Information Fusion 61 (2020) 124–138

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Explainable decision forest: Transforming a decision forest into an interpretable tree

Omer Sagi*, Lior Rokach

Department of Information Systems and Software Engineering, Ben-Gurion University of the Negev, P.O.B. 653, Beer-Sheva 8410501, Israel

ARTICLE INFO

Keywords:
Classification Trees
Decision forest
Ensemble learning

ABSTRACT

Decision forests are considered the best practice in many machine learning challenges, mainly due to their superior predictive performance. However, simple models like decision trees may be preferred over decision forests in cases in which the generated predictions must be efficient or interpretable (e.g. in insurance or health-related use cases). This paper presents a novel method for transforming a decision forest into an interpretable decision tree, which aims at preserving the predictive performance of decision forests while enabling efficient classifications that can be understood by humans. This is done by creating a set of rule conjunctions that represent the original decision forest; the conjunctions are then hierarchically organized to form a new decision tree. We evaluate the proposed method on 33 UCI datasets and show that the resulting model usually approximates the ROC AUC gained by random forest while providing an interpretable decision path for each classification.

1. Introduction

Decision forest is an umbrella term for ensemble methods that combine multiple decision trees in supervised machine learning tasks. Their ability to aggregate different hypotheses rather than search for a local optima, along with their robustness to different sample sizes and feature spaces, make them popular in many data science challenges [1–3]. However, despite decision forest's high degree of accuracy, other models may be preferable for two main reasons. First, classifications of decision forests are usually inefficient compared to single classifier models as many decision trees are applied to generate a single classification. This attribute becomes a serious vulnerability in real-time predictive systems [4,5]. Second, it is not easy to intuitively explain the rational behind the classifications of decision forests as each classification consists of the results of many trees. This issue usually prevents the use of decision forests in domains that require a clear explanation for individual decisions (e.g., medicine, insurance, etc.) [6].

Previous studies that addressed the above mentioned vulnerabilities of ensemble models can be categorized into two main approaches: ensemble pruning methods and ensemble derived models. The objective of ensemble pruning methods is to search for a subset of ensemble members that performs at least as good as the source ensemble [7]. These methods were shown to significantly improve ensemble performance in terms of complexity and accuracy. The problem of interpretability nevertheless remains unsolved when using such methods, as the resulting ensemble still cannot be interpreted. The notion of deriving a single intelligible model from a given decision forest was tested in a few studies as well. One approach is to train a simple model, using a large set of synthetic or unlabeled data that was classified by a previously trained decision forest [8,9]. However, this approach depends on unlabeled data, which limits its usage to cases where unlabeled data is available or with an unbiased procedure for generating a synthetic dataset. Another approach for transforming a decision forest into a single intelligible classifier is to include a post-processing step. In this step, the decision tree is derived from the structure of the given decision forest [10,11]. A substantial limitation of existing post-processing methods is their high complexity, which prevents their application for large decision forests. In addition, many hyperparameters must be tuned in order to find a suitable setting for a given case.

This paper presents a scalable method for transforming a decision forest into a single decision tree. The resulting decision tree approximates the predictive performance of the original decision forest while providing intelligible and faster predictions. A decision tree has been selected to be the outputted model as it was shown to be interpretable both in terms of its graphical model structure as well as its decomposability, i.e. - each node and decision path can be corresponded to a plain textual description [12]. As opposed to similar methods, the proposed method is suitable for forests of any size and does not require complex

* Corresponding author.
E-mail address: sagyome@post.bgu.ac.il (O. Sagi).

https://doi.org/10.1016/j.inffus.2020.03.013
Received 11 October 2019; Received in revised form 22 March 2020; Accepted 22 March 2020
Available online 26 March 2020

hyperparameter tuning. The method includes two main stages. In the first stage, we create a conjunction set that represents the original decision forest. In the second stage, we build a decision tree that organizes the conjunction set in a tree structure. The remainder of the paper is structured as follows: In Section 2 we lay the scientific background and describe related work. In Section 3, we present the developed method. Section 4 presents an experimental evaluation and discusses its results. Section 5 concludes and suggests future research directions.

2. Background

Ensemble models and specifically decision forests are considered the best practice in many supervised machine-learning tasks, mainly due to their superior predictive performance compared to other models [1,2,13]. Nonetheless, simple models like decision trees might be preferred over decision forests under some circumstances [6,14,15]. Building an interpretable decision tree that approximates the predictive performance of a given decision forest is the subject of this work. The following section provides the relevant scientific background by reviewing interpretable machine-learning models, ensemble learning and related attempts for simplifying decision forests.

2.1. Interpretable machine-learning models

The increasing adoption of machine-learning models in critical areas, like health-care and justice, have stressed the necessity of interpretable machune-learning models [12,16]. Interpertable machine-learning models are models that allow humans to fully understand their entire logic [17]. Alternatively, a model can be defined as interpertable if it is possible to explain the operation of the model using either a simple visualization or a plain text [18]. While predictive performance was always a desired property of machine-learning models, the subject of model interpretability was rarely addressed in past studies [19].

Recently, the ubiquitous presence of machine-learning models has highlighted the fact that there are some scenarios in which model interpretability is equally important as its predictive performance. In one example, it was shown that the COMPAS model, used by several justice institutes to predict crime recidivism, was ethnically biased and led to injustices that could have been prevented by having a better understanding of model decisions. Another example is an Amazon's algorithm that unintentionally excluded minority neighbourhoods from a marketing campaign [20]. In the military domain, DRAPA has recently launched the Explainable AI (XAI) program which aims at producing interpretable models without impairing the prediction accuracy [21]. FICO has announced an XAI challenge of its own that claims to *"create models and techniques that both accurate and provide good trustworthy explanations"* [22]. Finally, in order to be compliant with the General Data Protection Regulation (GDPR) [23], organizations are required to provide interpretations for decisions that were made automatically and hence, to address the *"right to explanation"* [24].

Several studies, mainly from the past few years, have presented different techniques for making machine-learning models more interpretable. Some of the techniques aimed at providing local interpretability while others focused on developing globally interpretable models [17]. Local interpretability is achieved when it is possible to understand the rationale of a single prediction without necessarily understanding the entire structure of the model. LIME manifests this approach by learning an interpretable model locally around the prediction [18]. In SHAP (SHapley Additive exPlanations), a local model is built for determining feature importances in the context of a specific prediction [25]. Selecting a small number of â˘ærepresentativeâĝ samples may also provide local interpretability as domain experts are able to better understand why a given prediction has been made [26]. Local interpretability can be also gained by calculating the marginal contribution of individual input features per output using quasi-random sequences [27]. For some applications it is required to have explanations at cohort and global levels as well (for example: predicting disease progression) and therefore, local explanation models may not suffice [16]. A model is globally interpretable if it allows the understanding of its whole logic and the entire reasoning that leads to all of its possible outcomes. It is broadly agreed that a small set of globally interpretable models are recognized: decision trees, decision rules and linear models [8,28,29]. Several works have presented methods for generating decision rules and decision trees out of pre-trained deep neural networks [30–32] and other black-box models [28,33].

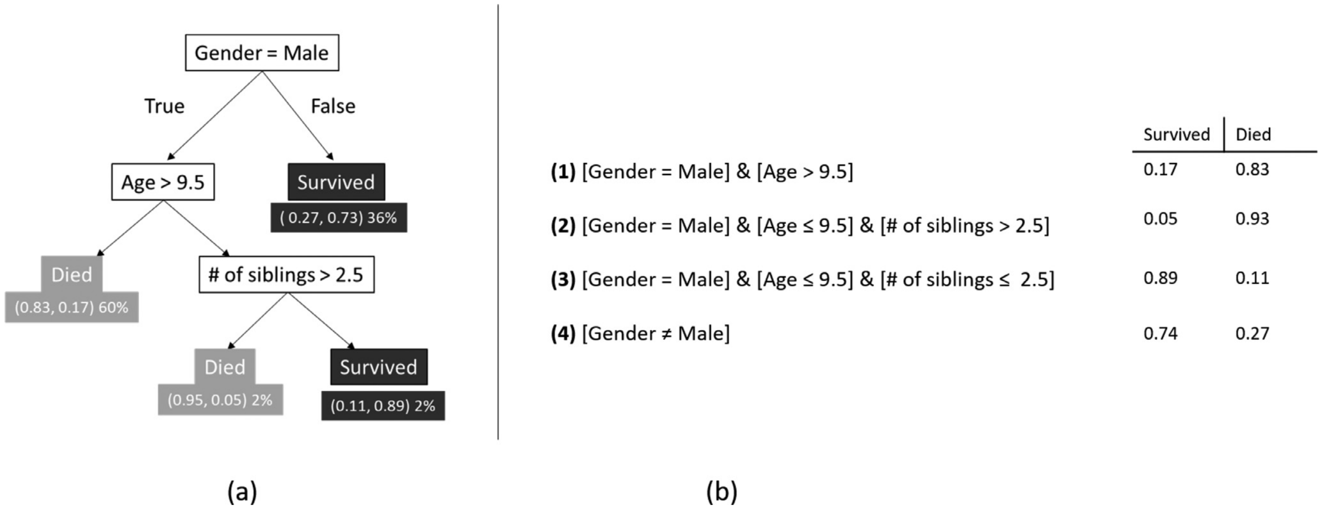
2.1.1. Decision-tree as an interpertable machine-learning model

A decision tree is a predictive model represented as a set of test conjunctions, where each test compares an attribute with a numerical threshold or a set of possible values [34]. Using the Titanic dataset as an example, the goal is to classify whether or not a given passenger will survive the sinking Titanic based on features such as age, gender, the passenger's ticket class or number of siblings, etc. In the decision tree constructed, nodes contain the class distribution and as a result, also the predicted class. Fig. 1a illustrates a decision tree of the Titanic dataset. Internal nodes contain tested features while leaves contain class distributions (the first dimension refers to "died" and the second dimension refers to "survived") and the proportion of associated instances from the training dataset. New instances are assigned to classes based on their path from the root to the leaf. Local decisions can be easily interpreted since each leaf is translated into a clear conjunction of attributes [35,36], as depicted in Fig. 1b. Due to this capability, decision trees are broadly used for machine-learning problems that require the understanding of the model structure as well as its outputs [37,38]. Furthermore, decision trees can contribute to the process of scientific discovery and domain exploration by revealing interesting relationships that have not been considered before [19,30].

