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

Soma Onishi *, Masahiro Nishimura, Ryota Fujimura, Yoichi Hayashi *

Posted Date: 1 February 2024

doi: 10.20944/preprints202401.2212.v1

Keywords: Interpretable machine learning; Explainable artificial intelligence; Rule extraction; Rule-based models; Decision lists; Decision trees; Tree ensemble approximators

Article

Why Do Tree Ensemble Approximators not outperform the Recursive-Rule eXtraction Algorithm?

Soma Onishi *, Masahiro Nishimura, Ryota Fujimura and Yoichi Hayashi

Dept. Computer Science, Meiji University, Kawasaki 214-8571, Japan; nishimura.0622@gmail.com (M.N.); ryotafujimura06@gmail.com (R.F.); hayashiy@cs.meiji.ac.jp (Y.H.)

* Correspondence: somaonishi4@gmail.com

Abstract: Machine learning models are increasingly being used in critical domains, but their complexity, lack of transparency, and poor interpretability remain problematic. Decision trees (DTs) and rule-based approaches are well-known examples of interpretable models, and numerous studies have investigated techniques for approximating tree ensembles using DTs or rule sets; however, tree ensemble approximators do not consider interpretability. These methods are known to generate three main types of rule sets: DT-based, unordered-based, and decision list-based. However, no known metric has been devised to distinguish and compare these rule sets. Therefore, the present study proposes an interpretability metric to allow comparisons of interpretability between different rule sets, such as decision list- and DT-based rule sets, and investigates the interpretability of the rules generated by the tree ensemble approximators. To provide new insights into the reasons why decision list-based and inspired classifiers do not work well for categorical datasets consisting of mainly nominal attributes, we compare objective metrics and rule sets generated by the tree ensemble approximators and the *Recursive-Rule eXtraction algorithm (Re-RX) with J48graft*. The results indicated that *Re-RX with J48graft* can handle categorical and numerical attributes separately, has simple rules, and achieves high interpretability, even when the number of rules is large.

Keywords: interpretable machine learning; explainable artificial intelligence; rule extraction; rule-based models; decision lists; decision trees; tree ensemble approximators

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].

Rudin [11] 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. In the field of health care, for example, it is not sufficient for a medical diagnosis model simply to be accurate; it must also be transparent to health professionals who use the output to make decisions about a given patient [6,12,13]. 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. 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 [14].

Tree ensembles are often used for tabular data. Bagging [15] and random forests (RFs) [16] are known as independent ensembles, whereas gradient boosting machines (GBMs) [17] such as *XGBoost* [18], *LightGBM* [19], and *CatBoost* [20] are known as dependent ensembles. Tree ensembles are extensively utilized in academic and research contexts. They are also applied in practical scenarios across a wide array of domains [21]. Recently, these models have been effective in many classification

 doi:10.20944/preprints202401.2212.v1

tasks. In fact, these models are used by most winners of Kaggle competitions [1]. 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.

Decision trees (DTs), rule-based approaches, and knowledge graph-based approaches are widely used as examples of interpretable models [22–29]. Techniques for approximating tree ensembles with DTs or rule sets have also been investigated [30–36]. 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 [28].

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. In the present study, we newly propose an interpretability metric, *Complexity of Rules with Empirical Probability* (*CREP*), to allow 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* [29].

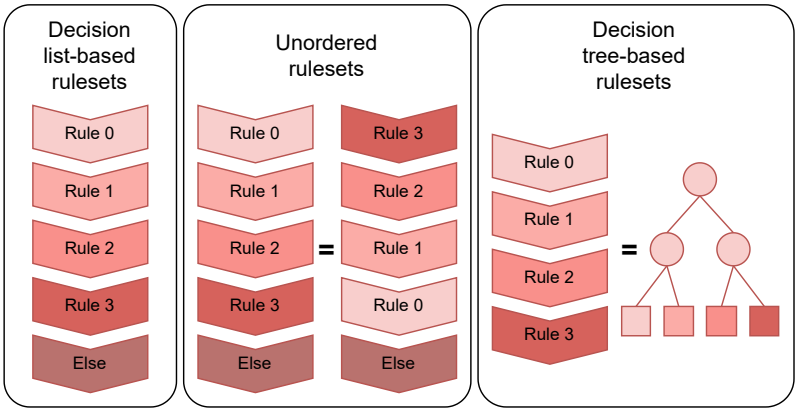


Figure 1. Concepts of the three types of rule sets. Decision list-based rule sets classify instances by sequentially referencing rules from top to bottom. Unordered rule sets classify instances by referencing rules in any order. Decision tree-based rule sets are variable to decision trees and differ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

2. Related Work

The *Re-RX* algorithm [28] 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* [29] is the extended version of *Re-RX*. Numerous studies have conducted research on *Re-RX* [37–42]. *RuleFit* [30] 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* [31] 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* [32]

[1] 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.

 doi:10.20944/preprints202401.2212.v1

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 [33], *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+*) [36], 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.* [35], *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 [43].

