Supplementary material related to the paper: **Forest-ORE**: Mining Optimal Rule Ensemble to interpret Random Forest models

Maissae Haddouchi[a,*], Abdelaziz Berrado[a]

[a]*AMIPS Research Team, Ecole Mohammadia d'Ingénieurs (EMI), Mohammed V University in Rabat, Morocco*

Purpose of the Document

This document serves as supplementary material for the paper titled "Forest-ORE: Mining Optimal Rule Ensemble to Interpret Random Forest Models." It contains additional details, analyses, methods, and results discussed in the main paper. Readers are encouraged to consult this document for further explanations and examples that complement the content of the paper.

Contents

*Corresponding author

Email addresses: maissaehaddouchi@research.emi.ac.ma (Maissae Haddouchi), berrado@emi.ac.ma (Abdelaziz Berrado)

1. Details related to the section Forest-ORE Framework

1.1. Rule Preselection algorithm and outputs

Algorithm 1 outlines the steps followed in the preselection stage

---

Algorithm 1 Rule Preselection

Let RFR denote RF rules and Data the training dataset. Let min_conf and min_class_cov denote the lower limits for rules confidence and class coverage. Let max_len and max_simil denote the upper limits for rule length and similarity. Let PSR denote the resulting Preselected Rules, and $PSRS$ denote similar rules removed.

Input: $RFR, Data, min\_conf, min\_class\_cov, max\_len, max\_simil$

Output: $PSR, PSRS$

Initialization: $PSR \leftarrow RFR$ and $PSRS \leftarrow null$

$RR \leftarrow redund(PSR)$ ▷$redund(D)$: function extracting redundant rules in a set D

$PSR \leftarrow PSR - RR$

$PSR \leftarrow \{R \in PSR \mid len(R) \leq max\_len\}$

for each $R \in PSR$ **do**

Compute $R_{conf} = conf(R)$ and $R_{class\_cov} = class\_cov(R)$

end for

$PSR \leftarrow \{R \in PSR \mid R_{class\_cov} \geq min\_class\_cov \text{ and } R_{conf} \geq min\_conf\}$

Compute Mat_{simil}, the $k * k$ matrix of rules' pairwise similarity

for each $row\ i\ in\ Mat_{simil}$ **do**

$S_{simil_i} \leftarrow \{R \in PSR \mid Mat_{simil}[i, j] \geq max\_simil, \quad j = 1...k\}$

end for

$S \leftarrow \{S_{simil_i} : i = 1...k\}$

for each $S_{simil} \in S$ **do**

$Best_{conf} \leftarrow \{argmax(conf(R)) \mid R \in S_{simil}\}$

$Best_{cov} \leftarrow \{argmax(cov(R)) \mid R \in Best_{conf}\}$

$Best_{att} \leftarrow \{argmin(att\_nbr(R)) \mid R \in Best_{cov}\}$

$R_{best} \leftarrow \{argmin(lev\_nbr(R)) \mid R \in Best_{att}\}$

$PSRS \leftarrow PSRS \cup \{S_{simil} - \{R_{best}\}\}$

$PSR \leftarrow PSR - \{S_{simil} - \{R_{best}\}\}$

end for

Table 1 illustrates a row in the RuleMetrics data frame, and Table 2 illustrates a row in CovOk/CovNok data frames.

Table 1: Illustration of a row in the RuleMetrics data frame

Id	Conf.	Cov.	Att. nbr	Lev. nbr	Att. nbr_S	Lev. nbr_S	Att.	Ypred	Condition
1	0.986	0.594	2	6	0.286	0.122	V6,V7	2	X[,6] in {1,2} & X[,7] in {1,2,3,5}

Table 2: Illustration of a row in CovOk/CovNok data frames

R1	R2	R3	R4	R5	R6	R7	R8	...	Rm
1	0	0	0	1	0	0	1	...	0

1.2. Rule Enrichment algorithm and outputs

Let $MetaR$ represent the nxm rule matrix where each line refers to an instance, and each column refers to a rule from the collection of preselected rules. This matrix links, for each instance, the rules covering it, as illustrated in table 3. Line 1, for example, means that the conditions of the rules R_1, R_2, and R_4 are applied to instance 1. In the Metarules methodology, each line from $MetaR$ is mapped to a transaction where each rule is considered an item. As example, line 1 from table 3 is mapped to the transaction: $\{R_1, R_2, R_4\}$. The Association Rule Mining (ARM) [1] approach is then applied to the n transactions in order to find the one-way association rules. The one-way association rules takes the format $R_i \rightarrow R_j$ and is called a metarule. The quality of the containment in the Metarules approach is monitored through the ARM confidence and support. The support of a metarule is computed by dividing the number of instances satisfying R_i and R_j by the total number of instances. The confidence is computed by dividing the number of instances satisfying R_i and R_j by the number of instances where R_i is applied. To ensure a quasi-total containment, we fix the ARM minimum confidence to a value near to 1. Accordingly, we guarantee that the R_i is not a generic rule applied to other regions different from the one delimited by the rule j. As for the support, it is recommended to set it to a value that avoids over-fitting.

We first extract rules interacting with the selected rules via the metarules approach. We then select the ones providing new information to each selected rule. Since each rule R_j is a combination of (variable, values) pairs that defines a subregion of the attribute space, the idea is to search rules defining the same subregion and using a set of variables different from the ones used in the rule R_j. From these rules, we choose the best ones based on the rate of their intersections with the rule R_j, confidence values, coverage values, and the number of attributes used. We define the intersection between the rules R_i and R_j "$intersect(R_i, R_j)$" as the size

of the set of instances covered by the rules R_i and R_j divided by the size the set of instances covered by the rule R_j.
Algorithm 2 describes the "Rule enrichment" steps. The dataframe of complementary rules takes the format illustrated in Table 4.

Algorithm 2 Rule Enrichment

Let PSR denote the Preselected rules, $PSRS$ the similar rules removed (ref Algorithm 1), SR the Selected Rules, and Data the training dataset. Let CR denote the set of complementary rules that will be returned. Let fix minimum confidence and minimum support for association rules mining ($arm\_minconf$, arm_minsup).

Input: $PSR, PSRS, SR, arm\_minconf, arm\_minsup$
Output: CR
Initialization: $CR \leftarrow null, PSR \leftarrow PSR \cup PSRS$
Compute $MetaR$, the Metarules matrix ▷rows represent instances, and columns represent PSR rules (see Table 3)
Convert $MetaR$ to transactions $Meta_{trans}$
Apply association rule mining to $Meta_{trans}$ to discover the rules applied to the subspaces covered by SR. Each discovered metarule is constrained to be an expression of the form $R_i \rightarrow R_j$ where $R_i \in PSR$ and $R_j \in SR$.
for each $R_j \in$ SR **do**
  Extract Att_{R_j} the list of attibutes used in R_j
  Extract $Meta_{R_j}$ the set of R_j metarules
  $RM \leftarrow \{R \in Meta_{R_j} \mid Att_R = Att_{R_j}\}$
  $Meta_{R_j} \leftarrow Meta_{R_j} - RM$
  Extract $U_{att} = unique(\{Att_R \mid R \in Meta_{R_j}\})$
  for each $Att \in U_{att}$ **do**
    $Rules_{Att} \leftarrow \{R \in Meta_{R_j} \mid Att_R = Att\}$
    $Best_{intersect} \leftarrow \{argmax(intersect(R, R_j)) \mid R \in Rules_{Att}\}$
    $Best_{conf} \leftarrow \{argmax(conf(R)) \mid R \in Best_{intersect}\}$
    $Best_{cov} \leftarrow \{argmax(cov(R)) \mid R \in Best_{conf}\}$
    $R_s \leftarrow \{argmin(att\_nbr(R)) \mid R \in Best_{cov}\}$
    $CR \leftarrow CR \cup \{R_s\}$
  end for
end for

Table 3: Simplified illustration of the Metarule matrix. Rows represent data instances, and columns represent rules.

