IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 March 2023, accepted 29 March 2023, date of publication 20 April 2023, date of current version 2 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3268866

RESEARCH ARTICLE

BELLATREX: Building Explanations Through a LocaLly AccuraTe Rule EXtractor

KLEST DEDJA[1,2], FELIPE KENJI NAKANO[1,2], KONSTANTINOS PLIAKOS[3], AND CELINE VENS[1,2]

[1]Department of Public Health and Primary Care, KU Leuven, 8500 Kortrijk, Belgium
[2]ITEC, imec Research Group at KU Leuven, 8500 Kortrijk, Belgium
[3]Department of Management, Strategy, and Innovation, KU Leuven, 3000 Leuven, Belgium

Corresponding author: Celine Vens (celine.vens@kuleuven.be)

This work was supported in part by the Flemish Government (AI Research Program), and in part by the Research Fund Flanders under Project G080118N and Project G0A2120N.

ABSTRACT Random forests are machine learning methods characterised by high performance and robustness to overfitting. However, since multiple learners are combined, they are not as interpretable as a single decision tree. In this work we propose a novel method that is Building Explanations through a LocalLy AccuraTe Rule EXtractor (Bellatrex), which is able to explain the forest prediction for a given test instance with only a few diverse rules. Starting from the decision trees generated by a random forest, our method: 1) pre-selects a subset of the rules used to make the prediction; 2) creates a vector representation of such rules; 3) projects them to a low-dimensional space; 4) clusters such representations to pick a rule from each cluster to explain the instance prediction. We test the effectiveness of Bellatrex on 89 real-world datasets and we demonstrate the validity of our method for binary classification, regression, multi-label classification and time-to-event tasks. To the best of our knowledge, it is the first time that an interpretability toolbox can handle all these tasks within the same framework. We also show that Bellatrex is able to approximate the performance of the corresponding ensemble model in all considered tasks, and it does so while selecting at most three rules from the whole forest. Finally, a comparison with similar methods in literature also shows that our proposed approach substantially outperforms other explainable toolboxes in terms of predictive performance.

INDEX TERMS Explainable AI, interpretable ML, multi-label classification, multi-target regression, random forest, random survival forest.

I. INTRODUCTION

Machine Learning (ML) models are nowadays employed in various domains with excellent performance, and the raise of complex but highly performing models (so called "black box" models) has characterised the past decades. However, in fields where the stakes for a single decision are high (healthcare, banking, etc.) [1], [2], [3] there is still scepticism in adopting such methods. The motivation for this lies behind the fact that professionals may be held accountable for a costly mistake, and that in such cases they want to fully understand and trust the outcomes of a ML model. An important element that leads to trust in ML models is to achieve more explainability [4].

In recent years, the ML community has therefore been developing tools that help ML practitioners and end-users to understand the predictions of black box models, leading to a booming of the field of Explainable AI (X-AI). As a result, X-AI methods nowadays make up a vast literature [5] and can be categorised in several ways. A common criterion distinguishes between the scope of the output explanations: *global* explanations [6], [7], whose aim is to identify trends in the data and explain the model as a whole, or *local* explanations [8], [9], whose aim is to understand why a certain prediction is made for a given instance. Another common differentiation is based on whether the method is

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

tailored to determined type of model and makes use of the model's internal architecture, or whether it does not. In the first case, we talk about a *model-specific* approach, as opposed to a *model-agnostic* (e.g. LIME [8] and SHAP [9]) approach, which can be applied to any type of model.

In this study, we propose Bellatrex: a local, model-specific method that condenses the predictions made by a random forest (RF) [10] into a few rules. By doing so, our method uses information directly extracted from the original RF. Furthermore, the procedure is optimised to follow closely the RF predictions for every instance, and maintains the original RF high performance. Additionally, Bellatrex provides different insights by selecting rules that are dissimilar from each other. Furthermore, given its local approach, the extracted rules are different for each instance.

The choice to specifically explain RF predictions is driven by the fact that RF has a great predictive performance and is easier to train than other highly performing methods, such as deep neural networks [11]. Moreover, RF has been extended to a wide range of learning tasks, including binary classification, multi-label classification, regression and time-to-event prediction [12]. We should highlight that our method is designed in such a way that it is independent of the learning task at hand, and consequently, it is capable of handling all these scenarios.

Bellatrex is built on top of a trained RF and consists of four main steps. First, for a given input instance, a subset of rules is selected based on how close their prediction is with respect to the original ensemble. Second, a vector representation is built by extracting information from the rule structure. Third, dimensionality reduction of such representations is performed and lastly, clustering on the projected vector representations is applied and a single prototype rule per cluster is eventually extracted. These final rules are then presented to the end user as an explanation for the forest prediction. By keeping the number of prototype rules low, we provide a few diverse explanations, which can be inspected by the user.

Preliminary results of this work were presented in an earlier workshop paper [13] and were restricted to binary and time-to-event predictions. Here, we extend this work to regression (both single- and multi-target) and multi-label classification, and improve the vector representation step. Furthermore, we also provide a thorough experimental evaluation of our method against competitor methods from the literature in predictive performance, explanation complexity and explanation diversity, followed by an ablation study which validates the steps in our method. Overall, our contributions to the field of X-AI can be summarised as follows:

- We propose a local approach for explaining RF predictions, with a method that extracts a few meaningful decision paths from a trained RF. Importantly, the extracted rules are tailored to the instance to be explained, which represents, to the best of our knowledge, a novelty in the field of explaining RF predictions.
- We evaluate our local explainer in five different scenarios: binary classification, single- and multi-target regression, multi-label classification and time-to-event prediction, where the latter three are seldom explored in existing explainability methods. The inclusion of the time-to-event analysis in particular, highlights a novelty of our work since, to the best of our knowledge, no existing model-specific interpretability method is compatible with an adaptation of RF to time-to-event predictions.
- We also compare, for each for each of the aforementioned scenarios, the complexity of the explanations generated by Bellatrex against methods with a similar output format. By doing so, we can quantitatively compare the degree of interpretability of various models; it is the first time that such an analysis is performed in depth.
- We make our method available to the ML community by providing a Python implementation through GitHub,[1] compatible with the scikit-learn and scikit-survival Python libraries.

The remainder of the manuscript is organised as follows: Section II positions our work within the framework of post-hoc, RF-specific explainers, and underlines the differences with the existing literature; Section III provides the background information for the method section (Section IV), where our model is introduced. Section V is dedicated to the experimental set-up, whereas Section VI reports the obtained results as well as the outcome of the ablation study. Two examples of explanations generated by Bellatrex are shown in Section VII; finally, we draw our conclusions and share future directions of this work in Section VIII.

II. RELATED WORK

As previously mentioned, Bellatrex falls in the category of local, model-specific methods, and explains the outcomes of a trained random (survival) forest model in a post-hoc fashion. Such family of RF-specific methods has produced a rich literature as shown in [14] and [15]. In this section, we discuss other model-specific approaches that use RFs as the underlying black-box model.

The earliest example of such is [16]. This work proposed a procedure for analysing a RF in binary classification tasks, and concepts such as *tree-metric*, a distance metric defined on the space of tree learners, are also introduced. Trees are then represented in a two-dimensional space by using Multi Dimensional Scaling on the tree-distance matrix. The authors then perform a qualitative study and mention the possibility of clustering the resulting tree representations.

A more recent work by [17] followed the path traced by [16] and suggested a range of tree similarity metrics and by performing clustering on such tree representations. Furthermore, the authors implicitly propose a vector representation of trees based on the covariates used to build them, which

[1] link: https://github.com/Klest94/Bellatrex (upon acceptance)

corresponds to an adaptation of the approach from [18]. Given such representation, the authors clustered and extracted a few trees out of a random forest and illustrate the validity of their framework "C443" on simulated and real world datasets. The resulting method has some features in common with Bellatrex, such as the idea of representing tree- (or rule-) distances for clustering purposes. However, the focus in C443 lays more on giving an overview of the possible approaches rather than optimising the performance of a surrogate model. Furthermore, the C443 method aims at global explanations (e.g. by looking at trends and sources of heterogeneity), rather than focusing at an instance-base level.

Other relevant approaches include [19]: an algorithm that computes how a covariate within a value range influences the final prediction of a model, while [20] proposes an optimisation-based approach exploits the bijective relationship between a tree leaf and a subspace of the input space. Similarly, recent work [21], [22] lead to the creation of "LionForest", where rules are generated in a local approach fashion, and path reduction approaches are run in order to provide shorter rules with more "conclusive" (i.e., stable) explanations.

A toolbox named "SIRUS" is introduced in [23], where a set of rules is extracted from an RF and shown as an explanation. More specifically, rules are first generated by large RF and a pre-selection step picks the most commonly occurring ones. After that, a "post-treatment" step narrows down the selection by eliminating rules associated to a linear combination of other rules of higher frequency. The procedure stops when the desired number of final rules is reached and is then shown to the end user. These final rules serve as an explanation of the original RF and their average prediction can be used as a the prediction of the obtained surrogate model. The resulting explanation is global and consists of a limited number of rules that are stable under different subsets of data, and whose outcome can be averaged to get a prediction.

More recently, "RuleCOSI+" [24] has been proposed as another global, rule-extraction method that greedily combines and simplifies the corresponding base trees. More specifically, the algorithm builds a decision list by iteratively creating new rules by merging or pruning the ones generated by the tree ensemble. At the end of each iteration step, the generated rule is added to the existing list under the condition that it is beneficial to the generalisation performance. When no extra rules can be added, an empty rule (playing the part of an "else" instance) is appended, and the final model consists of the set of rules found, structured as members of a decision list. That is, for a given instance to be explained, the rules are evaluated in order of appearance until one of them is found to hold and is used as the models' prediction. If no rules are fired, the final, empty "else" rule is used to make the prediction.

Finally, a global approach is represented by "Hierarchical Shrinkage" [25] (HS) algorithm. The idea is to increase performance and robustness of tree learners by shrinking the predictions of any leaf node towards the predictions of its ancestors. The shrinking operation is performed repeatedly and is controlled by a regularisation parameter. The resulting tree has a size that can be controlled and offers a global explanation to the model.

It is worth mentioning that the aforementioned RF-specific, post-hoc explainability methods follow a global approach and therefore show the same explanation regardless of the input instance. Bellatrex, on the other side, follows a local approach that adapts its explanations to the instance of interest. This leads to an increased flexibility of the output explanation and allows Bellatrex to closely follow the predictions of the underlying RF.

III. BACKGROUND

Random forest [10] (RF) is a computationally efficient ML method that delivers excellent predictive performance, and since its appearance, RFs have been extremely successful in performing classification and regression tasks. Moreover, RF has been extended to a variety of other learning tasks, including multi-label classification [26] and multi-target regression. Most of the extensions that have been proposed over the years are associated with the node splitting mechanism as well as the prototype function (i.e., the function that provides predictions in the leaves).

Another adaptation involves survival analysis, a recently explored topic in the ML community [27]. The goal in this setting is to predict the time until an event occurs (hence, this task is also known as time-to-event prediction) when the exact time of the event is not always observed, resulting in partial information (*censoring*). Decision Trees and RFs (as well as other machine learning methods) have been adapted to survival data, they are called (Decision) Survival Trees (SDT) [28] and Random Survival Forests (RSF) [12], respectively. The main splitting criterion being used is the *logrank* score [29], a criterion that guarantees good generalisation for censored time-to-event data [12].

IV. PROPOSED METHOD

Let $\mathcal{T}$ be a random forest and let $\boldsymbol{x}$ be an instance for which we want to find an explanation. Bellatrex uses these inputs to generate an explanation, and an overview of the procedure is illustrated in Fig. 1.