3. Materials and Methods

3.1. Datasets

We used 10 diverse datasets from the University of California, Irvine, Machine Learning Repository [44] to compare each method. The details of the datasets are shown in Table 1. 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 $10\times$ in the 10×10-fold cross-validation (CV) scheme described in Section 3.3.3.

Table 1. Dataset properties

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* [45] as simple DT-based methods. *J48graft* is a grafted (pruned or unpruned) *C4.5* [46] DT. *DT* generates a binary tree, while *J48graft*, capable of handling categorical attributes, generates a m-ary tree.

 doi:10.20944/preprints202401.2212.v1

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 [2] [3].

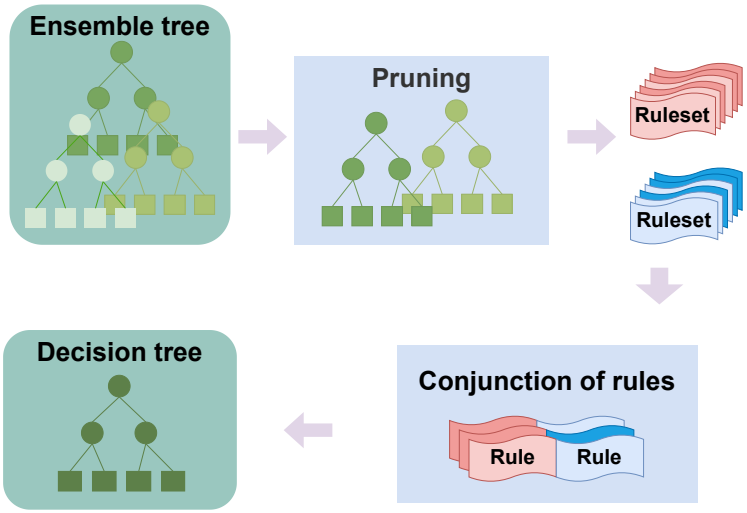


Figure 2. Overview of *FBTs*. (a) Fit the ensemble tree, (b) Pruning: remove trees that do not improve the accuracy from the ensemble tree, (c) Convert the tree to rules, (d) Conjunction of rules: generate the conjunction set by gradually merging the conjunction sets of the base trees into a single set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

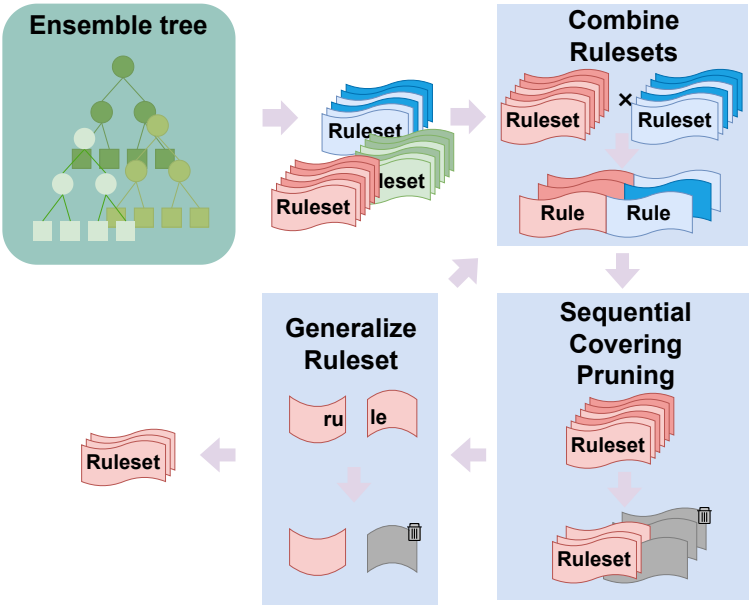


Figure 3. Overview of *RuleCOSI+*. (a) Fit the ensemble tree, (b) Convert the tree to rule sets, (c) Combine rule sets: greedily verify all combinations and determine the rules to adopt and create a new rule set, (d) Sequential covering pruning: simplify the rule set, (e) Generalize rule set: remove 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.