	$\mathbf{R_1}$	$\mathbf{R_2}$	$\mathbf{R_3}$	$\mathbf{R_4}$	$\mathbf{R_5}$	$\mathbf{R_6}$	$\mathbf{R_7}$
1	R_1	R_2		R_4			
2		R_2		R_4			
3		R_2	R_3		R_5		

Table 4 illustrates a containment between rule (id=102) and rule (id 97). The 1^{st} column is reserved for the selected rules IDs. The 2^{nd} column reports the complementary rules IDs. The 3^{rd} column is reserved for the conditions of the rules in the 2^{nd} column. The 5^{th} column reports the containment rate of the 2^{nd} column rule in the 1^{st} column rule. The remaining columns relate the characteristics of the 2^{nd} column rules. In this table, each bold line represents the characteristics of a selected rule.

Table 4: Illustration of the complementary rules dataframe

ID SR	ID Rule	Condition	Ypred	Intersect	Att.	Att. nbr	Lev. nbr	Conf.	Cov.
102	**102**	**X[,1] in {A2,A4} & X[,2] in {B2}**	**1**	**1.00**	**V1,V2**	**2**	**3**	**1**	**0.25**
102	97	X[,3] in {C2} & X[,5] in {E1,E4}	1	0.96	V3,V5	2	3	1	0.19

1.3. Parameter tuning in Forest-ORE

Parameter tuning in Forest-ORE allows users to adjust various settings to optimize the balance between interpretability, accuracy, and complexity based on their specific needs. Key tunable parameters include:

- Objective Function Weights: The objective function includes five weights that control different aspects of the rule ensemble:

    1. The first weight controls the size of the rule ensemble, where a higher value encourages smaller sets of rules.
    2. The second weight controls the cumulative prediction error, with a higher value favoring rules with higher confidence.
    3. The third weight controls the cumulative coverage of the rule ensemble, with a higher value promoting rules that cover a broader range of instances.

4. The fourth weight controls the cumulative rule lengths, while the fifth controls the cumulative sum of levels used in the rules. Higher values for these weights encourage shorter rules with fewer variables and levels. By default, equal weights (1) are assigned to the first three components, while the last two have lower weights (0.1 and 0.05). This setup prioritizes optimizing rule size, accuracy, and coverage while also favoring shorter, simpler rules. Users can adjust these weights according to their specific problem.

- Maximum Rule Length: The maximum length of rules can be limited during the preselection stage to reduce the number of input rules in the optimization stage, with a default setting of 6. This allows users to balance model complexity with computational time.

- Loss Accuracy Parameter: This parameter controls the upper bound for the allowable loss in accuracy compared to the RF accuracy, with a default value of 0.01. A higher value relaxes this constraint, allowing for greater accuracy loss, which can help prevent overfitting.

- Coverage Constraint Parameter: This parameter sets the upper bound for the allowable loss in overall coverage compared to the preselected rules' coverage (initially set to 1). The default value is 0.05, which controls the proportion of data instances not covered. A higher value increases the risk of overfitting and may result in a larger number of rules. Users can adjust this parameter to balance capturing more data patterns, ensuring robustness, and simplifying the rule set.

- Overlap Constraint Parameters: These parameters control the overall and instance-level overlaps between rules, with default values of 0.5 and 3, respectively. These constraints are crucial for ensuring that the rules remain distinct and that the model provides clear explanations for predictions. However, strict overlap constraints may be less suitable for datasets where classes are not well-separated. Adjusting these constraints can help tailor the model's interpretability to the desired or achievable level of rule overlap.

2. Details related to Experiments

2.1. Illustration of rules overlaps using the Upset method

In order to visualize the overlaps between the sets defined by the selected rules, one can use approaches available in literature, such as VennEuler [2], Upset [3] and Radial sets [4]. Venn diagram is the most intuitive tool for visualizing intersections and looking at what is shared between groups. However, as the number of sets increases, Venn diagram becomes complex and hard to interpret [5]. Upset and Radial sets are more suitable for multiple overlapping sets. Figure 1 uses the Upset method to visualize rules overlaps on the XOR dataset. The horizontal bar chart on the bottom left side shows the distribution of the instances over the rules. Their color reflects the rule classification. The vertical bar chart on the top right side shows the distribution of the instances depending on the combination of rules to which they belong. Their color reflects the true class of the instances. In the example of the XOR dataset, there is no overlap between the rule sets. We provide in the following the example of the Mushroom dataset to show an example of overlaps.

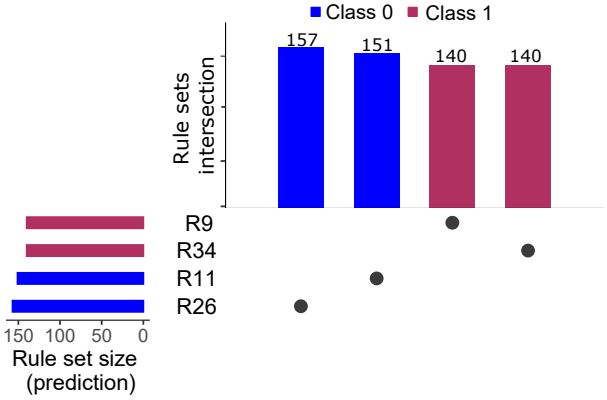

Figure 1: Upset visualization on XOR rule sets

We give the example of the Mushroom dataset (Figure 2) to show an example of overlaps. The mushroom dataset includes 8124 instances and 23 descriptive variables. Table 5 lists the selected rules resulting from applying Forest-ORE to the Mushroom dataset. In Figure 2, the 2nd vertical bar chart represents the size of instances applied exclusively to R82, the 3rd to instances applied exclusively to R43, and the 5th to the intersection between R43 and R82. This figure also shows that the "else" rule classifies instances as poisonous (horizontal bar chart), whereas, in fact, some of them are edible (vertical bar chart).