We first extract τ trees $\mathcal{T}_i \in \mathcal{T}$ that generate the most similar predictions $\mathcal{T}_i(\boldsymbol{x})$ compared to the ensemble model prediction $\hat{y} = \mathcal{T}(\boldsymbol{x})$ (Fig. 1, top-left). Next, we represent each of the selected trees $\mathcal{T}_i$ as a vector. Only a few studies [16], [17] have addressed the challenge of representing decision trees as a numerical vector, and they only focus on global representations. Given the local nature of Bellatrex, we propose a novel, path-based approach, where the vector representation for $\mathcal{T}_i$ also depends on the instance of interest $\boldsymbol{x}$. More specifically, we follow $\boldsymbol{x}$ as it traverses the tree and record the input covariates used to perform each split. Next, we assign for each split at node k a contribution to the vector representation that is proportional to the number of

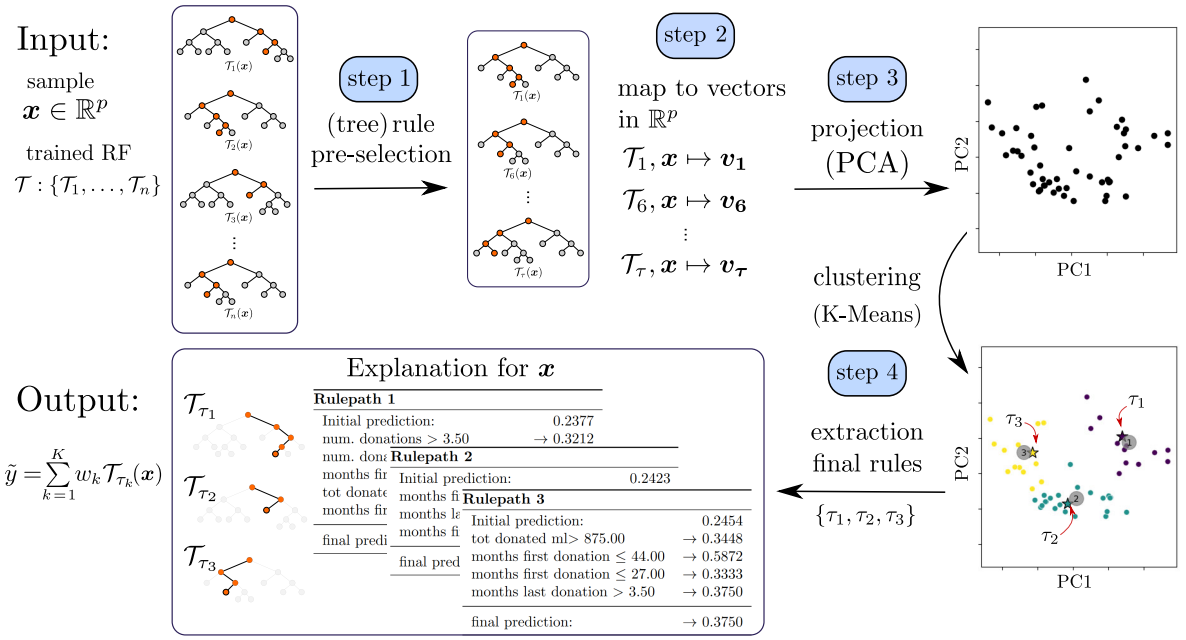


FIGURE 1. Schematic representation of Bellatrex. Given a trained RF and an instance x to be explained (starting top left, going clockwise), rule pre-selection is performed, followed by the vectorisation step. Such vectors are projected and clustered, the final selected rules are shown to the end-user.

instances $n(k)$ traversing the node during the training phase of the tree learner. In formulas, with the root node indexed as 0, our local, path-based representation becomes:

$$\mathcal{T}_i, \boldsymbol{x} \mapsto \boldsymbol{v} = [v_1, v_2, \ldots, v_p], \text{ where}$$
$$v_j = \sum_{k \in \text{path}(\boldsymbol{x})} \frac{n(k)}{n(0)} \cdot \mathbb{1}\{\text{split on covariate } j\} \qquad (1)$$

It is worth noting that the resulting node contribution weights $n(k)/n(0)$ are positive and upper bounded by 1, which is the weight assigned to the root node split; furthermore, the assigned weights decrease as the depth increases. By doing so, rules that differ (only) in the root node splitting covariate will be further apart in Euclidean distance compared to the ones whose decision paths differ (only) in the deeper nodes, which is in agreement with common intuition. From this moment onward, we will refer to $\mathcal{T}_i$ both to the i-th tree learner of the RF as well as the corresponding rulepath representation for an instance $\boldsymbol{x}$.

Next, we project such vector representations to a low-dimensional space using Principal Component Analysis (PCA) (top-right of Fig. 1). The idea is to remove noise, to improve computational efficiency for later steps, and to enable a better visualisation of the subsequent clustering. The representations are real-valued vectors of length d at this stage, where d is the number of output dimensions from the PCA step.

Subsequently, in order to obtain a diverse set of explanations, we perform clustering on the vector representations using a standard clustering method, such as K-Means++ [30] (see an example in Fig. 2). By doing so, we group the vectors into K clusters, we identify the vector closest to each cluster

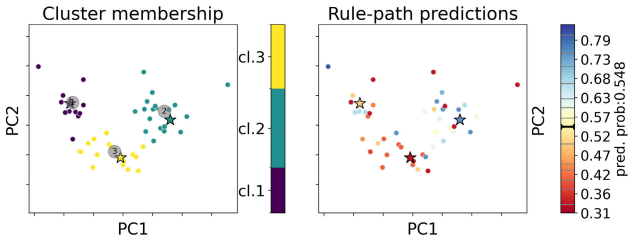


FIGURE 2. Left: Example of the clustering step performed on a test instance x of the blood dataset (Section VI-A for details). The centroids are shown in grey, and the final representative rules are plotted with a star shape. Right: colours according to individual predictions for x.

centre and pick the corresponding rule $\mathcal{T}_{\tau_k}$ as a representative for explaining the outcome of the model (Fig. 1 bottom-right). The rules extracted with this procedure correspond to what we call *final rules*, and the nature of the partitioning step guarantees the distance among the vector representations of the selected final rules is maximal.

It is worth noting that similar results can be obtained with other partitioning algorithms such as K-median [31], as long as the aim of the algorithm is to maximise between-cluster distance and minimise within-cluster distance. The optimal number of clusters K is identified during the hyperparameter tuning procedure, introduced at the end of this Section.

Finally, given the K clusters, the corresponding final rules $\mathcal{T}_{\tau_k}$, and the instance $\boldsymbol{x}$, we build (bottom-left Fig. 1) a surrogate model prediction $\tilde{y}$ as follows:

$$\tilde{y} = \sum_{k=1}^{K} w_k \mathcal{T}_{\tau_k}(\boldsymbol{x}) \qquad (2)$$

where w_k represents the weight given to the cluster k. We define w_k as the proportion of the τ rules (selected in the first step) that are part of the cluster. It follows that $\sum w_k = 1$, and that the surrogate model predicts a weighted average of the selected rules.

Our method requires three hyperparameters: the number of trees τ to keep in the pre-selection phase, the number of output components d for PCA, and the number of clusters K. The optimal values can be tuned for each test instance separately, by calculating the *fidelity* of the surrogate model to the original ensemble. This measure is an indicator of how closely the ensemble prediction is being imitated by our surrogate model, and has been listed as a desirable property for interpreting black-box models [32]. Given an instance $\boldsymbol{x}$, the prediction $\hat{y}$ made by the R(S)F, and the surrogate prediction $\tilde{y}$ from Bellatrex, we define and compute fidelity $\mathcal{F}(\boldsymbol{x})$ as:

$$\mathcal{F}(\boldsymbol{x}) = 1 - \|\hat{y} - \tilde{y}\|_2. \tag{3}$$

Depending on the predictive scenario, $\hat{y}$ and $\tilde{y}$ are either scalars for predicted probabilities (binary classification), estimated scalar values (regression), vectors of predicted class probabilities (multi-label classification) or real valued-vectors (multi-target regression), or scalar predicted risk scores (survival analysis). Note that $\mathcal{F}(\boldsymbol{x}) = 1$ for a given instance $\boldsymbol{x}$ indicates perfect fidelity of Bellatrex to the R(S)F model. As a result, the hyper-parameter combination (K, d, τ) that maximises $\mathcal{F}$ can be chosen.

An important advantage of our method is that it does not require an external validation set to tune these hyperparameters; Bellatrex looks instead at the *predicted* labels from the underlying RF model which at this stage serves as an oracle, in a procedure similar to the one described in [33]. The hyperparameters that, for a given test instance, yield the prediction closest to the underlying RF prediction are the ones selected. By doing so, we increase the fidelity of the surrogate model to the underlying black box, and we concurrently allow an efficient use of data, with a greater fraction used to train the underlying model instead of being left aside for the purpose of hyperparameter tuning.

Alternatively, the parameter values can be determined by the user. In the latter case, end-users can choose, for example, the number of clusters and explore the trade-off between having a simple (single) explanation against obtaining multiple (different) explanations for a given example.

A. COMPUTATIONAL COMPLEXITY

We now analyse the computational complexity of Bellatrex. Given the number of data instances n, the number of covariates p, and the number of learners m, the computational complexity of Bellatrex can be estimated by looking at the complexity of its four main steps:

- the sorting algorithm for rule pre-selection, with complexity $\mathcal{O}(m \log m)$;
- the vectorisation process of the pre-selected rules: $\mathcal{O}(m \log n)$;
- the dimensionality reduction step with PCA, where $\mathcal{O}(n^2_{max}\, n_{min})$ [34], where $n_{max} = \max(m, p)$ and $n_{min} = \min(m, p)$.
- the partitioning algorithm, namely K-Means++, with $\mathcal{O}(d\, m)$, where d is the number of output dimensions from PCA. Follows a nearest neighbour search, performed with KDTree [35], with $\mathcal{O}(d\, m \log(m))$.

We conclude that the computational complexity of Bellatrex is mainly driven by the dimensionality reduction step.

Translating the above analysis into practice, we observe that querying Bellatrex for an explanation is quite fast and takes a couple of seconds to run on a laptop. Moreover, we notice that the computational complexity of Bellatrex is virtually independent of the number of samples n in the data, whereas it is sensitive to the number of covariates p for high dimensional data.

V. EXPERIMENTS

We evaluate our method across multiple datasets; unless otherwise specified, the datasets are selected from publicly available repositories such as UCI[2] and MULAN,[3] or downloaded from sklearn's library Overall, we include as many as 89 datasets spanning five different prediction scenarios, more specifically:

- 24 datasets for binary classification;
- 14 datasets for survival analysis (time-to-event prediction);
- 19 datasets for regression;
- 13 datasets for multi-label classification;
- 19 datasets for multi-target regression.

A number of pre-processing steps were performed before running the experiments in Section V-C. More specifically, categorical variables were one-hot encoded, instances and covariates with more than 30% missing values were dropped, and the remaining missing values were imputed with MICE [36]. Finally, for single target and multi-target regression tasks, we normalised the target labels to the [0, 1] interval for a better comparison of prediction errors. The resulting post-processed datasets and related properties (size, dimensionality, label distribution) are also shared in the repository.

A. COMPETING METHODS

The proposed algorithm is compared against several competing methods. More specifically we include: four tree-based models of various degree of interpretability, two linear models that are interpretable off the shelf, and four methods from literature that provide post-hoc explanations to RF predictions.

1) TREE-BASED METHODS

The first family of models that we compare against includes tree-based learners such as DT and RF. These methods represent the two extremes of the performance-interpretability

[2] https://archive.ics.uci.edu/ml/datasets.php
[3] http://mulan.sourceforge.net/datasets-mlc.html

trade-off. In addition, we propose two approaches, namely ''Small RF'' and ''OOB Trees'', which position themseves somewhere in between. The Small RF method consists of training a Random (Survival) Forest with K base tree learners, where K is chosen for each test instance to be equal to the one used by Bellatrex. The OOB Trees method consists of averaging the prediction of K selected trees from a full R(S)F, where the K trees with smallest out-of-bag (OOB) error are selected. The value K is again set to the same value as the number of final rules extracted by Bellatrex.

2) LINEAR, ANTE-HOC EXPLAINABLE METHODS

Next, we perform a performance comparison with competing methods that are *ante-hoc* explainable, that is, models that are interpretable by design. We include regularised regression models (LR) with elastic net penalty for binary classification, multi-label classification, regression, and multi-target regression tasks. Additionally, we consider a regularised Cox proportional hazard (Cox-PH) [37] model for time-to-event data.

3) POST-HOC COMPETITORS

Finally, we compare our method against recent work in literature. For this purpose we select from the related work Section II the methods that have a publicly available source code. The selection narrows down to C443 [17], SIRUS [23], RuleCOSI+ [24] and HS [25]. It is worth mentioning that only SIRUS and HS fully support regression tasks, and that only the latter is adapted to multi-target regression and multi-label classification. Moreover, none of the selected competing methods is adapted to a survival analysis scenario.

To begin with, to set up a comparison between C443 (available for binary classification tasks only, within the homonym package in R) and Bellatrex, we train the underlying RF black-box model with the same number of learners and stopping criterion as in Section V-C. Next, we run C443, collect the output trees, and average their prediction (weighted proportionally to cluster size) for a close comparison. As for the number of clusters we run C443 with $K = 3$, which is the upper bound for the number of rules extracted by Bellatrex.

To run the comparison with SIRUS, we keep the parameters suggested by the authors. More specifically, the underlying RF model is trained with 5000 trees and the number of final rules to be extracted is set to 10. For RuleCOSI+, we run the proposed method after increasing the maximum rule length to 5 and the rule confidence to be at least 0.9, we do so to achieve better performance. We therefore report RuleCOSI+ results for this particular choice of hyperparameters. Finally, we consider Hierarchical Shrinkage [25] (HS) method in its default configuration. Such configuration sets the maximum number of leaves of the underlying decision tree learner to 20.

B. EVALUATION METRICS

We consider several metrics to validate our method, as well as to compare it, whenever possible, against similar methods in literature. More specifically, we firstly look at predictive performance, which is also a desirable property for model interpretability [32], [38]; next, we look at the complexity of the explanations generated by our method. Finally, we check against redundancy of the explanations by looking at the dissimilarity of the generated rulesets.

1) PERFORMANCE

The predictive performance of the models is computed through commonly used measures in each of the five tasks. More specifically, we evaluate the AUROC for binary classification, the weighted average AUROC in multi-label classification (averaged over each label with weight proportional to the number of positive instances of the label), mean absolute error (MAE) for single target and multi-target regression tasks (in the latter case averaged over the targets), and concordance index (C-index) for the survival analysis data.