---

2 https://github.com/sagyome/XGBoostTreeApproximator
3 https://github.com/jobregon1212/rulecosi

 doi:10.20944/preprints202401.2212.v1

As a rule-based method, we used *Re-RX with J48graft*, an overview of which is shown in Figure 4. In learning a multilayer perceptron (*MLP*) in *Re-RX with J48graft*, we apply one-hot encoding [4] to categorical attributes to enable efficient learning. With the application of the one-hot encoding to categorical attributes, we modify 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 $\overline{\mathcal{C}}_i$ 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: $\forall j \in \{j = 0, ..., |\overline{\mathcal{C}}_i| - 1\}$, if $w_{j,*} = 0$, then $\mathcal{D} \leftarrow \mathcal{D} \setminus \{c_i\}$.

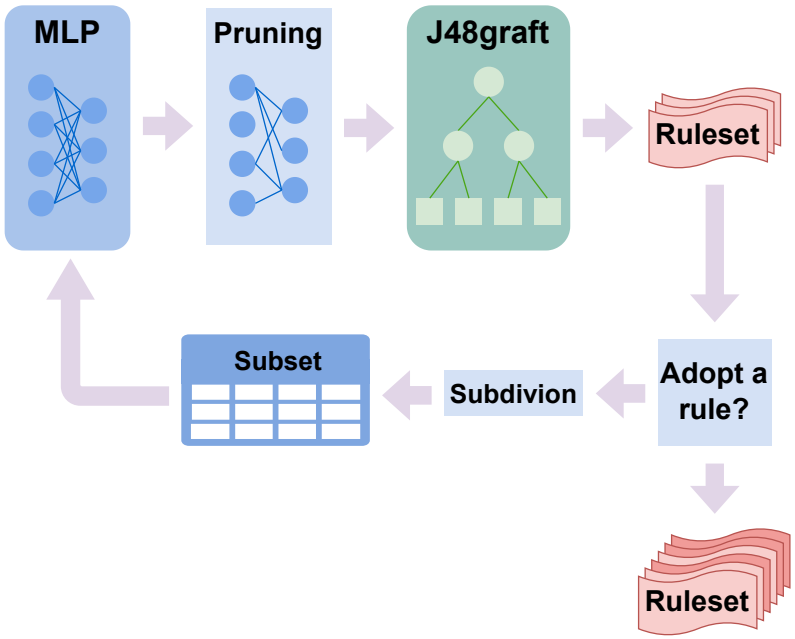


Figure 4. Overview of *Re-RX with J48graft*. (a) Fit a multilayer perceptron, (b) Pruning: reduce the number of attributes, (c) Fit *J48graft*, (d) Convert the tree to a rule set, (e) Adopt a rule?: select a rule to adopt as is, (f) Subdivision: recursive (a–e) for a rule that is not adopted using data and attributes not included in the rule.

All fitted methods are converted to the `RuleSet` [5] module implemented by Obregon and Jung [36].

3.3. Experimental Design

In this section, we present the experimental design for comparing methods.

3.3.1. 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 the numeric attributes, we applied standardization only to the training and prediction of the *MLP* in *Re-RX with J48graft*.

---

[4] We used OneHotEncoder from scikit-learn [47]
[5] https://github.com/jobregon1212/rulecosi/blob/master/rulecosi/rules.py

 doi:10.20944/preprints202401.2212.v1

3.3.2. 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 newly propose an 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:

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

where N_{cond_r} is the number of conditions in rule r, and $cov(r, \mathcal{D})$ is the coverage of rule r on 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. 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, 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 Eq. (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 calcu-lating *CREP* for each class. We redefine as follows:

$$\begin{aligned} \text{micro}-CREP &= \sum_{r \in R} N_{cond_r} \cdot cov(r, \mathcal{D}) \\ CREP_c &= \sum_{r \in R_c} N_{cond_r} \cdot cov(r, \mathcal{D}) \\ \text{macro}-CREP &= \frac{1}{|C|} \sum_{c \in C} CREP_c \end{aligned}$$

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*.