Table 5: Selected rules provided by applying “Forest-ORE” to the Mushroom dataset

id	confidence	coverage	class_coverage	att. nbr	lev. Nbr	cond	Ypred	att.
43	1	0.45	0.86	4	18	X[,5] in {a,l,n} & X[,15] in {g,n,o,p,w} & X[,18] in {n,o} & X[,20] in {b,h,k,n,o,r,u,y}	’e’	V5,V15,V18,V20
44	1	0.47	0.97	1	6	X[,5] in {c,f,m,p,s,y}	’p’	V5
82	1	0.48	0.93	3	16	X[,8] in {b} & X[,15] in {b,e,g,n,o,p,w,y} & X[,20] in {b,k,n,o,u,w,y}	’e’	V8,V15,V20

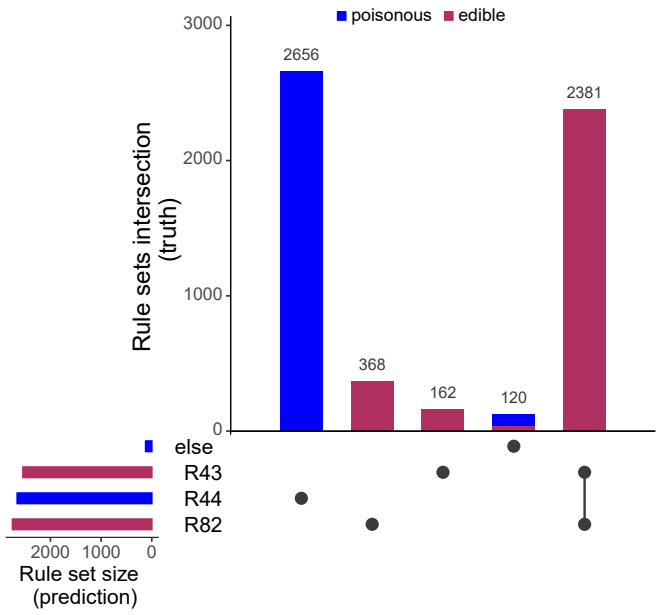

Figure 2: Illustration of rules overlaps representation using the Upset method. Upset provides an efficient way to visualize intersections in the multiple sets explaining the Mushroom dataset. In this case, the overlap is between rules 82 et 43 and concerns the edible class.

2.2. Cohen’s Kappa results on benchmark datasets

Cohen’s Kappa [6] is a statistical measure used to compare multi-class and imbalanced class data. It is known as a measure of reliability. It informs about how well a classifier is performing compared to the performance of a classifier that simply guesses at random according to the frequency of each class. Cohen’s kappa values ranges from -1 to 1. According to Landis and Koch [7], values less than 0 indicate that the classifier is useless, while values ranging between 0 and

0.20 qualify its usefulness as slight, those between 0.21 and 0.40 as fair, those between 0.41 and 0.60 as moderate, those between 0.61 and 0.80 as substantial, and those between 0.81 and 1 as almost perfect. Figure 3 shows the variation of the average a kappa on the testing sets over the benchmark datasets. The corresponding Wilcoxon signed-rank test results are provided in Table 6.

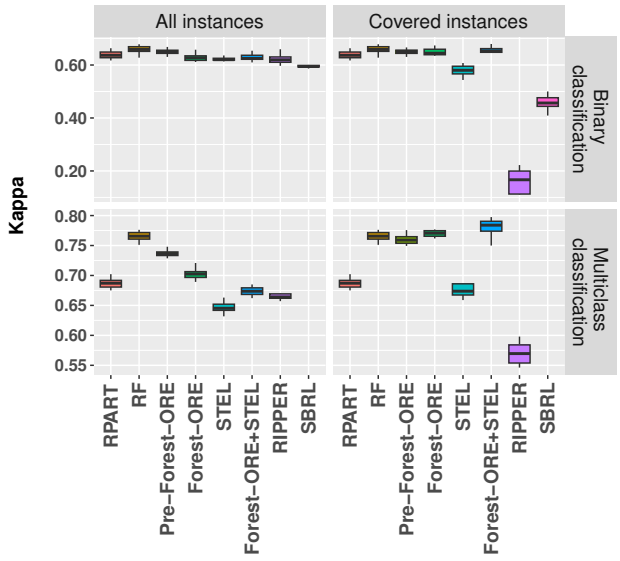

Figure 3: Boxplots of the mean Kappa on the testing sets across the benchmark datasets. Left: Global Kappa. Right: Kappa for instances covered by the classifier rules.

Table 6: Wilcoxon signed-rank test results for Kappa on the testing sets. Left: Global scores. Right: Scores for instances covered by the classifier rules.

	All instances				Covered instances			
	Method	**Wins**	**Ties**	**Losses**	**Method**	**Wins**	**Ties**	**Losses**
Binary classification	RF	7	0	0	RF	5	2	0
	Pre-Forest-ORE	6	0	1	Forest-ORE	4	3	0
	Forest-ORE	1	4	2	Forest-ORE+STEL	4	3	0
	Forest-ORE+STEL	1	4	2	Pre-Forest-ORE	4	2	1
	RPART	1	4	2	RPART	2	1	4
	STEL	1	4	2	STEL	2	1	4
	RIPPER	1	4	2	SBRL	1	0	6
	SBRL	0	0	7	RIPPER	0	0	7
Multi. classification	RF	6	0	0	Forest-ORE+STEL	5	1	0
	Pre-Forest-ORE	5	0	1	RF	4	2	0
	Forest-ORE	3	1	2	Forest-ORE	3	2	1
	RPART	2	2	2	Pre-Forest-ORE	3	1	2
	Forest-ORE+STEL	2	1	3	RPART	1	1	4
	STEL	0	1	5	STEL	1	1	4
	RIPPER	0	1	5	RIPPER	0	0	6

2.3. Results per dataset

Tables 7, 8, 9 , 10, 11 , and 12 report the average accuracy and coverage metrics and their ranking per dataset, and per classification issue. Table 13 reports the p-value of the Friedman test on these average results.

Table 7: Average accuracy metrics and their ranking per dataset on binary classification (all instances)

