Unboxing Tree Ensembles for interpretability: a hierarchical visualization tool and a multivariate optimal re-built tree

Giulia Di Teodoro[a], Marta Monaci[a], Laura Palagi[a]

[a]*Department of Computer, Control and Management Engineering Antonio Ruberti (DIAG), Sapienza University of Rome, 00185, Rome, Italy*
{*giulia.diteodoro;marta.monaci;laura.palagi*}*@uniroma1.it*

Abstract

The interpretability of models has become a crucial issue in Machine Learning because of algorithmic decisions' growing impact on real-world applications. Tree ensemble methods, such as Random Forests or XgBoost, are powerful learning tools for classification tasks. However, while combining multiple trees may provide higher prediction quality than a single one, it sacrifices the interpretability property resulting in "black-box" models. In light of this, we aim to develop an interpretable representation of a tree-ensemble model that can provide valuable insights into its behavior. First, given a target tree-ensemble model, we develop a hierarchical visualization tool based on a heatmap representation of the forest's feature use, considering the frequency of a feature and the level at which it is selected as an indicator of importance. Next, we propose a mixed-integer linear programming (MILP) formulation for constructing a single optimal multivariate tree that accurately mimics the target model predictions. The goal is to provide an interpretable surrogate model based on oblique hyperplane splits, which uses only the most relevant features according to the defined forest's importance indicators. The MILP model includes a penalty on feature selection based on their frequency in the forest to further induce sparsity of the splits. The natural formulation has been strengthened to improve the computational performance of mixed-integer software. Computational experience is carried out on benchmark datasets from the UCI repository using a state-of-the-art off-the-shelf solver. Results show that the proposed model is effective in yielding a shallow interpretable tree approximating the tree-ensemble decision function.

Keywords:
Tree ensembles; Visualizing Forests; Optimal Trees; Mixed-Integer Programming; Machine Learning; Interpretability.

1. Introduction

When building Machine Learning (ML) models in supervised learning, it is becoming more and more important to balance accuracy and interpretability. There is a growing number of sensitive domains where a detailed understanding of the model and the outputs is as important as the accuracy in prediction [8]. Following [10], interpretability is defined as "the ability to explain or to present the decision in understandable terms to a human". Generally, models with high accuracy are more complex and less interpretable. This trade-off is evident when comparing Decision trees (DT) and Tree ensembles (TE) (e.g., Random Forests, XGBoost). Decision

Trees (DT) (see [6, 25, 26]) have high interpretability since their construction process is simple, intuitive, and can be easily visualized. They are white-box models easily explained by Boolean logic; indeed, the prediction made by the tree is a conjunction of predicates. The disadvantages are that they often overfit, do not have good out-of-sample predictive capabilities, can grow exponentially, and have high variability. On the other hand, Tree Ensemble (TE) models such as Random Forests [4], and XGboost [13] constitute one of the most widely used techniques for regression and classification tasks. By aggregating many decision trees, tree ensemble techniques substantially increase predictive capabilities. Indeed, including the randomness element and the ensemble procedure used to aggregate the individual tree predictions reduces the model variance. Thus, TEs can achieve high levels of accuracy at the expense of lower interpretability. Indeed, a TE is a black-box model, not interpretable through parameters or its functional form. TE models are used in many fields, such as medical or financial, where opaque and redundant decisions could be highly harmful, and an easily understood interpretation of the model's predictions could significantly impact the final decision. Hence the ability to understand the interactions between predictors and responses used in a TE model is an important issue when undertaking a decision-making process.
In [15], a list of desiderata of a surrogate predictive model is reported as follows.

1. Interpretability: to what extent the model and/or its predictions are human-understandable. We adopt as a measure of interpretability the complexity of the predictive model in terms of the model size (depth or the number of leaves) and the sparsity of the predictors used to construct the decision rules.
2. Accuracy: to which extent the model accurately predicts out-of-sample instances.
3. Fidelity: to what extent can the model accurately imitate a black-box predictor.

Measures of Accuracy and Fidelity are based on standard KPIs such as error rate, and F1-score having as target values the ground truth and the predicted values, respectively.
The paper is organized as follows. In Section 2, we review state of the art in Interpretative models for Tree Ensemble, including the main feature's VI indicators. In Section 3, we focus on our contribution. In Section 4, we state the problem and the main basic definitions and tools used. Section 5 is devoted to the description of the visualization toolbox, and Section 6 defines the model for the optimal re-built tree. The formulation is strengthened in Section 7. Finally Section 8 reports numerical results on a standard benchmark of test problems from the UCI collection.

2. State of the art on Interpretative models for Tree Ensemble

Several tools and methods in the literature are used to visualize and interpret tree ensembles and, in particular, Random Forests (RFs). In this section, we aim to review only some of the possible approaches, and we refer to the recent survey [1] for a more detailed comparison. According to [1], the approaches can be mainly divided into:

- *Internal processing*, which includes methods that aim to provide a global overview of the model through measures helpful in interpreting the results obtained;
- *Post-Hoc approaches*, which aim to identify a relationship structure among response and predictors and include the construction of a single surrogate model that approximates the original TE prediction function.

Internal processing. Among measures for interpreting the prediction results of a TE, the feature importance approach assigns each feature j a score to indicate its impact on predicting the output y. These approaches produce a features ranking which is usually obtained by modifying in some way the value of each feature at the time and evaluating the impact on the quality of the tree ensemble. Mean Decrease accuracy (MDA) and Mean Decrease Impurity (MDI) [4, 14, 22] and Partial Dependence Plot (PDP) [13] provide examples of such measures.
A different approach to calculating the importance of variable j, presented in [19], consists in randomly changing the assignment of the samples to the children nodes whenever the variable j is used in the splitting rule of a node and evaluating the change in the accuracy of the predictions. The more significant the change in the prediction, the more the importance of the variable j. The underlying idea is that a random assignment of samples to nodes at shallower depths leads to more significant perturbations to the final assignment of samples to leaves and, thus, in the classification.
In [20], the Minimal Depth (MD) is introduced as an alternative measure of the importance based solely on the structure of the trees in the forest rather than the goodness of fit of the prediction. The features' importance is determined by the level at which they are used for splitting in the trees' nodes, where the more frequent use at shallow depths, the more important. The idea is that the higher the probability of a feature appearing in the first levels of the trees, the larger the impact on accuracy and also the importance accordingly to the noisy definition in [19]. The Surrogate Minimal Depth, proposed in [28], generalizes MD by using surrogate variables, as defined by Breiman in [6], to obtain the minimal depth.
Another way to provide additional information about the tree ensemble is the *Proximity measure* of pair of samples, which is a weighted average of the number of trees in the TE model in which the samples end up in the same leaf. This concept was introduced first for Random Forest [5], where weights can be all equal, and later extended to Gradient Boosted Trees [30]. The proximity measure takes values between 0 and 1. Associated with proximity, a similarity measure among samples can be defined as *TE distance* of pairs of samples. The TE distance is calculated as (1- TE proximity) and is a pseudo-metric [30]. The basic idea is that similar observations should be found more frequently in the same terminal nodes than dissimilar observations. Based on this measure, training samples can be visualized through a Multi-dimensional Scaling (MDS) plot so that users can intuitively observe data clusters and outliers identified by the TE model. Other visualization toolkits have been developed to graphically represent the measures shown above specifically for Random Forests: *randomForest* tool in R [21], *ggRandomForests* package in R [12], and more recently *iForest* tool that allows interactive visualization of RF in Python [32]. These tools provide visualization of feature importance and partial dependence information, ranges of splitting values for each feature, similarities, and structure of decision paths, distribution of training data, and an interactive inspection of the model.

Post-Hoc approaches. Another methodology used to aid the interpretability of TEs is represented by post-hoc approaches that aim to identify the relationship between the predicted output and the predictors. These include approaches that do *size-reduction* of the forest, those that *extract rules*, and those that aim for *local explainability*.
The model we propose in this paper falls into the size-reduction class. Several approaches have been proposed in the literature to make forests interpretable, as the many trees in the forest make it difficult to understand the decision path leading to predictions. These approaches aim to reduce the size of TE while maintaining its predictive capability. This is an NP-hard

problem [29]; in some cases, the smaller ensemble may even perform better [33]. A huge branch of literature is available on these topics, and we refer to [1] and references therein for a recent review of the main approaches

We are interested mainly in the stream of "born-again" trees, which involves building a single decision tree, called *representer tree*, that mirrors the behavior of a pre-existing tree ensemble over all its feature space. Born-again trees were originally proposed in [7] as the problems of: "giving a probability distribution $\mathcal{P}$ on the space of input variables x, find the tree that best represents $f(x)$" where $f(x)$ is a given prediction. TEs are used to manufacture new pairs (x^i, y^i), and a representer tree is constructed with a prediction accuracy on the manufactured samples close to the tree ensemble's accuracy rather than the ground truth. However, the tree can grow too much and might result in a not interpretable model.

Recently, in [31], the concept of the born again tree has been explored from an optimization viewpoint. *Faithfulness* is defined as the capacity of the representer tree to reproduce precisely the decision function $f(x)$ of the TE on the whole space of the features and not only on the samples. In both approaches [7, 31], a univariate tree is constructed where the depth or number of leaves is controlled by optimizing exactly or approximately the complexity and by post-pruning to reduce the size. In [31], it is proved that the problem of determining a faithful optimal decision tree of minimal size (where either depth or number of leaves, or any hierarchy of these two are used as complexity measures) is $\mathcal{NP}$-hard and that the depth of such an optimal tree is bounded above by the sum of the depth of the trees in the forest (the bound is tight). As the Problem is NP-hard, the computational time of the dynamic programming algorithm proposed in [31] will eventually increase exponentially with the number of features, and a heuristic approach is proposed, which is guaranteed to be faithful but not necessarily minimal in size. A post-pruning phase is used on the set of six problems studied that seems not to affect the quality of the predictions but to significantly simplify the born-again trees. Authors observed that born-again decision trees contain many inexpressive regions which do not contribute to effectively classifying samples. The purpose of these regions and their contribution to the generalization capabilities of random forests is not clear yet.

3. Our contribution

Our contribution is twofold and follows the line of research aimed at creating an interpretable representation of a tree-ensemble model able to provide valuable insights into its behavior. Following the classification proposed in [1] of interpretative methods for Random Forests, we can divide our contribution into two parts interacting with each other.

In particular, given a Tree Ensemble (TE) model, we provide

- `VITE`: a hierarchical VIsualization tool for TE that depends solely on the structure of the tree and aims to visualize how features are used within the forest;
- `MIRET`: a surrogate Multivariate Interpretable RE-built optimal Tree to gain interpretability on the relation between the input features and the TE outcomes.

The visualization tool `VITE` fits into the framework of the visualization tools developed in the literature to better understand the dynamics inside the TE. We drew our inspiration from the concept of *Minimal Depth*, where the feature's importance is determined by the smallest depth at which it is used in a node's splitting rule for the first time. Indeed, since variable selection at splitting nodes is based on impurity indicators, those more frequently used at high

levels produce the most significant impurity decrease. However, we are interested in having a hierarchical view of the role of features in the trees composing the TE. Indeed, in a tree, the *Minimal Depth* (MD) of a feature j is a nonnegative random variable taking values $\{0, \dots, D\}$, where D is the depth of the tree. MD measures the distance from the root node to the root of the closest maximal subtree having j as a splitting feature. Essentially, it measures how far a sample moves down in the tree before encountering the first split on j. In this way, the information on how often a feature is used in the tree, namely how many subtrees having j as root nodes are present, is wholly lost. Indeed, it is possible for a feature to appear multiple times at lower levels in the tree after its initial appearance, which defines the Minimum Depth, or conversely, it may not appear again after its first appearance. Following this idea, we propose using the frequency of each feature's usage at each node or level in the trees of the TE as a measure of the feature's importance. This is based on the notion that features appearing more frequently at nodes closer to the root are likely to play a significant role in prediction outcomes. We derive a tool for visualizing TE features' frequency on a single tree that gives a view at a glance of the hierarchical structure of the TE. Our tool is based on the heatmaps construction considering features' frequency. The TE visualization of `VITE` is intended to be an addition to the existing visualization tools for further facilitation of model interpretation in a graphic key. `VITE` gives an immediate glimpse of the overall structure of the forest without the need to analyze individual decision pathways to extrapolate what might be the reason for different classifications. In addition, it allows seeing in a single view which are the features used most in the trees of the forest and the split point range of each feature, allowing one to reflect on the different ranges' effect on the predictions.

The Multivariate Interpretable RE-built Tree `MIRET` fits in the framework of born-again trees, defining an *optimal representer tree*. A representer tree aims to replace a tree ensemble classifier with a newly constructed single tree that can reproduce, in some sense, the behavior of the TE. Differently from preceding approaches in [7, 31], we propose constructing a tree with fixed depth, thus preventing it from growing excessively and becoming difficult to understand. To this aim, we consider surrogate trees with fixed depth that uses oblique splits to partially regain the freedom lost by fixing complexity.
More in detail, based on the target TE, we present a mixed-integer linear programming (MILP) formulation for learning a Multivariate Interpretable RE-built tree with the *same maximum depth* D of TE, and that accounts for the information derived from the target TE.
To this end, we extract knowledge from the target TE and inject information into our tree model. In particular:

- Following the TE voting procedure, which leads to the prediction, we maximize fidelity to TE. Namely, we minimize the misclassification of each sample with respect to its predicted class extracted from the ensemble model.