2) COMPLEXITY

Next to predictive performance, we investigate the complexity of the generated explanations, a concept that is inversely related to model interpretability. The notion of interpretability is itself difficult to measure objectively as no general definition of interpretability exists [38], [39], and many metrics have been proposed [40]. However, when it comes to specifically interpreting tree-based models, it is common to report the total number of rule-splits shown by the model explainer as a measure of interpretability [20], [24], [41], where a lower number of rule-splits (complexity) corresponds to a higher interpretability. Formally, consider an explanation made of rules $\mathcal{R} = \{r_1, \ldots, r_m\}$, and let $\{\text{len}(r_1), \ldots, \text{len}(r_m)\}$ be their respective length (that is, number of split tests), then the *complexity* of the explanation is defined as:

$$\mathcal{C} = \sum_{r_i \in \mathcal{R}} \text{len}(r_i) \tag{4}$$

In practice, which set of rules $\mathcal{R}$ constitute an explanation is not always well defined and can depend on the user needs. In the context of our study, since we are focusing on local explanations, we include in $\mathcal{R}$ only the rules that are effectively used for the final instance prediction. More specifically:

- In the case where one or more decision trees are shown to the end-user as an explanation, we estimate $\mathcal{C}$ by counting only the rule(s) $r_{t_1}, \ldots, r_{t_n}$ that lead to a leaf node, and we define $\mathcal{R} = \{r_{t_1}, \ldots r_{t_n}\}$. This is the case of HS, DT, C443, OOB Trees, and Small RF.
- In case, such as in RULECOSI+, the end-user is presented an ordered decision list with $m - 1$ disjoint rules and a final rule r_m that serves as an ''else'' clause, then one of the two scenarios holds:

  - the activated rule is $r_n \neq r_m$. In this case, we consider that the activated rule is a sufficient explanation, hence we set $\mathcal{R} = \{r_n\}$;
  - the activated rule is r_m, which is not a sufficient explanation since it has no antecedent and its prediction is used because none of the previous rules is activated. The explanation for the end-user is complete only if all rules are shown, therefore we set $\mathcal{R} = \{r_1, \dots r_m\}$.
- In case the explanation consists of a collection of decision rules $\{r_1, \dots, r_m\}$ such as in Bellatrex and SIRUS, we have $\mathcal{R} = \{r_1, \dots, r_m\}$.

The above definition suggests that the complexity of an explanation increases as the rules get longer or as the cardinality of $\mathcal{R}$ increases, which is in agreement with human intuition. Regarding this, we notice that, let d be the maximum depth of a tree and K the number of extracted rules by Bellatrex for a given instance, we have $\mathcal{C} \leq Kd$. This means that the complexity of the final explanation can be tuned by either adjusting the maximum depth of the underlying RF, or by tuning the maximum value of K in the rule extraction step of Bellatrex. Such two-fold tuning can also be performed with C443, but cannot be as effective with other methods such as SIRUS where rules are already extremely short and $d \approx 1$, or DT and HS, where $K = 1$.

3) DISSIMILARITY

Finally, we consider the dissimilarity of the extracted rules. For this purpose, we need to define a distance on the tree-representation space, and for this purpose we use a generalised version of the Jaccard similarity index [17]. Namely, given the vector representations $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$ of two rules, we compute their similarity $\mathcal{S}$ as:

$$\mathcal{S}(\boldsymbol{v_i}, \boldsymbol{v_j}) = \frac{\sum_{k=1}^{p} \min(v_{ik}, v_{jk})}{\sum_{k=1}^{p} \max(v_{ik}, v_{jk})}. \tag{5}$$

Given an instance $\boldsymbol{x}$ and the vector representation of the associated final rules $\boldsymbol{v}_{i_1}, \dots, \boldsymbol{v}_{i_K}$, we obtain the rule dissimilarity $\mathcal{D}$ for $\boldsymbol{x}$ by computing the average pairwise dissimilarity $1-\mathcal{S}$ between the K final rule vector representations:

$$\mathcal{D} = \frac{1}{K(K-1)} \sum_{l \neq j}^{l,j \in \{i_1, \dots i_K\}} 1 - \mathcal{S}(\boldsymbol{v_l}, \boldsymbol{v_j}) \tag{6}$$

Finally, $\mathcal{D}$ is computed for every test instance, and the average is reported in the results (Section VI). When $K = 1$, dissimilarity in (6) is not defined, therefore such instances do not contribute to the computation of the average.

C. EXPERIMENTAL SET-UP

In all experiments, we trained the underlying RF with 100 base learners. As stopping criterion, we required that split-nodes included at least 5 instances, or at least 10 for the time-to-event scenario, no rule post-pruning was performed.

We report cross validated average predictive performance, complexity, and dissimilarity along 5 folds. The test sample size for datasets exceeding 500 instances is limited to 100 for computational reasons.[4] In the multi-label classification scenario, we drop labels that do not occur in the training or testing folds, since AUROC would otherwise not be defined.

TABLE 1. Possible choices for the hyperparameters in Bellatrex. For every instance x to be explained, the combination with the highest achieved fidelity $\mathcal{F}$ is chosen.

name	description	values
τ	nb. of pre-selected trees	{20, 50, 80}
d	nb. of dimensions after PCA	{2, 5, no PCA}
K	nb. of clusters and final rules	{1, 2, 3}

With regards to hyperparameter tuning of the steps of our method, values are tested in a grid search fashion and are shown in Table 1. As explained earlier, these values are optimised for each test instance individually according to fidelity (3). The rationale behind this grid is that the possible values for τ and d cover a wide range while using at most 3 different values and limit computation time. Furthermore, the values for K are chosen so that at most 3 final rules are selected, keeping the model fairly explainable.

VI. RESULTS

In this Section, we report the average predictive performance, complexity, and dissimilarity of the explanations generated by our method across the different datasets. On the predictive performance side, we compare Bellatrex against all the competing methods introduced in Section V-A. Next, the complexity (as in (4)) of Bellatrex output explanations is compared against rule extracting algorithms where the total number of splits can be counted. Finally, the average dissimilarity (6) of Bellatrex rulesets is compared against methods that extract a similar number of rules from a RF.

The dataset-specific results for the binary classification data are presented in this section, whereas, to avoid excessive repetition, only the scenario-wide average results are reported for the remaining cases. Dataset-specific results for the remaining scenarios are shared in the Appendix. Additionally, we test the statistical significance of the observed differences (in performance, in dissimilarity and in complexity) among Bellatrex and the other competing methods. We do so by conducting a post-hoc Friedman-Nemenyi test, setting the significance level to 0.05, as recommended in [42]. Results are visualised with critical difference diagrams, which connect methods that are not statistically significantly different by horizontal line segments. In this manuscript we share the results of the Friedman-Nemenyi for the binary classification, whereas for the remaining scenarios we refer to the Appendix. Finally, we present an ablation study; that is, we verify whether the main steps of the proposed procedure are of added value to the method.

[4]Note that the algorithmic procedure of rule extraction, including hyperparameter tuning, is performed for every test instance separately.

TABLE 2. Average predictive performance (AUROC) across binary classification datasets. Our proposed methods is shown under the name "Bellatrex" in its two approaches. The highest achieved performances except for the "black box" RF are shown in bold.

	RF	**Bellatrex**	OOB Trees	Single DT	Small RF	LR	SIRUS	C443	Rule COSI+	HS
blood	0.7080	0.7041	0.6474	0.7048	0.6413	**0.7394**	0.7327	0.6256	0.6419	0.7209
breast cancer diagn.	0.9836	**0.9867**	0.9393	0.9601	0.9526	0.9645	0.9806	0.9748	0.9400	0.9655
breast cancer original	0.9958	0.9954	0.9665	0.9626	0.9696	0.9970	**0.9974**	0.9923	0.9749	0.9625
breast cancer progn.	0.5015	0.5348	0.5689	0.5893	0.5230	**0.6370**	0.5896	0.5504	0.5374	0.6333
breast cancer coimba	0.7569	0.7508	0.6846	0.7108	0.6085	**0.8108**	0.7415	0.6762	0.5938	0.7200
Colonoscopy Green	0.9333	**0.9449**	0.8128	0.6962	0.7526	0.6718	0.7974	0.8333	0.7154	0.7897
Colonoscopy Hinselm.	0.5917	0.6229	0.5875	0.6146	0.4667	0.4833	**0.6458**	0.5813	0.5812	0.5792
Colonoscopy Schiller	0.6277	0.5923	0.5369	0.5600	0.5385	0.6615	0.6662	**0.6969**	0.6000	0.5246
divorce	0.9941	0.9471	0.9412	0.9349	0.9405	**1.0000**	0.9920	0.9716	0.9325	0.9294
Flowmeters	0.9943	**0.9771**	0.9271	0.9086	0.7800	0.4771	0.9286	0.9157	0.9514	0.9314
haberman	0.6897	0.6786	0.6400	0.6185	0.6269	0.6856	**0.6883**	0.6763	0.5550	0.6175
hcc-survival	0.8331	**0.8250**	0.6688	0.7162	0.7377	0.4192	0.7831	0.6369	0.6008	0.7138
ionosphere	0.9828	**0.9812**	0.9204	0.9276	0.9067	0.9020	0.9406	0.9420	0.8251	0.9386
LSVT voice	0.8824	**0.8882**	0.7404	0.7699	0.7228	0.6191	0.7993	0.7846	0.7022	0.7566
mamographic	0.8523	0.8510	0.8247	0.8355	0.8249	0.8316	0.8439	0.8352	0.8072	**0.8539**
musk	0.9511	**0.9475**	0.8300	0.7716	0.8088	0.9132	0.7562	0.8633	0.7158	0.7983
parkinson	0.9349	**0.9128**	0.7912	0.7819	0.7766	0.3051	0.7424	0.8099	0.7663	0.7789
risk factors	0.9709	0.9339	0.9050	0.9379	0.8745	**0.9801**	0.9420	0.7176	0.8179	0.9408
simulation crashes	0.9284	0.9012	0.8335	0.7594	0.8027	**0.9654**	0.8663	0.5799	0.7939	0.7700
sonar	0.9163	**0.9117**	0.7591	0.7498	0.7347	0.8455	0.8045	0.7493	0.7617	0.7622
SPECT	0.7695	0.7489	0.7411	0.6957	0.7037	**0.7870**	0.7305	0.7682	0.7169	0.7502
SPECTF	0.8203	**0.7920**	0.6193	0.6485	0.7273	0.7442	0.7838	0.7087	0.6656	0.6816
vertebral column	0.9550	0.9477	0.9002	0.8907	0.8530	**0.9562**	0.8954	0.8423	0.8962	0.8762
wholesale	0.9554	**0.9525**	0.9056	0.9322	0.9141	0.9023	0.9382	0.9364	0.9163	0.9390
Average	0.8554	**0.8470**	0.7788	0.7782	0.7578	0.7625	0.8161	0.7779	0.7504	0.7889

A. PREDICTIVE PERFORMANCE

We compute the average test-set performance for every considered dataset over 5-fold cross validation. We report the average across datasets for each of the five prediction tasks, and we include the dataset-specific performance results for the binary classification case in Table 2.

1) PREDICTION PERFORMANCE IN BINARY CLASSIFICATION

We observe that Bellatrex considerably outperforms all competing methods and its performance is often on par, and sometimes better, than the original RF. In most of the cases, Bellatrex performs better than the other competing methods. Some exceptions to this trend are provided by the LR learner which, despite achieving non-remarkable performance on average, appears to be the best model in some datasets (e.g. "blood" and "Colonoscopy Schiller"). This phenomenon might be explained by the intrinsically different (linear) nature of LR predictions as opposed to the other tree-based models. In the middle range of the performance spectrum, we encounter competing methods from literature such as HS, with C443 and SIRUS scoring a couple of "wins" and performing better than LR on average. On the lower end, we have the remaining methods: RuleCOSI+, Small RF, OOB Trees and DT. The trends from this qualitative analysis are confirmed by the Friedman-Nemenyi test ($\alpha = 0.05$) and the resulting diagram is shown in Fig. 3.

We observe that the difference in performance between RF and Bellatrex is small and is not statistically significant, meaning that we reduce the size of the output explanation without significant loss in predictive performance. It is also remarkable that Bellatrex is ranked on top of all the competitor methods and is also statistically significantly different to RuleCOSI+, Small RF, OOB Trees, as well as the single DT. This is to be expected as the competing methods offer a global approach for explanations, whereas Bellatrex follows a local approach that provides a more flexible modelling.

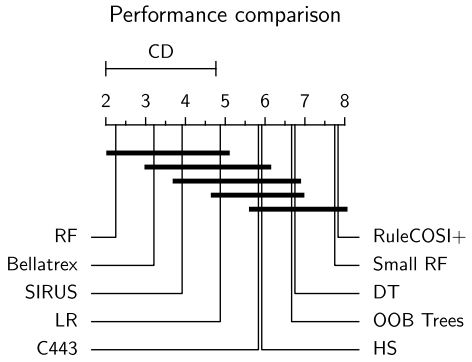


FIGURE 3. Friedman-Nemenyi test, results regarding performance on binary classification data.

2) PREDICTION PERFORMANCE IN TTE, REGRESSION, MLC, MTR

Similar trends are highlighted when running Bellatrex on time-to event (TTE), regression, multi-label classification (MLC), and multi-target regression (MTR) data, with the average results being shown in Table 3. The competitiveness of Bellatrex against all competing methods is confirmed as

TABLE 3. Average performance across remaining datasets. In bold, the best achieved performance, excluding the black-box R(S)F.

Method	predictive scenario			
	survival	multi-label	regression	multi-target
R(S)F	0.7274	0.8473	0.0679	0.0531
Bellatrex	**0.7275**	**0.7894**	**0.0674**	**0.0511**
OOB (S)Trees	0.6719	0.7524	0.0772	0.0562
Small R(S)F	0.6724	0.7441	0.0788	0.0591
(S)DT	0.6696	0.7378	0.0832	0.0593
linear model [a]	0.7040	0.5076	0.0942	0.0758
SIRUS	-	-	0.1090	-
HS	-	0.7585	0.0890	0.0715

[a] Cox-PH for time-to-event tasks, LR for the other tasks

TABLE 4. Average complexity of the explanations across binary classification datasets.

Method	avg. complexity (avg. n. rules)
Bellatrex	8.62 (1.7)
OOB Trees	9.26 (1.7)
Small RF	9.03 (1.7)
Single (S)DT	4.44 (1.0)
SIRUS	11.83 (10.0)
RuleCOSI+	6.00 (2.5)
C443	15.17 (3.0)
HS	**4.34** (1.0)

Bellatrex outperforms all considered competitors. Furthermore, Bellatrex is outperforming the original black-box in regression and MTR tasks, and shows a lower performance for MLC tasks.