	RPART	STEL	Pre-Forest-ORE	Forest-ORE	Forest-ORE+STEL	RF	RIPPER	SBRL
APPENDICITIS	0.863 (4)	0.847 (2.5)	0.875 (7)	0.866 (5.5)	0.866 (5.5)	0.878 (8)	0.847 (2.5)	0.819 (1)
BANKNOTE	0.966 (7)	0.964 (5)	0.956 (2)	0.963 (3.5)	0.963 (3.5)	0.970 (8)	0.965 (6)	0.945 (1)
BRCANCER	0.941 (3)	0.952 (6)	0.959 (7)	0.948 (5)	0.945 (4)	0.974 (8)	0.939 (2)	0.937 (1)
CRYOTHERAPY	0.707 (1)	0.715 (2)	0.756 (5.5)	0.741 (4)	0.763 (7)	0.767 (8)	0.730 (3)	0.756 (5.5)
HABERMAN	0.733 (8)	0.727 (4)	0.729 (5)	0.731 (7)	0.730 (6)	0.721 (3)	0.720 (2)	0.716 (1)
HEART	0.806 (6)	0.786 (3.5)	0.833 (8)	0.786 (3.5)	0.778 (1)	0.828 (7)	0.779 (2)	0.798 (5)
HYPOTHYROID	0.981 (6)	0.981 (6)	0.978 (3)	0.976 (1)	0.977 (2)	0.983 (8)	0.980 (4)	0.981 (6)
INDIAN	0.709 (5)	0.707 (3.5)	0.717 (7.5)	0.707 (3.5)	0.700 (2)	0.717 (7.5)	0.693 (1)	0.715 (6)
ION	0.899 (5)	0.916 (6)	0.924 (7)	0.895 (4)	0.894 (3)	0.927 (8)	0.882 (2)	0.802 (1)
MAMMOGRAPHIC	0.824 (7)	0.820 (4)	0.825 (8)	0.821 (5)	0.818 (3)	0.822 (6)	0.800 (2)	0.761 (1)
MUSHROOM	0.995 (2)	0.996 (4)	0.998 (5.5)	0.995 (2)	0.995 (2)	0.998 (5.5)	1.000 (7.5)	1.000 (7.5)
MUTAGENESIS	0.685 (8)	0.670 (3)	0.674 (4)	0.677 (6)	0.676 (5)	0.683 (7)	0.669 (2)	0.665 (1)
PHONEME	0.761 (2)	0.755 (1)	0.771 (3)	0.772 (4)	0.774 (5)	0.787 (6)	0.814 (8)	0.803 (7)
TICTACTOE	0.908 (1)	0.985 (7)	0.914 (2)	0.974 (4.5)	0.974 (4.5)	0.932 (3)	0.976 (6)	1.000 (8)
VOTE	0.953 (2)	0.946 (1)	0.958 (5)	0.960 (6.5)	0.960 (6.5)	0.968 (8)	0.957 (3.5)	0.957 (3.5)
WDBC	0.944 (4.5)	0.947 (6)	0.960 (7)	0.944 (4.5)	0.942 (2)	0.964 (8)	0.943 (3)	0.909 (1)
WILT	0.951 (5)	0.946 (2.5)	0.952 (6.5)	0.946 (2.5)	0.946 (2.5)	0.952 (6.5)	0.946 (2.5)	0.953 (8)
WISCONSIN	0.937 (1)	0.953 (5.5)	0.963 (7)	0.950 (3)	0.951 (4)	0.971 (8)	0.940 (2)	0.953 (5.5)
XOR	1.000 (5)	1.000 (5)	1.000 (5)	1.000 (5)	1.000 (5)	0.904 (1)	1.000 (5)	1.000 (5)
Total Rank	**82.5**	**77.5**	**105**	**80**	**73.5**	**124.5**	**66**	**75**

Table 8: Average accuracy metrics and their ranking per dataset on binary classification (covered instances)

	RPART	STEL	Pre-Forest-ORE	Forest-ORE	Forest-ORE+STEL	RF	RIPPER	SBRL
APPENDICITIS	0.863 (4)	0.847 (3)	0.875 (7)	0.870 (5)	0.873 (6)	0.878 (8)	0.695 (1)	0.835 (2)
BANKNOTE	0.966 (4)	0.955 (2)	0.956 (3)	0.967 (5.5)	0.967 (5.5)	0.970 (8)	0.968 (7)	0.934 (1)
BRCANCER	0.941 (2)	0.960 (4)	0.959 (3)	0.966 (6)	0.963 (5)	0.974 (8)	0.914 (1)	0.968 (7)
CRYOTHERAPY	0.707 (1)	0.720 (2)	0.756 (4)	0.753 (3)	0.783 (6)	0.767 (5)	0.809 (7)	0.900 (8)
HABERMAN	0.733 (6)	0.729 (4.5)	0.729 (4.5)	0.740 (7)	0.744 (8)	0.721 (3)	0.460 (1)	0.712 (2)
HEART	0.806 (4)	0.788 (2)	0.833 (7)	0.811 (5)	0.802 (3)	0.828 (6)	0.771 (1)	0.846 (8)
HYPOTHYROID	0.981 (4)	0.986 (7)	0.979 (3)	0.985 (6)	0.989 (8)	0.983 (5)	0.818 (2)	0.776 (1)
INDIAN	0.709 (3)	0.735 (8)	0.720 (6)	0.717 (4.5)	0.727 (7)	0.717 (4.5)	0.431 (1)	0.579 (2)
ION	0.899 (3)	0.921 (4.5)	0.924 (7)	0.922 (6)	0.921 (4.5)	0.927 (8)	0.840 (1)	0.885 (2)
MAMMOGRAPHIC	0.824 (4)	0.840 (8)	0.825 (5)	0.831 (6)	0.837 (7)	0.822 (3)	0.780 (2)	0.769 (1)
MUSHROOM	0.995 (1.5)	0.995 (1.5)	0.998 (3.5)	1.000 (6.5)	1.000 (6.5)	0.998 (3.5)	1.000 (6.5)	1.000 (6.5)
MUTAGENESIS	0.685 (8)	0.673 (3)	0.674 (4)	0.679 (6)	0.678 (5)	0.683 (7)	0.539 (1)	0.660 (2)
PHONEME	0.761 (2)	0.817 (8)	0.772 (3)	0.781 (4)	0.786 (5)	0.787 (6)	0.683 (1)	0.802 (7)
TICTACTOE	0.908 (1)	1.000 (7.5)	0.914 (2)	0.993 (5)	0.995 (6)	0.932 (3)	0.972 (4)	1.000 (7.5)
VOTE	0.953 (2)	0.963 (5)	0.958 (3.5)	0.964 (6.5)	0.964 (6.5)	0.968 (8)	0.930 (1)	0.958 (3.5)
WDBC	0.944 (3)	0.951 (4)	0.960 (7)	0.957 (6)	0.955 (5)	0.964 (8)	0.915 (1)	0.929 (2)
WILT	0.951 (3.5)	0.961 (7)	0.953 (6)	0.951 (3.5)	0.972 (8)	0.952 (5)	0.466 (1)	0.936 (2)
WISCONSIN	0.937 (2)	0.958 (3)	0.963 (4)	0.965 (5)	0.966 (6)	0.971 (7)	0.897 (1)	0.977 (8)
XOR	1.000 (5)	1.000 (5)	1.000 (5)	1.000 (5)	1.000 (5)	0.904 (1)	1.000 (5)	1.000 (5)
Total Rank	**63**	**89**	**87.5**	**101.5**	**113**	**107**	**45.5**	**77.5**

Table 9: Average coverage metrics and their ranking per dataset on binary classification