- According to the measures used in the visualization tool, we detect each feature's usage frequency along the forest's trees. Through penalization in the objective function, we drive the selection of features based on their frequencies. Further, we give our model only a subset of the most frequent features for each tree level. In this way, we promote selecting the most representative features of the TE while further inducing the sparsity of the branching hyperplanes. Indeed, sparsity is a core component of interpretability [27], and having sparser decision rules allows the end user to identify better the key factors influencing the outcome.

- To encourage the partition of the feature space as performed by the TE, we calculate the proximity of each sample pair in the TE. Based on this, we ensure that pairs of samples with proximity greater than a specified threshold are placed in the same final leaf of our tree, i.e., in the same final feature space partition.

In section 7, a strengthened MILP formulation is proposed to improve the computational performance of MILP algorithms. This approach provides a simple yet effective way to interpret the predictions of a complex ensemble model, yielding a shallow interpretable tree able to give insights about the features that most affect the classification while approximating the RF decision function.

4. Basic definition and preliminaries

We focus on binary classifiers, and we assume that we are given a training dataset

$$\{(x^i, y^i) \in \mathbb{R}^{|\mathcal{J}|} \times \{-1, 1\},\ i \in \mathcal{I}\},$$

where $\mathcal{J}$ is the index set of the features, and w.l.o.g. we normalize the feature values $x^i, i \in \mathcal{I}$ in the interval $[0, 1]$.
The training data is used to construct a predictor $f(x) : \mathbb{R}^{|\mathcal{J}|} \to \{-1, 1\}$. We are interested in Tree (T) and Tree Ensemble (TE) models, and in this section, we review the basic concepts of Ts and TEs that are needed in the following Sections.
Decision trees yield a partition of the feature space $[0, 1]^{|J|}$ by applying hierarchical disjunctive splittings. A tree is characterized by a maximum depth D so that nodes are organized into at most D levels in $\mathcal{D} = \{0, 1, \ldots, D\}$. A decision tree is composed of *branch nodes* $\mathcal{B}$ and *leaf nodes* $\mathcal{L}$. A branch node $t \in \mathcal{B}$ applies a splitting rule on the samples in the node, while a leaf node $\ell \in \mathcal{L}$ acts as a collector of samples. In each leaf node, a class is assigned to all the samples contained within it using a simple rule, such as the majority vote.
The most common decision trees are univariate, employing axis-aligned splits. In this case, the splitting rule in a branch node refers to the selection of a single feature and threshold such that the training samples are partitioned among the child nodes to maximize the decrease of node impurity, measured by some indicators (e.g. the Gini index). In recent years, starting from the very first paper [2], multivariate (or oblique) decision trees have been proposed which may involve multiple features per split. Each branching rule is defined by hyperplane $h_t(x) = a_t^T x^i + b_t$, where $a_t \in \mathbb{R}^{|\mathcal{J}|}$ and $b_t \in \mathbb{R}$ (and the apex ^{T} denotes transposition of the vector). These multivariate splits are much more flexible than univariate ones, at the expense of lower interpretability. However, the problem of determining them is much more complicated, and it can be tackled using Mixed Integer Linear Programming. Of course, univariate models are a subclass of multivariate ones, thus in general, the splitting rule at a node $t \in \mathcal{B}$ can be represented as

$$\text{if } h_t(x^i) \begin{cases} \leq 0 & x^i \text{ follows the left branch of } t \\ > 0 & x^i \text{ follows the right branch of } t \end{cases}. \tag{1}$$

The hierarchical tree structure uses hyperplane splits that recursively partition the feature space into disjoint regions, each of which corresponds to a leaf node in the tree. The obtained tree is then used to classify unseen data. A root-to-leaf unique path (decision path), which is a conjunction of predicates, leads to the prediction $\hat{y}_T \in \{-1, 1\}$ made by the tree.

We define a *Tree Ensemble* TE as a collection of tree estimators $\mathcal{E}$ with associated weights w_e, with $e \in \mathcal{E}$. The TE decision function $F_{TE} : \mathbb{R}^{|\mathcal{J}|} \to \{-1, 1\}$ is obtained as the weighted majority vote of the decision function of its trees (ties are usually broken in favor of the smaller index).

5. VITE: a hierarchical VIsualization tool for TE

The underlying idea of our graphical representation has been inspired by the definition of variable importance given by Ishwaran [18] and the definition of Minimal Depth [20]. The importance of a variable is related to the level at which it is used as variable splitting, assuming that splitting at the root node or nodes at shallower tree levels has a greater impact on variable importance than those used at deeper levels. Indeed features that are used overwhelmingly at the shallower depths are those that decrease impurity the most and therefore play a greater role in the classification of the samples. We aim to generalize this principle and consider all the times a feature is used in the trees, and not just the first appearance. This gives a hierarchical perspective on the role played by features in the trees that make up the TE. Indeed, a feature can show up multiple times at lower levels in the tree after its initial appearance, which determines its Minimal Depth. On the other hand, it may not reappear after its initial appearance. We aim to gain insight into the role of features along all the levels of the trees.
Drawing inspiration from this concept, we consider for each feature the percentage frequency with which it is selected, at each level, in the forest's trees. We define the $j-$th feature *level frequency* for each $d \in \mathcal{D}$ as the ratio between the number of times the feature j is used at level d of all the trees in $\mathcal{E}$ and the total number of nodes that effectively apply a split at level d, thus not accounting for the possible leaves appearing at level d (pruned nodes). In this definition, we must account for the type of TE considered. Indeed, as remarked in [30], in Random Forest each tree is generated by an identical and independent process and contributes equally to the prediction. However, this is no longer true in boosted trees where individual trees are obtained by a boosting process which makes trees not independent and with a different contribution to the overall prediction. This aspect can be managed by weighting the contribution of each tree similarly to the approach proposed in [30] for the Proximity measure. Furthermore, each decision tree in a TE can be allowed to use only a limited random subset of features with cardinality as candidates for splitting [17]. Again, in this case, a weight can be used to compensate for the bias as proposed for the Minimal Depht in [20]. To formally introduce the level frequency of a feature, we assume, for the sake of simplicity, that nodes in each tree are numbered according to breadth-first indexing (increasing from left to right at each level d), starting from the root node, which is numbered zero, so that $t \in \mathcal{T} = \{0, 1, 2, \ldots, 2^{D+1} - 1\}$. Let us define the indicator function for each $j \in \mathcal{J}, t \in \mathcal{T}, e \in \mathcal{E}$, as

$$\mathbb{1}(j, t, e) = \begin{cases} 1 & \text{if feature } j \text{ is used at node } t \text{ of the tree } e \\ 0 & \text{otherwise} \end{cases}$$

and for each tree $e \in \mathcal{E}$, let $\mathcal{B}^e(d)$ be the set of nodes at level d of the tree e which effectively apply a splitting rule. Thus, we have the following definition.

Definition 5.1 (Level Frequency)**.** *The frequency $f_{d,j}$ of a feature j at a level $d \in \{0, \ldots, D-1\}$ is*

$$f_{j,d} = \frac{1}{\sum_{e \in \mathcal{E}} |\mathcal{B}^e(d)|} \sum_{e \in \mathcal{E}} w^e \sum_{t \in \mathcal{B}^e(d)} \mathbb{1}(j, t, e), \tag{2}$$

where w^e $\forall e \in \mathcal{E}$ are non-negative weights that take into account (i) the probability of a feature being sampled as a candidate for the splitting rule in a level of the tree $e \in \mathcal{E}$ and (ii) the contribution of each tree $e \in \mathcal{E}$ in predicting the outcome.

According to this definition, we can define the level frequency matrix of dimension respectively $|\mathcal{J}| \times D$ with elements $\{f_{j,d}\}_{i\in\mathcal{J},d\in\mathcal{D}}$. Each column of the matrix sums up to 100%.
We represent this matrix with a heatmap where the darker colors represent the more used features at the level d.
To obtain a deeper view of the features' use in the TE we can consider a more specific definition of the frequency for each node t in the tree. We define the $j-$th feature *node frequency* for each $t \in \mathcal{T}$ as the ratio between the number of times the feature j is used at node t in all the trees in $\mathcal{E}$ and the total number of nodes t that effectively splits. As in the definition 5.1, we can weigh the contribution of each tree to account for a possible different role in the TE. Thus, we have the following definition.

Definition 5.2 (Node Frequency)**.** *The frequency $f_{t,j}^{node}$ of a feature $j \in \mathcal{J}$ at a node $t \in \mathcal{T}$ is*

$$f_{t,j}^{node} = \frac{1}{\sum_{e\in\mathcal{E}}\sum_{j\in\mathcal{J}} \mathbb{1}(j,t,e)} \sum_{e\in\mathcal{E}} w^e \mathbb{1}(j,t,e), \tag{3}$$

where $\sum_{e\in\mathcal{E}}\sum_{j\in\mathcal{J}} \mathbb{1}(t,j)$ is equal to the number of times the node t splits in the forest and w^e $\forall e \in \mathcal{E}$ are non-negative weights that take into account (i) the probability of a feature being sampled as a candidate for the splitting rule in a node of the tree $e \in \mathcal{E}$ and (ii) the contribution of each tree $e \in \mathcal{E}$ in predicting the outcome.

We use this definition to obtain a visualization of the TE in the form of a single tree of depth D where for each node $t \in \mathcal{T}$ we report the heatmap of the features' node frequency matrix with elements $\{f_{t,j}^{\text{node}}\}_{i\in\mathcal{J}}$. Also in this case, for each $t \in \mathcal{T}$ we have $\sum_{j\in\mathcal{J}} f_{t,j}^{\text{node}} = 100\%$. Of course, the leaf nodes which are at level D are not represented since there are no splits. However, in case a terminal node appears in upper levels, from 0 to $D-1$, it is reported with values $f_{t,j}^{\text{node}} = 0$ for all $j \in \mathcal{J}$. As additional information, we recover from the TE the ranges of the threshold b_j for each feature j used in the trees of the TE at each node $t \in \mathcal{T}$. This is reported in the visualized tree, as an interval $[l_j, u_j]$ near the name of each feature x_j which is used at node t. When $l_j = u_j$, a single value is reported. The interval gives us additional information: for each tree in the TE, whenever a features j is used at node t we trivially have that

$$x_j^i \begin{cases} \leq l_j & x^i \text{ always goes to the left branch} \\ \in (l_j, u_j] & \text{uncertain (grey choice)} \\ > u_j & x^i \text{always goes to the right branch} \end{cases}$$

This can give additional insight that can be analyzed by experts in the field under study. Indeed it may allow identifying easily understandable decision paths for samples x^i whose features j fall into the external intervals of the $(-\infty, l_j]$ or $(u_j, +\infty)$ for all $j \in \mathcal{J}$.
We report a toy example of the use of `VITE` for the construction of the frequency matrices and of the representative tree in Figure 1 (without considering the thresholds). Here we assume to have a simple Random Forest composed of $E = 3$ trees with depth $D = 3$, and samples $x^i \in \mathbb{R}^4$, namely $\mathcal{J} = \{1, 2, 3, 4\}$. We use weights $w^e = 1$ in the definitions of frequencies. In

the example, feature x_1 and feature x_2 are the ones that are most frequently used in the root node and at shallower depths. Feature x_3 appears only on level $d = 1$ and feature x_4 is not used at all in the forest. From this representation, we can derive considerations of the importance of features in a similar way as compared to the previous heatmap. This illustration allows us to have a greater level of detail in that it is possible to analyze how many variables are used in the nodes splitting rule.

As a more significant example, we apply the `VITE` tool for visualizing a Random Forest with 100 tree estimators of maximum depth $D = 3$ trained on Cleveland dataset [11], which has $x \in \mathbb{R}^{13}$, inhibiting the random sampling of features in the trees of the forest. In Figure 2, we report the heatmap representing the level frequency matrix; in Figure 3 we report the representative tree of depth $D = 3$ where at each node t, the node frequency of each feature j is reported together with the interval $[l_j, u_j]$.

The source code and the data to use the VITE tool are available at `https://github.com/gditeodoro/VITE`.