Methods like HS are based on a single DT and offer an overly simplified prediction process, while methods such as LR fail to capture non-linearity and label correlations. Finally, SIRUS has been implemented for binary and regression tasks, but has not been tested for the latter.

B. COMPLEXITY

Next, we consider the complexity of explanations and compare Bellatrex against other competing methods. It is worth reminding that our definition of complexity in (4) counts the number of split-rules performed across (possibly multiple) extracted rules. This allows to compare the interpretability of Bellatrex not only against tree-based methods such as DTs, Small RF and OOB Trees, but also against methods from literature that output rule-based explanation in a different format. More specifically, we can compare Bellatrex against SIRUS, HS, RuleCOSI+, and C443. We do not report the complexity $\mathcal{C}$ from RF since its complexity is much higher than the other reported methods.

1) EXPLANATION COMPLEXITY IN BINARY CLASSIFICATION

Specifically for binary classification tasks, we compare Bellatrex against all aforementioned methods. To determine which rules are considered as part of an explanation, we refer to our guidelines in Section V-B. We report average complexity $\mathcal{C}$ in Table 4, and we also include the average number of rules extracted by Bellatrex. Additionally, a dataset-specific comparison is available in the Appendix, in Table 17.

The results show that Bellatrex needs 8.62 splits on average to generate an explanation. This finding, combined with the fact that 1.7 rules are extracted on average suggests that the final rules are usually fairly short and can be considered more interpretable than a single rule of similar total complexity. Furthermore, we also notice that the datasets for which Bellatrex extracts the least amount of rules (such as ''divorce'', ''breast cancer original'' and ''risk factors'') are also the ones whose binary label is easy to predict (AUROC $\geq$ 0.90) for Bellatrex (cfr. Table 2). This suggests that Bellatrex is able to output non-redundant explanations: if a single rule is enough to make an accurate prediction, our method does not need to extract any more rules.

When comparing Bellatrex with RuleCOSI+, C443, DT, and HS, we observe that the latter two generate shorter explanations. This is to be expected since these methods use a single rule to perform predictions for a given instance. It is worth noting however, that this comes at the expense of losing performance: we are witnessing another instance of interpretability-performance trade-off as highlighted in Fig. 4. The aforementioned trade-off is also present when comparing the results of SIRUS against HS and RuleCOSI+: SIRUS outputs rules with higher total complexity, and at the same time shows a slightly higher performance. It is worth noting that SIRUS extracts 10 rules by default, and since its average number of rule-splits is 11.83, it means that the vast majority of the extracted rules has length 1, consistent with what is reported in the original paper [23]. As for C443, the achieved performance is similar to HS, whereas its total complexity is comparable to SIRUS and higher than Bellatrex. The latter results is to be expected given that C443 always extracts 3 trees and therefore 3 rules are contributing to the computation of $\mathcal{C}$, as opposed to the average 1.7 rules in Bellatrex.

Finally, OOB Trees and Small RF achieve similar complexity compared to Bellatrex, which is to be expected given that the same number of rules is extracted from a similarly constructed underlying RF.

2) EXPLANATION COMPLEXITY IN TTE, REGRESSION, MLC, MTR

Continuing on the comparison of the (proxy for) interpretability, we compare the complexity of explanations $\mathcal{C}$ on the remaining four scenarios when a competing method implementation is available. This translates to comparing Bellatrex against HS for regression, MLC, and MTR tasks,

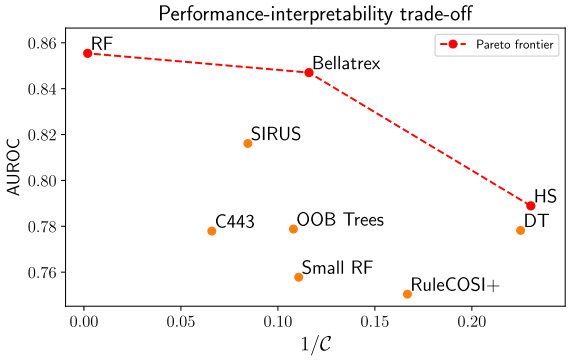


FIGURE 4. Visualisation of the performance-interpretability trade-off, where interpretability is approximated as $1/C$. The Pareto optimum frontier is shown in red.

TABLE 5. Average number of rule-splits used as explanation, comparison across datasets. In bold, the shortest average explanation per scenario.

Method	predictive scenario			
	survival	multi-label	regression	multi-target
Bellatrex	19.23	63.00	22.53	23.10
OOB Trees	18.73	61.77	22.32	23.48
Small RF	18.80	60.82	22.30	23.54
Single (S)DT	10.69	15.56	8.61	8.00
SIRUS	-	-	13.53	-
HS	-	**5.14**	**4.27**	**4.15**

and against SIRUS for regression tasks. OOB Trees and Small RF, and (S)DT are also considered, and an overview is given in Table 5, whereas detailed results can be found in Tables 18-21.

The complexity of the rules extracted by Bellatrex and OOB Trees and Small RF is similar across all scenarios, in line with what is observed with the binary classification datasets. We also observe that Bellatrex generates on average longer explanations on MLC tasks, likely due to the fact that such datasets are more difficult to learn and need larger trees to fit. For this case, we suggest the end user to select one (or a few) label(s) of interest in the training phase, and run Bellatrex on the reduced label set to get shorter explanations.

Finally, it is worth mentioning that the increase in average rule length observed in multi-label datasets is common to Bellatrex, DT, OOB Trees and Small RF, whereas it does not affect HS, whose rules are significantly shorter.

C. DISSIMILARITY

Finally, we compare (when possible) the average dissimilarity $\mathcal{D}$ of the final rules extracted by Bellatrex against the ones generated by Small RF, the ones picked according to the OOB Trees method, and the ones extracted by the C443 method. These methods are indeed the only ones that extract a comparable number of independent learners; HS on the other hand builds a single decision tree whereas SIRUS extracts 10 rules by default. Finally, RULECOSI+ does not generate independent rules by design: exactly one of them is activated, under the condition that the previous ones have not been used. The results across the five predicting scenarios are shown in Table 6, with the dataset-specific details available in the Appendix.

TABLE 6. Average dissimilarity comparison across datasets. In bold: highest dissimilarity per scenario.

Method	predictive scenario				
	binary	survival	multi-label	regression	multi-target
Bellatrex	0.9044	**0.8420**	**0.9569**	**0.5459**	**0.7145**
C443	**0.9286**	-	-	-	-
OOB Trees	0.8268	0.8219	0.9240	0.4171	0.5977
Small RF	0.8556	0.7945	0.9380	0.4483	0.6326

We observe that C443 and Bellatrex achieve the highest dissimilarity and consistently achieve higher dissimilarity compared to Small RF and OOB Trees. This means that the Bellatrex algorithm is successful at picking dissimilar enough rulesets, and that these are not redundant. We can verify how this dissimilarity in rules translates in practice in Section VII.

The Friedman-Nemenyi test (Fig. 16) indicates that C443 and Bellatrex generate significantly more dissimilar rules compared to OOB Trees and to Small RF, whose rule dissimilarity is driven by the intrinsic randomness of the RF algorithm. Also, the difference in dissimilarity between rulesets in C443 and rulesets in Bellatrex is not statistically significant.

D. ABLATION STUDY

As described above, Bellatrex consists of four main steps: 1) pre-selection of the most relevant decision rules within the random forest, 2) mapping of the selected rules to a vector representation, 3) projection of the vector representations to a low-dimensional space and 4) the clustering step followed by the final rule extraction. The vector representation and the clustering steps are fundamental for our approach and cannot possibly be removed, therefore we focus on evaluating the added value of the rule pre-selection mechanism and of the dimensionality reduction step. To do so, we measure the predictive performance of Bellatrex in the following four configurations:

- We apply our proposed method in full, including all steps described above;
- We apply dimensionality reduction but no rule pre-selection (i.e., no step 1);
- We apply rule pre-selection but no dimensionality reduction (no step 3);
- We apply neither dimensionality reduction (PCA) nor rule pre-selection (i.e. both steps 1 and 3 are removed);

We present the results related to binary classification datasets in Table 7, while the results related to the other tasks are provided in the Appendix (Tables 27-30). As shown, the best average performance is achieved when all steps of the proposed approach are included, affirming the added

TABLE 7. Ablation study results on the binary datasets. In bold, the best achieved performance.

	Bellatrex (original)	no step 1	no step 3	no step 1 & no step 3
blood	0.7041	**0.7050**	0.7005	0.6975
breast cancer diagn.	0.9867	**0.9872**	0.9865	0.9823
breast cancer original	**0.9954**	0.9934	0.9953	0.9922
breast cancer progn.	0.5348	0.5226	**0.5441**	0.4822
breast cancer coimba	0.7508	0.7677	**0.7723**	0.7600
Colonoscopy Green	**0.9449**	0.8987	0.8385	0.9205
Colonoscopy Hinselm.	0.6229	0.4708	0.6042	**0.6396**
Colonoscopy Schiller	0.5923	**0.6754**	0.6077	0.6569
divorce	**0.9471**	0.9457	0.9412	0.9401
Flowmeters	**0.9771**	0.9514	0.9500	0.9286
haberman	0.6786	0.6726	0.6769	**0.6832**
hcc-survival	**0.8250**	0.7638	0.8058	0.7412
ionosphere	**0.9812**	0.9768	0.9737	0.9444
LSVT voice	**0.8882**	0.8654	0.8243	0.8618
mamographic	**0.8510**	0.8467	0.8494	0.8396
musk	**0.9475**	0.9298	0.9230	0.8757
parkinson	**0.9128**	0.8824	0.9005	0.8340
risk factors	0.9339	0.9309	**0.9351**	0.9339
simul. crashes	**0.9012**	0.8875	0.8629	0.7775
sonar	**0.9117**	0.8928	0.8864	0.8141
SPECT	0.7489	0.7524	0.7500	**0.7680**
SPECTF	0.7920	**0.8177**	0.7918	0.7478
vertebral	0.9477	**0.9501**	0.9499	0.9414
wholesale	0.9525	**0.9549**	0.9529	0.9455
performance	**0.8470**	0.8351	0.8343	0.8212

value of each of them. Furthermore, we observe the most substantial decrease in average performance when we remove both steps 1 and 3, whereas the intermediate scenarios (one step removed) fall somewhere in between.

VII. ILLUSTRATION OF BELLATREX

In this Section, we show how Bellatrex works in practice, and to do so we pick two examples from the "boston housing" [43] regression dataset. This dataset contains 506 neighbourhoods, each described by 14 covariates. The target variable is the median house value, normalised to the [0, 1] interval.

Our first example consists of a test instance with a relatively low house value of 0.206, for which Bellatrex predicts a value of 0.244 (same value predicted by RF). More specifically, Bellatrex reaches the prediction by discarding 20 rules in the pre-selection step and projecting the vector representation of the remaining rules in 2 dimensions; such representations are grouped in a single cluster and the rule closest to the centre is selected as the final representative. Further insights are shown in Fig. 5, where the rule representations are shown.

The left plot of the Figure shows that a single cluster is selected, and that the rules on the right side of the plot are treated as outliers. On the right-side plot, the same rules are coloured according to their prediction, and these are shown to vary between 0.20 and 0.29. The final representative rule is situated in the central position and is associated to a prediction close to the average value of 0.24. Inspection of such rule is shown in Example 1, where we show how the prediction, starting from an initial estimated equal to the

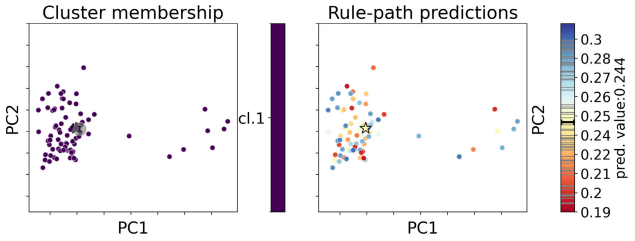


FIGURE 5. Bellatrex performed on a test instance of the boston housing dataset. On the left, the centre of the cluster is in grey; on the right, rules are coloured according to their predicted value. On both sides, the final extracted rule is shown in a star shape.

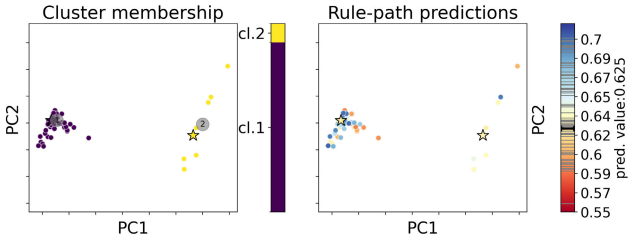


FIGURE 6. Bellatrex performed on a second test instance of the boston housing data. On the left, the centres of the cluster are in grey; on the right, rules are coloured according to their predicted value. On both sides, the final extracted rules are shown in a star shape.

TABLE 8. Overview of the 24 datasets used in the binary classification scenario.

	data size (n, p)	% positive class
blood	(748 , 4)	24
breast cancer diagn.	(569, 30)	37
breast cancer original	(699, 10)	34
breast cancer progn.	(198, 33)	24
breast cancer coimba	(116, 9)	55
Colonoscopy Green	(98, 62)	68
Colonoscopy Hinselm.	(97, 62)	85
Colonoscopy Schiller	(92, 62)	73
divorce	(170, 54)	49
Flowmeters	(87, 36)	60
haberman	(306, 3)	26
hcc-survival	(165, 49)	62
ionosphere	(351, 34)	36
LSVT voice	(126, 310)	67
mamographic	(961, 5)	46
musk	(476, 166)	43
parkinson	(756, 753)	75
risk factors	(858, 35)	6
crashes	(540, 18)	91
sonar	(208, 60)	53
SPECT	(267, 22)	79
SPECTF	(267, 44)	79
vertebral	(310, 6)	68
wholesale	(440, 7)	32

average (bootstrapped) value of the root node, moves until its final value indicated by the corresponding leaf.