The generation of decision trees can be conducted using multiple types of induction algorithms [39–41]. Most induction algorithms search for the best splitting attribute following each partition of the tree, based on a pre-defined splitting criteria (e.g., Gini coefficient, information gain, etc.) [42]. Despite their popularity, decision trees also have several limitations and drawbacks; most of them are related the myopic nature of their induction algorithms [43]. First, they tend to perform well only when there are a few relevant attributes and when there are no complex interactions among the attributes. Specifically, the splitting criterion evaluates attributes based on their immediate descendants - such a strategy may overlook important combinations of attributes. Decision trees are also over sensitive to changes in the training set, which makes them relatively unstable. Finally, decision trees have a fragmentation problem. This usually happens if many attributes are tested along the path. In this case, leaves may consist of a relatively small number of instances, and therefore, their prediction confidence is limited.

2.2. Ensemble Learning and decision forests

Ensemble learning refers to the generation and integration of multiple models to solve a machine learning task. The main premise of this approach is that by combining multiple models, the errors of a single model are likely to be compensated for, and as a result, the overall predictive performance of the model will be improved [13]. These methods are considered the best practice for challenges involved with relational datasets, where each sample (e.g., person) consists of meaningful features of different nature (e.g., age, gender) [44,45]. As opposed to Deep neural networks, which perform better in image or sequence-like datasets [46], ensemble models can work sufficiently well with small datasets [1]. Decision forests manifest the ensemble learning approach by training multiple decision trees and provide an aggregation mechanism for transforming their results into a single decision [1]. Research performed in the past few years presented the successful use of decision



	Survived	Died
(1) [Gender = Male] & [Age > 9.5]	0.17	0.83
(2) [Gender = Male] & [Age ≤ 9.5] & [# of siblings > 2.5]	0.05	0.93
(3) [Gender = Male] & [Age ≤ 9.5] & [# of siblings ≤ 2.5]	0.89	0.11
(4) [Gender ≠ Male]	0.74	0.27

Fig. 1. Titanic survival decision tree.

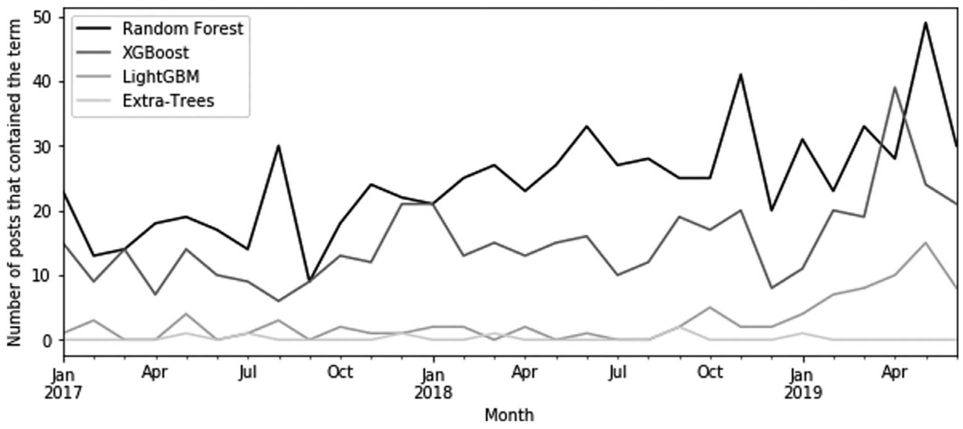


Fig. 2. Number of StackExchange data-science related discussions per decision forest algorithm over time.

forests in object detection [47], precision agriculture [48], malware detection [49] and many other domains [50]. There are several reasons for the superior predictive performance of ensemble methods that is often observed. First, the risk of obtaining a local minimum is decreased when combining several learners. In addition, when there is only a small amount of data, a single learning algorithm is prone to choosing an incorrect hypothesis. Finally, combining different models may extend the search space, especially in cases where the optimal hypothesis is outside of any single model's scope. Ensemble inducers are usually divided into two frameworks: the dependent framework and the independent framework. In the dependent framework, the output of each inducer affects the construction of the next inducer. In the independent framework each inducer is built independently from other inducers. AdaBoost [51], gradient boosting machines and specifically XGBoost [2] are the most popular algorithms belonging to the dependent framework, while bagging [52], and random forest are the most popular independent framework algorithms. Random forest [53,54] is the most popular decision forest model [55], primarily due to its stability and robustness with datasets of any size [56]. As of 2019, random forest was the most discussed ensemble algorithm in the StackExchange technical forum discussions, as illustrated in Fig. 2. When training a random forest, individual decision trees are trained using different samples of the instances, where at each split, the learning algorithm randomly samples a subset of the attributes and chooses the best split among them. A contemporary comparison found that random forest performs better then gradient boosting machines when the number of features is 10 or less while the later is exceptional when the number of features is between 11 and 50 [45].

2.3. Simplifying ensemble models

Despite their high predictive performance, there are cases in which models with a higher error rate might be preferred over ensemble models. For example, large ensembles may raise latency issues in real-time systems, as each prediction requires the aggregation of many individual models. In such cases, it is more reasonable to select models that allow faster predictions [4,57]. Another example involves cases in which predictions should be justified by a clear and rational explanation (e.g., medicine, insurance, etc.) [6], and hence, intelligible models like decision trees, logistic regression, or k-nearest-neighbors may be preferred, despite their higher error rate. Approaches developed in order to mitigate these issues can be divided into two categories: ensemble pruning methods and ensemble derived models [55].

2.3.1. Ensemble pruning methods

Ensemble pruning methods, also known as ensemble selection methods, were developed with the aim of reducing the complexity of ensemble models. These methods search for a subset of ensemble members that performs at least as well as the original ensemble [7]. Classifiers may be pruned or selected by either ranking methods or by search methods. Ranking methods rank the individual members based on predetermined criteria. Ranking may be based on accuracy-guided forward search, cross-validation based search, collective-agreement based pruning that takes into account both performance and redundancy, etc [5,58,59]. Search-based methods conduct a heuristic search in the classifiers' space and evaluate the collective performance of different classifiers' sub-sets [60–62]. Ensemble pruning methods have been shown to significantly improve ensemble performance in terms of complexity. However, they do not solve the interpretation problem - i.e., the pruned ensemble and its predictions are still not human understandable.

2.3.2. Ensemble derived models

Ensemble derived models are created by leveraging the properties of a given ensemble model to generate a single model. The goal of the new model is to preserve the predictive performance of the inputted ensemble while providing faster, and often intelligible, classifications. One family of methods that manifest this approach trains a single model to learn the classifications of a pretrained ensemble model. One example is a method that uses an ensemble model to label a vast amount of unlabeled data and then uses the new labeled data to fit a neural network [9]. In CMM [8], a new model is trained by learning the data partitioning implicitly from the ensemble, using artificial data. Following training an ensemble with a labelled dataset, artificial unlabelled data is generated by using a selected data generation method (e.g., kernel estimate [63]) The artificial data is then labelled by an ensemble model that was trained using the available training data. The final decision tree is than trained using the manufactured data labels. This approach has been successfully applied in the medical domain in diagnosing depression based on patient questionnaires [33]. It is important to mention that CMM is highly dependent on the availability of unlabelled data or a data generation method that suits the training set. Another approach for deriving a single model from a given ensemble is to add a post-processing stage that manipulates the ensemble's inner structure to compose a new model. An early work utilizing this approach presented an algorithm that generates if-then rules from each of the ensemble classifiers and converts the rules into binary vectors. These vectors are then used as training data for learning a new decision tree [64]. Van Assche and Blockeel proposed the interpretable single model (ISM), an algorithm for constructing a single decision tree that iteratively chooses the most informative node for splitting the new tree based on the structure of the source decision forest [10]. Genetic extraction of a single, interpertable model(GENESIM) is a recently developed method that applies genetic algorithms to transform ensemble inducers into a single decision tree [11]. The complexity of both ISM and GENESIM is exponential with the size of the decision forest and the size of the data. Therefore, these methods cannot be applied on forests that contain more than a few trees. The evaluation of GENESIM was performed using a decision forest of 25 trees as the source model, whereas for ISM, the authors used a forest of 33 trees. Another limitation of these algorithms is their high sensitivity to different hyperparameter values.

3. Forest based tree (FBT)

This section presents a method that uses a trained decision forest for generating a single decision tree in a post-processing manner. We name the new model forest based tree (FBT). The main contribution of FBT is in expanding the range of models that can be used in cases where there is a trade-off between predictive performance to prediction time or prediction interpertability. In addition, in contrast to existing methods, this method can be applied on large decision forests without requiring hyperparameter tuning.

3.1. Approach overview

Given a dataset of n examples, m features and c different classes $(D = \{(x_i, y_i)\}\ |D| = n, x_i \in R^m, y_i \in 1, \dots, c)$, a decision forest aggregates K additive functions and maps an m-dimensional feature vector into a c-dimensional probability vector as follows:

$$\phi(x_i) = \frac{\sum_{k=1}^{|T|} t_k(x_i)}{|T|}, t_k \in R^c \tag{1}$$

where T is a set of decision trees contained in the decision forest. The proposed method aims at building a new tree $\hat{t}$ such that:

$$\forall x_i, \hat{t}(x_i) \approx \phi(x_i) \tag{2}$$

The method consists of two main stages:

1. Building a set of rule conjunctions from the given decision forest.
2. Organizing the set of conjunctions in a tree structure that will enable fast predictions for unseen instances.