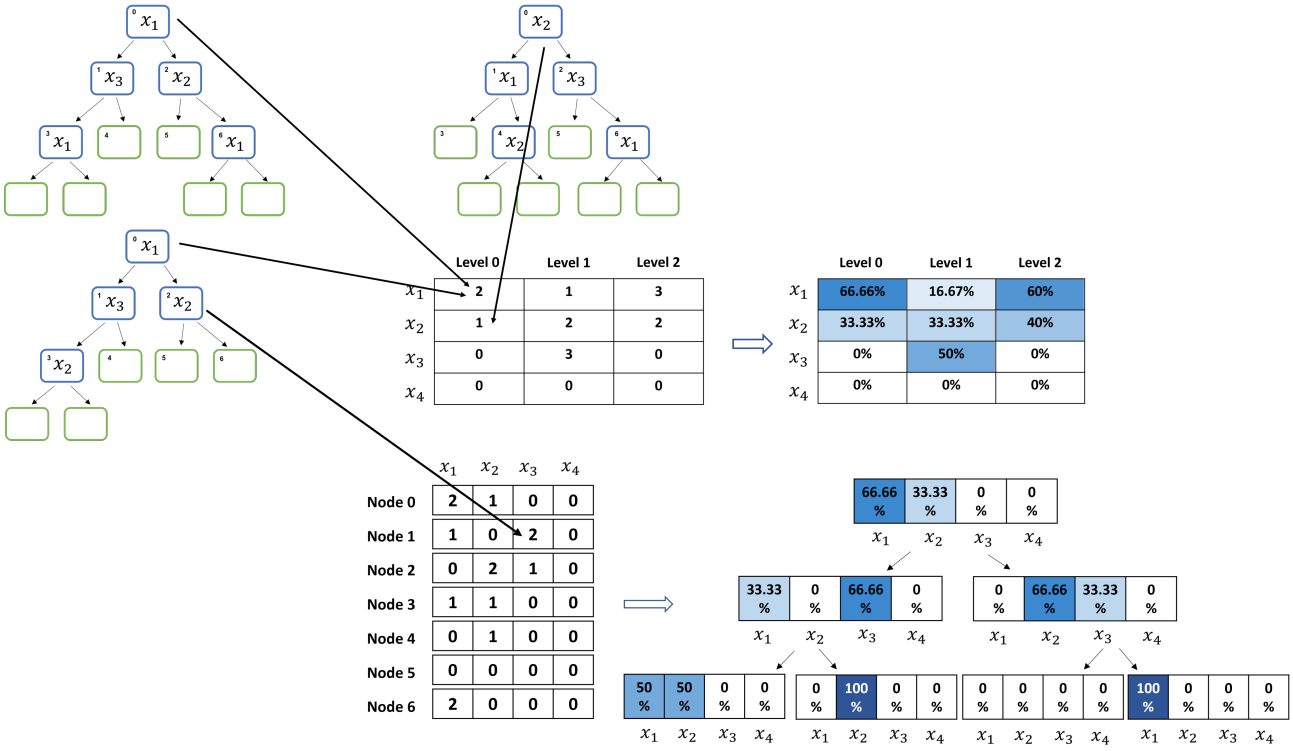

Figure 1: Construction of the features' usage heatmap at different depths in the TE

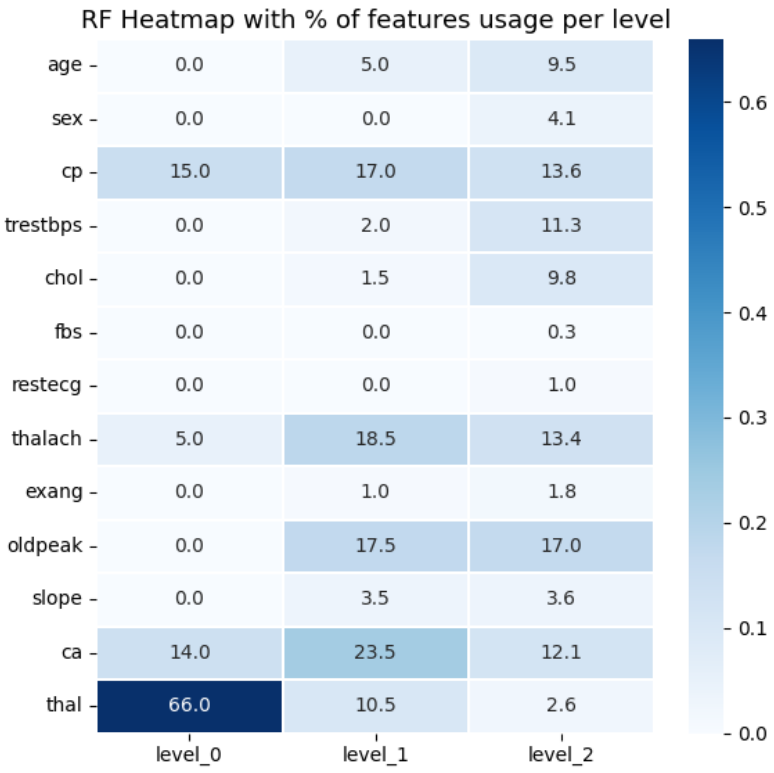

Figure 2: Heatmap of features frequency (%) at different tree levels in the Random Forest for the Cleveland dataset

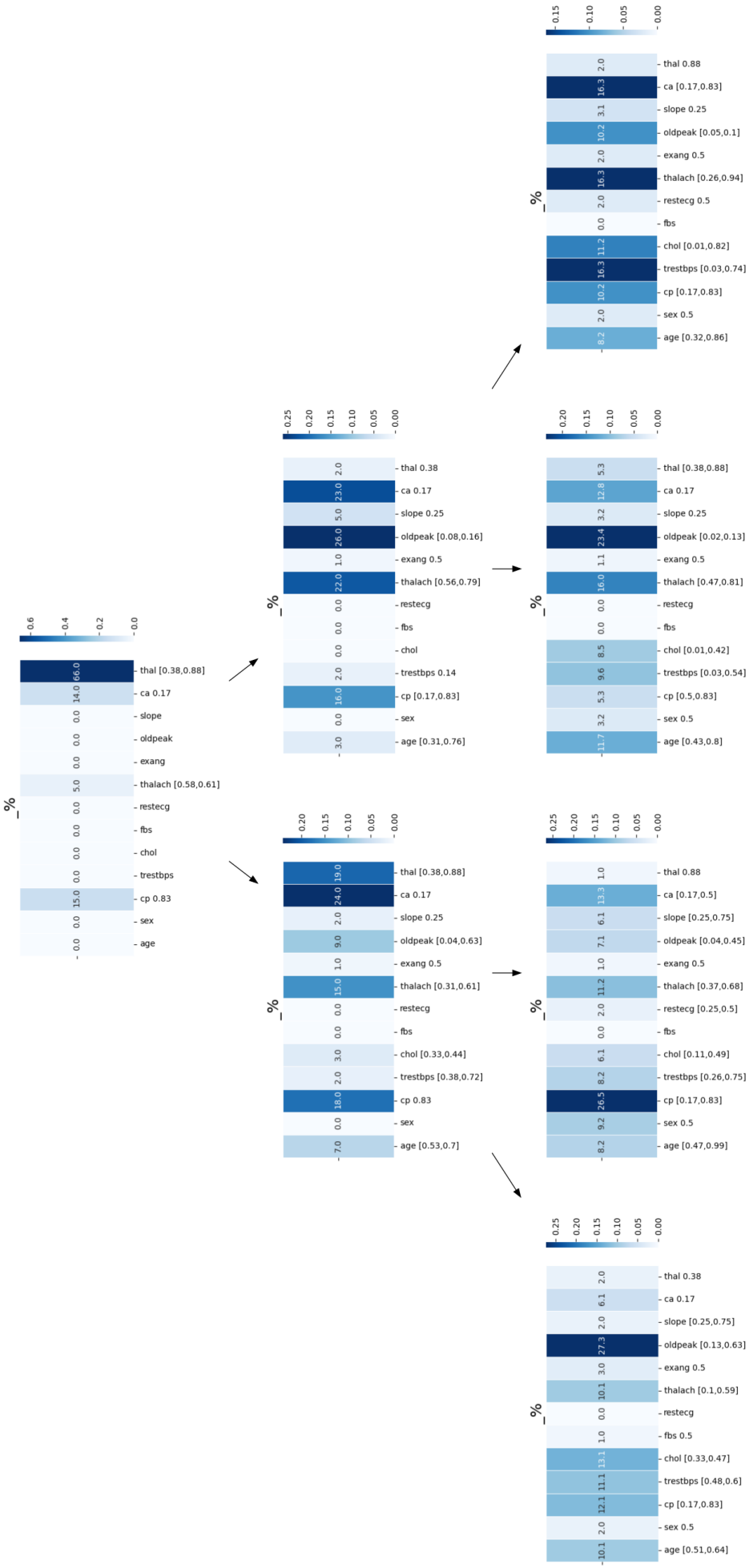

Figure 3: Visualize the features' usage on a single tree with given depth for the Cleveland dataset

6. MIRET: a Multivariate Interpratable REbuilt optimal Tree

6.1. Introduction

In this section, we proposed a Mixed Integer Linear Programming (MILP) formulation for learning a single optimal multivariate tree as a surrogate model of the ensemble classifier.
In more detail, let TE be a *target Tree Ensemble model*, with $|\mathcal{E}|$ tree estimators of maximal depth D trained on the given dataset and let $F_{TE} : \mathbb{R}^{|\mathcal{J}|} \to \{-1, 1\}$ be the decision function of TE.
The aim is to provide a shallow and interpretable surrogate multivariate tree model T, of the same maximal depth D of TE, with decision function $F_T : \mathbb{R}^{|\mathcal{J}|} \to \{-1, 1\}$, which is able to reproduce as much as possible the TE predictions on the training set. Further, we aim to enforce in the splitting hyperplanes the use of the most informative features.
As a first step, we formalize the definition of *fidelity* we are using.

Definition 6.1 (Training data fidelity)**.** *Given a tree ensemble TE, we say that a multivariate decision tree T of depth D is faithful to training data w.r.t. TE, when the prediction $F_{TE}(x^i) = F_T(x^i)$ for all $i \in \mathcal{I}$.*

This definition is different from the one proposed in [31], where instead, authors construct a *whole faithful* representer tree which aims to reproduce exactly the decision function on the whole space of the features, namely $F_{TE}(x) = F_T(x)$, for all $x \in \mathbb{R}^{|\mathcal{J}|}$, and not only on the training samples $x^i, i \in \mathcal{I}$, or from the one given in [4] where manufactured data are used in the definition of fidelity. In those approaches, the complexity (depth and/or number of splits) is minimized, exactly or heuristically, but in principle, it can grow exponentially. Instead, in our approach, we aim to find a partition of the space which produces the same classification on the available samples keeping the depth of the representer tree fixed to D, the same as the trees in the TE. Since depth D is fixed, the existence of such a faithful T on the training data is not straightforward and, in general, cannot be guaranteed. Thus, we chose to maximize the fidelity and to include a penalization term to favor sparsity in the splits. According to the taxonomy reported in [8], we refer to the following definition of *global surrogate fitting model*. Giving a loss function $S : \{-1, 1\}^{|\mathcal{J}|} \times \{-1, 1\}^{|\mathcal{J}|} \to \mathbb{R}$ which measures the error in binary classification and a penalty function $V(T)$, which measures the number of features used by the tree T, we aim to find a tree T^* in a restricted set of decision trees $\mathcal{F}_T$ such that

$$T^* = \arg\min_{T \in \mathcal{F}_T} \left(\sum_{i \in \mathcal{I}} S(F_T(x^i), F_{TE}(x^i)) + V(T) \right)$$

In our approach, trees in the class $\mathcal{F}_T$ are characterized by

- fixed depth D;
- branching at node t according to rule (1) with a multivariate splitting

$$h_t(x) = a_t^T x + b_t, \tag{4}$$

  with $a_t \in \mathbb{R}^n$ and $b_t \in \mathbb{R}$;
- constraints derived from TE information.

The training phase outputs an optimal classification tree T^* defined by coefficients a_t^* and b_t^* for each branching node $t \in \mathcal{B}$.

Notation. In order to present our formulation, we introduce the main concepts and notation about optimal trees, as first described in [3]. Since the tree has fixed depth D, levels can be numbered as $\mathcal{D} = \{0, \dots, D\}$ being 0 the root node and D the terminal level of the leaf nodes. As usual, nodes are divided into *branch* and *leaf* nodes. A branch node t applies the multivariate splitting rule (4) on the subset of samples $\mathcal{I}_t \subseteq \mathcal{I}$ assigned to it, partitioning them among the left or right branch and hence among the two child nodes according to (1). Following [9], branching nodes $\mathcal{B}$ at level $d < D$ always apply a split which can be an effective or a dummy one. Indeed, if a node t at a level $d < D$ does not need to partition the samples further, we define two dummy children anyway such that only one of the two contains all the samples $\mathcal{I}_t$. We assume that nodes in the tree are numbered according to breadth-first indexing, starting from the root node, which is numbered zero and increasing from left to right at each level d, so that the tree has an overall number of nodes $\sum_{k=0}^{D} 2^k - 1 = 2^{D+1} - 2$.
We make use of the following notation, which is pictured in Figure 4:

- $\mathcal{B}$: the set of *branch nodes* where an oblique splitting rule is applied; $\mathcal{B}$ are numbered $\{0, \dots, 2^D - 2\}$.
- $\mathcal{L}$: the set of *leaf nodes* where a class is assigned to a sample; $\mathcal{L}$ are numbered $\{2^D - 1, \dots, 2^{D+1} - 2\}$.
- $\mathcal{B}(d)$ the set of nodes at level d, numbered as $\{0, \dots, 2^D - 2\}$. ;
- $\mathcal{B}' = \bigcup_{d=0}^{D-2} \mathcal{B}(d)$: the set of branch nodes not adjacent to the leaves;
- $\mathcal{B}'' = \mathcal{B}(D-1)$: the set of branch nodes adjacent to the leaves;
- $\mathcal{S}(t)$ the set of leaf nodes in the subtree rooted at node $t \in \mathcal{B}$
- $\mathcal{S}_L(t)$, $\mathcal{S}_R(t)$ the set of leaf nodes following the left and right branch of the subtree rooted at node $t \in \mathcal{B}$;
- the class assignment c_ℓ for each $\ell \in \mathcal{L}$. Being in a binary classification setting, we pre-assigned a class label to each leaf node, labeling as -1 the odd leaves and as $+1$ the even ones.