	RPART	STEL	Pre-Forest-ORE	Forest-ORE	Forest-ORE+STEL	RF	RIPPER	SBRL
APPENDICITIS	1.000 (7)	0.872 (3)	1.000 (7)	0.984 (5)	0.941 (4)	1.000 (7)	0.103 (1)	0.512 (2)
BANKNOTE	1.000 (7)	0.449 (2)	1.000 (7)	0.980 (4.5)	0.980 (4.5)	1.000 (7)	0.437 (1)	0.778 (3)
BRCANCER	1.000 (7)	0.918 (3)	1.000 (7)	0.956 (4.5)	0.956 (4.5)	1.000 (7)	0.344 (1)	0.791 (2)
CRYOTHERAPY	1.000 (7)	0.885 (3)	1.000 (7)	0.930 (5)	0.900 (4)	1.000 (7)	0.367 (1)	0.415 (2)
HABERMAN	1.000 (7)	0.931 (3)	1.000 (7)	0.975 (5)	0.936 (4)	1.000 (7)	0.113 (1)	0.570 (2)
HEART	1.000 (7)	0.978 (5)	1.000 (7)	0.914 (3.5)	0.914 (3.5)	1.000 (7)	0.416 (1)	0.764 (2)
HYPOTHYROID	1.000 (7.5)	0.983 (5)	0.998 (6)	0.972 (4)	0.962 (3)	1.000 (7.5)	0.043 (1)	0.059 (2)
INDIAN	1.000 (7.5)	0.847 (3)	0.987 (6)	0.937 (5)	0.908 (4)	1.000 (7.5)	0.131 (1)	0.589 (2)
ION	1.000 (7)	0.942 (3)	1.000 (7)	0.948 (5)	0.947 (4)	1.000 (7)	0.366 (1)	0.542 (2)
MAMMOGRAPHIC	1.000 (7)	0.926 (3)	1.000 (7)	0.965 (5)	0.938 (4)	1.000 (7)	0.472 (1)	0.811 (2)
MUSHROOM	1.000 (7)	0.657 (2)	1.000 (7)	0.982 (4.5)	0.982 (4.5)	1.000 (7)	0.482 (1)	0.808 (3)
MUTAGENESIS	1.000 (7)	0.974 (3)	1.000 (7)	0.979 (5)	0.976 (4)	1.000 (7)	0.066 (1)	0.756 (2)
PHONEME	1.000 (7.5)	0.657 (2)	0.997 (6)	0.971 (5)	0.952 (4)	1.000 (7.5)	0.295 (1)	0.938 (3)
TICTACTOE	1.000 (7)	0.333 (1)	1.000 (7)	0.963 (5)	0.961 (4)	1.000 (7)	0.344 (2)	0.653 (3)
VOTE	1.000 (7)	0.752 (3)	1.000 (7)	0.981 (4.5)	0.981 (4.5)	1.000 (7)	0.398 (1)	0.637 (2)
WDBC	1.000 (7)	0.948 (3)	1.000 (7)	0.972 (4.5)	0.972 (4.5)	1.000 (7)	0.383 (1)	0.623 (2)
WILT	1.000 (7.5)	0.755 (3)	0.999 (6)	0.977 (5)	0.902 (4)	1.000 (7.5)	0.009 (1)	0.715 (2)
WISCONSIN	1.000 (7)	0.922 (3)	1.000 (7)	0.961 (4.5)	0.961 (4.5)	1.000 (7)	0.369 (1)	0.701 (2)
XOR	1.000 (6.5)	0.486 (2)	1.000 (6.5)	1.000 (6.5)	0.524 (3)	1.000 (6.5)	0.476 (1)	0.753 (4)
Total Rank	**134.5**	**55**	**128.5**	**91**	**76.5**	**134.5**	**20**	**44**

Table 10: Average accuracy metrics and their ranking per dataset on multiclass classification (all instances)

	RPART	STEL	Pre-Forest-ORE	Forest-ORE	Forest-ORE+STEL	RF	RIPPER
ANNEAL	0.885 (3)	0.861 (1)	0.908 (7)	0.887 (4)	0.894 (5)	0.907 (6)	0.870 (2)
AUTO	0.626 (1.5)	0.698 (3)	0.752 (6)	0.718 (5)	0.716 (4)	0.782 (7)	0.626 (1.5)
BANANA	0.743 (6)	0.717 (3)	0.711 (1)	0.740 (4.5)	0.740 (4.5)	0.748 (7)	0.714 (2)
CAR	0.935 (4)	0.859 (2)	0.943 (6)	0.936 (5)	0.918 (3)	0.955 (7)	0.846 (1)
DERMA	0.932 (2)	0.939 (4)	0.973 (6)	0.935 (3)	0.942 (5)	0.979 (7)	0.881 (1)
ECOLI	0.791 (4)	0.788 (3)	0.793 (5)	0.776 (2)	0.775 (1)	0.798 (6)	0.805 (7)
GLASS	0.628 (2)	0.675 (5)	0.708 (6)	0.657 (3)	0.658 (4)	0.729 (7)	0.609 (1)
IRIS	0.938 (1.5)	0.949 (6)	0.940 (3)	0.942 (4.5)	0.942 (4.5)	0.938 (1.5)	0.964 (7)
NEWTHYROID	0.871 (1)	0.891 (4)	0.920 (7)	0.892 (5)	0.889 (3)	0.914 (6)	0.886 (2)
PAGEBLOCKS	0.950 (5)	0.942 (1.5)	0.953 (6)	0.944 (3)	0.942 (1.5)	0.958 (7)	0.947 (4)
TEXTURE	0.781 (4)	0.549 (1)	0.791 (5)	0.772 (3)	0.619 (2)	0.922 (7)	0.900 (6)
THYROID	0.949 (6)	0.944 (3.5)	0.947 (5)	0.944 (3.5)	0.942 (2)	0.952 (7)	0.940 (1)
TITATNIC	0.786 (4.5)	0.784 (1.5)	0.785 (3)	0.788 (6.5)	0.788 (6.5)	0.786 (4.5)	0.784 (1.5)
VERTEBRAL	0.845 (7)	0.815 (3)	0.832 (6)	0.809 (2)	0.817 (4)	0.829 (5)	0.717 (1)
VOWEL	0.499 (2.5)	0.480 (1)	0.749 (6)	0.724 (5)	0.499 (2.5)	0.840 (7)	0.628 (4)
WINE	0.877 (1)	0.925 (2.5)	0.966 (6)	0.942 (5)	0.940 (4)	0.979 (7)	0.925 (2.5)
WIRELESS	0.933 (2)	0.923 (1)	0.955 (5)	0.940 (3)	0.941 (4)	0.956 (6)	0.961 (7)
Total Rank	**57**	**46**	**89**	**67**	**60.5**	**105**	**51.5**

Table 11: Average accuracy metrics and their ranking per dataset on multiclass classification (covered instances)