3.3.3. Model Evaluation and Hyperparameter Optimization

We performed the experiment using a stratified 10×10-fold CV [6] scheme, which is a 10-fold CV repeated 10 times. In each cv-fold, we performed hyperparameter optimization using Optuna [48]. 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, other hyperparameters were optimized using multi-objective optimization [7] to maximize both classification performance and interpretability simultaneously. For both *DT* and *J48graft*, the first step was skipped

[6] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html
[7] https://optuna.readthedocs.io/en/stable/tutorial/20_recipes/002_multi_objective.html

 doi:10.20944/preprints202401.2212.v1

because these methods do not have a base model. We used the area under the receiver operating characteristics curve (AUC-ROC) [49] for the classification performance metric, and the inverse of N_{rules} ($1/N_{rules}$) for the interpretability metric. See Section A for details on the hyperparameters for 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:

$$k = -\log_2 N_{rules} + \alpha \cdot AUC \tag{2}$$

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 4 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. Results

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

4.1. Classification Results

The classification results are presented in Table 2. *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 2. Results for classification performance ($\mu \pm \sigma$). All metrics are reported as AUCs. 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 [50])). 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	*FBTs*	*RuleCOSI+*	*Re-RX with J48graft*	*J48graft*	*DT*
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

 doi:10.20944/preprints202401.2212.v1

4.2. Interpretability Results

The interpretability results and the number of rules in N_{rules} and *CREP* are presented in Tables 3 and 4. *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 4.4 and 5.4. *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.

Table 3. Results for the number of rules.

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

Table 4. Results for CREP.

dataset	*FBTs*	*RuleCOSI+*	*Re-RX with J48graft*	*J48graft*	*DT*
`heart`	2.68 ± 1.59	5.01 ± 3.00	$\mathbf{1.25 \pm 0.48}$	1.47 ± 0.58	1.53 ± 0.89
`australian`	1.13 ± 0.72	2.62 ± 1.22	$\mathbf{0.97 \pm 0.25}$	1.05 ± 0.21	$\mathbf{1.03 \pm 0.23}$
`mammographic`	1.70 ± 0.92	2.18 ± 0.80	$\mathbf{1.03 \pm 0.14}$	1.27 ± 0.24	$\mathbf{1.10 \pm 0.37}$
`tic-tac-toe`	3.94 ± 2.84	5.62 ± 1.44	$\mathbf{1.16 \pm 0.56}$	$\mathbf{1.11 \pm 0.36}$	1.83 ± 1.32
`german`	3.27 ± 1.88	4.42 ± 2.50	$\mathbf{1.43 \pm 0.48}$	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	$\mathbf{1.38 \pm 0.75}$
`banknote`	2.88 ± 1.35	3.90 ± 0.93	2.13 ± 0.64	2.33 ± 0.55	$\mathbf{1.82 \pm 0.75}$
`bank-marketing`	3.01 ± 2.03	3.18 ± 0.58	$\mathbf{1.29 \pm 0.44}$	1.65 ± 0.28	2.19 ± 0.90
`spambase`	$\mathbf{1.56 \pm 0.94}$	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	$\mathbf{1.02 \pm 0.06}$	1.89 ± 0.01	$\mathbf{1.03 \pm 0.15}$
ranking	3.5	4.7	1.5	3.1	2.2

 doi:10.20944/preprints202401.2212.v1

4.3. Summary of Comparative Experiments

Table 5 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 5. Summary of classification and interpretability performance ($\mu \pm \sigma$) across all datasets.

method	**AUC**	$\mathbf{N_{rules}}$	**CREP**
FBTs	78.04 ± 13.05	24.52 ± 85.43	2.52 ± 1.99
RuleCOSI+	$\mathbf{80.22 \pm 10.20}$	$\mathbf{2.97 \pm 1.63}$	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	$\mathbf{1.57 \pm 0.85}$

4.4. 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 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" ≤ 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 $x \neq a$.

4.4.1. `bank-marketing`

Table 6 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:

$$r_{\text{class 1}} := \neg r_0 \wedge \neg r_1 \rightarrow [1] \tag{3}$$

By contrast, the rule set generated by *Re-RX with J48graft* consists exclusively of rules based on categorical attributes, resulting in relatively high interpretability. Furthermore, it is simpler by post-processing, as shown in Table 7. The rule set generated by *DT*, while containing numerical attributes, was composed of simple rules.

 doi:10.20944/preprints202401.2212.v1

Table 6. Rule sets generated from the `bank-marketing` dataset.

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

Table 7. Post-processed rule set generated from the `bank-marketing` dataset in *Re-RX with J48graft*.