Variables. The straightforward variables are the coefficient of the hyperplane h_t for each $t \in \mathcal{B}$ which are continuous and w.l.o.g. can be assumed normalized so that

$$a_t = \{a_{t,j}\}_{j \in \mathcal{J}} \in [-1, 1]^{|J|}, \; b_t \in [-1, 1] \quad \forall \, t \in \mathcal{B}.$$

As done in OCTs [3], we also need the binary variables $z_{i,\ell}$ for all samples $i \in \mathcal{I}$ and leaves $\ell \in \mathcal{L}$, defined as follows:

$$z_{i,\ell} = \begin{cases} 1 & \text{if sample } i \text{ is assigned to leaf } \ell \in \mathcal{L} \\ 0 & \text{otherwise} \end{cases}.$$

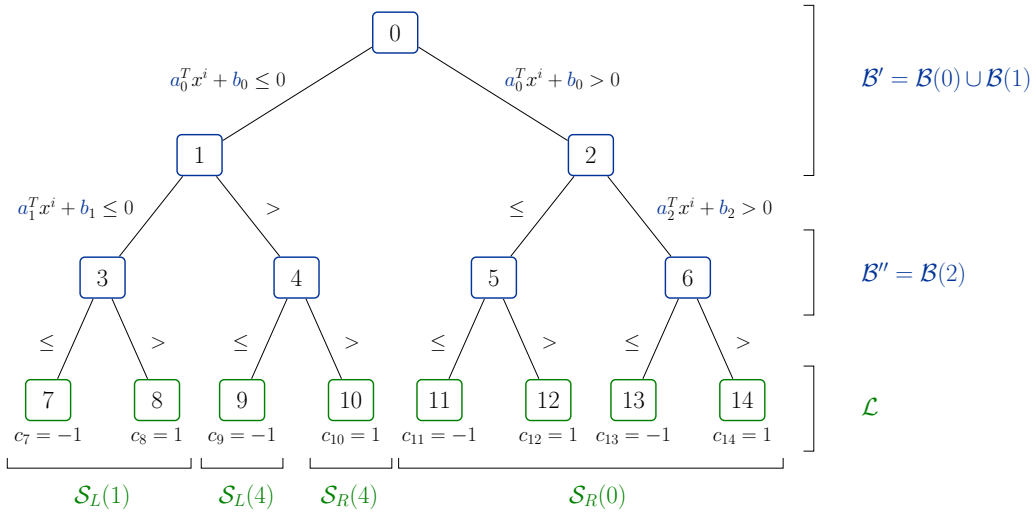

Figure 4: Representation of structure and notation for a tree with depth $D = 3$.

Since we are interested in controlling the use of features along the tree, we introduce other binary variables $s_{t,j} \in \{0, 1\}$, for each $t \in \mathcal{B}$ and $j \in \mathcal{J}$, as follows:

$$s_{t,j} = \begin{cases} 1 & \text{if feature } j \text{ is } \textit{used} \text{ at node } t \ (a_{t,j} \neq 0) \\ 0 & \text{otherwise} \end{cases}.$$

Table 1 presents an overview of the variables used in the model.

Variable	**Description**
$a_{t,j} \in [-1, 1]$	hyperplane coefficient at node t of feature j
$b_t \in [-1, 1]$	hyperplane intercept at node t
$s_{t,j} \in \{0, 1\}$	1 if feature j is selected at node t, 0 otherwise
$z_{i,\ell} \in \{0, 1\}$	1 if sample i is assigned to leaf ℓ, 0 otherwise

Table 1: Overview of decision variables in `MIRET`

Tree structure based Constraints. We need to state constraints to recover the tree structure as in MILP-based OCT formulation (e.g. [3, 9].
The first set of constraints forces each sample x^i, $i \in \mathcal{I}$ to be assigned to one and only one leaf node $\ell \in \mathcal{L}$. *Assignment constraints* are stated as follows:

$$\sum_{\ell \in \mathcal{L}} z_{i,\ell} = 1 \qquad \forall\ i \in \mathcal{I}. \tag{5}$$

We further need to model disjunctive conditions on samples that model the routing rules defined in (1) and ensure that the hyperplane splits are designed accordingly to the assignment z of samples to the leaf nodes. The *routing constraints* on the sample x^i are defined at each node $t \in \mathcal{B}$ but must apply only to that $i \in \mathcal{I}_t$, which is determined during the optimization process itself.

As in [9], we use the observation that whenever $i \in \mathcal{I}_t$, then x^i must end up in one of the leaves of the subtree rooted at t, and more specifically, either in the subset of the left or of the right subtree rooted at node t. This condition is expressed as

$$\text{either} \sum_{\ell \in \mathcal{S}_L(t)} z_{i,\ell} = 1 \qquad \text{or} \qquad \sum_{\ell \in \mathcal{S}_R(t)} z_{i,\ell} = 1.$$

Thus, we can write the *routing constraints* as

$$a_t^T x^i + b_t \leq M_{\mathcal{H}_L}\Big(1 - \sum_{\ell \in \mathcal{S}_L(t)} z_{i,\ell}\Big) \qquad \forall i \in \mathcal{I},\ \forall t \in \mathcal{B}, \tag{6}$$

$$a_t^T x^i + b_t - \varepsilon \geq -M_{\mathcal{H}_R}\Big(1 - \sum_{\ell \in \mathcal{S}_R(t)} z_{i,\ell}\Big) \qquad \forall i \in \mathcal{I},\ \forall t \in \mathcal{B} \tag{7}$$

where $\varepsilon > 0$ is a sufficiently small positive value to model strict inequalities in (1).
Indeed, if a sample $i \notin \mathcal{I}_t$, we have that $\sum_{\ell \in \mathcal{S}_L(t)} z_{i,\ell} = \sum_{\ell \in \mathcal{S}_R(t)} z_{i,\ell} = 0$, and thus both the constraints do not impose any routing conditions on sample i. Instead, if sample $i \in \mathcal{I}_t$ and $\sum_{\ell \in \mathcal{S}_L(t)} z_{i,\ell} = 1$, thus, only the constraint (6) is activated for sample i at node t, while (7) is deactivated. Analogous considerations can be done if a sample $i \in \mathcal{I}_t$ follows the right branch. The Big-M values can be easily obtained because $x^i \in [0,1]^{|\mathcal{J}|}$, $a_t \in [-1,1]^{|\mathcal{J}|}$ and $b_t \in [-1,1]$ for all $t \in \mathcal{B}$. Hence we can set the Big-M values to

$$M_{\mathcal{H}_L} = |\mathcal{J}| + 1 \qquad M_{\mathcal{H}_R} = |\mathcal{J}| + 1 + \varepsilon.$$

On the other hand, the choice of the ε parameter is critical because small values may lead to numerical issues, and large ones may cut feasible solutions.
Additionally, we need to include constraints among $s_{t,j}$ and $a_{t,j}$ to force the conditions $s_{t,j} = 0 \iff a_{t,j} = 0$. This is easily modelled by *sparsity constraints* as

$$-s_{t,j} \leq a_{t,j} \leq s_{t,j}. \tag{8}$$

6.2. Incorporating TE-driven information

We embed information derived from the TE both in the constraints and in the objective function of the model `MIRET`. In particular, we use

- the proximity measure m_{ik} among samples x^i, x^k;
- the frequency $f_{d,j}$ of each feature j at each level d;
- the probability p^i of the TE class prediction $\widehat{y}^i$,

and we report here the definitions needed.
The *proximity measure* m_{ik} of a pair of samples x^i, x^k in the TE is the number of trees $e \in \mathcal{E}$ in the TE model in which the two samples end up in the same leaf. Let us define the indicator function for each pair $i, k \in \mathcal{I}$ and each tree $e \in \mathcal{E}$ as

$$\mathbb{1}(i,k,e) = \begin{cases} 1 & \text{if samples } x^i, x^k \text{ end in the same leaf } \ell \text{ of the tree } e \\ 0 & \text{otherwise} \end{cases}$$

Thus, we have the following definition [5, 30]:

$$m_{i,k} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} w^e \sum_{e \in \mathcal{E}} \mathbb{1}(i,k,e), \tag{9}$$

where $w^e \; \forall e \in \mathcal{E}$ are non-negative weights that play a similar role as in the definitions 2 and 3. The proximity measure is considered a measure of the distance in space. Being two samples in the same leaf in all the trees indicates that the two samples always belong to the same space partition. We use it in the constraints.

The *level frequency* $f_{d,j}$ of each feature j at each level d across the trees in TE is calculated according to (2).
In our optimization model, we grow a full tree with depth D, and all the branch nodes at level d are potential splitting nodes. In total, nodes are $|\mathcal{E}| \cdot 2^d$, where 2^d is the number of nodes at level d. Since we aim to measure the spread of features among nodes in the trees, we use in (2) the denominator equal to $|\mathcal{E}| \cdot 2^d$, and we have for all $d \in \mathcal{D}$, $j \in \mathcal{J}$:

$$f_{d,j} = \frac{1}{|\mathcal{E}| 2^d} \sum_{e \in \mathcal{E}} w^e \sum_{t \in \mathcal{B}^e(d)} \mathbb{1}(j,t,e). \tag{10}$$

We use it both in the constraints and in the objective function.

The *class probability* p^i of a sample x^i is the highest estimated probability of the prediction given by TE. It is calculated as

$$p^i := \max_{c \in \{-1,1\}} p^i_c,$$

where

$$p^i_c := \frac{1}{E} \sum_{e=1}^{E} \frac{N_{\ell_e,c}}{N_{\ell_e}},$$

with N_{ℓ_e} the number of samples assigned to leaf ℓ in the tree e of the TE model and as $N_{\ell_e,c}$ the number of samples with class label c assigned to the same ℓ in e. We use it in the objective function.

TE-driven Constraints. We use the *proximity measure* to constrain the space partitioning of the representative tree to be similar to the TE. To this aim, we define the set $\mathcal{U}_{TE}$ of pairs of samples with proximity measure larger than a given *proximity threshold* $\overline{m}_{TE} \leq 1$ which is a hyperparameter of our model:

$$\mathcal{U}_{TE}(\overline{m}_{TE}) := \{(i,k) \in \mathcal{I} \times \mathcal{I} : \; i < k \; \wedge \; m_{i,k} \geq \overline{m}_{TE}\}, \tag{11}$$

and we include the *proximity constraints*:

$$z_{i,\ell} = z_{k,\ell} \qquad \forall \ell \in \mathcal{L}, \quad \forall (i,k) \in \mathcal{U}(\overline{m}_{TE})$$

which ensures that pairs of samples with a proximity measure over the threshold are assigned to the same leaf in the model.

The *level frequency* $f_{d,j}$ is used both to define hard constraints and to drive the selection of the features according to their frequency. Let us define the index sets

$$\mathcal{J}_\gamma(d) := \{j \in \mathcal{J} : f_{d,j} > \gamma_d\}, \qquad \forall d \in \mathcal{D}, \tag{12}$$

where γ_d for $d \in \mathcal{D}$ are given *frequency thresholds* and they are treated as hyperparameters of our model.
The *frequency constraints* force the model to select at level d only features in the subset $\mathcal{J}_\gamma(d)$ by imposing the following conditions:

$$\begin{aligned} &a_{t,j} = 0 && \forall j \in \mathcal{J} \setminus \mathcal{J}_\gamma(d) && \forall t \in \mathcal{B}(d), && \forall d \in \mathcal{D}, \\ &s_{t,j} = 0, && \forall j \in \mathcal{J} \setminus \mathcal{J}_\gamma(d) && \forall t \in \mathcal{B}(d), && \forall d \in \mathcal{D} \end{aligned}$$

This way, we hardly induce local sparsity of the splitting hyperplanes.