The extracted rule is aligned with human intuition: the root node-split shows that a smaller average number of rooms (n. rooms) per dwelling is associated to a lower house value, with the prediction dropping from an initial estimate of

TABLE 9. Overview of the 14 datasets used in the time-to-event scenario.

	data size (n, p)	censoring rate (%)
addicts	(238, 3)	37
breast cancer survival	(198, 80)	74
DBCD	(295, 4919)	73
DLBCL	(240, 7399)	43
echocardiogram	(130, 9)	32
FLChain	(7874, 8)	72
gbsg2	(686, 8)	56
lung	(228, 8)	28
NHANES I	(9931, 18)	65
PBC	(403, 19)	56
rotterdam (excl. recurr)	(2982, 11)	57
rotterdam (incl. recurr)	(2982, 12)	57
veteran	(137, 9)	7
whas500	(500, 14)	57

TABLE 10. Overview of the 19 datasets used in the regression scenario.

	data size (n, p)
airfoil	(1503, 5)
Ames Housing	(2915, 283)
auto mpg	(398, 7)
bike sharing	(731, 12)
boston housing	(506, 14)
california housing	(20640, 8)
car imports	(201, 63)
Computer hardware	(209, 6)
concrete compress	(1030, 8)
concrete slump	(103, 9)
ENB2012 cooling	(768, 8)
ENB2012 heating	(768, 8)
forest fires	(517, 12)
PRSA data	(41757, 13)
slump dataset	(103, 9)
students maths	(395, 43)
wine quality all	(6497, 12)
wine quality red	(1599, 11)
wine quality white	(4898, 11)

TABLE 11. Overview of the 13 datasets used in the multi-label classification scenario.

	data size (n, p)	number of labels
birds	(645, 260)	19
CAL500	(502, 68)	174
emotions	(593, 72)	6
enron	(1702, 1001)	53
flags	(193, 19)	7
genbase	(662, 1185)	27
langlog	(1460, 1004)	75
medical	(978, 1449)	45
ng20	(4379, 1006)	20
scene	(2407, 294)	6
slashdot	(3782, 1079)	22
stackex chess	(1675, 585)	227
yeast	(2417, 103)	14

0.390 to 0.317. Similarly, the splits involving lower status show that houses situated in a poorer neighbourhood (that is, with high values of lower status), are associated to lower prices. According to this rule, the median house price of the town increases with a low crime rate, and decreases with high proportion of houses being built before 1940 (high value for the age variable).

Our second example involves a test instance of high value $y = 0.660$, whose Bellatrex prediction is 0.625 (and underlying RF prediction is 0.633), and two rules are selected as the final prediction, as shown in Fig. 6.

TABLE 12. Overview of 19 datasets used in the multi-target regression scenario.

	data size (n, p)	number of targets
andro	(49, 30)	6
atp1d	(337, 370)	6
atp7d	(296, 370)	6
edm	(154, 16)	2
enb	(768, 8)	2
ENB2012	(768, 8)	2
jura	(359, 15)	3
oes10	(403, 298)	16
oes97	(334, 263)	16
osales	(639, 401)	12
rf1	(9125, 64)	8
rf2	(4332, 576)	8
scm1d	(9803, 280)	16
scm20d	(8966, 61)	16
scpf	(1137, 23)	3
sf1	(323, 10)	3
sf2	(1066, 9)	3
slump	(103, 7)	3
wq	(1060, 16)	14

TABLE 13. Average predictive performance (C-index) across time-to-event datasets. In bold, the best achieved performance, excluding the black-box RSF.

	RSF	**Bella-trex**	OOB S. Trees	SDT	Small RSF	Cox PH
addicts	0.6497	0.6512	0.6065	0.6323	0.6205	**0.6563**
b. c. survival	0.6524	**0.6417**	0.5619	0.5863	0.5619	0.5708
DBCD	0.7549	**0.7534**	0.5940	0.5574	0.6686	0.7206
DLBCL	0.6352	**0.6357**	0.5697	0.5796	0.5456	0.6160
echocardiogr.	0.4145	0.4166	0.4427	**0.4958**	0.4785	0.4096
FLChain	0.8312	**0.8330**	0.8105	0.7683	0.7858	0.8218
gbsg2	0.7022	**0.7035**	0.6850	0.6434	0.6419	0.6876
lung	0.6216	**0.6236**	0.5816	0.5904	0.5386	0.6056
NHANES I	0.8205	**0.8209**	0.7358	0.7435	0.7584	0.8043
PBC	0.8469	**0.8467**	0.7977	0.7601	0.7780	0.8192
rotterdam [a]	0.7954	**0.7961**	0.7581	0.7518	0.7731	0.7668
rotterdam [b]	0.9038	**0.9047**	0.8774	0.8418	0.8652	0.8996
veteran	0.7349	0.7349	0.6547	0.6983	0.6546	**0.7452**
whas500	0.7449	**0.7458**	0.6922	0.6730	0.6763	0.7332
Average	0.7220	**0.7224**	0.6691	0.6659	0.6676	0.7040

[a] excluding the recurr variable
[b] including the recurr variable

Example 1. Example of output Bellatrex explanation.

Extracted rule	(weight=1)	initial estimate = 0.390
Instance	split test	prediction
(n. rooms = 5.64)	n. rooms $\leq$ 6.80	$\rightarrow$ 0.317
(lower status = 18.34)	lower status $>$ 14.30	$\rightarrow$ 0.209
(lower status = 18.34)	lower status $\leq$ 19.08	$\rightarrow$ 0.271
(age = 94.70)	age $>$ 93.90	$\rightarrow$ 0.238
(crime rate = 0.88)	crime rate $\leq$ 2.15	$\rightarrow$ 0.275
(lower status = 18.34)	lower status $>$ 17.07	$\rightarrow$ **0.244** (leaf)

TABLE 14. Average performance (MAE) across regression datasets. In bold, the best achieved performance, excluding RF.

	RF	**Bellatrex**	OOB Trees	DT	Small RF	LR	SIRUS	HS
airfoil	0.0373	**0.0374**	0.0492	0.0645	0.0493	0.1069	0.1195	0.0896
AmesHousing	0.0229	0.0230	0.0280	0.0308	0.0297	**0.0226**	0.0504	0.0356
auto mpg	0.0525	**0.0525**	0.0615	0.0644	0.0664	0.0676	0.0871	0.0692
bike sharing	0.0481	**0.0481**	0.0586	0.0635	0.0594	0.0783	0.0933	0.0713
boston housing	0.0577	**0.0576**	0.0650	0.0747	0.0724	0.0824	0.0842	0.0694
california housing	0.0613	**0.0611**	0.0726	0.0848	0.0791	0.1040	0.1438	0.1045
car imports 1985	0.0362	**0.0360**	0.0436	0.0456	0.0417	0.0384	0.0666	0.0465
Computer hardware	0.0291	**0.0297**	0.0327	0.0423	0.0356	0.0354	0.0440	0.0369
concrete compressive strength	0.0397	**0.0396**	0.0497	0.0602	0.0534	0.1053	0.1126	0.0807
concrete slump data	0.0735	0.0734	0.0816	0.0852	**0.0669**	0.0903	0.1223	0.0890
ENB2012 cooling	0.0339	**0.0338**	0.0350	0.0386	0.0363	0.0724	0.0768	0.0434
ENB2012 heating	0.0112	**0.0111**	0.0114	0.0132	0.0127	0.0644	0.0807	0.0247
forest fires	0.3040	**0.3032**	0.3100	0.3256	0.3248	0.3190	0.3170	0.3116
PRSA data	0.0359	**0.0361**	0.0460	0.0491	0.0463	0.0882	0.1022	0.0855
slump dataset	0.0682	0.0680	0.0778	0.0797	0.0901	**0.0506**	0.0957	0.0744
students final math	0.1556	**0.1550**	0.1855	0.1851	0.1666	0.1764	0.1611	0.1624
wine quality all	0.0741	**0.0710**	0.0879	0.0893	0.0871	0.0930	0.1025	0.0961
wine quality red	0.0843	**0.0808**	0.0934	0.1008	0.0978	0.1029	0.1123	0.1067
wine quality white	0.0645	**0.0623**	0.0774	0.0833	0.0810	0.0923	0.0996	0.0938
Average	0.0679	**0.0674**	0.0772	0.0832	0.0788	0.0942	0.1090	0.0890

Example 2. Example of output Bellatrex explanation.

Extracted rule 1	($w_1 = 0.90$)	initial estimate $= 0.399$
Instance	split test	prediction
(n. rooms = 7.18)	n. rooms > 6.98	$\rightarrow 0.774$
(n. rooms = 7.18)	n. rooms ≤ 7.44	$\rightarrow 0.647$
(black = 393)	black > 386	$\rightarrow 0.6642$
(n. rooms = 7.18)	n. rooms ≤ 7.26	$\rightarrow 0.679$
(crime rate = 0.03)	crime rate ≤ 0.03	$\rightarrow 0.625$ (leaf)
Extracted rule 2	($w_2 = 0.10$)	initial estimate $= 0.379$
Instance	split test	prediction
(lower status = 4.03)	lower status ≤ 9.71	$\rightarrow 0.549$
(n. rooms = 7.18)	n. rooms ≤ 7.44	$\rightarrow 0.500$
(n. rooms = 7.18)	n. rooms > 6.73	$\rightarrow 0.603$
(pupils ratio = 17.8)	pupils ratio < 18.9	$\rightarrow 0.624$
(age = 61.10)	age ≤ 87.00	$\rightarrow 0.612$
(black = 393)	black > 392	$\rightarrow 0.641$
(age = 61.10)	age > 41.95	$\rightarrow 0.608$
(n. rooms = 7.18)	n. rooms > 6.93	$\rightarrow 0.630$
(pupils ratio = 17.8)	pupils ratio > 15.50	$\rightarrow 0.628$ (leaf)

TABLE 15. Average performance (AUROC) across multi-label datasets. In bold, the best achieved performance, excluding RF.

	RF	**Bella-trex**	OOB Trees	DT	Small RF	LR	HS
birds	0.9199	**0.8190**	0.7862	0.7371	0.7818	0.4563	0.7565
CAL500	0.5781	**0.5428**	0.5327	0.5206	0.5349	0.5085	0.5427
emotions	0.8520	**0.8205**	0.7586	0.7453	0.7498	0.4399	0.7846
enron	0.8192	**0.7707**	0.7322	0.7190	0.7164	0.5071	0.7349
flags	0.7458	**0.7237**	0.6882	0.6799	0.6336	0.6372	0.6938
genbase	1.0000	**0.9977**	0.9928	0.9971	0.9944	0.5019	0.9976
langlog	0.7575	0.6075	0.6012	0.6107	0.5991	0.5099	**0.6940**
medical	0.9841	**0.9497**	0.9102	0.9279	0.8954	0.5182	0.9461
ng20	0.9631	**0.9225**	0.8525	0.8360	0.8630	0.4996	0.8180
scene	0.9477	**0.9195**	0.8437	0.7865	0.8451	0.5023	0.8353
slashdot	0.8691	**0.8186**	0.7669	0.7716	0.7754	0.4978	0.7263
stackex	0.8330	**0.6928**	0.6670	0.6538	0.6653	0.5016	0.6916
yeast	0.7455	**0.6777**	0.6491	0.6063	0.6197	0.5180	0.6393
Average	0.8473	**0.7894**	0.7524	0.7378	0.7441	0.5076	0.7585

The plot shows that two final rules are extracted have predictions close to the average despite the fact that they are following different decision splits, as shown in Example 2

TABLE 16. Average performance (weighted MAE) across multi-target datasets. In bold, the best achieved performance, excluding the black-box RF.

	RF	**Bella-trex**	OOB Trees	DT	Small RF	LR	HS
andro	0.0959	**0.0911**	0.1078	0.1162	0.1102	0.1388	0.1374
atp1d	0.0484	**0.0479**	0.0556	0.0589	0.0563	0.0741	0.0648
atp7d	0.0470	**0.0443**	0.0489	0.0503	0.0597	0.0871	0.0580
edm	0.1194	**0.1068**	0.1072	0.1189	0.1259	0.1656	0.1323
enb	0.0227	**0.0225**	0.0228	0.0244	0.0241	0.0630	0.0335
ENB2012	0.0235	**0.0234**	0.0249	0.0271	0.0258	0.0649	0.0353
jura	0.0678	**0.0678**	0.0783	0.0808	0.0801	0.0717	0.0808
oes10	0.0218	**0.0220**	0.0246	0.0268	0.0263	0.0271	0.0294
oes97	0.0293	**0.0296**	0.0321	0.0343	0.0339	0.0379	0.0342
osales	0.0378	**0.0361**	0.0422	0.0431	0.0433	0.0867	0.0461
rf1	0.0023	**0.0019**	0.0027	0.0036	0.0030	0.0324	0.0454
rf2	0.0038	**0.0031**	0.0051	0.0063	0.0053	0.0288	0.0444
scm1d	0.0253	**0.0249**	0.0323	0.0362	0.0330	0.0368	0.0640
scm20d	0.0309	**0.0296**	0.0402	0.0419	0.0388	0.0664	0.0817
scpf	0.0113	**0.0105**	0.0138	0.0113	0.0163	0.0122	0.0135
sf1	0.0772	**0.0726**	0.0797	0.0727	0.0769	0.0751	0.0768
sf2	0.0321	0.0318	0.0324	**0.0312**	0.0328	0.0322	0.0339
slump	0.1451	0.1433	**0.1403**	0.1643	0.1501	0.1551	0.1649
wq	0.1681	**0.1610**	0.1774	0.1791	0.1806	0.1849	0.1828
Average	0.0531	**0.0511**	0.0562	0.0593	0.0591	0.0758	0.0715

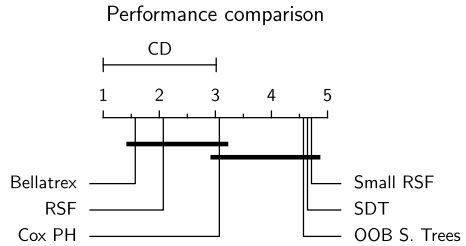


FIGURE 7. Nemenyi-test for time-to-event data.