	RPART	STEL	Pre-Forest-ORE	Forest-ORE	Forest-ORE+STEL	RF	RIPPER
ANNEAL	0.885 (3)	0.865 (2)	0.911 (5)	0.913 (6)	0.916 (7)	0.907 (4)	0.734 (1)
AUTO	0.626 (1)	0.729 (3)	0.752 (4)	0.785 (7)	0.78 (5)	0.782 (6)	0.641 (2)
BANANA	0.743 (5)	0.788 (7)	0.712 (2)	0.741 (3)	0.742 (4)	0.748 (6)	0.696 (1)
CAR	0.935 (3)	0.818 (2)	0.943 (4)	0.969 (6)	0.975 (7)	0.955 (5)	0.629 (1)
DERMA	0.932 (2)	0.957 (3.5)	0.973 (6)	0.957 (3.5)	0.969 (5)	0.979 (7)	0.882 (1)
ECOLI	0.791 (2)	0.808 (7)	0.793 (3)	0.803 (5.5)	0.803 (5.5)	0.798 (4)	0.726 (1)
GLASS	0.628 (1)	0.699 (3.5)	0.708 (5)	0.699 (3.5)	0.711 (6)	0.729 (7)	0.639 (2)
IRIS	0.938 (1.5)	0.944 (4)	0.94 (3)	0.946 (5)	0.952 (6)	0.938 (1.5)	0.986 (7)
NEWTHYROID	0.871 (1)	0.901 (3)	0.92 (7)	0.909 (5)	0.907 (4)	0.914 (6)	0.875 (2)
PAGEBLOCKS	0.95 (2.5)	0.95 (2.5)	0.955 (4)	0.962 (6)	0.964 (7)	0.958 (5)	0.77 (1)
TEXTURE	0.781 (1)	0.801 (2)	0.916 (4)	0.929 (6)	0.984 (7)	0.922 (5)	0.911 (3)
THYROID	0.949 (2)	0.964 (4)	0.979 (5)	0.987 (6)	0.995 (7)	0.952 (3)	0.742 (1)
TITATNIC	0.786 (2.5)	0.81 (6)	0.785 (1)	0.787 (4)	0.79 (5)	0.786 (2.5)	0.89 (7)
VERTEBRAL	0.845 (7)	0.83 (4)	0.832 (5)	0.828 (2)	0.838 (6)	0.829 (3)	0.585 (1)
VOWEL	0.499 (1)	0.543 (2)	0.755 (4)	0.799 (6)	0.79 (5)	0.84 (7)	0.683 (3)
WINE	0.877 (1)	0.947 (3)	0.966 (6)	0.959 (4.5)	0.959 (4.5)	0.979 (7)	0.939 (2)
WIRELESS	0.933 (1)	0.94 (2)	0.955 (5)	0.952 (3)	0.954 (4)	0.956 (6)	0.959 (7)
Total Rank	**37.5**	**60.5**	**73**	**82**	**95**	**85**	**43**

Table 12: Average coverage metrics and their ranking per dataset on multiclass classification

	RPART	STEL	Pre-Forest-ORE	Forest-ORE	Forest-ORE+STEL	RF	RIPPER
ANNEAL	1.000 (6.5)	0.948 (2)	0.993 (5)	0.960 (4)	0.952 (3)	1.000 (6.5)	0.235 (1)
AUTO	1.000 (6)	0.921 (4)	1.000 (6)	0.892 (3)	0.887 (2)	1.000 (6)	0.665 (1)
BANANA	1.000 (6.5)	0.695 (2)	0.999 (5)	0.982 (4)	0.971 (3)	1.000 (6.5)	0.416 (1)
CAR	1.000 (6)	0.765 (2)	1.000 (6)	0.952 (4)	0.899 (3)	1.000 (6)	0.360 (1)
DERMA	1.000 (6.5)	0.905 (2)	0.999 (5)	0.947 (4)	0.941 (3)	1.000 (6.5)	0.672 (1)
ECOLI	1.000 (6)	0.949 (2)	1.000 (6)	0.952 (4)	0.950 (3)	1.000 (6)	0.565 (1)
GLASS	1.000 (6)	0.849 (2)	1.000 (6)	0.917 (4)	0.898 (3)	1.000 (6)	0.580 (1)
IRIS	1.000 (6)	0.809 (2)	1.000 (6)	0.982 (4)	0.900 (3)	1.000 (6)	0.640 (1)
NEWTHYROID	1.000 (6)	0.926 (2)	1.000 (6)	0.977 (4)	0.937 (3)	1.000 (6)	0.262 (1)
PAGEBLOCKS	1.000 (6.5)	0.981 (4)	0.992 (5)	0.967 (3)	0.961 (2)	1.000 (6.5)	0.088 (1)
TEXTURE	1.000 (6.5)	0.584 (2)	0.793 (4)	0.759 (3)	0.537 (1)	1.000 (6.5)	0.896 (5)
THYROID	1.000 (6.5)	0.897 (4)	0.901 (5)	0.877 (3)	0.857 (2)	1.000 (6.5)	0.027 (1)
TITATNIC	1.000 (6)	0.842 (2)	1.000 (6)	0.982 (4)	0.959 (3)	1.000 (6)	0.145 (1)
VERTEBRAL	1.000 (6)	0.860 (2)	1.000 (6)	0.957 (4)	0.955 (3)	1.000 (6)	0.468 (1)
VOWEL	1.000 (6.5)	0.816 (2)	0.983 (5)	0.885 (4)	0.566 (1)	1.000 (6.5)	0.836 (3)
WINE	1.000 (6)	0.834 (2)	1.000 (6)	0.966 (4)	0.958 (3)	1.000 (6)	0.591 (1)
WIRELESS	1.000 (6)	0.828 (2)	1.000 (6)	0.970 (3.5)	0.970 (3.5)	1.000 (6)	0.751 (1)
Total Rank	**105.5**	**40**	**94**	**63.5**	**44.5**	**105.5**	**23**

Table 13: Friedman test on the average results per dataset

Classification issue	**coverage**	**metric**	**Friedman test pvalue**
Binary classification	all	accuracy	9.26E-04
	covered	accuracy	4.94E-05
	covered	coverage	3.69E-24
Multiclass classification	all	accuracy	2.86E-06
	covered	accuracy	1.20E-05
	covered	coverage	1.56E-17

2.4. Analysis of computational time required by Forest-ORE

Table 14 reports the execution time (s) spent by the Forest-ORE on the benchmarking datasets.

Table 14: Forest-ORE execution time in seconds (measured on an Intel Core i7, Windows environment)