6.3. Objective function

The objective function is obtained as a combination of the two terms which measure fidelity and sparsity. In order to measure fidelity, we need to define a loss function $S : \{-1,1\}^{|\mathcal{J}|} \times \{-1,1\}^{|\mathcal{J}|} \to \mathbb{R}$ to measure the error $S(F_T(x^i), F_{TE}(x^i))$. The values $F_{TE}(x^i) = \widehat{y}^i$ are the class label predicted by TE for sample x^i, based e.g. on majority voting.
The value $F_T(x^i)$ can be easily obtained as

$$F_T(x^i) = \sum_{\ell \in \mathcal{L}} c_\ell z_{i,\ell}$$

where c_ℓ is the vector of assigned classes to leaves. Thus

$$\frac{1}{2}\widehat{y}^i\Big(\widehat{y}^i - F_T(x^i)\Big) = \begin{cases} 1 & \text{if } F_{TE}(x^i) = F_T(x^i) \\ 0 & \text{otherwise} \end{cases}.$$

We use class probability p^i of the sample x^i to weight misclassification with the aim of penalizing more the error on samples predicted with high probability by TE; we define the following loss function

$$S(F_{TE}(x^i), F_T(x^i)) = \frac{1}{2}p^i F_{TE}(x^i)\Big(F_{TE}(x^i) - F_T(x^i)\Big) = \frac{1}{2}p^i\widehat{y}^i\Big(\widehat{y}^i - \sum_{\ell \in \mathcal{L}} c_\ell z_{i,\ell}\Big).$$

We also aim to promote the use of sparser hyperplanes, enhancing the interpretability of the tree model. Thus we penalize the selection of features according to the level frequencies with which they are used at each level d in TE. In particular, for each feature $j \in \mathcal{J}$ we count how many times it is used in the model at level d as $\sum_{t \in \mathcal{B}(d)} s_{t,j}$. At each level d, the use of feature j is weighted by the reciprocal of its level frequency $f_{d,j}$, in a way that the more the feature j is used at level d in the TE, the less is the penalization for using it in the optimal tree. We restrict this penalization only to the most used features per level, namely for $j \in \mathcal{J}_\gamma(d)$. The penalization term is

$$V(T) = \sum_{d \in \mathcal{D}} \sum_{j \in \mathcal{J}_\gamma(d)} \frac{1}{f_{d,j}} \sum_{t \in \mathcal{B}(d)} s_{t,j}.$$

Thus, the overall objective function is:

$$\min \frac{1}{2}\sum_{i\in\mathcal{I}} p^i\hat{y}^i\Big(\hat{y}^i - \sum_{\ell\in\mathcal{L}} c_\ell z_{i,\ell}\Big) + \alpha \sum_{d\in\mathcal{D}}\sum_{j\in\mathcal{J}_\gamma(d)} \frac{1}{f_{d,j}} \sum_{t\in\mathcal{B}(d)} s_{t,j},$$

where α is a penalty hyperparameter to control the trade-off between the two objectives.

6.4. The basic MILP model of `MIRET`

The MILP formulation of the basic-`MIRET` is the following:

$$
\begin{aligned}
(\texttt{b-MIRET}) \quad \min_{a,b,z,s} \quad & \frac{1}{2}\sum_{i\in\mathcal{I}} p^i\hat{y}^i\Big(\hat{y}^i - \sum_{\ell\in\mathcal{L}} c_\ell z_{i,\ell}\Big) + \alpha \sum_{d\in\mathcal{D}}\sum_{j\in\mathcal{J}_\gamma(d)} \frac{1}{f_{d,j}} \sum_{t\in\mathcal{B}(d)} s_{t,j} \\
\text{s.t.} \quad & a_t^T x^i + b_t - \varepsilon \geq -(|J|+1+\varepsilon)\Big(1 - \sum_{\ell\in\mathcal{S}_R(t)} z_{i,\ell}\Big) && \forall t\in\mathcal{B}, \quad \forall i\in\mathcal{I}, \\
& a_t^T x^i + b_t \leq (|J|+1)\Big(1 - \sum_{\ell\in\mathcal{S}_L(t)} z_{i,\ell}\Big) && \forall t\in\mathcal{B}, \quad \forall i\in\mathcal{I}, \\
& \sum_{\ell\in\mathcal{L}} z_{i,\ell} = 1 && \forall i\in\mathcal{I}, \\
& z_{i,\ell} = z_{k,\ell} && \forall \ell\in\mathcal{L}, \quad \forall i,k\in\mathcal{U}_{TE}(\overline{m}_{TE}), \\
& a_{t,j} = 0, && \forall d\in\mathcal{D}, \quad \forall j\in\mathcal{J}\setminus\mathcal{J}_\gamma(d), \quad \forall t\in\mathcal{B}(d), \\
& s_{t,j} = 0 && \forall d\in\mathcal{D}, \quad \forall j\in\mathcal{J}\setminus\mathcal{J}_\gamma(d), \quad \forall t\in\mathcal{B}(d), \\
& -s_{t,j} \leq a_{t,j} \leq s_{t,j} && \forall t\in\mathcal{B}, \quad \forall j\in\mathcal{J}, \\
& -1 \leq a_{tj} \leq 1 && \forall t\in\mathcal{B} \quad \forall j\in\mathcal{J}, \\
& -1 \leq b_t \leq 1 && \forall t\in\mathcal{B}, \\
& z_{i,\ell}\in\{0,1\} && \forall \ell\in\mathcal{L}, \quad \forall i\in\mathcal{I}, \\
& s_{t,j}\in\{0,1\} && \forall t\in\mathcal{B}, \quad \forall j\in\mathcal{J}.
\end{aligned}
$$

Solving this model gives an optimal T^* defined for each node $t\in\mathcal{B}$ by the splits

$$\text{if } a_t^{*T}x + b_t^* \begin{cases} \leq 0 \\ > 0 \end{cases},$$

Following the path from the root to leaves gives the decision rules that lead to a classification of a sample. In the next section, we slightly modify the formulation in order to gain on the performance of out-of-shell MILP solvers.

7. Improving the basic `MIRET` formulation

7.1. Existence of at least one split

As a first step, we aim to avoid the existence of a feasible solution that does not provide any split, namely a dummy tree. Indeed, we observe that there exists a feasible (obviously not optimal) solution with $s_{j,t} = 0$ for all $j\in\mathcal{J}$ and $t\in\mathcal{B}$ (so that the penalty term $V(T) = 0$) and $a_t = 0$ for all $t\in\mathcal{B}$ where $b_t, z_{i,\ell}$ can be fixed to any feasible value. All these feasible solutions

do not apply a partition of the samples, thus resulting in a dummy tree. In particular, this also implies that when using a continuous linear relaxation to compute a lower bound, we can obtain a fractional feasible solution $z_{i,\ell} \in [0,1]$ such that the assignment constraints are satisfied and $\hat{y}^i = \sum_{\ell \in \mathcal{L}} c_\ell z_{i,\ell}$ which is of course optimal for the linear relaxation since the value of the objective function is zero. This gives a trivial lower bound.
In order to avoid such a trivial solution, we add the constraint

$$\sum_{t \in \mathcal{B}} \sum_{j \in \mathcal{J}} s_{t,j} \geq 1$$

which forces at least one branch node in the tree to use at least one feature (the tree has at least one univariate split). Of course, we could enforce a more stringent constraint by increasing the right-hand side to a value greater than 1 which is usually the case in optimal tree solutions.

7.2. Breaking symmetries

In a MILP model, symmetries refer to the existence of multiple solutions that have the same objective value but differ only in the values assigned to a set of integer variables [23]. The presence of symmetries can make it difficult to solve the problem, as they can lead to a large number of equivalent solutions, and to a huge branching tree to be explored by the optimization algorithm.
Our basic formulation is affected by this issue in a specific way. Indeed, our classification tree is symmetric in the sense that the same value of the objective function is obtained for every solution that permutes components of z variables in such a way that

(i) the final partition of samples, and so the final predictions, remains unchanged and,
(ii) the same s variables activate, i.e. the same features are used at each level d of the tree.

Indeed, in any branch node $t \in \mathcal{B}'$, samples assigned to its right and left sub-trees can be swapped using the same setting of the variables s obtaining equivalent solutions.
However, this is no longer true for nodes at the branching level adjacent to the leaves $\mathcal{B}''$ due to the pre-assigned labels c to leaves. Hence, for each node $t \in \mathcal{B}''$, the subset of samples I_t is forced to follow either the left or the right branch of the node by the leaves labels.
In order to avoid the presence of multiple equivalent solutions obtained by swapping branches, we require the following symmetry-breaking constraints:

$$b_t \geq 0, \qquad \forall t \in \mathcal{B}'$$

This set of constraints enforces a non-negative intercept for the splitting hyperplanes at node $t \in \mathcal{B}'$ thus avoiding the presence of symmetric solutions obtainable by reversing the splitting rules. In this way, we reduce the set of feasible solutions by removing redundant solutions which produce the same final partitions of samples using the same features in the tree.

7.3. Adding branching rules

Exploiting the nested tree structure and the routing constraints (6) (7), we introduce for each $t \in \mathcal{B}'$ and each sample $i \in \mathcal{I}$ two additional boolean variables $q^i_L(t), q^i_R(t) \in \{0,1\}$ with the constraints

$$q^i_L(t) - \sum_{\ell \in \mathcal{S}_L(t)} z_{i,\ell} = 0; \qquad q^i_R(t) - \sum_{\ell \in \mathcal{S}_R(t)} z_{i,\ell} = 0.$$

Hence, we have

$$q^i_{L/R}(t) = \begin{cases} 1 & \text{if sample } i \text{ is assigned to a leaf } \ell \in \mathcal{S}_{L/R}(t) \\ 0 & \text{otherwise} \end{cases},$$

and routing constraints (7)-(6) become

$$a_t^T x^i + b_t \leq M_{\mathcal{H}_L}\Big(1 - q^i_L(t)\Big), \qquad \forall t \in \mathcal{B}, \qquad \forall i \in \mathcal{I}$$
$$a_t^T x^i + b_t - \varepsilon \geq -M_{\mathcal{H}_R}\Big(1 - q^i_R(t)\Big), \qquad \forall t \in \mathcal{B}, \qquad \forall i \in \mathcal{I}$$

and the assignment constraints (5) are written as

$$q^i_L(0) + q^i_R(0) = 1, \qquad \forall i \in \mathcal{I}.$$

In the branching procedure, fixing a variable $q^i_L(t) = 0$ (or $q^i_R(t) = 0$) forces all the $z_{i,\ell} = 0$ with $\ell \in \mathcal{S}_L(t)$ (or $S_R(t)$), thus avoiding the need to explore branching on the single variables $z_{i,\ell}$. Although the large increase in variables and constraints, the numerical experiments that we performed seem to confirm the effectiveness of having such variables.
We also add parenting constraints from node $t \in \mathcal{B}(d)$ to its children $2t+1, 2t+2 \in \mathcal{B}(d+1)$ that connects the q_L and q_R variables along the subtree rooted at t. These are $|\mathcal{I}| \cdot |\mathcal{B}'|$ conditions expressed as follow

$$q^i_L(t) = q^i_L(2t+1) + q^i_R(2t+1), \qquad \forall t \in \mathcal{B}', \qquad \forall i \in \mathcal{I},$$
$$q^i_R(t) = q^i_L(2t+2) + q^i_R(2t+2), \qquad \forall t \in \mathcal{B}', \qquad \forall i \in \mathcal{I}.$$

We report in Table 2 the additional variables.

Variable	**Description**
$q^i_L(t) \in \{0,1\}$	if sample i is assigned to a leaf $\ell \in \mathcal{S}_L(t)$
$q^i_R(t) \in \{0,1\}$	if sample i is assigned to a leaf $\ell \in \mathcal{S}_R(t)$

Table 2: Additional decision variables in improved `MIRET`

The improved MILP formulation `MIRET` is the following:

$$
\begin{aligned}
(\texttt{MIRET}) \quad \min_{a,b,z,s,q_L,q_R} \quad & \frac{1}{2}\sum_{i\in\mathcal{I}} p^i\hat{y}^i\Big(\hat{y}^i - \sum_{\ell\in\mathcal{L}} c_\ell z_{i,\ell}\Big) + \alpha \sum_{d\in\mathcal{D}}\sum_{j\in\mathcal{J}_\gamma(d)} \frac{1}{f_{d,j}} \sum_{t\in\mathcal{B}(d)} s_{t,j} \\
\text{s.t.} \quad & a_t^T x^i + b_t - \varepsilon \geq -(|J|+1+\varepsilon)\Big(1 - q_R^i(t)\Big) && \forall t\in\mathcal{B}, \quad \forall i\in\mathcal{I}, \\
& a_t^T x^i + b_t \leq (|J|+1)\Big(1 - q_L^i(t)\Big) && \forall t\in\mathcal{B}, \quad \forall i\in\mathcal{I}, \\
& q_L^i(0) + q_R^i(0) = 1 && \forall i\in\mathcal{I}, \\
& q_L^i(t) = q_L^i(2t+1) + q_R^i(2t+1), && \forall t\in\mathcal{B}', \quad \forall i\in\mathcal{I}, \\
& q_R^i(t) = q_L^i(2t+2) + q_R^i(2t+2), && \forall t\in\mathcal{B}', \quad \forall i\in\mathcal{I}, \\
& q_L^i(t) = \sum_{\ell\in\mathcal{S}_L(t)} z_{i,\ell} && \forall t\in\mathcal{B}, \quad \forall i\in\mathcal{I}, \\
& q_R^i(t) = \sum_{\ell\in\mathcal{S}_R(t)} z_{i,\ell} && \forall t\in\mathcal{B}, \quad \forall i\in\mathcal{I}, \\
& \sum_{t\in\mathcal{B}}\sum_{j\in\mathcal{J}} s_{t,j} \geq 1, \\
& b_t \geq 0, && \forall t\in\mathcal{B}', \\
& z_{i,\ell} = z_{k,\ell} && \forall \ell\in\mathcal{L}, \quad \forall i,k\in\mathcal{U}_{TE}(\overline{m}_{TE}), \\
& a_{t,j} = 0, && \forall d\in\mathcal{D}, \quad \forall t\in\mathcal{B}(d), \qquad \forall j\in\mathcal{J}\setminus\mathcal{J}_\gamma(d), \\
& s_{t,j} = 0 && \forall d\in\mathcal{D}, \quad \forall t\in\mathcal{B}(d), \qquad \forall j\in\mathcal{J}\setminus\mathcal{J}_\gamma(d), \\
& -s_{t,j} \leq a_{t,j} \leq s_{t,j}, && \forall t\in\mathcal{B}, \quad \forall j\in\mathcal{J}, \\
& -1 \leq a_{tj} \leq 1 && \forall t\in\mathcal{B} \quad \forall j\in\mathcal{J}, \\
& -1 \leq b_t \leq 1 && \forall t\in\mathcal{B}, \\
& z_{i,\ell}\in\{0,1\} && \forall \ell\in\mathcal{L}, \quad \forall i\in\mathcal{I}, \\
& s_{t,j}\in\{0,1\} && \forall j\in\mathcal{J}, \quad \forall t\in\mathcal{B}.
\end{aligned}
$$

8. Computational experience

In this section, we present different computational results in order to evaluate the performances of the approach proposed. We use a Random Forest as TE, and in particular, we use the RF method as implemented in `sklearn.ensemble.RandomForestClassifier` [24] with the following setting:

- maximum depth $D \in \{2, 3, 4\}$;
- $|\mathcal{E}| = 100$;
- random sampling of the features deactivated so that weights $w^e = 1$, $e \in \mathcal{E}$ in (2) and (9).