The first rule follows human intuition when assigning higher house value for higher n. rooms and for lower crim. rate. Additionally, it also reflects the racial bias of the 1970 US Census when assigning higher value to ethnically homogeneous neighbourhoods (high values of black covariate) [44]. The second rule shows a similar

TABLE 17. Average complexity of explanations in binary classification data, with the shortest explanations in bold.

	Bellatrex	OOB Trees	Small RF	DT	SIRUS	Rule COSI+	C443	HS
blood	15.13	14.94	14.40	6.39	12.00	4.09	22.59	**3.39**
breast cancer diagnostic	5.49	5.79	5.93	4.13	13.40	**3.48**	13.10	4.05
breast cancer original	5.53	6.13	5.00	4.01	11.60	**3.45**	13.48	4.85
breast cancer prognostic	9.79	10.56	11.19	**4.13**	10.00	9.13	15.61	4.33
breast cancer coimba	9.37	9.57	9.36	**3.39**	11.20	8.17	13.89	4.32
Colonoscopy Green	7.03	8.44	8.23	**3.76**	10.20	5.98	13.07	4.82
Colonoscopy Hinselmann	6.28	6.94	6.31	**2.71**	10.20	4.23	10.92	4.28
Colonoscopy Schiller	6.70	7.97	7.08	**3.03**	10.20	4.83	11.08	3.41
divorce	1.47	1.45	1.73	**1.29**	10.40	1.53	4.64	1.39
Flowmeters	4.89	5.44	4.76	**2.61**	11.60	2.76	8.27	2.78
haberman	13.48	12.88	13.12	5.73	13.20	7.25	19.71	**5.22**
hcc-survival	9.56	10.87	11.00	4.56	10.00	7.32	15.64	**4.32**
ionosphere	8.01	10.37	9.27	**5.08**	14.60	7.40	17.54	5.27
LSVT voice	6.60	7.20	7.06	**3.22**	10.20	5.51	10.48	3.72
mamographic	16.21	16.34	14.89	7.82	12.00	8.36	21.33	**4.08**
musk	12.48	13.72	13.79	6.09	12.80	11.59	21.32	**5.76**
parkinson	13.39	14.14	13.59	9.42	10.80	8.58	23.54	**4.10**
risk factors	7.27	8.73	9.25	5.49	15.60	**5.22**	23.54	6.57
simulation crashes	6.48	6.71	6.58	**3.61**	13.00	3.78	15.86	4.46
sonar	8.29	9.33	9.62	**3.74**	10.40	9.42	14.04	4.02
SPECT	11.55	12.04	12.04	**4.17**	15.00	5.40	11.87	4.77
SPECTF	7.85	8.62	8.78	**4.17**	10.80	7.27	13.48	5.32
vertebral	7.25	7.42	7.46	**3.70**	11.80	6.17	14.20	4.37
wholesale	6.74	6.65	6.31	4.35	13.00	**2.96**	14.87	4.52
Average	8.62	9.26	9.03	4.44	11.83	6.00	15.17	**4.34**

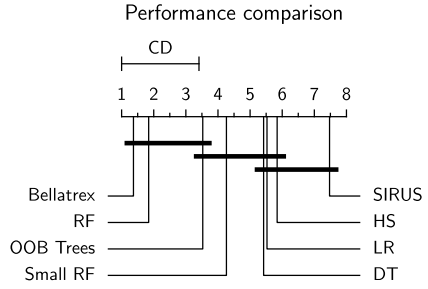


FIGURE 8. Nemenyi-test for regression tasks.

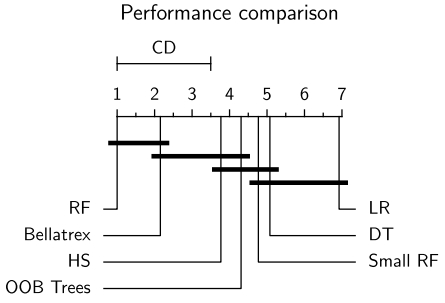


FIGURE 9. Nemenyi-test for multi-label classification data.

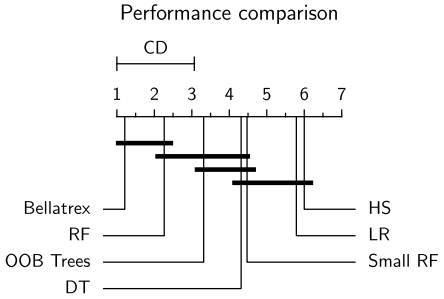


FIGURE 10. Nemenyi-test for multi-target regression data.

TABLE 18. Average complexity of explanations in time-to event data, with the shortest explanations in bold.

	Bellatrex	OOB STrees	Small RSF
addicts	10.51	**9.46**	10.07
B. C. survival	**10.82**	11.02	11.73
DBCD	21.11	**16.93**	19.64
DLBCL	26.98	23.61	**23.53**
echocardiogram	10.15	10.18	**9.73**
FLChain	31.73	32.15	**31.69**
gbsg2	19.04	**17.79**	18.01
lung	12.86	**11.88**	12.30
NHANES I	29.85	29.47	**28.94**
PBC	**14.70**	15.07	14.91
rotterdam (excl. recurr)	21.88	21.83	**21.79**
rotterdam (incl. recurr)	21.83	22.90	**21.48**
veteran	**9.90**	10.80	11.11
whas500	**15.76**	15.89	15.79
average	18.42	**17.85**	17.97

perspective by using a different set of tests: the trend highlighted by n. rooms is consistent with previous findings, whereas conditions including lower status, pupils ratio, and age appear for the first time in this example. Consistently with human intuition lower values of pupils-to-teacher ratio (pupils ratio) are associated with higher house value, whereas the age value for this instance has a small negative effect on house pricing. Finally, consistently with the previous example, a lower value for lower status is associated to higher median house value.

TABLE 19. Average complexity of explanations in regression datasets, with the shortest explanations in bold.

	Bellatrex	OOB Trees	Small RF	SIRUS	HS
airfoil	23.10	22.62	22.46	13.80	**4.17**
AmesHousing	29.85	29.11	28.85	14.00	**3.85**
auto mpg	16.75	16.44	16.78	12.00	**3.73**
bike sharing	21.86	21.57	21.49	14.80	**4.46**
boston housing	21.33	21.77	20.57	12.20	**3.58**
california	38.60	37.74	37.82	13.00	**3.92**
car imports	14.93	15.69	15.77	14.20	**4.76**
Computer	16.83	17.67	17.61	14.80	**4.19**
compress. strength	22.77	22.25	22.79	12.20	**4.13**
concrete slump	11.76	11.32	11.68	13.40	**4.63**
ENB2012 cooling	17.69	17.70	17.64	14.20	**3.92**
ENB2012 heating	17.32	17.34	17.25	15.00	**4.36**
forest fires	20.31	20.40	20.24	10.00	**5.88**
PRSA data	41.26	41.30	41.30	15.60	**4.20**
slump dataset	12.27	12.07	12.55	13.40	**4.86**
students final math	19.34	19.11	18.58	14.80	**4.54**
wine quality all	29.28	28.45	28.09	14.20	**3.71**
wine quality red	23.70	22.85	23.32	11.20	**4.08**
wine quality white	29.23	28.77	28.82	14.20	**4.20**
average	22.54	22.32	22.30	13.53	**4.27**

TABLE 20. Average complexity of explanations in multi-label data, with the shortest explanations in bold.

	Bellatrex	OOB Trees	Small RF	HS
birds	23.43	23.97	24.72	**4.82**
CAL500	29.29	31.29	29.87	**4.26**
emotions	19.48	21.84	22.61	**4.40**
enron	57.84	59.84	56.64	**4.72**
flags	16.51	17.65	18.10	**4.27**
genbase	22.64	20.48	21.68	**5.79**
langlog	42.98	39.50	38.19	**7.02**
medical	59.49	56.71	56.08	**6.06**
ng20	127.97	122.78	119.65	**6.30**
scene	25.95	26.71	26.88	**3.38**
slashdot	248.33	244.98	240.03	**5.50**
stackex chess	111.52	100.02	100.56	**5.14**
yeast	33.51	37.18	35.60	**5.10**
average	63.00	61.77	60.82	**5.14**

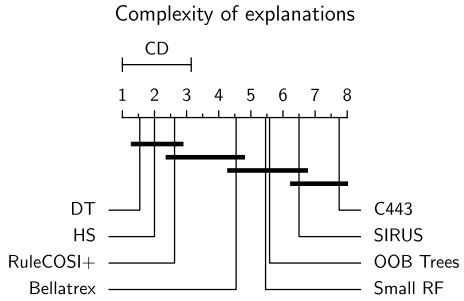


FIGURE 11. Binary classification datasets.

TABLE 21. Average complexity of explanations in multi-target data, with the shortest explanations in bold.

	Bellatrex	OOB Trees	Small RF	HS
andro	8.77	9.62	9.18	**5.42**
atp1d	19.01	19.86	19.41	**3.65**
atp7d	18.97	20.05	20.24	**3.65**
edm	9.25	9.27	9.30	**3.72**
enb	18.55	18.58	18.65	**3.92**
ENB2012	18.32	18.33	18.44	**4.13**
jura	20.87	20.71	20.53	**4.44**
oes10	24.49	24.38	24.40	**4.40**
oes97	23.26	23.73	23.05	**4.25**
osales	49.87	55.07	57.53	**4.72**
rf1	30.40	30.94	30.67	**3.43**
rf2	27.48	27.98	27.98	**3.98**
scm1d	33.82	33.65	33.46	**3.77**
scm20d	34.84	34.84	35.14	**4.01**
scpf	27.75	27.13	27.71	**3.37**
sf1	12.42	12.34	12.88	**4.55**
sf2	17.05	17.11	16.44	**4.55**
slump	13.72	13.21	13.30	**4.89**
wq	30.08	29.28	29.04	**4.09**
average	23.10	23.48	23.54	**4.15**

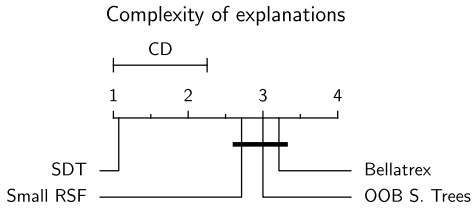


FIGURE 12. Time-to-event datasets.

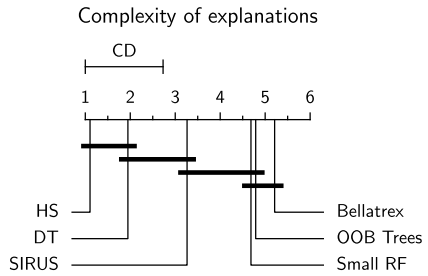


FIGURE 13. Regression datasets.

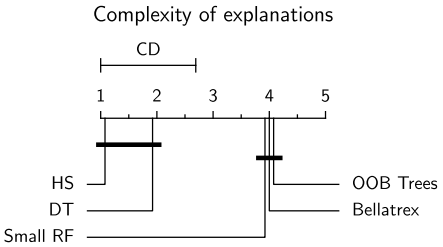


FIGURE 14. Multi-label classification datasets.

VIII. CONCLUSION AND FUTURE WORK

We proposed a novel method (Bellatrex) that interprets RF predictions by extracting, for each given instance, a surrogate model consisting of the most representative rules. Although we have focused on RF's, our proposed approach is able to accommodate other tree-ensemble models such as Extremely Randomized Trees [45] or such as Bagging trees [46]. Furthermore, Bellatrex is compatible with other variations

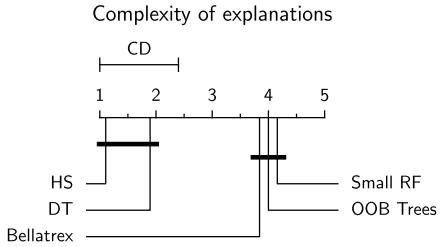


FIGURE 15. Multi-target regression datasets.

TABLE 22. Average rule dissimilarity for binary classification datasets.