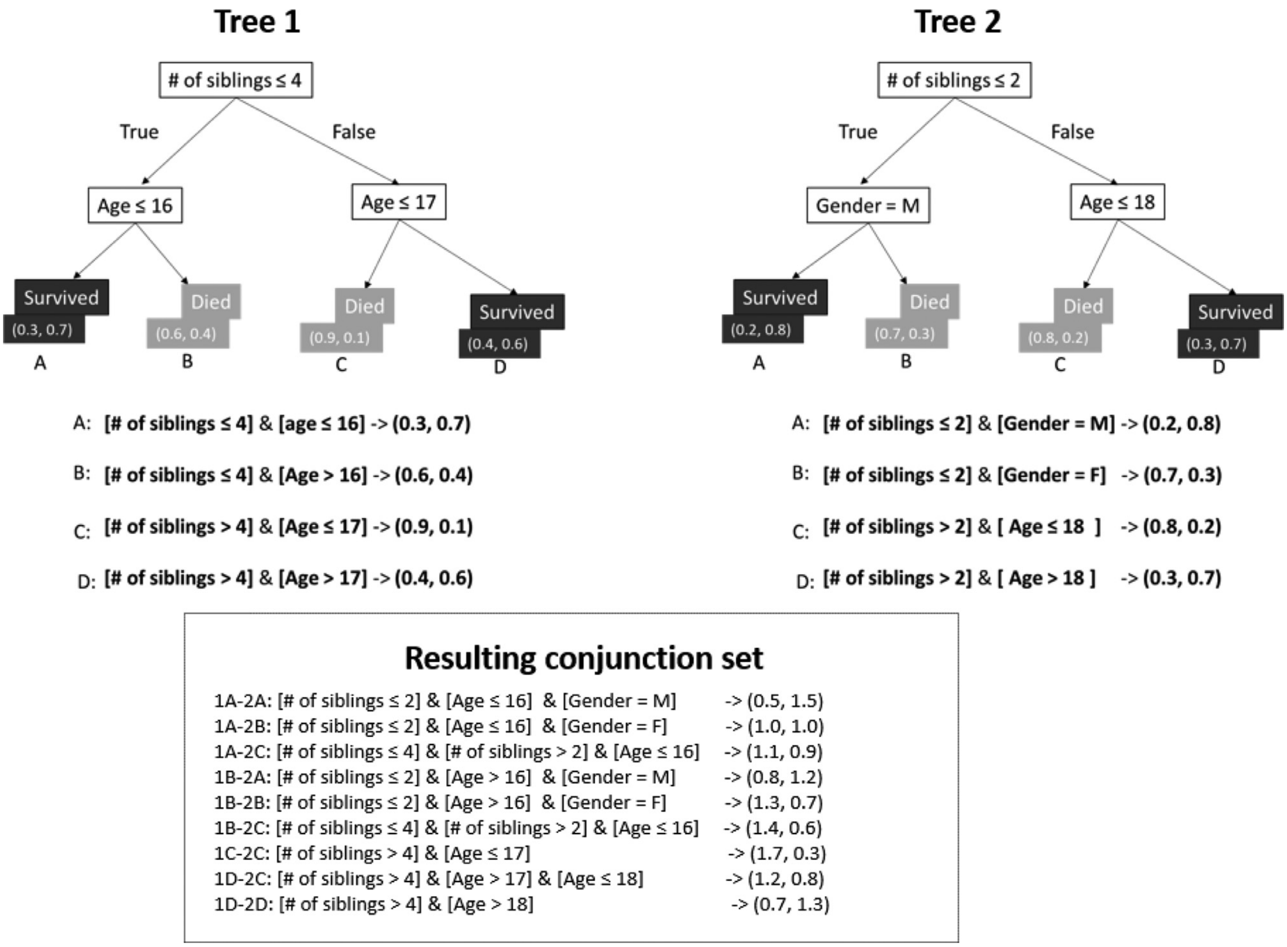
The main premise behind this method is that both decision forests and decision trees can be represented as sets of disjoint conjunctions of rules. Once a conjunction set representation of a decision forest has been created, conjunctions can be organized in a tree structure which will enable intelligible and faster predictions. It is important to add that the proposed method does not consider any dependencies across different trees. Therefore, it is more reasonable to use this method for independent decision forests (e.g., random forest, rotation forest, etc.).

3.2. Conjunction set generation

In this stage we divide the decision forest into a set of rule conjunctions where each rule conjunction corresponds to a possible output of the decision forest. As a preliminary step, we ignore the hierarchical structure of a decision tree t_i that is included in decision forest T. Instead, we view the tree as a set of rule conjunctions CS_i. Each conjunction $(c_{ij}, \hat{y}_{c_{ij}})$ in CS_i is a sequence of logical rules c_{ij} mapped to $\hat{y}_{c_{ij}}$, a K-dimensional vector where K is the number of labels and each cell holds the probability of the corresponding class. Merging two conjunction sets CS_1 and CS_2 can be done by applying a Cartesian product where each conjunction $(c_{1j}, \hat{y}_{c_{1j}})$ is merged with conjunction $(c_{2j}, \hat{y}_{c_{2j}})$ to create a new conjunction $(c_{1j} \wedge c_{2j}, \hat{y}_{c_{1j}} + \hat{y}_{c_{2j}})$. Fig. 3 illustrates the merging of two simple decision trees trained for predicting the survival of Titanic passengers. We take the merging of leaves 1A (leaf A of tree 1) and 2A (leaf A of tree 2) into a new conjunction 1A-2A as an example; these two conjunctions include four rules that can be reduced into three rules since [# of siblings < 4] is contained in [# of siblings < 2]. It is important to note that although there are four conjunctions in each set, the combined set consists of nine conjunctions and not sixteen. This is the result of excluding conjunctions that contain incompatible rules. For example: conjunctions 1A and 2D cannot be merged as the resulting conjunction will include incompatible age rules.

3.2.1. Addressing complexity challenges

Extending this method to fit forests of any size is theoretically trivial, since more trees can always be added to the resulting conjunction set. However, such an extension raises practical challenges as the complexity of merging a forest of K trees, where each tree contains M leaves, will be bounded in $O(M^K)$. This complexity prevents the usage of the proposed method for forests of many trees or for machine-learning challenges with a large feature space. We address this issue by proposing a heuristic that limits the size of the conjunction set in each iteration. More specifically, we define a threshold L as the maximum allowed conjunctions in each iteration, we estimate the probability of a given conjunction $\{r_1 \wedge r_2 \dots \wedge r_n\}$ as the product of its rules' independent probabilities $\{P(r_1) \cdot P(r_2) \dots \cdot P(r_n)\}$ where $P(r_n)$ is calculated as the empirical probability of having the rule in the training set. Using Fig. 3 as an example, if 10% of the Titanic passengers had more than Four siblings and 20% of the passengers were older than Eighteen, then $P(1D - 2D)$ would be $(0.1 \times 0.2 = 0.02)$. In each iteration we include only the top L conjunctions in terms of conjunction probability so the complexity of the algorithm is $O(LK)$. The developed heuristic considers the existence



A: **[# of siblings ≤ 4] & [age ≤ 16] -> (0.3, 0.7)**

B: **[# of siblings ≤ 4] & [Age > 16] -> (0.6, 0.4)**

C: **[# of siblings > 4] & [Age ≤ 17] -> (0.9, 0.1)**

D: **[# of siblings > 4] & [Age > 17] -> (0.4, 0.6)**

A: **[# of siblings ≤ 2] & [Gender = M] -> (0.2, 0.8)**

B: **[# of siblings ≤ 2] & [Gender = F] -> (0.7, 0.3)**

C: **[# of siblings > 2] & [Age ≤ 18] -> (0.8, 0.2)**

D: **[# of siblings > 2] & [Age > 18] -> (0.3, 0.7)**

Resulting conjunction set

1A-2A: [# of siblings ≤ 2] & [Age ≤ 16] & [Gender = M] -> (0.5, 1.5)
1A-2B: [# of siblings ≤ 2] & [Age ≤ 16] & [Gender = F] -> (1.0, 1.0)
1A-2C: [# of siblings ≤ 4] & [# of siblings > 2] & [Age ≤ 16] -> (1.1, 0.9)
1B-2A: [# of siblings ≤ 2] & [Age > 16] & [Gender = M] -> (0.8, 1.2)
1B-2B: [# of siblings ≤ 2] & [Age > 16] & [Gender = F] -> (1.3, 0.7)
1B-2C: [# of siblings ≤ 4] & [# of siblings > 2] & [Age ≤ 16] -> (1.4, 0.6)
1C-2C: [# of siblings > 4] & [Age ≤ 17] -> (1.7, 0.3)
1D-2C: [# of siblings > 4] & [Age > 17] & [Age ≤ 18] -> (1.2, 0.8)
1D-2D: [# of siblings > 4] & [Age > 18] -> (0.7, 1.3)

Fig. 3. An example of merging two conjunction sets (illustrative numbers).

of redundant conjunctions, i.e., conjunctions that are not likely to be used for unseen instances due to their low likelihood. For example: assuming there are very few people with more than four siblings on the Titanic and even fewer seventeen year old, then it is very unlikely that conjunction 1D-2C will be applied in predicting new instances. Furthermore, we demonstrate this phenomenon with a simple experiment. We trained random forests that contain 5, 10 and 20 trees for three different datasets (Iris, Breast cancer and Ecoli). We used a separate test set for conducting predictions and measured three main metrics:

1. the number of possible conjunctions stemming from the decision forest
2. the number of conjunctions that were assigned to at least one instance from the test set, and
3. the ratio of conjunctions used.

We repeated this experiment 30 times for each setting. Table 1 presents the average score for each metric. In this table we can see evidence for cases in which a linear increase in the forest size resulted in an exponential increase in the conjunction set size, as well as in an exponential decrease of the proportion of relevant conjunctions. The intuition for selecting the product of rulesâ independent probabilities is illustrated in Fig. 4. This example presents the distribution of conjunction probabilities for conjunctions that were applied in predicting at least one instance versus conjunctions that were not applied. The underlying model is a random forest, consisting of 20 trees, that was trained on the Iris dataset. As shown in Fig. 4, conjunctions that were used for prediction were significantly more probable (in terms of the empirical product of probabilities) than conjunctions that were not used.

Another way to increase the relevance of conjunctions included is to eliminate redundant trees by using ensemble pruning methods. In this work we use a greedy method [65] that iteratively adds base models that increase the AUC (area under the curve) of the forest until no improvement is achieved. The greedy search is done iteratively in the implementation that is presented in the experimental evaluation section. However, This stage can be parallelized so increasing the number of estimators would not result with a dramatic increase of training time. It is important to note that the proposed heuristics for pruning decision trees, as well as for filtering the conjunction set, can be replaced with other approaches and may serve as the subject of future research.