	Extract rules		Pre-select rules		Prepare opt. Inputs		Build opt. model		Run opt. Model	
	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE
ANNEAL	4.28	0.04	13.40	0.14	26.49	0.44	39.54	0.65	5.53	1.88
APPENDICITIS	0.88	0.02	1.92	0.05	1.76	0.03	4.13	0.08	0.76	0.07
AUTO	4.27	0.04	11.81	0.16	14.88	0.20	18.49	0.29	4.12	0.85
BANANA	0.66	0.01	17.67	0.64	18.14	0.32	37.25	0.76	293.17	42.86
BANKNOTE	1.53	0.03	9.37	0.23	11.39	0.12	29.48	0.34	0.85	0.11
BRCANCER	0.99	0.02	5.66	0.09	13.52	0.30	34.86	0.77	25.81	4.93
CAR	7.50	0.03	37.92	0.51	93.46	0.59	157.01	1.06	41.31	7.45
CRYOTHERAPY	1.00	0.03	1.88	0.05	1.99	0.05	4.71	0.12	0.66	0.11
DERMA	2.26	0.05	7.07	0.12	15.95	0.46	20.55	0.58	1.49	0.47
ECOLI	1.89	0.03	8.04	0.10	19.36	0.33	19.65	0.32	1.31	0.21
GLASS	3.67	0.05	7.65	0.11	10.99	0.27	13.66	0.32	1.23	0.25
HABERMAN	0.88	0.02	2.12	0.04	3.88	0.05	9.55	0.12	0.34	0.06
HEART	2.02	0.04	6.82	0.12	11.14	0.12	26.92	0.36	102.56	22.20
HYPOTHYROID	4.71	0.14	49.38	1.02	53.50	0.76	129.07	1.76	89.24	27.66
INDIAN	6.20	0.08	10.02	0.25	11.83	0.36	27.67	0.58	85.96	23.40
ION	2.42	0.05	7.08	0.08	10.48	0.25	23.32	0.21	181.01	45.86
IRIS	0.50	0.02	0.78	0.03	1.23	0.05	2.31	0.08	0.10	0.04
MAMMOGRAPHIC	3.16	0.05	17.01	0.17	22.71	0.47	60.68	1.20	1.00	0.21
MUSHROOM	1.02	0.02	340.93	10.37	134.50	2.32	366.40	6.68	54.53	1.77
MUTAGENESIS	1.29	0.03	13.31	0.21	16.24	0.29	49.20	0.99	4.49	1.05
NEWTHYROID	1.09	0.03	2.63	0.05	3.89	0.09	7.42	0.18	0.49	0.08
PAGEBLOCKS	7.65	0.11	540.69	16.98	185.05	1.85	331.24	3.34	27.89	5.56
PHONEME	5.97	0.03	485.75	27.44	105.49	2.43	329.81	6.15	663.57	162.03
TEXTURE	35.05	0.30	601.95	13.91	446.32	6.58	401.26	5.31	560.01	220.89
THYROID	12.61	0.18	424.87	11.25	134.45	2.54	282.47	4.72	46.18	17.90
TICTACTOE	3.11	0.02	15.02	0.13	28.58	0.19	78.36	0.66	3.84	0.79
TITATNIC	0.32	0.03	1.69	0.02	5.15	0.09	9.89	0.19	1.60	0.09
VERTEBRAL	2.47	0.06	7.50	0.12	12.17	0.18	24.91	0.30	1.54	0.32
VOTE	0.98	0.01	5.58	0.05	10.96	0.13	27.96	0.35	0.64	0.10
VOWEL	19.26	0.12	50.24	0.61	202.25	1.94	183.55	1.89	25.34	3.81
WDBC	1.53	0.02	7.15	0.11	13.16	0.39	30.97	0.71	98.64	73.55
WILT	3.73	0.06	122.78	5.83	75.43	0.87	185.81	2.09	31.37	4.22
WINE	1.06	0.02	3.16	0.08	5.38	0.17	10.41	0.32	1.16	0.15
WIRELESS	4.13	0.03	58.46	1.53	171.40	1.73	288.91	2.60	23.71	7.62
WISCONSIN	1.11	0.05	6.03	0.12	14.16	0.44	35.47	0.72	8.44	2.41
XOR	0.34	0.00	1.28	0.02	2.05	0.04	4.19	0.04	0.05	0.00

One of the critical aspects of assessing any machine learning method is its scalability, particularly with increasing dataset size and feature dimensionality. We conduct in the following an analysis to understand the relationship between dataset size, feature dimensionality, and computational running time. We also analyze the relationship between the number of input rules, their average length, and computational running time. We began by examining how the running time scales across

the benchmarking datasets, considering both the number of instances and the number of attributes (Figure 4). We then examined the impact of the number of input rules and their length on execution time. The following graphs illustrate these relationships, followed by a detailed discussion of key findings (Figure 5)

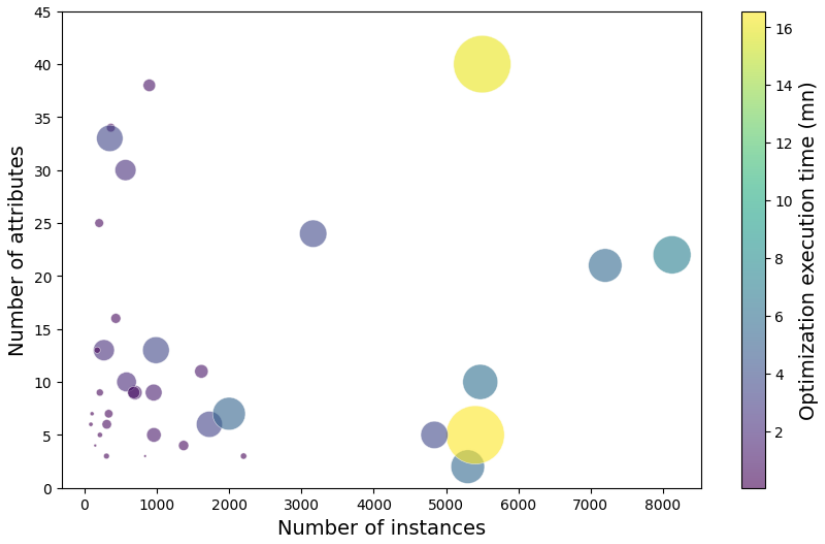

Figure 4: Number of instances vs Number of attributes, sized and colored by Optimization execution time(mn)

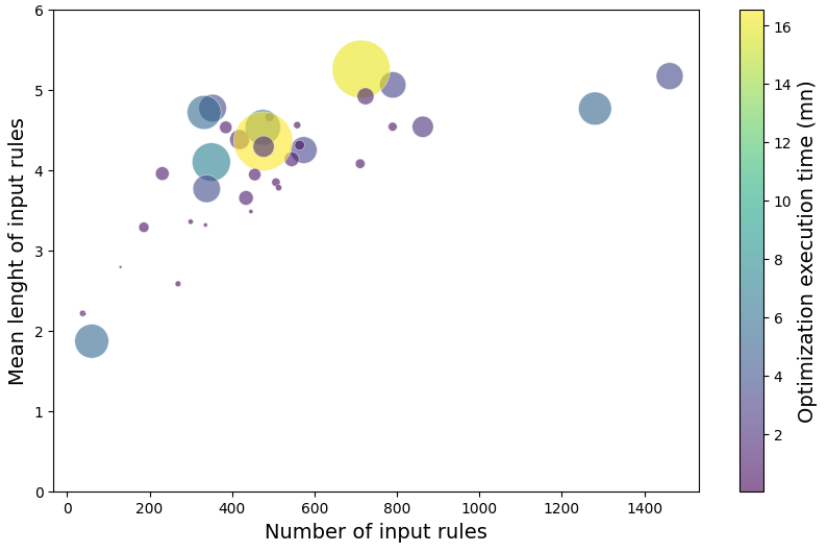

Figure 5: Number of input rules vs Mean lenght of input rules, sized and colored by Optimization execution time(mn)

Key Points Based on the Graphs:

1. Running Time and Feature Dimensionality: Contrary to the expectation that more features would lead to longer run times, we observed that datasets with fewer attributes sometimes require more computational time. This could be due to factors such as data distribution, overlap between classes, or complex relationships between features, which can make rule extraction or optimization more challenging.
2. Running Time and Number of Instances: We also noted that datasets with the largest number of instances did not require proportionally higher running times compared to those with fewer instances. This suggests that the relationship between instance size and running time is non-linear and likely depends on other factors, such as the internal structure of the dataset.
3. Running Time and Input Rules Complexity: The graph suggests that both the number of input rules and their mean length can influence the optimization execution time, with rule complexity (longer rules) often being associated with increased computational time. However, there are exceptions where datasets with the highest number of rules and the longest mean lengths require less execution time. While simplifying rules can help reduce computational effort, a deeper analysis of data characteristics and the nature of the optimization problem is necessary to fully understand what drives execution time in different scenarios.

