, M.; Galinovic, I.; Khalil, A.A.; Fiebach, J.B.; Frey, D. Opening the black box of artificial intelligence for clinical decision support: A study predicting stroke outcome. *PLoS ONE* **2020**, *15*, e0231166. [[CrossRef](#)]
21. Yang, C.C. Explainable Artificial Intelligence for Predictive Modeling in Healthcare. *J. Healthc. Inform. Res.* **2022**, *6*, 228–239. [[CrossRef](#)]
22. Carmona, P.; Dwekat, A.; Mardawi, Z. No more black boxes! Explaining the predictions of a machine learning XGBoost classifier algorithm in business failure. *Res. Int. Bus. Financ.* **2022**, *61*, 101649. [[CrossRef](#)]
23. Lipton, Z.C. The Mythos of Model Interpretability. *arXiv* **2017**. [[CrossRef](#)]
24. Qian, H.; Ma, P.; Gao, S.; Song, Y. Soft reordering one-dimensional convolutional neural network for credit scoring. *Knowl. Based Syst.* **2023**, *266*, 110414. [[CrossRef](#)]
25. Mahbooba, B.; Timilsina, M.; Sahal, R.; Serrano, M. Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model. *Complexity* **2021**, *2021*, 6634811. [[CrossRef](#)]
26. Shulman, E.; Wolf, L. Meta Decision Trees for Explainable Recommendation Systems. In Proceedings of the AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, 7–9 February 2020; pp. 365–371. [[CrossRef](#)]
27. Blanco-Justicia, A.; Domingo-Ferrer, J.; Martínez, S.; Sánchez, D. Machine learning explainability via microaggregation and shallow decision trees. *Knowl. Based Syst.* **2020**, *194*, 105532. [[CrossRef](#)]
28. Sachan, S.; Yang, J.B.; Xu, D.L.; Benavides, D.E.; Li, Y. An explainable AI decision-support-system to automate loan underwriting. *Expert Syst. Appl.* **2020**, *144*, 113100. [[CrossRef](#)]
29. Yang, L.H.; Liu, J.; Ye, F.F.; Wang, Y.M.; Nugent, C.; Wang, H.; Martinez, L. Highly explainable cumulative belief rule-based system with effective rule-base modeling and inference scheme. *Knowl. Based Syst.* **2022**, *240*, 107805. [[CrossRef](#)]
30. Li, H.; Wang, Y.; Zhang, S.; Song, Y.; Qu, H. KG4Vis: A Knowledge Graph-Based Approach for Visualization Recommendation. *IEEE Trans. Vis. Comput. Graph.* **2022**, *28*, 195–205. [[CrossRef](#)]
31. Setiono, R.; Baesens, B.; Mues, C. Recursive Neural Network Rule Extraction for Data With Mixed Attributes. *IEEE Trans. Neural Netw.* **2008**, *19*, 299–307. [[CrossRef](#)] [[PubMed](#)]
32. Hayashi, Y.; Nakano, S. Use of a Recursive-Rule eXtraction algorithm with J48graft to achieve highly accurate and concise rule extraction from a large breast cancer dataset. *Inform. Med. Unlocked* **2015**, *1*, 9–16. [[CrossRef](#)]
33. Friedman, J.H.; Popescu, B.E. Predictive learning via rule ensembles. *Ann. Appl. Stat.* **2008**, *2*, 916–954. [[CrossRef](#)]
34. Deng, H. Interpreting tree ensembles with inTrees. *Int. J. Data Sci. Anal.* **2019**, *7*, 277–287. [[CrossRef](#)]
35. Hara, S.; Hayashi, K. Making Tree Ensembles Interpretable: A Bayesian Model Selection Approach. In Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, Playa Blanca, Spain, 9–11 April 2018; Volume 84, pp. 77–85.