3.3. *Tree generation*

The resulting conjunction set can be used as a classification model by searching for the conjunction that fits a given instance as input. However, taking this approach would be extremely inefficient, since numerous conjunctions might be tested before matching the right conjunction. In addition, since some conjunctions are filtered during the construction process, it is very likely that some instances would not be assigned to any of the remaining conjunctions. At this stage we introduce an algorithm that addresses these issues whose objective is to organize the conjunction set in a tree structure that enables efficient classification and provides a mechanism for matching the most suitable subset of conjunctions to unseen instances, even in cases where none of the conjunctions perfectly matches the given instance. As a preliminary stage, a dataset that represents the conjunction set is created where each row stands for a single conjunction. In this dataset, each numerical feature is mapped

Table 1
Conjunction usage ratio.

Dataset	Forest size	Number of conjunctions	Number of conjunctions used	Usage ratio
Breast cancer	5	204.95	59.73	0.34
	10	3224.90	103.00	0.04
	20	27916.43	116.27	0.01
Ecoli	5	464.87	57.35	0.15
	10	11150.23	89.73	0.01
	20	100332.93	94.63	0.00
Iris	5	71.21	13.50	0.20
	10	556.58	20.85	0.05
	20	1810.37	23.15	0.02

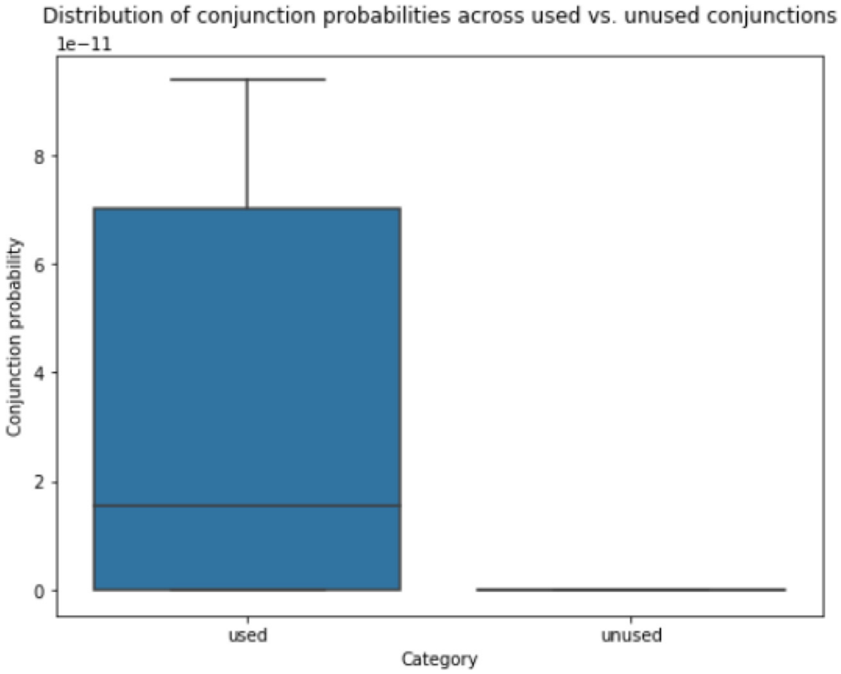


Fig. 4. Box plot of used vs. unused conjunction probabilities for an example of a random forest of 20 trees that was trained for predicting iris species.

Table 2
Conjunction set - represented as a dataset for constructing the new tree.

	age_lower_bound	age_upper_bound	gender	number_of_siblings_lower_bound	number_of_siblings_upper_bound	probs	prediction
1A-2A	-infinity	16	M	-infinity	2	(0.5, 1.5)	1
1A-2B	-infinity	16	F	-infinity	2	(1, 1)	0
1A-2C	-infinity	16	(M,F)	2	4	(1.1, 0.9)	0
1B-2A	16	infinity	M	-infinity	2	(0.8, 1.2)	1
1B-2B	16	infinity	F	-infinity	2	(1.3, 0.7)	0
1B-2C	-infinity	16	(M,F)	2	4	(1.4, 0.6)	0
1C-2C	-infinity	17	(M,F)	4	infinity	(1.7, 0.3)	0
1D-2C	17	18	(M,F)	4	infinity	(1.2, 0.8)	0
1D-2D	18	infinity	(M,F)	4	infinity	(0.7, 1.3)	1

into two columns: one for its upper bound and one for its lower bound at the corresponding conjunction. Categorical features on the other hand, are mapped into one column where each cell contains the possible feature values of the corresponding conjunction. In addition, the dataset stores the class probabilities and thereby, the predicted class of each conjunction. Table 2 is an example of a dataset that represents the conjunction set presented in Fig. 3.

Constructing the decision tree is initiated by defining a root node that contains the entire set of conjunctions. Splitting a node according to a selected rule r is done by routing all the conjunctions that are relevant for instances that fit r to the left node and routing conjunctions that are relevant for instances that contradict r to the right node. Fig. 5 presents an example of using the gender as a splitting criterion for the dataset provided in Table 2. In this example conjunctions $1A-2A$ and $1B-2A$ explicitly fit males while conjunctions $1A-2B$ and $1B-2B$ explicitly fit females. The rest of the conjunctions, do not have an explicit gender requirement and therefore, they are routed to both descendants. The entropy of a node that contains conjunction set CS_i is calculated as the entropy of the prediction vector (the prediction column in Table 2). Given a rule r that splits a conjunction set CS_i into two conjunctions sets CS_{i1} and CS_{i2}, the information gain of the split is calculated as:

$$IG(CS_i, R) = \frac{|CS_{i1}| \cdot entropy(CS_{i1}) + |CS_{i2}| \cdot entropy(CS_{i2})}{|CS_{i1}| + |CS_{i2}|} - entropy(CS_i) \quad (3)$$

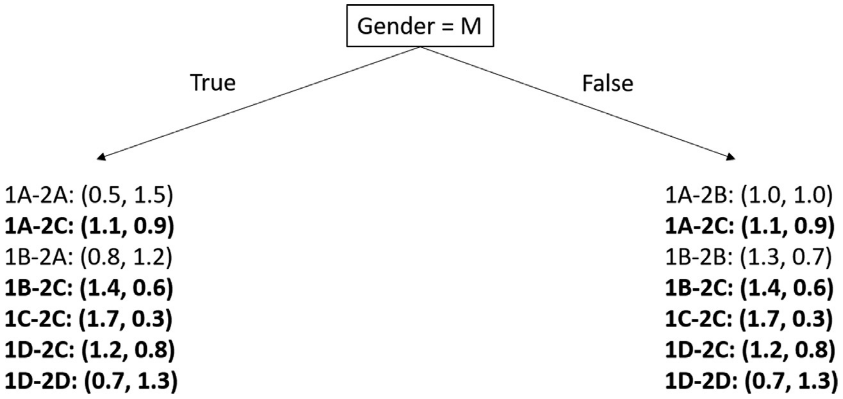


Fig. 5. Splitting the conjunction set by gender. Conjunctions included on both sides are marked in bold.

At each node we select the rule with the highest information gain as the splitting criterion. Splitting a node that contains CS_i is stopped when the class with the highest probability is the same for all of the conjunctions ($entropy(CS_i) = 0$). More specifically, we define a node as a leaf if all of its conjunctions point to the same predicted class or when the best splitting rule does not reduce the number of conjunctions in at least one of the resulting descendant nodes. Applying the resulting tree to predict the class probabilities of a new instances is done by routing the instance to its associated leaf and averaging the class vectors of all of the conjunctions that are included in the leaf.

3.4. Algorithm

The pseudo-code of our method is presented in Algorithm 1 . As inputs, the method gets a decision forest T, a dataset $D_{pruning}$ for the pruning stage, the $forest\_min\_size$ parameter as the lower bound for the pruned forest size, and the L parameter which limits the algorithm complexity by serving as an upper bound for the conjunction set size. In lines 1–4, a pruned decision forest T' is created by iteratively adding trees that maximize the AUC of T' for the pruning set, $D_{pruning}$. In lines 5–21 a conjunction set that represents the decision forest is generated iteratively. The first conjunction set is simply defined as the conjunction set CS_0 of the first decision tree that was added to T'. Each conjunction c_{ij} in CS_i is a logical conjunction of K rules $\{r_1, r_2 \ldots .r_k\}$ that is mapped into an M dimensional vector $\hat{y}$ where M is the number of classes and each dimension holds the corresponding class probability. CS_i is built by merging conjunctions of CS_{i-1} with conjunctions of t_i (also referred as leaves). In lines 18–19 we calculate the probability $P(c_{ij})$ of each conjunction as the product of its rules' independent probabilities. We then exclude all of the conjunctions c_{ij} that are ranked below the top L conjunctions. Lines 22–34 describe the hierarchical ordering of $CS_{|T'|}$ to decision nodes. The entropy of a node is calculated as the entropy of $(argmax(\hat{y}_1), argmax(\hat{y}_2), argmax(\hat{y}_K))$ of all of the conjunctions included in the node. $argmax(\hat{y}_k)$ denotes the class with the highest probability in conjunction k. The split is made by routing the conjunctions that may be assigned to instance that satisfy the splitting rule into the left node and routing all of the conjunctions that may be assigned to instances that do not satisfy the splitting rule into the right node. Predicting class probabilities for an unseen instance is done by routing the instance to its associated leaf and returning the average vector of the leafâs class vectors as an output.