Conclusion: The running time appears to depend on more nuanced factors, such as:

- The separability or distribution of the data (well-separated datasets may require less time to process).
- The complexity of the relationships between features, rather than simply the number of features or instances.
- The nature of the optimization problem itself, where certain datasets may present more difficult optimization challenges, resulting in longer running times despite having fewer features or instances.

2.5. Ablation Studies

In this section, we analyze the effect of the different weights used in the objective function on the quality of the set of rules and their predictive performance and coverage. We also analyze the impact of suppressing the preselection stage on these measures. The study includes 20 datasets; ten of them concern binary

classification, and ten concern multi-class classification.
Similarly to the ablation methodology used in [8], we obtain the first four ablation models by excluding the weights from the objective function one at a time. The fifth model is obtained by suppressing the preselection stage from Forest-ORE processing. Ablation models are described in Table 15.
Table 16 shows that "abl_conf" and "abl_cov" models induce a decrease in the average rule confidence and average rule coverage. This result demonstrates that excluding W_0 and W_1 weights lowers the quality of the rules. Similar observations can be made about excluding W_2 and W_3, which induce an increase in the number of attributes and modalities per rule. This result demonstrates that excluding the W_2 and W_3 increase the complexity of the rule ensemble.
Table 17, reports the average results on the predictive performance. This table shows that the predictive performance of "abl_lenght" and "abl_preselect" are slightly better than "no_abl" model. However, The "abl_lenght" model induces an increase in the complexity of the model (increase in the length of the rules). On the other hand, the "abl_preselect" model induces a increase in the computational time (Table 18).
These results show, on the one hand, that each term in the objective function contributes to improving the quality of the final rule ensemble. On the other hand, it demonstrates the importance of the preselection stage. Using the preselection stage induces a very small loss in predictive performance (0.003 on average accuracy), but an important gain in computational time (divised by 4.7 on average).

Table 15: Description of the ablation models

Ablation model	Abbrev.	Description	Setting
No high confidence	abl_conf	is obtained by excluding the term which encourages rules with high confidence	$W_0 = 0$
No high coverage	abl_cov	is obtained by excluding the term that encourages rules with high coverage	$W_1 = 0$
No reduced length	abl_length	is obtained by excluding the term which encourages rules with few attributes	$W_2 = 0$
No reduced modalities	abl_mod	is obtained by excluding the term which encourages rules with few levels	$W_3 = 0$
No preselection stage	abl_preselect	is obtained by excluding the preselection stage, which is intended to reduce the size of the initial set of rules by selecting the best RF rules, based on their individual performance	Remove preselection stage

Table 16: Results of the ablation on the quality of the rules

	Confidence		Coverage		Class coverage		Number of variables		Number of levels	
abl_model	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE
no_abl	0.930	0.005	0.190	0.007	0.485	0.015	3.016	0.068	6.986	0.309
abl_conf	0.912	0.006	0.196	0.007	0.501	0.015	2.969	0.066	6.827	0.293
abl_cov	0.932	0.005	0.177	0.007	0.460	0.015	2.978	0.068	6.663	0.301
abl_length	0.930	0.005	0.191	0.007	0.487	0.015	3.089	0.069	7.230	0.324
abl_mod	0.930	0.005	0.190	0.007	0.487	0.015	3.031	0.068	7.103	0.312
abl_preselect	0.928	0.005	0.197	0.008	0.510	0.016	3.002	0.074	6.971	0.325

Table 17: Results of the ablation on the predictive performance of the rules

	Accuracy		F1 score		Fidelity		Coverage	
	Mean	SE	Mean	SE	Mean	SE	Mean	SE
no_abl	0.859	0.003	0.825	0.004	0.952	0.002	0.958	0.001
abl_conf	0.860	0.004	0.826	0.005	0.945	0.003	0.958	0.002
abl_cov	0.860	0.003	0.826	0.004	0.955	0.002	0.955	0.002
abl_length	0.862	0.003	0.827	0.004	0.954	0.002	0.957	0.001
abl_mod	0.860	0.003	0.826	0.004	0.954	0.002	0.958	0.001
abl_preselect	0.862	0.003	0.828	0.004	0.951	0.002	0.956	0.004

Table 18: Results of the ablation on the exection time

	Size of the set of rules				Execution time (s)							
	Initial set		Final set		Extract/preselect rules		Prepare opt inputs		Build opt. model		Run Opt.	
	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE
no_abl	445	6	9.76	0.46	10.52	0.05	15.17	0.05	28.05	0.10	24.31	2.98
abl_conf	445	6	9.77	0.46	10.52	0.05	15.17	0.05	28.09	0.10	36.00	4.07
abl_cov	445	6	9.78	0.46	10.52	0.05	15.17	0.05	28.10	0.09	18.30	1.34
abl_length	445	6	9.79	0.46	10.52	0.05	15.17	0.05	28.08	0.09	22.20	2.83
abl_mod	445	6	9.80	0.47	10.52	0.05	15.17	0.05	28.14	0.09	25.82	2.74
abl_preselect	1978	20.00	9.13	0.42	6.49	0.03	140.36	1.80	155.12	0.31	62.13	8.82

References

[1] R. Agrawal, T. Imieliński, A. Swami, Mining association rules between sets of items in large databases, SIGMOD Rec. 22 (1993) 207–216. doi:10.1145/170036.170072.

[2] L. Wilkinson, Exact and Approximate Area-Proportional Circular Venn and Euler Diagrams, IEEE Transactions on Visualization and Computer Graphics 18 (2012) 321–331. doi:10.1109/TVCG.2011.56.

[3] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, H. Pfister, UpSet: Visualization of Intersecting Sets, IEEE Transactions on Visualization and Computer Graphics 20 (2014) 1983–1992. doi:10.1109/TVCG.2014.2346248.

[4] B. Alsallakh, W. Aigner, S. Miksch, H. Hauser, Radial Sets: Interactive Visual Analysis of Large Overlapping Sets, IEEE Transactions on Visualization and Computer Graphics 19 (2013) 2496–2505. doi:10.1109/TVCG.2013.184.

[5] S. Y. Ho, S. Tan, C. C. Sze, L. Wong, W. W. B. Goh, What can Venn diagrams teach us about doing data science better?, International Journal of Data Science and Analytics 11 (2021) 1–10. doi:10.1007/s41060-020-00230-4.

[6] J. Cohen, A Coefficient of Agreement for Nominal Scales, Educational and Psychological Measurement 20 (1960) 37–46. doi:10.1177/001316446002000104.

[7] J. R. Landis, G. G. Koch, The Measurement of Observer Agreement for Categorical Data, Biometrics 33 (1977) 159. doi:10.2307/2529310.

[8] H. Lakkaraju, S. H. Bach, J. Leskovec, 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, ACM, San Francisco California USA, 2016, pp. 1675–1684. doi:10.1145/2939672.2939874.