We denote the target forest as $\widehat{RF}$. The mixed-integer programming model `MIRET` is coded in Python and ran on a server Intel(R) Xeon(R) Gold 6252N CPU processor at 2.30 GHz and 96 GB of RAM. The MILP is solved using Gurobi 10.0.0 with standard settings. We set a time limit of 1 hour for each model optimization.

8.1. Datasets

We selected 10 datasets from UCI Machine Learning repository [11] related to binary classification tasks, and we normalized the feature values of each dataset in the interval 0-1. Information about the datasets considered is reported in Table 3. Each dataset was split into training (80%) and test (20%) sets.

Dataset	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{J}\rvert$	Class (%)
Cleveland	297	13	53.9/46.1
Diabetes	768	8	65.1/34.9
German	1000	20	30/70
Heart	270	13	55.6/44.4
IndianLiver	579	10	71.5/28.5
Ionosphere	351	34	35.9/64.1
Parkinson	195	22	24.6/75.4
Sonar	208	60	53.4/46.6
Wholesale	440	7	32.3/67.7
Wisconsin	569	30	37.3/62.7

Table 3: Characteristics of the datasets.

In order to have a glimpse view of the role of the $\widehat{RF}$ information that we used in the model, we calculate the distribution of the proximity measures and of the probability class on the training set, and we plot them using violin plots [16] in Figures 5 and 6 respectively. As regard the level frequencies, we also calculate them for the RF, and we report them together with those used in the optimal tree obtained with `MIRET` in Figure 7.

8.2. Hyperparameters setting

Regarding the hyperparameters in the `MIRET` model, we fixed

- $\overline{m}_{TE} = 1$ in (11),
- $\varepsilon = 0.001$ in (6), (7) (see [3]).

For the other hyperparameters γ_d, $d \in \mathcal{D}$, and α we performed a tailored tuning of the model using a grid search within a k-fold cross-validation.
In particular, the grid values on the thresholds γ_d in (12) are defined considering a percentage of the frequencies values of features used in the level d of the $\widehat{RF}$. Let $\mathcal{F}(d)^+ = \{f_{d,j} : f_{d,j} > 0,\ j \in \mathcal{J}\}$ be the set of positive level frequencies, then γ_d, $d \in \mathcal{D}$ are computed as the h-th percentile of $\mathcal{F}(d)^+$. Further, we also include the value $\gamma_d = 0$, to consider the case when `MIRET` can use all the features used in $\widehat{RF}$, namely any feature in $\mathcal{F}(d)^+$.
We perform a grid search using these values:

- penalty parameter $\alpha \in \{0.2, 0.4, 0.5, 0.6, 0.8\}$;
- $\gamma_d = 0$ and γ_d as the h-th percentile of $\mathcal{F}(d)^+$ with $h \in \{100/2, 100/3, 100/4\}$.

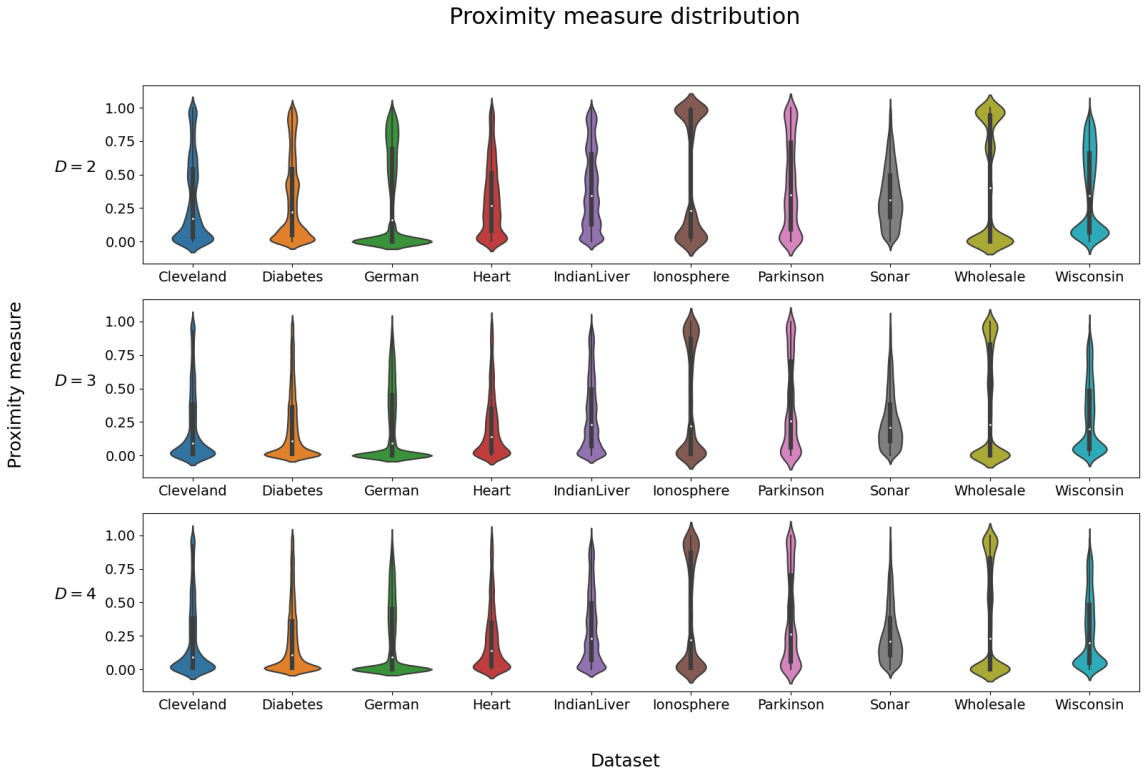

Figure 5: Distribution of proximity measures of pair of training set samples

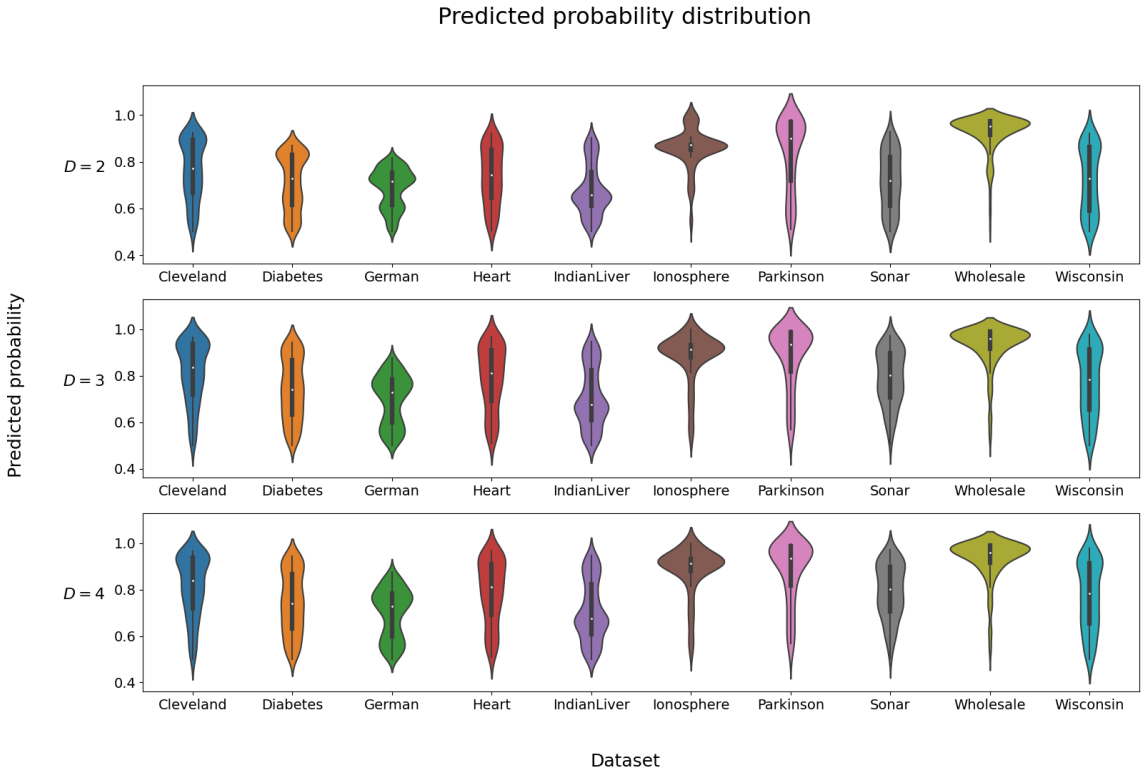

Figure 6: Distribution of predicted probabilities of training set samples

We set $k = 4$, and we evaluated the average validation fidelity and the average sparsity of the trained model. In the end, we selected hyperparameters that provide the best balance between the two objectives. The final hyperparameters setting used in the computational results are reported in the appendix in Table A.9.

8.3. Results

We analyze both the optimization performance, namely the viability and efficiency of using a MILP formulation for finding an optimal representative tree T^* of the random forest $\widehat{RF}$, and the predictive performance. We select depths up to $D = 4$. Indeed, aiming to obtain an interpretable tree model and accounting for the fact that we can have multivariate splittings, depth $D = 2, 3$ are the most significant to get real interpretability. However, to check scalability in the solution of the model, we also use $D = 4$.
Once an optimal tree T^* is obtained for $D = 2, 3, 4$ solving the MILP problem `MIRET`, following [15] we evaluate the performances both with respect to the $\widehat{RF}$ predictions (fidelity), and with respect to the ground truth labels (accuracy) both on the training and on the test sets.

- Fidelity is measured as

$$\text{FID}_{\widehat{RF}} = 1 - \frac{1}{2|\mathcal{I}|} \sum_{i \in \mathcal{I}} F_{\widehat{RF}}(x^i) \Big(F_{\widehat{RF}}(x^i) - F_{T^*}(x^i) \Big);$$

- `MIRET` accuracy with respect to the ground truth y^i, calculated as

$$\text{ACC}_{\texttt{MIRET}} = 1 - \frac{1}{2|\mathcal{I}|} \sum_{i \in \mathcal{I}} y^i \Big(y^i - F_{T^*}(x^i) \Big).$$

We also evaluated as a term of comparison the $\widehat{RF}$ accuracy with respect to the ground truth y^i, calculated as

$$\text{ACC}_{\widehat{RF}} = 1 - \frac{1}{2|\mathcal{I}|} \sum_{i \in \mathcal{I}} y^i \Big(y^i - F_{\widehat{RF}}(x^i) \Big).$$

In Table 4, we compare the optimization performance of the basic `MIRET` model and the improved `MIRET` version. For each dataset and each D (a total of 30 MILP problems), we report the computational time (s) and the optimality gap for solving the corresponding MILP. The returned solution T^* is not always certified as the global optimum of the problem. As expected, problem hardness increases with the size of the problem. However, the improved version of `MIRET` has better performance because it closes the gap on 4 additional problems, improves the gap on more than 85% of the not closed problem, and improves the time on about 30% of the closed ones. In particular, for $D = 2$, we obtain gap zero on nine of the ten datasets; for $D = 3$ optimality is certified only on four of the ten datasets and on none of the problems for $D = 4$. Despite the gap is not zero, the quality of the solutions in terms of predictive performance on the training set is outstanding, as reported in Table 5.
We report in Table 6 the out-of-sample predictive performances. The average fidelity of `MIRET` (FID-$\widehat{RF}$) is quite high, meaning that it fairly mimics the $\widehat{RF}$ classification, with an average of 91.71% at depth 2, 94.08% at depth 3, and 89.3% at depth 4.
Additionally, our model has retained on average the generalization capabilities of $\widehat{RF}$. As a matter of fact, it can be easily seen that the accuracy of `MIRET` is similar to the accuracy of $\widehat{RF}$

	$D=2$				$D=3$				$D=4$			
	Time		Gap		Time		Gap		Time		Gap	
Dataset	b-MIRET	MIRET	b-MIRET	MIRET	b-MIRET	MIRET	b-MIRET	MIRET	b-MIRET	MIRET	b-MIRET	MIRET
Cleveland	**28.4**	44.6	0.0	0.0	3600	**1942.2**	43.6	**0.0**	3600	3600	99.8	**96.0**
Diabetes	3600	**1373.3**	39.5	**0.0**	3600	3600	100.0	**96.1**	3600	3600	100.0	**98.8**
German	11.8	12.3	0.0	0.0	3600	3600	99.6	**73.6**	3600	3600	100.0	**99.4**
Heart	160.0	**17.1**	0.0	0.0	3600	3600	87.6	**73.2**	3600	3600	100.0	**94.8**
IndianLiver	3600	3600	80.5	**75.6**	3600	3600	100.0	**97.7**	3600	3600	100.0	**98.2**
Ionosphere	4.4	**3.4**	0.0	0.0	**107.4**	204.4	0.0	0.0	3600	3600	**96.6**	97.9
Parkinson	5.6	**4.6**	0.0	0.0	3600	3600	58.2	**33.8**	3600	3600	99.9	**95.9**
Sonar	3600	**117.1**	10.1	**0.0**	3600	3600	**89.3**	93.2	3600	3600	100.0	**89.9**
Wholesale	1.4	**0.7**	0.0	0.0	70.8	**18.0**	0.0	0.0	3600	3600	94.2	**83.0**
Wisconsin	26.4	**19.1**	0.0	0.0	3600	**1114.5**	48.0	**0.0**	3600	3600	**92.8**	94.9

Table 4: Comparison of b-MIRET and MIRET models in computational times (s) and MIP Gap values (%); in boldface the winning values. Winner on computational time is defined when the value is at least 10% better.