Algorithm 1: Generate a decision tree using an existing decision forest.

```
   Input: $T$ (A set of $n$ decision trees $\{t_0, t_1, \ldots t_n\}$), $D_{pruning}$ (pruning
          dataset), $forest\_min\_size$ (minimum forest size), $L$
          (Maximum size of conjunctions at each iteration)
   Output: $DT$ (a decision tree)
1  $T' = \{\}$ - Pruned decision forest, $AUC_{T'}(D_{pruning}) = 0$
2  while $AUC_{T'}(D_{pruning})$ was improved or $|T'| < forest\_min\_size$ do
3      Add to $T'$ the decision tree $t$ from $T$ that maximizes
       $AUC_{T'}(D_{pruning})$
4  end
5  for each decision tree $t_i$ in $T'$ do
6      if $i = 0$ then
7          $CS_0$ =conjunction set of $t_0$
           ($CS_i = \{(c_{i1}, \hat{y}_{c_{i1}}), (c_{i2}, \hat{y}_{c_{i2}}) \ldots (c_{iJ}, \hat{y}_{c_{iJ}})\}$ where $c_{ij}$ is a
           conjunction of rules and $\hat{y}_{c_j}$ is a vector of classes
           probabilities
8      end
9      else
10         $CS_i = \{\}$;
11         for leaf $c_j$ in $t_i$ do
12             for conjunction $c_k$ in $CS_{i-1}$ do
13                 if $c_j$ does not contradict $c_k$ then
14                     add $(c_j \wedge c_k, \hat{y}_{c_j} + \hat{y}_{c_k})$ to $CS_i$
15                 end
16             end
17         end
18         order $CS_i$ by $P(c_{ij})$
19         $CS_i$ = top $L$ conjunctions in $CS_i$
20     end
21 end
22 Define $DT$ as a single node that contains $CS_{|T'|}$
23 for each node at $DT$ that wasn't splitted or defined as a leaf do
24     for each splitting candidate rule $r_{ij}$ in $\{r_{11}, r_{12} \ldots r_{iL}\}$ do
25         Find Information Gain $IG$ from split($CS_{node}, r_{ij}$)
26     end
27     define $r'$ as the attribute with the highest information gain
28     if $IG(r') = 0$ then
29         define current node as a leaf
30     end
31     else
32         split current node by $r'$
33     end
34 end
```

4. Experimental evaluation

The effectiveness of the proposed forest based tree was evaluated by carrying out an experimental study as described below. The experimental study compared different variations of the forest based tree with several benchmark classifiers by considering two evaluation criteria: predictive performance and classification complexity. The predictive performance was assessed using the multiclass extension of the ROC AUC measure that aggregates the ROC AUC values over each pair of classes [66]. ROC AUC was proven to be a better measure than generalized accuracy as it is independent of the decision threshold and invariant to a priori class probability distributions [67]. We also measured the generalized accuracy as an additional predictive performance metric. In order to measure the classification complexity, we used the average prediction depth [68,69], which was calculated as the number of rule tests that each instance had to pass to generate its associated prediction. We selected this measure as it is independent of the implementation efficiency as opposed to classification runtime.

4.1. Experiment settings

The algorithm implementation was conducted with Python, using scikit-learn's random forest (RF) and extra-trees classifier [70] (ET) as decision forest models, trained with CARET trees as forest inducers. The number of base-models in each of the forests was set to be 100. The maximum depth of random forest's base trees was set to be five while the maximum depth of the extra-tree classifiers was set to be three. We trained two versions of the forest based tree. One was derived from the random forest model (RF-FBT) and the second was derived from the extra-trees classifier (ET-FBT). For both version the maximum allowed number of conjunctions per iteration was defined as 3000 and the minimum size of the pruned forest was set to be 10. The implementation that was used for this experiment, as well as the experiment setting and the datasets are publicly available in the following link: https://github.com/sagyome/forest_based_tree

Several models have been trained to serve as benchmarks to the forest based trees. A CARET decision tree (DT), implemented as a part of Python's Sklearn, was trained by applying grid-search that selects the best model for the given training data, based on 10-fold cross validation. Different hyper-parameters were considered when selecting the best decision tree such as maximum depth, splitting criterion, and minimum number of samples per leaf. Another model that was used as a benchmark is Domingos's combined multiple models (CMM) [8]. The model is built by generating a synthetic dataset that is twenty times larger than the original training set. The synthetic dataset was generated by using an open source code that implements a method that consider the correlations among the features when generating the synthetic data [71]. Following the generation of the synthetic data, the pre-trained random forest (RF) classifies the new instances that are then served as input data to train a simple decision tree. GENESIM and ISM were trained as an ensemble-derived benchmarks by running the open-source code that was published by the GENESIM authors. Default hyper-parameters were applied when training these two models. It is important to note that as GENESIM and ISM are not scalable for high-dimensional datasets, there were several cases where these algorithms did not manage to train a decision tree. The result tables were filled with NA values in cells that are corresponding to these cases and the statistical tests for the overall difference among the models do not include these datasets. The CN2 algorithm [72] was applied for training a classification rules benchmark. It was done by using Python's Orange implementation [73]. It is worthwhile mentioning that the complexity of CN2 was not evaluated as it does not have a tree structure or a forest structure like the other methods. Finally, we defined the augmented forest-based tree (AFBT) as a combination of the RF-FBT and DT model in the following way: in cases where the AUC of the decision tree was less than 99% of RF AUC, the value of AFBT was defined to be the value of RF-FBT. Otherwise, the value of AFBT was set to be the value of DT. The motivation for defining this approach was to test whether there are scenarios in which it would be preferable to opt the forest based tree method over a regular decision tree.

The evaluation criteria were compared for 33 classification tasks from the UCI repository. Table 3 summarizes the characteristics of the datasets that were used in this empirical study. It is important to note that since training time is dependent on code optimization, it was not measured and compared across the different methods. For each of the 33 dataset, we randomly divided the dataset into train and test sets in a 80/20 ratio respectively. The pruning of random forest for the FBT algorithms was conducted by using the training data as the pruning set. We measured the evaluation bi-criteria (ROC AUC, generalized accuracy and prediction depth) based on the test set predictions. This procedure was repeated 10 times.

Table 3
Datasets characteristics. # catg - Number of categorical features, # Real - Number of continous features.

Dataset	# Catg	# Real	# Instances	# Labels	Majority label proportion
Abalone	0	8	4177	28	0.16
Acute-inflam	0	6	120	2	0.51
Acute-nephitis	0	6	120	2	0.58
Aust_credit	0	14	690	2	0.56
Balance_scale	0	4	625	3	0.46
Bank	9	7	4521	2	0.88
Banknote	0	4	1372	2	0.56
Biodeg	0	41	1055	2	0.66
Breast cancer	0	9	683	2	0.65
Car	6	0	1728	4	0.7
Credit	9	6	642	2	0.53
Cryotherapy	0	6	90	2	0.53
Divorce	0	54	170	2	0.51
Ecoli	0	7	336	8	0.43
Forest	0	27	523	4	0.37
German	13	7	1000	2	0.7
Glass	0	9	214	6	0.36
Haberman	0	3	306	2	0.74
Internet	4	0	322	2	0.7
Iris	0	4	150	3	0.33
Kohkiloyeh	5	0	100	2	0.68
Liver	1	9	579	2	0.72
Magic	0	10	2000	2	0.66
Mamographic	0	5	830	2	0.51
Nurse	8	0	90	4	0.7
Occupancy	0	5	2665	2	0.64
Pima	0	8	768	2	0.65
Seismic	4	14	2584	2	0.93
Spambase	0	57	4601	2	0.61
Tic-tac-toe	9	0	958	2	0.65
Vegas	12	5	504	5	0.45
Winery	0	13	178	3	0.4
Zoo	0	16	101	7	0.41

4.2. Results

Table 4 presents the ROC AUC of the decision forest methods and the different variations of the proposed forest based tree. Each cell stores the average and standard deviation of 10 runs in which a given model has been trained and tested on the corresponding dataset. Similarly, Table 5 presents the ROC AUC for the benchmark decision trees and rule classifiers. Table 6 summarizes the generalized accuracy values of decision forests and forest based trees while Table 7 presents the generalized accuracy of the remaining benchmark methods. The complexity results, measured as the average number of rules that were applied per prediction (i.e., the prediction depth), are summarized in Table 8 for decision forests and forest based trees. Table 9 summarizes the complexity results for the rest of the benchmark models. In all of the above mentioned tables, we mark in bold the cells of the model with the best average value for the corresponding dataset as well as cells of models that approximate the best value based on a Wilcoxon signed-rank test. $p\_value = 0.05$ serves as the upper threshold for significant difference between two models. In order to evaluate the overall difference among the different models, we followed a procedure described in [74]. We first conducted an adjusted Friedman test that rejected the null hypothesis that all models performed the same in terms of AUC ($F_f = 142.92, p \approx 0$),

Table 4
AUC comparison – Decision forests and forest based trees.