	Re-RX with J48graft	coverage
r_1	$(V16 \neq success) \rightarrow [0]$	0.971
r_3	$(V16 = success) \wedge (V5 = yes) \rightarrow [0]$	0.0
r_4	$(V16 = success) \wedge (V5 \neq yes) \rightarrow [1]$	0.029

4.4.2. `german`

Table 8 shows an example of the rule set generated by each method. As in section 4.4.1, the rule set generated by *RuleCOSI+* contained a complex mixture of numerical and categorical attributes, and the rule for class 1 was Eq. (3), 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 a numerical attribute *duration* to improve the accuracy of the rule for *checking_status* $= 0$<=X<200. Also, as shown in Table 9, the rule set can be simpler, as in Section 4.4.1. The rule set generated by *DT*, while containing numerical attributes, was composed of simple rules.

 doi:10.20944/preprints202401.2212.v1

Table 8. 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	$(checking\_status = >=200) \rightarrow [0]$	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	$(checking\_status = 0<=X<200) \wedge (duration \leq 26) \rightarrow [0]$	0.19
r_9	$(checking\_status = 0<=X<200) \wedge (duration > 26) \rightarrow [1]$	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 9. 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	$(checking\_status = >=200) \rightarrow [0]$	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	$(checking\_status = 0<=X<200) \wedge (duration \leq 26) \rightarrow [0]$	0.19
r_6	$(checking\_status = 0<=X<200) \wedge (duration > 26) \rightarrow [1]$	0.079

5. Discussion

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

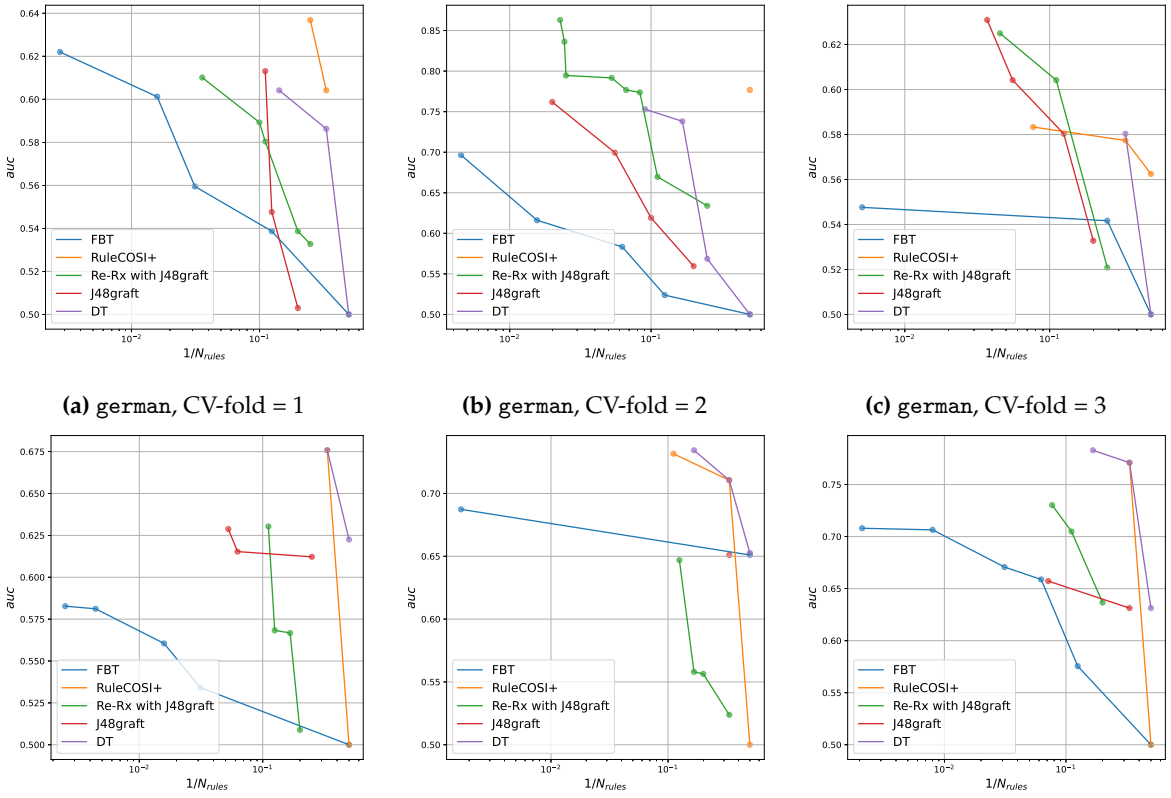
Many methods that have been proposed to enhance interpretability cannot handle categorical and numerical attributes separately. If we could adequately express a rule using only categorical attributes, the use of numerical attributes would reduce interpretability. Conditions for categorical attributes are easy to understand intuitively because they are categorized into a finite group. On the other hand, 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 $checking\_status \in \{no\ checking, <0, 0<=X<200, >=200\}$, whereas the conditions for the attribute $credit\_amount \in \mathbb{N}$ are infinite. In the division of the condition $credit\_amount \leq 10841.5$ in r_2 in *RuleCOSI+* in Table 8, it is difficult to understand why the value 10841.5 was chosen. Furthermore, rules that contain many numerical attribute conditions make it more difficult