	$D=2$			$D=3$			$D=4$		
		ACCURACY			ACCURACY			ACCURACY	
Dataset	FID-$\widehat{RF}$	MIRET	$\widehat{RF}$	FID-$\widehat{RF}$	MIRET	$\widehat{RF}$	FID-$\widehat{RF}$	MIRET	$\widehat{RF}$
Cleveland	96.6	81.9	81.9	97.9	85.7	87.8	93.7	85.7	92.0
Diabetes	91.9	73.5	75.4	95.4	75.9	79.2	91.2	76.7	83.2
German	98.0	66.8	66.0	95.4	68.5	71.6	88.6	69.9	78.0
Heart	77.8	72.2	84.3	95.4	86.6	89.4	92.1	86.1	94.0
IndianLiver	97.2	67.6	67.8	93.7	67.4	70.6	92.7	71.7	77.3
Ionosphere	98.6	89.6	91.1	95.0	89.6	94.6	92.9	89.6	96.8
Parkinson	93.6	84.6	88.5	89.1	85.9	95.5	93.6	93.6	98.7
Sonar	84.9	77.7	90.4	80.1	80.1	100.0	77.7	77.7	100.0
Wholesale	99.7	92.6	92.3	99.7	92.6	92.9	97.2	92.3	95.2
Wisconsin	99.1	95.8	96.7	96.9	95.6	98.7	98.0	97.4	99.3

Table 5: Predictive performance on the training set. FID-$\widehat{RF}$: Fidelity with respect to $\widehat{RF}$ (%); MIRET ACCURACY and $\widehat{RF}$ ACCURACY with respect to the ground truth (%).

along all the datasets. Thus, our model gains in interpretability as it provides a single optimal tree with more straightforward decision paths while being capable of fairly closely reproducing the predictive performances of the random forest.

In order to better understand the role of the TE-driven information inserted in the model, we analyze the features' use and the proximity measure among samples, comparing the MIRET tree with respect to the $\widehat{RF}$.

In particular, we report in Figure 7 a one-to-one comparison of each problem and for each D of the level frequency of the features used in the $\widehat{RF}$ and in the MIRET model. For $\widehat{RF}$, the feature's level frequency is calculated as in (10).

For MIRET, the heatmap reports the feature level frequency as the fraction of times a feature is

	$D=2$			$D=3$			$D=4$		
		ACCURACY			ACCURACY			ACCURACY	
Dataset	FID-$\widehat{RF}$	MIRET	$\widehat{RF}$	FID-$\widehat{RF}$	MIRET	$\widehat{RF}$	FID-$\widehat{RF}$	MIRET	$\widehat{RF}$
Cleveland	98.3	78.3	80.0	95.0	76.7	81.7	93.3	76.7	83.3
Diabetes	89.6	75.3	68.8	95.5	77.3	76.6	91.6	72.1	76.6
German	99.0	65.5	65.5	97.0	68.5	67.5	86.5	68.5	72.0
Heart	68.5	64.8	85.2	90.7	79.6	85.2	87.0	79.6	85.2
IndianLiver	94.8	65.5	65.5	92.2	64.7	65.5	95.7	66.4	63.8
Ionosphere	100.0	94.4	94.4	95.8	94.4	93.0	95.8	94.4	93.0
Parkinson	97.4	76.9	79.5	100.0	79.5	79.5	79.5	87.2	82.1
Sonar	76.2	76.2	81.0	83.3	83.3	81.0	73.8	76.2	78.6
Wholesale	97.7	90.9	90.9	96.6	89.8	90.9	97.7	90.9	90.9
Wisconsin	95.6	89.5	93.9	94.7	89.5	94.7	92.1	94.7	95.6

Table 6: Predictive performance on the test set: fidelity of the MIRET model with respect to $\widehat{RF}$, accuracy with respect to the ground truth of the MIRET and $\widehat{RF}$ models.

used in a split at level d; thus, it is calculated as

$$f_{d,j}^{\texttt{MIRET}} = \frac{1}{|\mathcal{B}(d)|} \sum_{t \in \mathcal{B}(d)} s_{t,j}^*.$$

From this Figure, we can also see at a glance the number of features used at each level which gives a rough view of the multi-dimension of the splits of the optimal tree. The deeper the $\widehat{RF}$ goes, the more features it tends to use with a low frequency. On the other hand, MIRET uses fewer features at each level. Thus, the penalization of the features' use in the objective function allows us to reduce the number of features used in the optimal tree, which are less than those used overall in the forest, and encourage the use of sparse multivariate splits.
As regard the proximity measure, we use pair of samples with maximal proximity in the $\widehat{RF}$ to define hard constraints. Of course, this implies that on the training data, these samples end up in the same leaves. This might not be true on the test set. Indeed, the multivariate structure of the splits of MIRET does not allow for replicating the exact partition of the space made by the $\widehat{RF}$, which uses univariate splits. However, proximity constraints aim to encourage clusters of samples that are in the same leaf in all trees of the $\widehat{RF}$, to belong to the same partition of space.
Using definition (9), we consider the proximity sets of samples in the test set for both the $\widehat{RF}$ and the MIRET tree with the threshold set to 1. Let denotes the two sets as $\mathcal{U}_{\widehat{RF}}(1)$ and $\mathcal{U}_{\texttt{MIRET}}(1)$ which can be easily obtained

$$\mathcal{U}_{\texttt{MIRET}}(1) := \{(i,k) \in \mathcal{I} \times \mathcal{I} : i < k \wedge z_{i,\ell}^* = z_{k,\ell}^* \text{ for all } \ell \in \mathcal{L}\}.$$

Further, we also consider the sets where the proximity is zero, namely pairs of samples that never end up in the same leaves. The sets are defined as

$$\overline{\mathcal{U}}_{\widehat{RF}}(0) := \{(i,k) \in \mathcal{I} \times \mathcal{I} : i < k \wedge m_{i,k} = 0\},$$

$$\overline{\mathcal{U}}_{\texttt{MIRET}}(0) := \{(i,k) \in \mathcal{I} \times \mathcal{I} : i < k \wedge \exists \bar{\ell} \in \mathcal{L} : z^*_{i,\bar{\ell}} \neq z^*_{k,\bar{\ell}} \}.$$

Thus, we can calculate the fractions U and $\overline{U}$, which measure the percentage of the samples that end up in the same leaf both in the $\widehat{RF}$ and the MIRET tree, and the percentage of the samples that end up in different leaves both in the $\widehat{RF}$ and the MIRET tree, respectively

$$U = \frac{|\mathcal{U}_{\widehat{RF}}(1) \cap \mathcal{U}_{\texttt{MIRET}}(1)|}{|\mathcal{U}_{\widehat{RF}}(1)|} \qquad \overline{U} = \frac{|\overline{\mathcal{U}}_{\widehat{RF}}(0) \cap \overline{\mathcal{U}}_{\texttt{MIRET}}(0)|}{|\overline{\mathcal{U}}_{\widehat{RF}}(0)|}$$

In Table 7, we report the cardinality of the sets $\mathcal{U}_{\widehat{RF}}(1)$ and $\mathcal{U}_{\widehat{RF}}(0)$. We report the values of U and $\overline{U}$ for all datasets on the test set in Figure 8. For some datasets, $|\mathcal{U}_{\widehat{RF}}(1)| = 0$, so the value of U cannot be calculated, and it is not shown in the plots. The higher the values of U and $\overline{U}$, the more the MIRET tree mimics the partition of the samples in $U_{\widehat{RF}}(1) \cup \overline{U}_{\widehat{RF}}(0)$ of the $\widehat{RF}$. The samples in the test set that have the highest proximity measure in $\widehat{RF}$ always end up in the same leaf in the surrogate model.

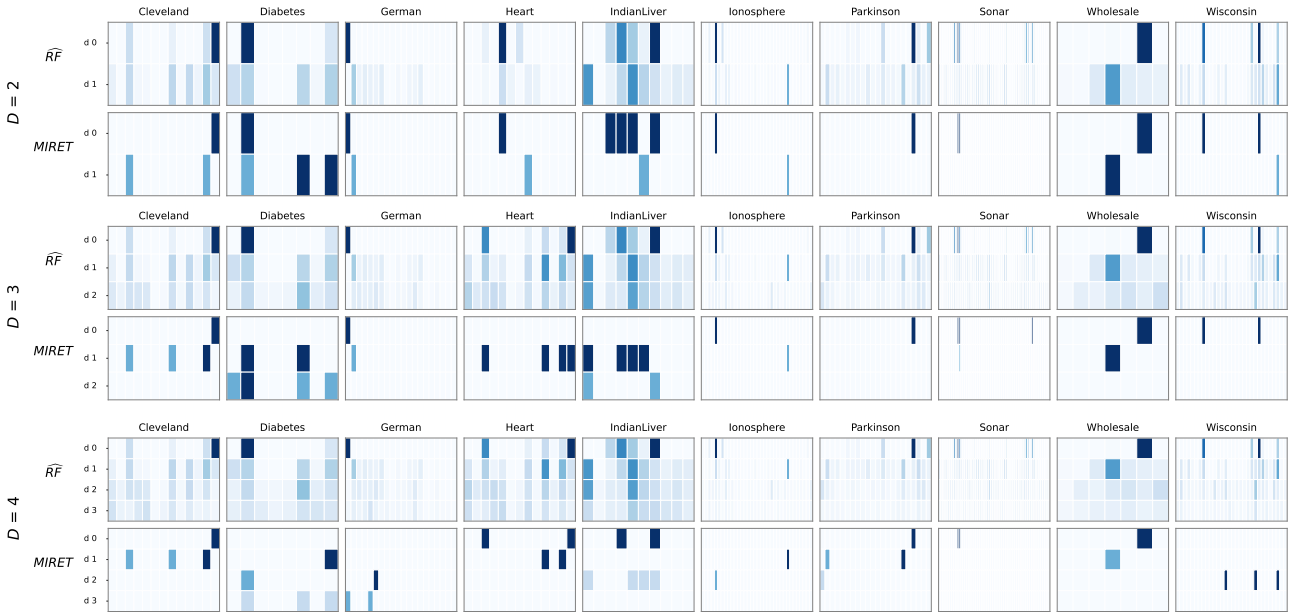

Figure 7: Level frequency of features in $\widehat{RF}$ and MIRET model for $D = 2, 3, 4$

As a final example of the MIRET model, we report in Figure 9 the tree generated with $D = 3$ on the Cleveland problem, which we visualized using VITE with the same setting of $\widehat{RF}$ in Figure 3. We observe the actual depth of the tree is 2 although $D = 3$. Indeed the nodes at level 2 generate only dummy children that are used only to define the class of all the samples. Thus we have the following classification function

$$T^*(x) = \begin{cases} 1 & \text{if } (x_{12} \leq 0 \wedge -0.009x_2 - 0.003x_{11} \leq -0.01) \vee (x_{12} > 0 \ \wedge \ -x_7 + x_{11} > -0.641) \\ -1 & \text{if } (x_{12} \leq 0 \wedge -0.009x_2 - 0.003x_{11} > -0.01) \vee (x_{12} > 0 \ \wedge \ -x_7 + x_{11} \leq -0.641) \end{cases}$$

It happens on several problems that the MIRET tree has an actual depth smaller than the maximum depth D of $\widehat{RF}$. This can be easily derived from Figure 7, where the presence of