Dataset	RF	ET	ET-FBT	RF-FBT	AFBT
Abalone	**91.02 ± 0.4**	89.32 ± 0.32	90.07 ± 0.31	89.73 ± 0.54	90.14 ± 0.49
Acute-inflam	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**
Acute-nephitis	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**
Aust credit	**90.49 ± 1.54**	89.43 ± 1.03	89.52 ± 1.67	89.09 ± 2.69	**90.67 ± 0.96**
Balance_scale	**91.06 ± 2.5**	88.29 ± 2.74	79.13 ± 3.41	**91.22 ± 2.37**	**91.22 ± 2.37**
Bank	**96.2 ± 0.41**	94.97 ± 0.64	**95.77 ± 0.49**	95.42 ± 0.78	95.06 ± 0.19
Banknote	**99.9 ± 0.04**	98.31 ± 0.51	98.94 ± 1.4	97.24 ± 3.56	98.94 ± 0.36
Biodeg	**92.81 ± 0.32**	87.18 ± 0.61	87.59 ± 1.9	84.05 ± 3.03	84.05 ± 3.03
Breast cancer	**99.09 ± 0.28**	**99.39 ± 0.15**	**99.06 ± 0.25**	**99.2 ± 0.37**	97.99 ± 0.94
Car	97.76 ± 0.53	95.24 ± 0.88	97.8 ± 0.52	**98.43 ± 0.42**	**98.5 ± 0.34**
Credit	**86.1 ± 0.43**	84.68 ± 0.32	84.14 ± 0.43	83.67 ± 1.98	84.71 ± 1.22
Cryotherapy	**99.35 ± 0.89**	98.77 ± 0.67	98.95 ± 1.89	**99.37 ± 0.75**	**99.58 ± 0.55**
Divorce	**98.69 ± 0.68**	**98.77 ± 0.59**	96.86 ± 2.57	**97.96 ± 2.03**	**97.96 ± 2.03**
Ecoli	**97.09 ± 0.21**	95.26 ± 0.49	96.5 ± 0.41	95.91 ± 1.04	95.91 ± 1.04
Forest	**96.43 ± 0.33**	95.66 ± 0.53	87.33 ± 3.91	91.51 ± 2.79	91.51 ± 2.79
German	**82.64 ± 1.84**	80.63 ± 1.37	79.27 ± 0.94	79.63 ± 2.1	79.63 ± 2.1
Glass	**94.56 ± 0.65**	90.57 ± 1.04	87.31 ± 3.35	88.34 ± 1.57	88.34 ± 1.57
Haberman	**80.67 ± 3.87**	**80.41 ± 1.78**	77.59 ± 3.69	79.12 ± 3.8	79.08 ± 3.81
Internet	**100.0 ± 0.0**	99.66 ± 0.41	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**
Iris	**99.82 ± 0.37**	**99.99 ± 0.02**	**99.87 ± 0.2**	**99.86 ± 0.18**	99.64 ± 0.87
Kohkiloyeh	91.33 ± 3.6	87.63 ± 4.43	91.5 ± 3.4	**94.78 ± 1.24**	**93.89 ± 2.65**
Liver	**78.78 ± 0.42**	77.07 ± 0.83	76.75 ± 1.44	76.95 ± 1.63	77.0 ± 1.21
Magic	**89.94 ± 1.04**	85.06 ± 0.89	87.84 ± 1.34	86.52 ± 2.58	86.52 ± 2.58
Mamographic	**89.52 ± 0.23**	86.41 ± 0.39	88.06 ± 2.17	89.27 ± 0.37	89.11 ± 0.6
Nurse	84.12 ± 2.34	83.33 ± 1.18	84.58 ± 1.65	83.89 ± 2.83	**86.08 ± 0.99**
Occupancy	99.53 ± 0.07	99.02 ± 0.2	99.53 ± 0.29	**99.64 ± 0.11**	98.8 ± 0.18
Pima	**87.73 ± 2.78**	84.72 ± 2.98	86.58 ± 2.81	83.81 ± 3.05	84.95 ± 3.36
Seismic	**97.33 ± 0.04**	**97.34 ± 0.09**	97.25 ± 0.16	97.04 ± 0.22	96.72 ± 0.33
Spambase	**96.9 ± 0.2**	83.43 ± 0.78	89.72 ± 1.89	91.34 ± 1.77	91.34 ± 1.77
Tic-tac-toe	90.45 ± 0.94	81.47 ± 1.54	86.24 ± 1.3	92.6 ± 2.99	94.01 ± 1.6
Vegas	**80.81 ± 1.53**	80.87 ± 0.38	79.86 ± 0.27	77.61 ± 0.95	77.59 ± 0.94
Winery	**99.94 ± 0.06**	**99.94 ± 0.06**	93.29 ± 3.14	97.45 ± 2.64	97.68 ± 2.57
Zoo	**100.0 ± 0.0**	99.95 ± 0.03	**100.0 ± 0.01**	**100.0 ± 0.0**	**100.0 ± 0.01**
Average	93.33 ± 6.6	91.88 ± 7.8	91.28 ± 8.1	92.27 ± 7.8	92.48 ± 7.6

Marked with bold - The best score and its equivalent scores ($p\_value \geq 0.05$) * Cases where optimized DT AUC is equivalent to the best FBT model ($p\_value \geq 0.05$) ** Cases where optimized DT AUC is higgher than the AUC of the best FBT model ($p\_value \leq 0.05$)

accuracy ($F_f = 88.56, p \approx 0$) and average prediction depth per sample ($F_f = 198.96, p \approx 0$). We proceeded with a post-hoc Nemenyi test of the AUC, the generalized accuracy and the average prediction depth for the different models, depicted in Table 10, Tables 11 and 12 respectively. The datasets that did not have results for ISM and GENESIM were excluded when conducting the adjusted Friedman tests, as well as the post-hoc Nemenyi tests. As a result, these tests were applied using 28 out of the 33 datasets.

The results obtained shows that in 12 out of the 30 datasets the forest based tree, derived from random forest (RF-FBT) made a pareto improvement [75] when compared to random forest. Specifically, it decreased the number of applied test rules without impairing the AUC. In three of the cases, RF-FBT improved random forest's AUC. Compared to the single optimized decision tree (DT), the new model had a superior AUC for 20 out of the 30 datasets while the decision tree used less test rules when inferencing new samples.

4.3. Discussion of results

The results presented in Table 10 implies that decision forests did not significantly outperform forest based trees in terms of ROC AUC. This evidence is valid for both the comparison of extra-trees based tree (ET-FBT) with extra-trees (ET) and for comparing random forest based tree (RF-FBT) with random forest (RF). On the other hand, the ROC AUC of RF was significantly superior to the ROC AUC values of all of the other decision tree models that were used as benchmarks. Fig. 6 depicts the average ranks of the different models in terms of ROC AUC using a critical difference diagram. A line is drawn between two algorithms that were not significantly differed according to Wilcoxon-Holm test. It illustrates the AUC closeness of RF to FBT, but also the insignificant difference between FBT and a regular decision tree (DT). However, the augmented forest-based tree (AFBT) significantly outperformed DT and may be considered as the best practice when a decision tree is required for a certain classification task. AFBT can be viewed as a rule of thumb for selecting the best model among FBT and DT. More specifically, whenever the AUC of a regular decision tree is lower than 0.98 of random forest's AUC, we assign FBT value to AFBT. Otherwise – DT value is assigned to AFBT. This approach stems from the fact that FBT seems to be more effective for cases where there is a large gap between the performance of decision forests and the performance of regular decision trees. In other cases, where the difference is small, constructing a more complex tree would not necessarily improve the predictive performance and may even impair it in some cases. Fig. 9 presents an empirical evidence that supports this intuition, using the datasets that were included in the experiment. This box-plot shows that whenever there is no large gap between random forest and a regular decision tree, a forest-based tree is not expected to be more accurate than the regular decision tree. It is to be noted that in two out of the 33 datasets, GENESIM and ISM achieved the highest performance in terms of AUC.

Fig. 7 illustrates the average accuracy ranks of the different models. Similarly to the ROC AUC, Random forest (RF) and the augmented forest based tree (AFBT) have achieved the best performance. However, the difference between these models to some of the additional benchmarks is not significant. In addition, there are shifts in average ranks for algorithms like extra-trees (ET), extra-tress based tree (ET-FBT) and random -forest based tree (RF-FBT) while other algorithms are ranked

Table 5
AUC comparison – Decision trees and rule classifier benchmarks.

Dataset	DT	CN2	CMM	ISM	GENESIM
Abalone	90.14 ± 0.49	58.06 ± 0.89	63.48 ± 1.0	*NA*	*NA*
Acute-inflam	97.92 ± 2.2	90.72 ± 10.47	**100.0 ± 0.0**	90.42 ± 5.74	**100.0 ± 0.0**
Acute-nephitis	**100.0 ± 0.0**	97.11 ± 6.68	**100.0 ± 0.0**	93.23 ± 4.05	97.46 ± 2.74
Aust_credit	89.91 ± 2.37	75.53 ± 3.71	85.99 ± 2.77	88.6 ± 2.92	**90.14 ± 2.85**
Balance_scale	78.82 ± 2.91	88.16 ± 1.82	89.4 ± 2.27	83.71 ± 3.32	87.66 ± 2.66
Bank	95.06 ± 0.19	92.35 ± 0.79	50.0 ± 0.0	*NA*	*NA*
Banknote	98.94 ± 0.36	94.84 ± 2.19	98.69 ± 0.31	98.66 ± 0.49	98.72 ± 0.91
Biodeg	85.69 ± 1.59	82.81 ± 2.85	77.49 ± 3.03	*NA*	*NA*
Breast cancer	97.17 ± 0.82	94.58 ± 2.3	95.86 ± 0.92	95.12 ± 1.58	96.54 ± 1.67
Car	**98.5 ± 0.34**	97.56 ± 0.55	89.6 ± 1.54	**98.3 ± 0.88**	**98.38 ± 0.86**
Credit	84.4 ± 0.96	65.36 ± 4.63	83.39 ± 1.12	78.49 ± 4.54	79.97 ± 4.66
Cryotherapy	80.93 ± 8.14	77.47 ± 8.28	97.01 ± 5.29	87.09 ± 6.82	88.08 ± 5.04
Divorce	93.19 ± 2.74	95.37 ± 2.37	93.53 ± 8.51	**97.35 ± 1.8**	**97.2 ± 2.0**
Ecoli	92.69 ± 2.22	87.95 ± 2.63	91.15 ± 1.68	89.58 ± 3.97	92.41 ± 5.28
Forest	93.24 ± 1.06	88.74 ± 2.07	80.02 ± 9.73	91.93 ± 1.51	93.31 ± 1.38
German	76.51 ± 2.32	73.59 ± 1.73	74.35 ± 2.35	74.99 ± 3.99	76.01 ± 2.03
Glass	82.06 ± 3.0	77.32 ± 3.29	83.0 ± 4.62	84.97 ± 4.24	87.14 ± 3.94
Haberman	72.01 ± 5.32	71.11 ± 4.13	73.9 ± 2.41	73.7 ± 8.13	76.62 ± 5.13
Internet	**100.0 ± 0.0**	50.0 ± 0.0	99.54 ± 1.46	95.37 ± 2.43	99.99 ± 0.02
Iris	97.72 ± 1.43	91.74 ± 3.28	98.0 ± 2.29	96.87 ± 2.36	96.55 ± 2.33
Kohkiloyeh	88.14 ± 2.76	57.43 ± 15.54	77.64 ± 11.25	80.02 ± 7.65	71.72 ± 10.47
Liver	75.21 ± 2.11	75.1 ± 2.18	65.2 ± 8.67	73.75 ± 4.98	75.21 ± 2.97
Magic	85.78 ± 0.82	79.63 ± 2.69	82.14 ± 1.55	*NA*	*NA*
Mamographic	87.81 ± 0.2	73.33 ± 4.18	82.64 ± 1.23	85.21 ± 3.76	87.49 ± 2.43
Nurse	**86.08 ± 0.99**	69.16 ± 5.54	76.65 ± 3.57	80.83 ± 6.06	80.7 ± 6.65
Occupancy	98.8 ± 0.18	98.53 ± 0.48	97.82 ± 0.1	99.33 ± 0.21	99.3 ± 0.28
Pima	85.79 ± 3.11	74.54 ± 3.46	81.34 ± 3.12	72.44 ± 3.07	78.51 ± 2.61
Seismic	96.72 ± 0.33	94.86 ± 0.45	50.0 ± 0.0	93.36 ± 0.73	95.86 ± 0.55
Spambase	93.82 ± 0.96	91.99 ± 1.08	69.17 ± 15.33	*NA*	*NA*
Tic-tac-toe	94.01 ± 1.6	74.46 ± 2.71	77.29 ± 2.53	**97.01 ± 1.33**	**96.74 ± 1.45**
Vegas	75.39 ± 1.22	61.43 ± 1.89	66.51 ± 7.06	72.5 ± 2.8	75.39 ± 3.03
Winery	93.73 ± 1.95	88.3 ± 3.59	99.05 ± 1.49	94.41 ± 2.47	95.37 ± 1.94
Zoo	**100.0 ± 0.00**	90.26 ± 4.74	99.44 ± 1.19	98.29 ± 1.63	97.68 ± 1.63
Average	89.85 ± 8.8	81.23 ± 13.3	85.96 ± 12.77	88.05 ± 9.1	89.64 ± 9.21