	OOB Trees	Small RF	**Bellatrex**	C443	average n. rules
blood	0.5398	0.5743	**0.6366**	0.6080	1.89
B.C. diagn.	0.8799	0.8966	0.9569	**0.9797**	1.25
B.C. original	0.7563	0.7349	0.8534	**0.9027**	1.20
B.C. progn.	0.9266	0.9251	0.9635	**0.9696**	2.03
B.C. coimba	0.7862	0.7894	0.8769	**0.8923**	2.18
Col. Green	0.9241	0.9561	0.9882	**0.9955**	1.85
Col. Hinselm.	0.8747	0.9636	0.9739	**0.9866**	1.74
Col. Schiller	0.9462	0.9659	**0.9933**	0.9892	1.88
divorce	0.8936	0.9349	**1.0000**	0.9959	1.04
Flowmeters	0.8513	0.9477	0.9901	**1.0000**	1.67
haberman	0.5380	0.5686	0.6589	**0.7616**	1.91
hcc-survival	0.9439	0.9381	0.9683	**0.9998**	2.18
ionosphere	0.8948	0.9068	0.9450	**0.9741**	1.49
LSVT voice	0.9826	0.9901	**0.9971**	0.9811	1.86
mamographic	0.5175	0.5605	0.6185	**0.6367**	1.67
musk	0.9773	0.9679	0.9894	**0.9979**	1.92
parkinson	0.9867	0.9922	0.9963	**0.9996**	1.66
risk factors	0.8167	0.8778	0.9414	**0.9832**	1.15
simul. crashes	0.7532	0.8415	0.9004	**0.9896**	1.34
sonar	0.9281	0.9555	0.9848	**0.9983**	2.01
SPECT	0.8617	0.8656	0.8979	**0.9389**	1.84
SPECTF	0.9231	0.9424	0.9608	**0.9823**	1.85
vertebral	0.6531	0.6970	0.7868	**0.8439**	1.56
wholesale	0.6871	0.7425	0.8277	**0.8813**	1.39
Average	0.8268	0.8556	0.9044	**0.9286**	1.69

TABLE 23. Average rule dissimilarity for time-to event data.

	OOB Trees	Small RSF	**Bellatrex**	average n. rules
addicts	**0.6032**	0.4419	0.5679	2.17
breast cancer survival	0.9657	0.9563	**0.9841**	2.26
DBCD	0.9969	0.9978	**0.9990**	2.26
DLBCL	0.9876	0.9971	**0.9994**	2.40
echocardiogram	0.7952	0.7905	**0.8567**	2.25
FLChain	0.6214	0.6761	**0.7430**	2.31
gbsg2	**0.7691**	0.7242	0.7633	2.33
lung	0.8182	0.7743	**0.8395**	2.29
NHANES I	0.7953	0.7945	**0.8391**	2.08
PBC	0.8495	0.8239	**0.8664**	2.28
rotterdam (excl. recurr)	**0.7958**	0.7333	0.7614	2.18
rotterdam (incl. recurr)	0.7501	0.7541	**0.8187**	2.22
veteran	0.7904	0.7461	**0.8167**	2.33
whas500	**0.8503**	0.7977	0.8305	2.35
average	0.8135	0.7863	**0.8347**	2.27

TABLE 24. Average rule dissimilarity for regression datasets.

	OOB Trees	Small RF	**Bellatrex**	average n. rules
airfoil	0.2391	0.2499	**0.3236**	2.30
AmesHousing	0.3034	0.3868	**0.4994**	2.35
auto mpg	0.6067	0.5759	**0.7403**	2.28
bike sharing	0.3403	0.4536	**0.6360**	2.31
boston housing	0.3728	0.4165	**0.5223**	2.34
california	0.1293	0.1833	**0.2721**	2.29
car imports	0.5419	0.4940	**0.6270**	2.31
Computer	0.4917	0.5141	**0.5566**	2.30
compress. strength	0.2432	0.3342	**0.4340**	2.38
concrete slump	0.3353	0.3921	**0.3989**	2.30
ENB2012 cooling	0.6035	0.6663	**0.7583**	2.31
ENB2012 heating	0.6049	0.6002	**0.7552**	2.26
forest fires	0.7254	0.7555	**0.7967**	2.27
PRSA data	0.2322	0.2378	**0.4673**	2.43
slump dataset	0.4581	0.5381	**0.6753**	2.41
students final math	0.6451	0.6821	**0.8208**	2.30
wine quality all	0.2955	0.2886	**0.3160**	2.02
wine quality red	0.4333	0.4422	**0.4544**	2.09
wine quality white	**0.3232**	0.3061	0.3185	2.11
average	0.4171	0.4483	**0.5459**	2.28

of random forests such as Random Survival Forest and Random Forest Regression and has been evaluated across 89 datasets belonging to five different predictive tasks: binary classification, time-to-event prediction, multi-label classification, regression, and multi-target regression. To our knowledge, no related work is as versatile as Bellatrex in providing explanations in so many different scenarios, or at least, no such experimental results have been reported. Bellatrex achieves high levels of predictive performance by extracting a small number of rules, which we limited to three in our experiments. Its performance is often on par with that of random forests, and even outperforms this benchmark in two out of five scenarios. We can therefore conclude that our method achieves its intended goal of explaining the prediction of a random forest with a few rules, without giving up on predictive performance. Furthermore, when compared against ante-hoc explainable methods such as LR and Cox-PH, Bellatrex comes on top. Finally, the comparison with state-of-the-art methods outline that Bellatrex outperforms, often significantly, all considered methods that extract rulesets from a trained RF, such as C443 [17], and SIRUS [23], or that manipulate tree learners such as HS [25].

The analysis of the explanation complexity shows that Bellatrex extracts on average 2 rules with a total combined complexity (i.e. test-splits used) that is normally below 25. Such numbers suggest that our extracted rules are fairly interpretable, even more so since two rules of length ℓ are more interpretable than a single rule of length 2ℓ. The systematic comparison of output complexity (and therefore interpretability) against other methods represents another novelty of this work. Our results confirm the existence of a performance-interpretability trade-off within tree based and rule extraction methods, with approaches such as HS and RuleCOSI+ generating shorter explanations but performing significantly worse. We also systematically studied the dissimilarity of the paths extracted by Bellatrex, where

TABLE 25. Average rule dissimilarity for multi-label data.

	OOB Trees	Small RF	**Bellatrex**	average n. rules
birds	0.9544	0.9664	**0.9859**	2.19
CAL5	0.9589	0.9494	**0.9731**	2.95
emotions	0.9318	0.9430	**0.9772**	2.63
enron	0.9652	0.9754	**0.9847**	2.73
flags	0.8615	0.8496	**0.9115**	2.76
genbase	0.9255	0.9364	**0.9465**	1.33
langlog	0.9690	0.9800	**0.9886**	2.40
medical	0.9507	0.9571	**0.9700**	2.57
ng20	0.8614	0.9093	**0.9169**	2.54
scene	0.9543	0.9768	**0.9908**	2.38
slashdot	0.8606	0.8830	**0.8873**	2.52
stackex chess	0.8862	0.9272	**0.9404**	2.78
yeast	0.9330	0.9406	**0.9662**	2.78
average	0.9240	0.9380	**0.9569**	2.51

TABLE 26. Average rule dissimilarity for multi-target data.

	OOB Trees	Small RF	**Bellatrex**	average n. rules
andro	0.7265	0.8548	**0.9324**	2.58
atp1d	0.8369	0.9082	**0.9483**	2.61
atp7d	0.8345	0.9477	**0.9731**	2.51
edm	0.5925	0.7311	**0.8482**	2.02
enb	0.5129	0.6408	**0.7467**	2.42
ENB2012	0.6717	0.6377	**0.7579**	2.40
jura	0.4259	0.4799	**0.5551**	2.57
oes10	0.9386	0.9285	**0.9506**	2.82
oes97	0.9387	0.9394	**0.9574**	2.82
osales	0.6833	0.7192	**0.7262**	2.66
rf1	0.1251	0.1270	**0.1683**	2.48
rf2	0.1772	0.1918	**0.2928**	2.45
scm1d	0.5864	0.6428	**0.8560**	2.55
scm20d	0.5261	0.4319	**0.6534**	2.64
scpf	0.4680	0.4429	**0.4682**	2.24
sf1	0.6775	0.6905	**0.7580**	1.89
sf2	0.4907	0.5014	**0.6181**	1.97
slump	0.4855	0.5849	**0.6054**	2.46
wq	0.6580	0.6191	**0.7598**	2.74
average	0.5977	0.6326	**0.7145**	2.47

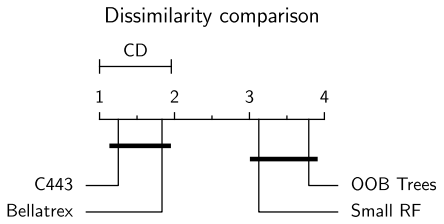


FIGURE 16. Binary data.

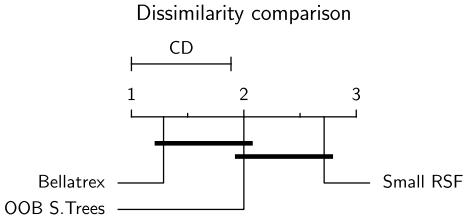


FIGURE 17. Time-to-event data.

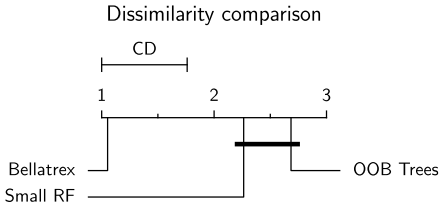


FIGURE 18. Regression tasks.

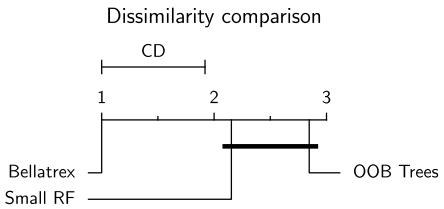


FIGURE 19. Multi-label classification data.

we showed that Bellatrex has (often significantly) greater dissimilarity compared to picking the same number of random trees from the ensemble, or picking the same number of best performing trees. This observed dissimilarity is remarkable, as Bellatrex proves to be able to extract diverse rules despite their optimal number K being optimised for increasing fidelity to the original RF instead. Furthermore, the high average dissimilarity suggests that the final extracted rules are not redundant, and that they give diverse possible explanations to the end user.

The limitations of our method are similar to those of other approaches that rely on tree ensembles. Firstly, the interpretability of the rules extracted by Bellatrex is challenged when such rules are too long. Another possible limitation lies in the computational complexity of the method: although a single prediction can be explained in a short time, running the procedure on a full dataset becomes computationally expensive. Directions of future work include the adaptation of Bellatrex to gradient boosting ensemble methods [47], [48], as well as extending Bellatrex to an even broader set of tasks, such as multi-event survival analysis [49], online learning [50], or network inference [51]. Additionally, it is our intention to explore new vector representations, as well as post-pruning procedures that can reduce the size of the rules without impacting on predictive performance.

APPENDIX

In the Appendix we include additional information regarding the employed datasets. Furthermore, we expand the information included in Tables 3-6, by reporting the dataset-specific results in terms of predictive performance, rule dissimilarity, complexity of the expalantions, and the outcome of the ablation study.

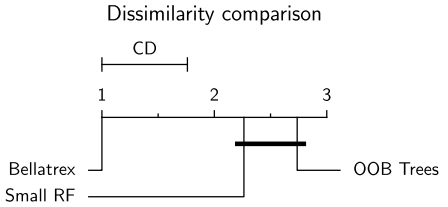


FIGURE 20. Multi-target regression data.

TABLE 27. Ablation study results on the time-to-event datasets. In bold, the best achieved performance.

	Bellatrex (original)	no pre selection	no dim. reduction	none of the two
addicts	**0.6512**	0.6411	0.6503	0.6451
breast cancer survival	0.6417	0.6060	**0.6440**	0.6245
DBCD	**0.7534**	0.7330	0.7072	0.5923
DLBCL	**0.6357**	0.5762	0.6118	0.5423
echocardiogram	0.4166	0.4199	0.4212	**0.4799**
FLChain	0.8330	**0.8331**	0.8317	0.8188
gbsg2	**0.7035**	0.7030	0.7025	0.6776
lung	0.6236	**0.6265**	0.6175	0.5949
NHANES I	0.8209	0.8136	**0.8215**	0.8070
PBC	**0.8467**	0.8417	0.8457	0.8076
rotterdam (excl. rec.)	**0.7961**	**0.7961**	0.7958	0.7870
rotterdam (incl. rec.)	0.9047	**0.9069**	0.9050	0.8945
veteran	0.7349	0.7277	**0.7389**	0.7130
whas500	0.7458	0.7412	**0.7466**	0.7373
average	**0.7224**	0.7122	0.7175	0.6946

TABLE 28. Ablation study results on the multi-label classification datasets. In bold, the best achieved performance.

	Bellatrex (original)	no pre selection	no dim. reduction	none of the two
birds	**0.8190**	0.7914	0.8184	0.7859
CAL500	**0.5428**	0.5375	0.5394	0.5320
emotions	**0.8205**	0.8000	**0.8205**	0.7523
enron	**0.7707**	0.7443	0.7576	0.7154
flags	0.7237	0.6877	**0.7317**	0.6854
genbase	0.9977	**0.9992**	0.9976	0.9984
langlog	**0.6075**	0.5848	0.5995	0.6071
medical	**0.9497**	0.9299	0.9365	0.9137
ng20	**0.9225**	0.9089	0.9082	0.8491
scene	**0.9195**	0.8993	0.9171	0.8377
slashdot	**0.8186**	0.7985	0.8035	0.7770
stackex chess	**0.6928**	0.6819	0.6776	0.6596
yeast	**0.6777**	0.6771	0.6623	0.6243
average	**0.7894**	0.7723	0.7823	0.7491

TABLE 29. Ablation study results on the regression datasets. In bold, the best achieved performance.