 doi:10.20944/preprints202401.2212.v1

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.

5.2. 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 Eq. (2). However, the Pareto optimal solution selected by Eq. (2) 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.



(a) `german`, CV-fold = 1 **(b)** `german`, CV-fold = 2 **(c)** `german`, CV-fold = 3

(d) `bank-marketing`, CV-fold = 1 **(e)** `bank-marketing`, CV-fold = 2 **(f)** `bank-marketing`, CV-fold = 3

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.

5.3. Decision Lists v.s. 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

 doi:10.20944/preprints202401.2212.v1

rule set with less interpretability than DTs. Therefore, if interpretability is important, DTs or DT-based rule sets are the better choice.

5.4. 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 4.4.1 and 4.4.2, has a high *CREP* value compared with the other methods in Table 4, observing that examples and the indicator are consistent. From the above, we conclude that *CREP* is an appropriate metric for evaluating the interpretability of rule sets.

5.5. Limitation

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 Eq. (2), but this equation is specialized for binary classification. In multi-class classification, effective results were not obtained using Eq. (2). 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 specialized for multi-class classification.

6. 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 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 low interpretability. On the other hand, *Re-RX with J48graft* obtained 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*, 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.

Acknowledgments: The authors thank FORTE Science Communications [8] for English language editing.

[8] https://www.forte-science.co.jp/

 doi:10.20944/preprints202401.2212.v1

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

Appendix A. Implementation details and hyperparameters

In this section, we provide the implementation details and hyperparameters for each method. See our repository [9] for more details.

Appendix A.1. XGBoost

Implementation. We used *XGBoost* [10] 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 A1, we provide the hyperparameter space.

Table A1. *XGBoost* hyperparameter space.

Parameter	Space
max_depth	UniformInt(1, 10)
η	LogUniform(1e-4,1.0)
# Iterations	50

Appendix A.2. FBTs

Implementation. We used the official implementation of *FBTs* [11]. We fixed and did not tune the following hyperparameters:

- $min\_forest\_size = 10$
- $max\_number\_of\_conjunctions = 1000$

In Table A2, we provide the hyperparameter space.

Table A2. *FBTs* hyperparameter space.

Parameter	Space
max_depth	UniformInt(1, 10)
$pruning\_method$	{auc, None}
# Iterations	50

Appendix A.3. RuleCOSI+

Implementation. We used the official implementation of *RuleCOSI+* [12]. In Table A3, we provide the hyperparameter space.

[9] https://github.com/somaonishi/InterpretableML-Comparisons
[10] https://xgboost.readthedocs.io/en/stable/python/python_api.html#xgboost.XGBClassifier
[11] https://github.com/sagyome/XGBoostTreeApproximator
[12] https://github.com/jobregon1212/rulecosi

 doi:10.20944/preprints202401.2212.v1

Table A3. *RuleCOSI+* hyperparameter space.

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

Appendix A.4. Re-RX with J48graft

Implementation. We used the repository [13] that we implemented for *Re-RX with J48graft*. We used $batch\_size = 2^{\lfloor log(d)+0.5 \rfloor}$, where d is the number of training data. In addition, we fixed and did not tune the following hyperparameters in the *MLP*:

- $epochs = 200$
- $early\_stopping = 10$
- $optimizer = AdamW$ [51]

In Table A4, 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 A5, we provide the hyperparameter space for the other parameters of *Re-RX with J48graft*.

Table A4. *MLP* hyperparameter space.

Parameter	Space
dim	UniformInt(1, 5)
learning_rate	LogUniform(5e-3, 0.1)
weight_decay	LogUniform(1e-6, 1e-2)
# Iterations	50

Table A5. *Re-RX with J48graft* hyperparameter space.