Marked with bold - The best score and its equvalent scores ($p\_value \geq 0.05$) NA is assigned when the given algorithm did not manage to train a classification model

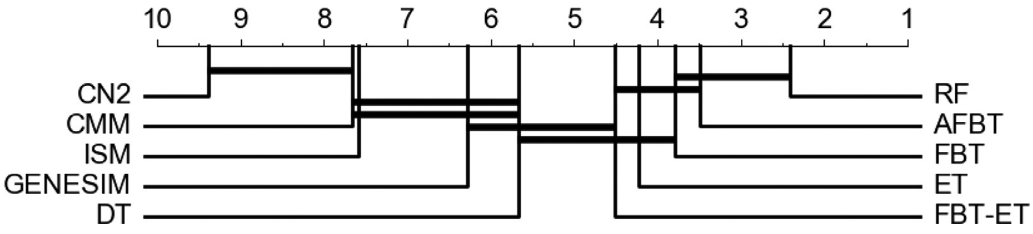


Fig. 6. Critical difference diagram of ROC AUC based on Wilcoxon–Holm post-hoc test.

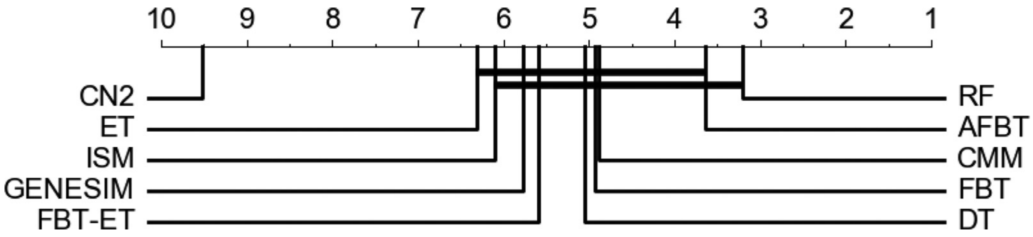


Fig. 7. Critical difference diagram of generalized accuracy based on Wilcoxon–Holm post-hoc test.

Table 6
Generallized accuracy - Decision forests and forest based trees.

Dataset	RF	ET	ET-FBT	RF-FBT	AFBT
Abalone	26.48 ± 0.64	24.95 ± 0.78	26.33 ± 0.74	24.2 ± 1.95	**27.73 ± 1.06**
Acute-inflam	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**
Acute-nephitis	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**
Aust_credit	83.12 ± 2.51	83.26 ± 2.03	83.84 ± 2.05	83.62 ± 2.54	83.48 ± 1.96
Balance_scale	**82.32 ± 3.03**	81.92 ± 3.42	63.04 ± 2.58	**82.96 ± 2.1**	**82.56 ± 2.09**
Bank	88.74 ± 0.68	88.91 ± 0.53	88.98 ± 0.58	88.92 ± 0.65	**89.8 ± 0.31**
Banknote	**98.84 ± 0.23**	92.25 ± 1.81	96.62 ± 3.3	92.95 ± 8.19	**98.76 ± 0.31**
Biodeg	**83.22 ± 1.05**	77.87 ± 0.92	79.72 ± 3.28	75.69 ± 3.06	75.97 ± 3.55
Breast cancer	**95.47 ± 0.9**	**95.47 ± 0.75**	94.74 ± 0.67	**95.55 ± 1.31**	**95.47 ± 1.23**
Car	85.23 ± 1.82	72.63 ± 2.06	87.23 ± 1.87	89.1 ± 1.25	**97.75 ± 0.51**
Credit	81.63 ± 1.04	80.31 ± 0.91	**82.95 ± 0.1**	80.16 ± 5.48	**82.95 ± 0.0**
Cryotherapy	96.11 ± 2.68	91.11 ± 3.88	95.56 ± 5.74	**97.22 ± 3.93**	**97.78 ± 3.88**
Divorce	92.35 ± 2.48	91.76 ± 1.86	93.82 ± 2.17	92.35 ± 2.84	**93.24 ± 2.42**
Ecoli	**84.56 ± 1.43**	71.47 ± 1.99	79.41 ± 4.27	81.76 ± 2.42	81.76 ± 2.42
Forest	**87.24 ± 1.97**	82.67 ± 1.25	60.1 ± 9.73	75.24 ± 3.54	79.9 ± 6.98
German	**72.6 ± 1.29**	70.7 ± 1.53	**71.7 ± 1.58**	**71.8 ± 1.57**	**72.25 ± 2.52**
Glass	**73.26 ± 3.68**	61.16 ± 5.81	52.09 ± 8.22	54.65 ± 7.54	54.65 ± 7.54
Haberman	**74.35 ± 1.78**	69.84 ± 1.53	69.84 ± 1.53	71.77 ± 3.06	71.94 ± 3.06
Internet	**100.0 ± 0.0**	94.62 ± 2.43	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**
Iris	98.33 ± 2.83	**99.33 ± 1.41**	98.0 ± 2.33	97.33 ± 2.63	97.67 ± 2.74
Kohkiloyeh	**82.5 ± 7.17**	78.0 ± 6.75	**85.5 ± 5.5**	**83.5 ± 4.74**	**82.5 ± 4.25**
Liver	**68.1 ± 1.15**	67.59 ± 1.09	67.59 ± 1.09	67.59 ± 2.61	**68.1 ± 0.41**
Magic	**82.2 ± 1.38**	74.05 ± 0.71	79.85 ± 1.87	80.3 ± 1.99	80.6 ± 1.11
Mamographic	82.65 ± 1.23	80.84 ± 0.84	79.28 ± 4.97	82.17 ± 0.86	**83.37 ± 0.76**
Nurse	**67.78 ± 2.34**	66.67 ± 0.0	**67.78 ± 3.51**	62.22 ± 6.31	**67.78 ± 2.34**
Occupancy	97.79 ± 0.08	94.48 ± 1.35	97.86 ± 0.1	97.8 ± 0.13	**98.8 ± 0.18**
Pima	**79.81 ± 3.67**	73.38 ± 3.06	78.77 ± 2.25	76.04 ± 3.3	78.18 ± 3.53
Seismic	**93.97 ± 0.33**	**93.89 ± 0.24**	**93.89 ± 0.24**	**93.97 ± 0.33**	93.58 ± 0.33
Spambase	91.13 ± 0.56	68.33 ± 1.83	80.25 ± 3.44	81.25 ± 2.88	**91.89 ± 0.62**
Tic-tac-toe	78.7 ± 2.12	73.07 ± 2.91	75.73 ± 2.53	84.27 ± 3.31	94.01 ± 1.6
Vegas	**52.67 ± 5.17**	51.39 ± 1.77	50.69 ± 3.19	45.45 ± 2.0	45.25 ± 1.87
Winery	97.5 ± 1.58	**98.06 ± 1.34**	78.33 ± 7.83	88.33 ± 9.51	88.89 ± 9.89
Zoo	**100.0 ± 0.0**	97.14 ± 2.46	99.52 ± 1.51	**100.0 ± 0.0**	**100.0 ± 0.0**
Average	85.96 ± 12.3	82.60 ± 13.3	82.28 ± 15.1	83.85 ± 14.3	85.37 ± 14.4

Marked with bold - The best score and its equivalent scores (*p_value* ≥ 0.05) NA is assigned when the given algorithm did not manage to train a classification model

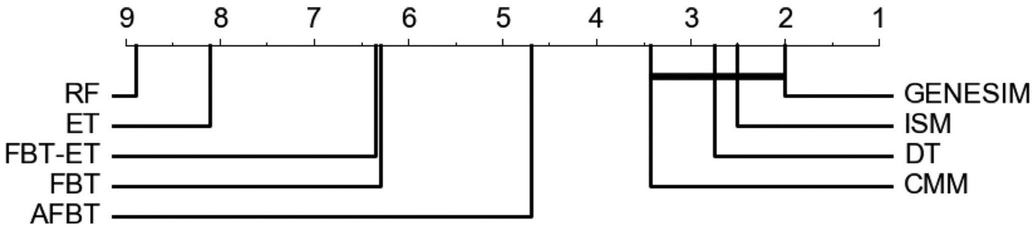


Fig. 8. Critical difference diagram of prediction average depth based on Wilcoxon–Holm post-hoc test.

higher. The gap between AUC and generalized accuracy ranks can be explained through the example of CMM that have a poor AUC performance for label imbalance classification problems. The label imbalance is aggravated following the generation of CMM's synthetic data; in some cases it may lead to a situation where a decision tree cannot be trained (For example: Siesmic and Bank datasets). Fig. 10 depicts the average AUC rank as a function of class impurity. It can be seen that like random forest and CARET decision trees, the forest based tree is likely to have a higher rank for label imbalance classification problems, as opposed to existing benchmarks like CMM, ISM and GENESIM that are likely to be ranked lower for challenges as such.