	$D=2$		$D=3$		$D=4$	
Dataset	$\lvert\mathcal{U}_{\widehat{RF}}(1)\rvert$	$\lvert\overline{\mathcal{U}}_{\widehat{RF}}(0)\rvert$	$\lvert\mathcal{U}_{\widehat{RF}}(1)\rvert$	$\lvert\overline{\mathcal{U}}_{\widehat{RF}}(0)\rvert$	$\lvert\mathcal{U}_{\widehat{RF}}(1)\rvert$	$\lvert\overline{\mathcal{U}}_{\widehat{RF}}(0)\rvert$
Cleveland	43	245	27	298	4	358
Diabetes	74	1955	3	2802	0	3736
German	105	8572	3	8617	0	8805
Heart	2	112	1	170	0	269
IndianLiver	4	336	1	888	0	1044
Ionosphere	311	176	53	271	46	510
Parkinson	5	85	4	99	3	120
Sonar	0	1	0	4	0	18
Wholesale	388	1592	112	1610	98	1669
Wisconsin	1415	1386	1013	1506	991	1720

Table 7: Cardinality of sets $\mathcal{U}_{\widehat{RF}}(1)$ and $\overline{\mathcal{U}}_{\widehat{RF}}(0)$ on the test set

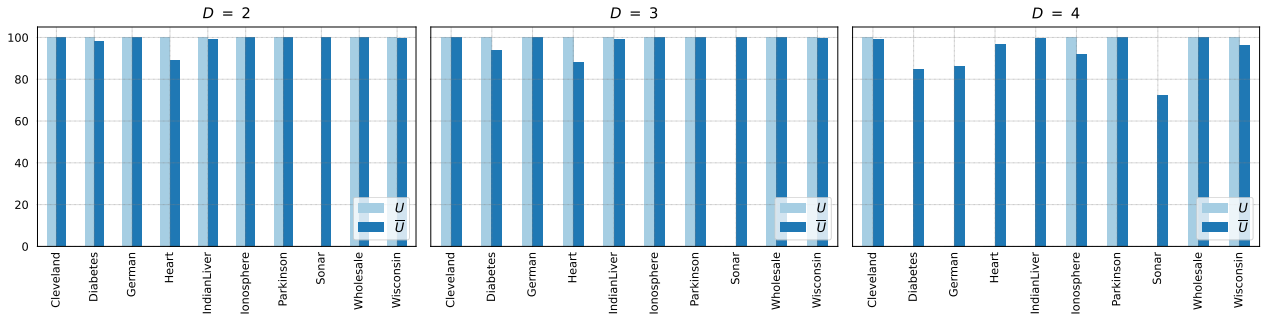

Figure 8: Plot of U and $\overline{U}$: accordance of samples partition of $\widehat{RF}$ and `MIRET` tree on the test set

a level d with no "colored" features in the `MIRET` level frequency plot means that $f_{d,j}^{\texttt{MIRET}} = 0$ $\forall j \in \mathcal{J}$, i.e. none of the nodes at level d applies a splitting rule. Thus, the size of the `MIRET` tree is often smaller than the maximum possible, leading to more interpretable trees.

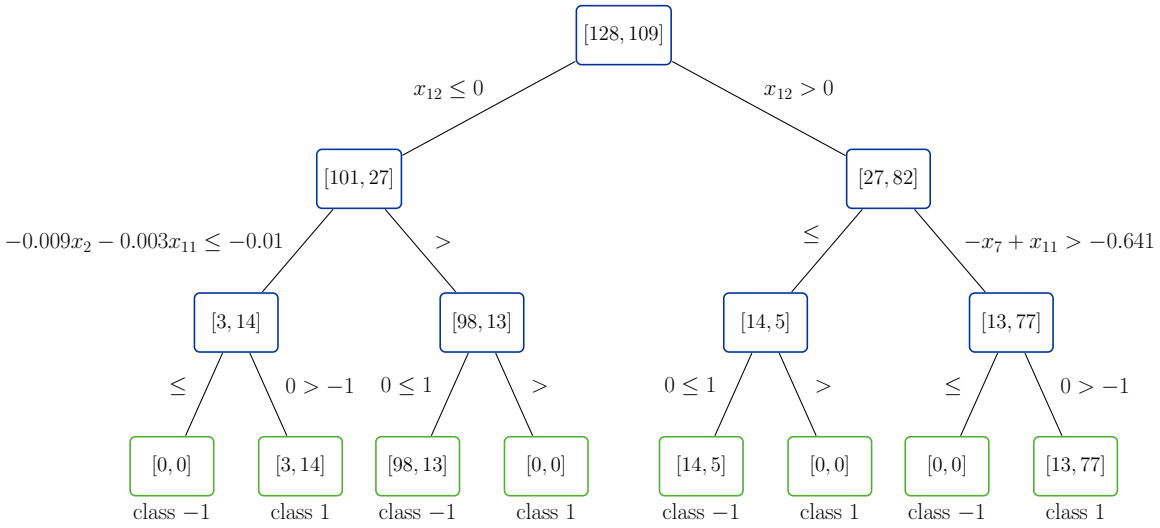

Figure 9: Optimal Tree obtained by `Miret` with $D = 3$ on Cleveland. For each node, we report the number of samples in the class defined by the ground truth: [# negative labels, # positive labels].

9. Conclusion

The paper falls in the field of interpretable representation of a tree-ensemble model, which aims to provide valuable insights into the relationship between the input features and the TE outcomes. Our contribution is twofold. Firstly, we propose a visualization tool `VITE` for ensemble tree models, which allows the user to capture the hierarchical role of features in determining predictions by showing the features' frequency use in the forest. The proposed tool is an addition to the existing visualization tools by helping to understand how features are used in the black-box tree-ensemble model.
Further, we present a mixed-integer linear formulation for learning an interpretable re-built tree (`MIRET`) from a target tree ensemble model (TE). `MIRET` is a multivariate tree with assigned depth D, which optimizes a weighted combination of fidelity to the TE and the number of the features used across the tree, gaining in interpretability at a fixed complexity of the tree. In order to improve consistency with the TE, we extracted information from it, such as level frequencies, proximity measures, and class probabilities, and we embed them into the MILP model. In this paper, we fixed the depth D of the `MIRET` tree to the one of the target TE. However, in principle, by adapting the definition of level frequencies, we can develop optimal trees with any desired depth.
Results on benchmark datasets show that the proposed model is effective in feature selection and yields a shallow interpretable tree while accurately approximating the tree-ensemble decision function. The `MIRET` model offers improved interpretability yielding a single optimal tree with intuitive decision paths, which fairly closely replicates the predictive capabilities of the target random forest model.

References

[1] Aria, M., Cuccurullo, C., and Gnasso, A. (2021). A comparison among interpretative proposals for random forests. *Machine Learning with Applications*, 6:100094.

[2] Bennett, K. P. (1992). Decision tree construction via linear programming. In *Proceedings*

of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conference, pages 97–101.

[3] Bertsimas, D. and Dunn, J. (2017). Optimal classification trees. *Machine Learning*, 106(7):1039–1082.

[4] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[5] Breiman, L. and Cutler, A. (2007 based on copyright year indicated in the authors' Fortran code). Random forests manual.

[6] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. (1984). *Classification and Regression Trees*. Chapman and Hall/CRC.

[7] Breiman, L. and Shang, N. (1996). Born again trees. *University of California, Berkeley, Berkeley, CA, Technical Report*, 1(2):4.

[8] Burkart, N. and Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317.

[9] D'Onofrio, F., Grani, G., Monaci, M., and Palagi, L. (2022). Margin optimal classification trees. *arXiv preprint arXiv:2210.10567*.

[10] Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

[11] Dua, D. and Graff, C. (2017). UCI machine learning repository.

[12] Ehrlinger, J. (2016). ggRandomForests: Exploring random forest survival. *preprint arXiv:11612.08974*.

[13] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232.

[14] Genuer, R., Poggi, J.-M., and Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern recognition letters*, 31(14):2225–2236.

[15] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42.

[16] Hintze, J. L. and Nelson, R. D. (1998). Violin plots: a box plot-density trace synergism. *The American Statistician*, 52(2):181–184.

[17] Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844.

[18] Ishwaran, H. (2007). Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*, 1:519 – 537.

[19] Ishwaran, H., Kogalur, U. B., Blackstone, E. H., and Lauer, M. S. (2008). Random survival forests. *The Annals of applied statistics*, 2(3):841–860.

[20] Ishwaran, H., Kogalur, U. B., Gorodeski, E. Z., Minn, A. J., and Lauer, M. S. (2010). High-dimensional variable selection for survival data. *Journal of the American Statistical Association*, 105(489):205–217.

[21] Liaw, A. and Wiener, M. (2002). Classification and Regression by randomForest. *R News*, 2(3):18–22.

[22] Louppe, G., Wehenkel, L., Sutera, A., and Geurts, P. (2013). Understanding variable importances in forests of randomized trees. *Advances in neural information processing systems*, 26.

[23] Margot, F. (2010). *Symmetry in Integer Linear Programming*, pages 647–686. Springer Berlin Heidelberg, Berlin, Heidelberg.

[24] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[25] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1:81–106.

[26] Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[27] Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., and Zhong, C. (2022). Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85.

[28] Seifert, S., Gundlach, S., and Szymczak, S. (2019). Surrogate minimal depth as an importance measure for variables in random forests. *Bioinformatics*, 35(19):3663–3671.

[29] Tamon, C. and Xiang, J. (2000). On the boosting pruning problem. In *Machine Learning: ECML 2000: 11th European Conference on Machine Learning Barcelona, Catalonia, Spain, May 31–June 2, 2000 Proceedings 11*, pages 404–412. Springer.

[30] Tan, S., Soloviev, M., Hooker, G., and Wells, M. T. (2020). Tree space prototypes: Another look at making tree ensembles interpretable. In *Proceedings of the 2020 ACM-IMS on foundations of data science conference*, pages 23–34.

[31] Vidal, T. and Schiffer, M. (2020). Born-again tree ensembles. In *International conference on machine learning*, pages 9743–9753. PMLR.

[32] Zhao, X., Wu, Y., Lee, D. L., and Cui, W. (2018). IForest: Interpreting random forests via visual analytics. *IEEE Trans. Vis. Comput. Graph.*, 25(1):407–416.

[33] Zhou, Z.-H., Wu, J., and Tang, W. (2002). Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, 137(1):239–263.

Appendix A. Additional tables

Table A.8 presents a summary on all the notation of sets, parameters and hyperparameters adopted in the paper and Table A.9 reports hyperparameters selected with the 4-fold cross validation.

Notation	Description
Sets	
$\mathcal{D}$	Tree levels
$\mathcal{E}$	Tree estimators of the TE
$\mathcal{B}$	Branch nodes
$\mathcal{L}$	Leaf nodes
$\mathcal{B}(d)$	Branch nodes at level d of the tree
$\mathcal{B}'$	Branch nodes not adjacent to leaves
$\mathcal{B}''$	Branch nodes adjacent to leaves
$\mathcal{S}(t)$	Sub-leaves, i.e. leaf nodes of the subtree rooted at node $t \in \mathcal{B}$
$\mathcal{S}_L(t)$	Left sub-leaves, i.e. leaves under the left branch of $t \in \mathcal{B}$
$\mathcal{S}_R(t)$	Right sub-leaves, i.e. leaves under the right branch of $t \in \mathcal{B}$
$\mathcal{I}$	Index set of data samples
$\mathcal{I}_t$	Index set of data samples assigned to node $t \in \mathcal{B}$
$\mathcal{J}$	Index set of features
$\mathcal{J}_\gamma(d)$	Index set of features with a level frequency at d in TE greater than γ_d
$\mathcal{U}(\overline{m}_{TE})$	Pairs of samples with a proximity measure in TE greater than $\overline{m}_{TE}$
Parameters	
p^i	Class probability of sample i predicted by TE
$f_{d,j}$	Frequency of feature j at level d in TE
c_ℓ	Class label pre-assigned to leaf node ℓ
ε	Parameter to model the closed inequality in routing constraints
$m_{i,k}$	Proximity measure between the pair of samples (x^i, x^k)
Hyperparameters	
D	Maximal depth of the tree
α	Penalty parameter for the feature selection
γ_d	Frequency threshold used to determine $\mathcal{J}_\gamma(d)$
$\overline{m}_{TE}$	Proximity threshold used to determine $\mathcal{U}(\overline{m}_{TE})$

Table A.8: Notation: sets, parameters and hyperparameters.

	$D = 2$		$D = 3$		$D = 4$	
Dataset	h	α	h	α	h	α
Cleveland	100/3	0.2	100/3	0.5	100/3	0.5
Diabetes	100/2	0.2	100/3	0.2	100/3	0.5
German	100/2	0.2	100/4	0.8	100/4	0.5
Heart	100/3	0.5	100/2	0.2	100/3	0.5
IndianLiver	0^*	0.2	100/2	0.2	100/2	0.2
Ionosphere	100/3	0.2	0^*	0.5	100/2	0.4
Parkinson	100/4	0.5	100/3	0.5	100/3	0.2
Sonar	100/4	0.6	100/4	0.5	0^*	0.8
Wholesale	100/4	0.2	100/2	0.2	0^*	0.4
Wisconsin	100/2	0.2	100/3	0.5	100/3	0.5

Table A.9: Hyperparameters selected with the 4-fold cross-validation. The apex * indicates that we set $\gamma_d = 0$.