	Bellatrex (original)	no pre selection	no dim. reduction	none of the two
airfoil	**0.0374**	0.0389	0.0377	0.0427
AmesHousing	0.0230	0.0243	**0.0228**	0.0277
auto mpg	**0.0525**	0.0545	0.0526	0.0571
bike sharing	**0.0481**	0.0487	**0.0481**	0.0521
boston housing	0.0576	0.0587	**0.0569**	0.0620
california	0.0611	0.0629	**0.0610**	0.0654
car imports	0.0360	0.0367	**0.0359**	0.0403
Computer	0.0297	0.0294	**0.0293**	0.0305
concrete compress	0.0396	0.0418	**0.0393**	0.0449
concrete slump	0.0734	0.0744	**0.0727**	0.0742
ENB2012 cooling	0.0338	**0.0335**	0.0336	0.0338
ENB2012 heating	0.0111	0.0112	**0.0110**	0.0117
forest fires	**0.3032**	0.3084	0.3044	0.3081
PRSA data	0.0361	0.0363	**0.0360**	0.0388
slump dataset	0.0680	0.0688	**0.0663**	0.0730
students maths	0.1550	0.1554	**0.1538**	0.1623
wine quality all	0.0710	0.0711	**0.0692**	0.0748
wine quality red	0.0808	0.0825	**0.0776**	0.0792
wine quality white	0.0623	0.0614	**0.0611**	0.0645
average	0.0674	0.0684	**0.0668**	0.0707

TABLE 30. Ablation study results on the multi-target regression datasets. In bold, the best achieved performance.

	Bellatrex (original)	no pre selection	no dim. reduction	none of the two
andro	0.0911	0.0968	0.0899	**0.0865**
atp1d	0.0479	0.0483	**0.0467**	0.0545
atp7d	0.0443	0.0437	**0.0413**	0.0720
edm	0.1068	0.1083	**0.0982**	0.1037
enb	0.0225	0.0231	**0.0224**	0.0232
ENB2012	0.0234	0.0236	**0.0231**	0.0240
jura	**0.0678**	0.0703	0.0687	0.0732
oes10	**0.0220**	0.0237	0.0225	0.0269
oes97	**0.0296**	0.0310	0.0298	0.0350
osales	**0.0361**	0.0376	0.0362	0.0413
rf1	**0.0019**	0.0022	**0.0019**	0.0025
rf2	0.0031	0.0033	**0.0030**	0.0043
scm1d	0.0249	0.0264	**0.0247**	0.0288
scm20d	0.0296	0.0307	**0.0292**	0.0353
scpf	0.0105	0.0104	0.0105	**0.0103**
sf1	**0.0726**	0.0731	0.0727	0.0741
sf2	0.0318	**0.0313**	**0.0313**	0.0316
slump	0.1433	0.1387	0.1426	**0.1357**
wq	0.1610	0.1639	**0.1608**	0.1696
average	0.0511	0.0519	**0.0503**	0.0543

A. DATASET OVERVIEW

The following subsection includes some general properties of the dataset employed in our experiments. More specifically, we share dataset size, dimensionality of the instances, and label distribution. The number of labels is shared in the multi-output cases.

B. PERFORMANCE

Here, we report the dataset-specific performance for time-to-event, multi-label classification, single target regression, and multi-target regression data. Additionally, we provide the outcomes of the statistical testing procedure of the scenarios. Note that the detailed results on binary classification data are reported in the main body, Section VI-A.

C. COMPLEXITY

Here, we report the dataset-specific results mentioned in Section VI-B, and regarding to complexity of the explanations (as defined in (4)). Additionally, we report the outcomes of the statistical testing procedure, for binary classification, time-to-event, multi-label classification, single target regression, and multi-target regression data.

D. DISSIMILARITY

Here, we report the dataset-specific results for dissimilarity (as defined in (6)), as well as the outcomes of the statistical

testing procedure for all five considered scenarios. We show in bold the rulesets with the highest average dissimilarity.

E. ABLATION STUDIES

Here, we report the dataset-specific results for the ablation study, as well as the outcomes of the statistical testing procedure, for time-to-event, multi-label classification, single target regression, and multi-target regression data. Detailed results on binary classification datasets are reported in the main body, Section VI-D.

ACKNOWLEDGMENT

The authors would like to thank Niels Libeer for his help in implementing the UI for Bellatrex.

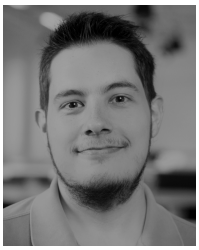
REFERENCES

[1] C. M. Cutillo, K. R. Sharma, L. Foschini, S. Kundu, M. Mackintosh, and K. D. Mandl, "Machine intelligence in healthcare: Perspectives on trustworthiness, explainability, usability, and transparency," *NPJ Digit. Med.*, vol. 3, no. 1, p. 47, 2020.
[2] D. Saraswat, P. Bhattacharya, A. Verma, V. K. Prasad, S. Tanwar, G. Sharma, P. N. Bokoro, and R. Sharma, "Explainable AI for healthcare 5.0: Opportunities and challenges," *IEEE Access*, vol. 10, pp. 84486–84517, 2022.
[3] A. Hanif, X. Zhang, and S. Wood, "A survey on explainable artificial intelligence techniques and challenges," in *Proc. IEEE 25th Int. Enterprise Distrib. Object Comput. Workshop (EDOCW)*, Oct. 2021, pp. 81–89.
[4] M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, and H. P. Beck, "The role of trust in automation reliance," *Int. J. Hum.-Comput. Stud.*, vol. 58, no. 6, pp. 697–718, Jun. 2003.
[5] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.
[6] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.
[7] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation," *J. Comput. Graph. Statist.*, vol. 24, no. 1, pp. 44–65, Jan. 2015.
[8] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?' Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.
[9] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017., pp. 1–10.
[10] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[11] P. Plonski. (2019). *Random Forest Vs Neural Network*. [Online]. Available: https://mljar.com/blog/random-forest-vs-neural-networkclassification/
[12] J. M. G. Taylor, "Random survival Forests," *J. Thoracic Oncol.*, vol. 6, no. 12, pp. 1974–1975, Dec. 2011.
[13] K. Dedja, F. K. Nakano, K. Pliakos, and C. Vens, "Explaining a random survival Forest by extracting prototype rules," in *Proc. ECML-PKDD, PharML Workshop*. Cham, Switzerland: Springer, 2021, pp. 1–8.
[14] M. Aria, C. Cuccurullo, and A. Gnasso, "A comparison among interpretative proposals for random Forests," *Mach. Learn. Appl.*, vol. 6, Dec. 2021, Art. no. 100094.
[15] M. Haddouchi and A. Berrado, "A survey of methods and tools used for interpreting Random Forest," in *Proc. Int. Conf. Smart Syst. Data Sci.*, 2019, pp. 1–6.
[16] H. A. Chipman, E. I. George, and R. E. Mcculloch, "Making sense of a Forest of trees," *Comput. Sci. Statist.*, vol. 1, pp. 84–92, Aug. 1998.
[17] A. Sies and I. Van Mechelen, "c443: A methodology to see a Forest for the trees," *J. Classification*, vol. 37, no. 3, pp. 730–753, Oct. 2020.
[18] M. Banerjee, Y. Ding, and A.-M. Noone, "Identifying representative trees from ensembles," *Statist. Med.*, vol. 31, no. 15, pp. 1601–1616, Jul. 2012.
[19] A. Moore, V. Murdock, Y. Cai, and K. Jones, "Transparent tree ensembles," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2018, pp. 1241–1244.
[20] S. Hara, "Making tree ensembles interpretable: A Bayesian model selection approach," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2018, pp. 77–85.
[21] I. Mollas, N. Bassiliades, and G. Tsoumakas, "Conclusive local interpretation rules for random Forests," *Data Mining Knowl. Discovery*, vol. 36, no. 4, pp. 1521–1574, Jul. 2022.
[22] N. Mylonas, I. Mollas, N. Bassiliades, and G. Tsoumakas, "Local multi-label explanations for random Forest," 2022, *arXiv:2207.01994*.
[23] C. Bénard, G. Biau, S. Da Veiga, and E. Scornet, "SIRUS: Stable and interpretable RUle set for classification," *Electron. J. Statist.*, vol. 15, no. 1, pp. 427–505, Jan. 2021.
[24] J. Obregon and J.-Y. Jung, "RuleCOSI+: Rule extraction for interpreting classification tree ensembles," *Inf. Fusion*, vol. 89, pp. 355–381, Jan. 2023.
[25] A. Agarwal, Y. S. Tan, O. Ronen, C. Singh, and B. Yu, "Hierarchical shrinkage: Improving the accuracy and interpretability of tree-based methods," 2022, *arXiv:2202.00858*.
[26] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," *Mach. Learn.*, vol. 73, no. 2, pp. 185–214, Nov. 2008.
[27] P. Wang, Y. Li, and C. K. Reddy, "Machine learning for survival analysis: A survey," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–39, 2019.
[28] M. Leblanc and J. Crowley, "Survival trees by goodness of split," *J. Amer. Stat. Assoc.*, vol. 88, no. 422, pp. 457–467, Jun. 1993.
[29] R. Peto and J. Peto, "Asymptotically efficient rank invariant test procedures," *J. Roy. Stat. Soc., Ser. A*, vol. 135, no. 2, p. 185, 1972.
[30] S. Vassilvitskii and D. Arthur, "K-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2006, pp. 1027–1035.
[31] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Exp. Syst. Appl.*, vol. 36, no. 2, pp. 3336–3341, 2009.
[32] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, Sep. 2019.
[33] Y. Zhou and G. Hooker, "Interpreting models via single tree approximation," 2016, *arXiv:1610.09036*.
[34] (2022). *Scikit-Learn Developers, Decomposing Signals in Components*. [Online]. Available: https://scikit-learn.org/1.1/modules/decomposition.html
[35] (2022). *Scikit-Learn Developers, Nearest Neighbors*. [Online]. Available: https://scikit-learn.org/1.1/modules/neighbors.html
[36] T. Raghunathan, J. Lepkowski, J. Hoewyk, and P. Solenberger, "A multivariate technique for multiply imputing missing values using a sequence of regression models," *Surv. Methodol.*, vol. 27, no. 1, pp. 85–96, 2000.
[37] D. R. Cox, "Regression models and life-tables," *J. Roy. Stat. Soc., Ser. B*, vol. 34, no. 2, pp. 187–202, Jan. 1972.
[38] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017, *arXiv:1702.08608*.
[39] C. Molnar, G. Casalicchio, and B. Bischl, "Interpretable machine learning—A brief history, state-of-the-art and challenges," *Commun. Comput. Inf. Sci.*, vol. 1323, no. 1, pp. 417–431, 2020.
[40] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, "Metrics for explainable AI: Challenges and prospects," 2018, *arXiv:1812.04608*.
[41] H. Deng, "Interpreting tree ensembles with inTrees," *Int. J. Data Sci. Anal.*, vol. 7, no. 4, pp. 277–287, Jun. 2019.
[42] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.
[43] D. Harrison, Jr., and D. L. Rubinfeld, "Hedonic housing prices and the demand for clean air," *J. Environ. Econ. Manag.*, vol. 5, no. 1, pp. 81–102, 1978.
[44] M. Carlisle. (2019). *Racist Data Destruction?* [Online]. Available: https://medium.com/docintangible/racist-data-destruction-113e3eff54a8
[45] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Apr. 2006.
[46] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
[47] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. Berlin, Germany: Springer, 2009.
[48] J. Browlee. (2020). *Gradient Boosting With Scikit-Learn, XGBoost, Lightgbm, and CatBoost*. [Online]. Available: https://machinelearningmastery.com/gradient-boosting-with-scikitlearn-xgboost-lightgbm-and-catboost/
[49] M. Crowder, *Multivariate Survival Analysis and Competing Risks*. Boca Raton, FL, USA: CRC Press, 2012.

[50] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem, and A. Bifet, "River: Machine learning for streaming data in Python," *J. Mach. Learn. Res.*, vol. 22, no. 1, pp. 4945–4952, 2021.
[51] K. Pliakos and C. Vens, "Network inference with ensembles of bi-clustering trees," *BMC Bioinf.*, vol. 20, no. 1, pp. 1–12, Dec. 2019.

KLEST DEDJA received the B.Sc. degree in mathematics from the University of Udine, in 2016, and the M.Sc. degree in applied mathematics from TU Delft, in 2019. He is currently pursuing the Ph.D. degree with the ITEC, imec Research Group at KU Leuven, Belgium. His research interests include machine learning applications to healthcare, explainable AI, and survival analysis.

FELIPE KENJI NAKANO received the bachelor's degree in computer science from Londrina State University, the master's degree in artificial intelligence from the Federal University of São Carlos, and the Ph.D. degree in biomedical sciences with emphasis on health and technology from KU Leuven. He provides theoretical contributions in the field of active learning, semi-supervised learning, structured-output prediction, and survival analysis. His research interest includes machine learning for (bio)-medical applications, such as prediction tasks in intensive care units, protein function prediction, and transposable elements.



KONSTANTINOS PLIAKOS received the Diploma degree in electrical and computer engineering and the M.Sc. degree in computational intelligence from AUTH, in 2011 and 2015, respectively, and the Ph.D. degree from KU Leuven, Belgium, in 2019. He is currently a Postdoctoral Researcher with KU Leuven. His research interests include multi-label and multi-target prediction, supervised and semi-supervised learning, dimensionality reduction, recommender systems, and biomedical network mining.

CELINE VENS received the Ph.D. degree in computer science (machine learning) from KU Leuven, Belgium, in 2007. She is currently an Associate Professor with the Faculty of Medicine, KU Leuven, and the ITEC-imec Research Group at KU Leuven. Her research interests include multi-label, multi-target, hierarchical prediction, recommender systems, tree ensemble learning, survival analysis, and biological network mining.