36. Sagi, O.; Rokach, L. Explainable decision forest: Transforming a decision forest into an interpretable tree. *Inf. Fusion* **2020**, *61*, 124–138. [[CrossRef](#)]
37. Sagi, O.; Rokach, L. Approximating XGBoost with an interpretable decision tree. *Inf. Sci.* **2021**, *572*, 522–542. [[CrossRef](#)]
38. Obregon, J.; Kim, A.; Jung, J.Y. RuleCOSI: Combination and simplification of production rules from boosted decision trees for imbalanced classification. *Expert Syst. Appl.* **2019**, *126*, 64–82. [[CrossRef](#)]
39. Obregon, J.; Jung, J.Y. RuleCOSI+: Rule extraction for interpreting classification tree ensembles. *Inf. Fusion* **2023**, *89*, 355–381. [[CrossRef](#)]
40. Nauck, D.D. Measuring interpretability in rule-based classification systems. In Proceedings of the 12th IEEE International Conference on Fuzzy Systems, FUZZ'03, St. Louis, MO, USA, 25–28 May 2003; IEEE: Piscataway, NJ, USA, 2003; Volume 1, pp. 196–201.
41. Lakkaraju, H.; Bach, S.H.; Leskovec, J. Interpretable decision sets: A joint framework for description and prediction. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1675–1684.
42. Souza, V.F.; Cicalese, F.; Laber, E.; Molinaro, M. Decision Trees with Short Explainable Rules. In *Advances in Neural Information Processing Systems*; Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A., Eds.; ICurran Associates, Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 12365–12379.
43. Margot, V.; Luta, G. A New Method to Compare the Interpretability of Rule-Based Algorithms. *AI* **2021**, *2*, 621–635. [[CrossRef](#)]
44. Hayashi, Y. Synergy effects between grafting and subdivision in Re-RX with J48graft for the diagnosis of thyroid disease. *Knowl. Based Syst.* **2017**, *131*, 170–182. [[CrossRef](#)]
45. Hayashi, Y.; Oishi, T. High accuracy-priority rule extraction for reconciling accuracy and interpretability in credit scoring. *New Gener. Comput.* **2018**, *36*, 393–418. [[CrossRef](#)]
46. Chakraborty, M.; Biswas, S.K.; Purkayastha, B. Recursive Rule Extraction from NN using Reverse Engineering Technique. *New Gener. Comput.* **2018**, *36*, 119–142. [[CrossRef](#)]
47. Hayashi, Y. Neural network rule extraction by a new ensemble concept and its theoretical and historical background: A review. *Int. J. Comput. Intell. Appl.* **2013**, *12*, 1340006. [[CrossRef](#)]
48. Hayashi, Y. Application of a rule extraction algorithm family based on the Re-RX algorithm to financial credit risk assessment from a Pareto optimal perspective. *Oper. Res. Perspect.* **2016**, *3*, 32–42. [[CrossRef](#)]
49. Hayashi, Y.; Takano, N. One-Dimensional Convolutional Neural Networks with Feature Selection for Highly Concise Rule Extraction from Credit Scoring Datasets with Heterogeneous Attributes. *Electronics* **2020**, *9*, 1318. [[CrossRef](#)]

50. Hayashi, Y. Does Deep Learning Work Well for Categorical Datasets with Mainly Nominal Attributes? *Electronics* **2020**, *9*, 1966. [[CrossRef](#)]
51. Kelly, M.; Longjohn, R.; Nottingham, K. UCI Machine Learning Repository. Available online: <https://archive.ics.uci.edu> (accessed on 13 March 2024).
52. Webb, G.I. Decision Tree Grafting from the All-Tests-but-One Partition. In Proceedings of the IJCAI'99: 16th International Joint Conference on Artificial Intelligence, San Francisco, CA, USA, 31 July–6 August 1999; Volume 2, pp. 702–707.
53. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann: Burlington, MA, USA, 1993.
54. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
55. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019.
56. Bradley, A.P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* **1997**, *30*, 1145–1159. [[CrossRef](#)]
57. Welch, B. The generalization of students problem when several different population variances are involved. *Biometrika* **1947**, *34*, 28–35. [[CrossRef](#)] [[PubMed](#)]
58. Feurer, M.; van Rijn, J.N.; Kadra, A.; Gijbbers, P.; Mallik, N.; Ravi, S.; Müller, A.; Vanschoren, J.; Hutter, F. OpenML-Python: An extensible Python API for OpenML. *J. Mach. Learn. Res.* **2021**, *22*, 1–5.
59. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.