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

Appendix A.5. DT

Implementation. We used scikit-learn's DT [14]. In Table A6, we provide the hyperparameter space. We used the default hyperparameters of *scikit-learn* for the other parameters.

[13] https://github.com/somaonishi/rerx
[14] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

 doi:10.20944/preprints202401.2212.v1

Table A6. *DT* hyperparameter space.

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

Appendix A.6. J48graft

Implementation. We used *J48graft* as implemented in the `rerx` repository [15]. Table A7 shows the hyperparameter space.

Table A7. *J48graft* hyperparameter space.

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

Appendix B. Results for other metrics

We show the results for the other metrics in Tables A8–A11.

Table A8. 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

[15] https://github.com/somaonishi/rerx/blob/main/rerx/tree/tree.py

 doi:10.20944/preprints202401.2212.v1

Table A9. 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 A10. 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

 doi:10.20944/preprints202401.2212.v1

Table A11. 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

1. Saeed, W.; Omlin, C. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowl. Based Syst.* **2023**, *263*, 110273. doi:10.1016/j.knosys.2023.110273.
2. Arrieta, A.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; Chatila, R.; Herrera, F. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. doi:10.1016/j.inffus.2019.12.012.
3. Adadi, A.; Berrada, M. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* **2018**, *6*, 52138–52160. doi:10.1109/ACCESS.2018.2870052.
4. Zhang, Y.; Tiňo, P.; Leonardis, A.; Tang, K. A Survey on Neural Network Interpretability. *IEEE Trans. Emerg. Top Comput. Intell.* **2021**, *5*, 726–742. doi:10.1109/TETCI.2021.3100641.
5. Demajo, L.M.; Vella, V.; Dingli, A. Explainable AI for Interpretable Credit Scoring. Computer Science & Information Technology (CS & IT). AIRCC Publishing Corporation, 2020. doi:10.5121/csit.2020.101516.
6. 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. doi:10.1016/j.cjca.2021.09.004.
7. Weber, L.; Lapuschkin, S.; Binder, A.; Samek, W. Beyond explaining: Opportunities and challenges of XAI-based model improvement. *Inf. Fusion* **2023**, *92*, 154–176.
8. Vilone, G.; Longo, L. Classification of Explainable Artificial Intelligence Methods through Their Output Formats. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 615–661.
9. Cabitza, 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.
10. Deck, L.; Schoeffer, J.; De-Arteaga, M.; Kühl, N. A Critical Survey on Fairness Benefits of XAI, 2023, [arXiv:cs.AI/2310.13007].
11. 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.
12. 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*. doi:10.1371/journal.pone.0231166.
13. Yang, C.C. Explainable Artificial Intelligence for Predictive Modeling in Healthcare. *JOURNAL OF HEALTHCARE INFORMATICS RESEARCH* **2022**, *6*, 228–239. doi:10.1007/s41666-022-00114-1.
14. Lipton, Z.C. The Mythos of Model Interpretability, 2017, [arXiv:cs.LG/1606.03490].
15. Breiman, L. Bagging predictors. *MACHINE LEARNING* **1996**, *24*, 123–140. doi:10.1007/BF00058655.

16. Breiman, L. Random forests. *MACHINE LEARNING* **2001**, *45*, 5–32. doi:10.1023/A:1010933404324.
17. Mason, L.; Baxter, J.; Bartlett, P.; Frean, M. Boosting Algorithms as Gradient Descent. Advances in Neural Information Processing Systems; Solla, S.; Leen, T.; Müller, K., Eds. MIT Press, 1999, Vol. 12.
18. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; Association for Computing Machinery: New York, NY, USA, 2016; KDD '16, pp. 785–794. doi:10.1145/2939672.2939785.
19. 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. Proceedings of the 31st International Conference on Neural Information Processing Systems; Curran Associates Inc.: Red Hook, NY, USA, 2017; NIPS'17, pp. 3149–3157.
20. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: unbiased boosting with categorical features, 2019, [arXiv:cs.LG/1706.09516].
21. Sagi, O.; Rokach, L. Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery* **2018**, *8*, e1249, [https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1249]. doi:10.1002/widm.1249.
22. 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*, 1–11.
23. Shulman, E.; Wolf, L. Meta Decision Trees for Explainable Recommendation Systems. Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society; Association for Computing Machinery: New York, NY, USA, 2020; AIES '20, pp. 365–371. doi:10.1145/3375627.3375876.
24. 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. doi:10.1016/j.knosys.2020.105532.
25. 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*. doi:10.1016/j.eswa.2019.113100.
26. 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*. doi:10.1016/j.knosys.2021.107805.
27. Li, H.; Wang, Y.; Zhang, S.; Song, Y.; Qu, H. KG4Vis: A Knowledge Graph-Based Approach for Visualization Recommendation. *IEEE Transactions on Visualization and Computer Graphics* **2022**, *28*, 195–205. doi:10.1109/tvcg.2021.3114863.
28. Setiono, R.; Baesens, B.; Mues, C. Recursive Neural Network Rule Extraction for Data With Mixed Attributes. *IEEE Transactions on Neural Networks* **2008**, *19*, 299–307. doi:10.1109/TNN.2007.908641.
29. 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. *Informatics in Medicine Unlocked* **2015**, *1*, 9–16. doi:10.1016/j.imu.2015.12.002.
30. Friedman, J.H.; Popescu, B.E. Predictive learning via rule ensembles. *The Annals of Applied Statistics* **2008**, *2*, 916 – 954. doi:10.1214/07-AOAS148.
31. Deng, H. Interpreting tree ensembles with inTrees. *International Journal of Data Science and Analytics* **2019**, *7*, 277–287. doi:10.1007/s41060-018-0144-8.
32. Hara, S.; Hayashi, K. Making Tree Ensembles Interpretable: A Bayesian Model Selection Approach. Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics; Storkey, A.; Perez-Cruz, F., Eds. PMLR, 2018, Vol. 84, *Proceedings of Machine Learning Research*, pp. 77–85.
33. Sagi, O.; Rokach, L. Explainable decision forest: Transforming a decision forest into an interpretable tree. *Inf. Fusion* **2020**, *61*, 124–138. doi:10.1016/j.inffus.2020.03.013.
34. Sagi, O.; Rokach, L. Approximating XGBoost with an interpretable decision tree. *Information Sciences* **2021**, *572*, 522–542. doi:10.1016/j.ins.2021.05.055.
35. 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. doi:10.1016/j.eswa.2019.02.012.
36. Obregon, J.; Jung, J.Y. RuleCOSI+: Rule extraction for interpreting classification tree ensembles. *Inf. Fusion* **2023**, *89*, 355–381. doi:10.1016/j.inffus.2022.08.021.
37. 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. doi:10.1016/j.knosys.2017.06.011.

38. Hayashi, Y.; Oishi, T. High accuracy-priority rule extraction for reconciling accuracy and interpretability in credit scoring. *New Generation Computing* **2018**, *36*, 393–418.
39. Chakraborty, M.; Biswas, S.K.; Purkayastha, B. Recursive Rule Extraction from NN using Reverse Engineering Technique. *New Generation Computing* **2018**, *36*, 119–142. doi:10.1007/s00354-018-0031-9.
40. Hayashi, Y. NEURAL NETWORK RULE EXTRACTION BY A NEW ENSEMBLE CONCEPT AND ITS THEORETICAL AND HISTORICAL BACKGROUND: A REVIEW. *International Journal of Computational Intelligence and Applications* **2013**, *12*, 1340006. doi:10.1142/S1469026813400063.
41. 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. *Operations Research Perspectives* **2016**, *3*, 32–42. doi:10.1016/j.orp.2016.08.001.
42. 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*. doi:10.3390/electronics9081318.
43. Hayashi, Y. Does Deep Learning Work Well for Categorical Datasets with Mainly Nominal Attributes? *Electronics* **2020**, *9*.
44. Kelly, M.; Longjohn, R.; Nottingham, K. UCI Machine Learning Repository. https://archive.ics.uci.edu.
45. Webb, G.I. Decision Tree Grafting from the All-Tests-but-One Partition. Proceedings of the 16th International Joint Conference on Artificial Intelligence; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1999; Vol. 2, *IJCAI'99*, pp. 702–707.
46. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann, 1993.
47. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
48. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.
49. Bradley, A.P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* **1997**, *30*, 1145–1159. doi:10.1016/S0031-3203(96)00142-2.
50. WELCH, B. THE GENERALIZATION OF STUDENTS PROBLEM WHEN SEVERAL DIFFERENT POPULATION VARIANCES ARE INVOLVED. *BIOMETRIKA* **1947**, *34*, 28–35. doi:10.2307/2332510.
51. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. International Conference on Learning Representations, 2019.