Fig. 8 depicts the average ranks of the complexity metric, measured as the average amount of rules that were applied in order to predict a test instance. It is apparent that there are significant gaps between the different models. Decision forest models are significantly more complex than the different variations of the forest-based tree while the rest of the benchmarks are significantly more simpler. According to the obtained results, and as can be seen in Table 8, the amount of rules required to be tested by RF-FBT to predict a new instance is 3% of the amount of rules that are tested by RF (on average). GENESIM, ISM and CMM on the other hand, construct decision trees that are very parsimonious and simple but that are also rarely provide an improved predictive performance in relation to regular decision trees. Decision paths provided by FBT are usually more complex than paths given by simple decision trees. Improving the intelligibility in a future refinement can be done by limiting the maximum depth of the constructed trees and evaluate the effect of such a limitation on the predictive performance. However, it should be considered that in some cases, deep decision trees have the ability to identify novel patterns that cannot be discovered by simple classification trees. Therefore, deep decision trees might be preferred over simple trees for knowledge discovery in expert systems regardless their overall predictive performance [19].

Table 7
Generallized accuracy – Decision trees and rule classifier benchmarks.

Dataset	DT	CN2	CMM	ISM	GENESIM
Abalone	**27.73 ± 1.06**	17.41 ± 0.75	26.7 ± 0.86	*NA*	*NA*
Acute-inflam	97.92 ± 2.2	88.69 ± 9.41	**100.0 ± 0.0**	84.72 ± 5.89	**100.0 ± 0.0**
Acute-nephitis	**100.0 ± 0.0**	96.19 ± 7.11	100.0 ± 0.0	89.44 ± 5.2	96.94 ± 3.57
Aust_credit	82.75 ± 0.46	68.22 ± 3.63	81.67 ± 3.79	**85.29 ± 2.48**	**84.48 ± 2.78**
Balance_scale	71.76 ± 3.88	74.14 ± 2.41	**82.32 ± 3.03**	77.47 ± 3.53	77.35 ± 3.22
Bank	**89.8 ± 0.31**	85.08 ± 0.77	88.74 ± 0.68	*nan ± nan*	*nan ± nan*
Banknote	**98.76 ± 0.31**	92.23 ± 2.16	**98.69 ± 0.31**	97.86 ± 0.65	98.21 ± 0.95
Biodeg	78.91 ± 1.58	75.75 ± 2.62	69.76 ± 1.65	*NA*	*NA*
Breast cancer	94.31 ± 1.41	92.65 ± 1.77	**95.69 ± 1.0**	94.01 ± 1.42	94.8 ± 2.18
Car	**97.75 ± 0.51**	87.66 ± 1.86	84.83 ± 1.79	95.59 ± 1.81	94.27 ± 1.79
Credit	81.71 ± 2.61	60.85 ± 3.18	**82.87 ± 0.25**	77.25 ± 3.48	77.3 ± 3.53
Cryotherapy	81.67 ± 8.71	75.91 ± 6.25	95.56 ± 5.74	86.11 ± 5.72	83.02 ± 4.86
Divorce	92.06 ± 3.41	92.04 ± 2.62	88.82 ± 14.79	**96.08 ± 2.07**	**95.88 ± 2.35**
Ecoli	75.74 ± 2.33	70.27 ± 3.77	82.94 ± 2.1	81.49 ± 2.99	80.2 ± 4.04
Forest	86.38 ± 1.27	76.87 ± 3.74	53.81 ± 15.85	85.29 ± 2.37	85.52 ± 2.76
German	**71.5 ± 2.8**	66.95 ± 1.66	**71.8 ± 2.1**	**71.38 ± 1.99**	69.29 ± 2.54
Glass	64.42 ± 4.91	53.6 ± 4.98	60.23 ± 7.39	70.62 ± 5.6	66.25 ± 7.07
Haberman	69.19 ± 2.79	65.02 ± 4.25	**73.55 ± 2.31**	70.75 ± 5.36	**72.53 ± 4.66**
Internet	100.0 ± 0.0	69.73 ± 1.95	99.54 ± 1.46	92.78 ± 2.6	99.91 ± 0.3
Iris	96.33 ± 2.92	84.7 ± 5.95	97.0 ± 2.92	94.17 ± 3.39	92.59 ± 2.74
Kohkiloyeh	**81.5 ± 4.74**	55.68 ± 14.01	74.5 ± 10.39	76.97 ± 7.52	69.09 ± 8.96
Liver	**68.19 ± 0.27**	65.54 ± 2.71	67.93 ± 1.45	67.45 ± 4.18	67.71 ± 3.11
Magic	80.48 ± 0.58	72.35 ± 2.5	81.03 ± 1.57	80.17 ± *nan*	80.83 ± *nan*
Mamographic	**83.37 ± 0.76**	63.64 ± 4.71	82.65 ± 1.23	82.11 ± 3.06	82.69 ± 2.55
Nurse	**67.78 ± 2.34**	50.41 ± 6.49	67.78 ± 2.34	**70.0 ± 10.25**	66.67 ± 10.48
Occupancy	**98.8 ± 0.18**	94.11 ± 1.34	97.79 ± 0.08	98.53 ± 0.44	98.35 ± 0.37
Pima	78.83 ± 3.56	67.13 ± 3.18	80.32 ± 2.98	71.9 ± 1.91	71.62 ± 1.95
Seismic	93.58 ± 0.33	90.17 ± 1.02	**93.97 ± 0.33**	93.36 ± 0.73	93.17 ± 0.82
Spambase	**91.89 ± 0.62**	83.0 ± 1.5	61.97 ± 19.1	*NA*	*NA*
Tic-tac-toe	94.01 ± 1.6	69.55 ± 2.9	75.99 ± 2.3	**94.67 ± 2.23**	91.11 ± 1.89
Vegas	43.86 ± 1.81	35.0 ± 2.87	47.92 ± 4.74	42.78 ± 3.6	44.44 ± 5.18
Winery	91.67 ± 2.62	78.12 ± 5.1	97.22 ± 2.27	89.43 ± 2.7	90.57 ± 4.08
Zoo	99.52 ± 1.51	76.83 ± 7.84	98.57 ± 3.21	93.03 ± 4.07	88.79 ± 6.01
Average	84.41 ± 14.0	73.64 ± 15.1	83.35 ± 14.7	83.23 ± 12.6	83.31 ± 13.6

Marked with bold - The best score and its equivalent scores ($p\_value \geq 0.05$)
NA is assigned when the given algorithm did not manage to train a classification model

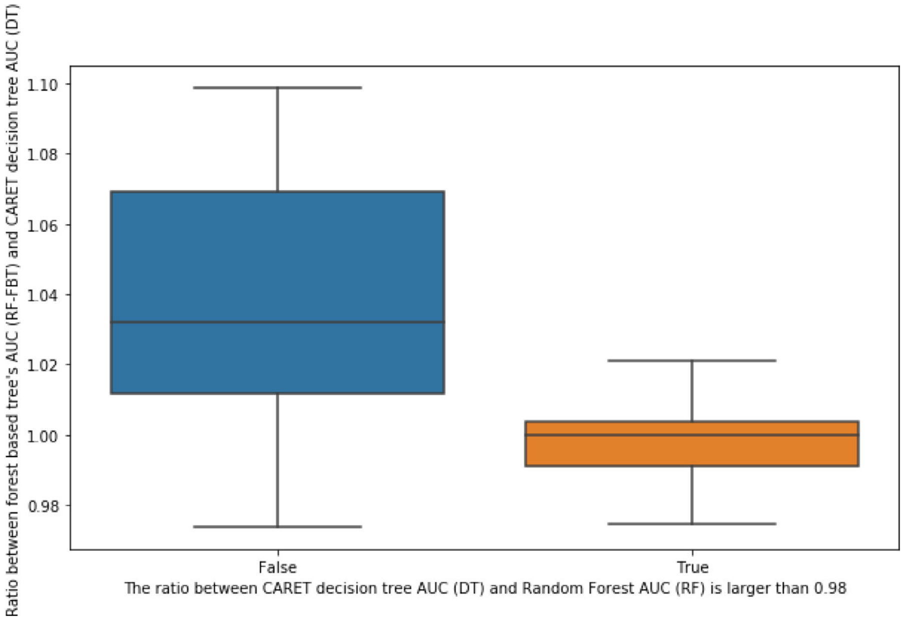


Fig. 9. A box-plot that illustrates the ROC AUC ratio of CARET decision tree and random forest based tree as a function of the AUC ratio between CARET decision tree and random forest.

Table 8
Number of rules per sample comparison – Decision forests and forest based trees.

Dataset	RF	ET	ET-FBT	RF-FBT	AFBT
Abalone	469.93 ± 0.44	210.88 ± 11.52	14.92 ± 0.64	13.15 ± 0.71	3.0 ± 0.0
Acute-inflam	258.15 ± 3.06	242.22 ± 2.31	8.1 ± 0.78	8.2 ± 0.77	5.1 ± 3.25
Acute-nephitis	197.88 ± 2.9	225.87 ± 3.58	8.15 ± 0.5	7.47 ± 0.48	1.53 ± 0.03
Aust_credit	452.14 ± 1.85	277.51 ± 5.64	13.08 ± 1.03	12.84 ± 0.4	3.95 ± 3.0
Balance_scale	446.89 ± 1.73	299.39 ± 0.21	5.36 ± 0.4	10.45 ± 0.13	10.45 ± 0.13
Bank	453.03 ± 2.02	295.01 ± 0.58	13.84 ± 1.04	14.0 ± 1.22	3.0 ± 0.0
Banknote	400.04 ± 2.23	273.71 ± 7.01	13.77 ± 0.96	10.82 ± 2.6	4.32 ± 0.37
Biodeg	456.68 ± 1.22	283.84 ± 5.22	13.27 ± 1.24	10.31 ± 0.45	10.31 ± 0.45
Breast cancer	351.57 ± 5.64	284.32 ± 2.87	13.29 ± 1.8	12.69 ± 0.92	4.86 ± 3.93
Car	363.11 ± 6.96	263.7 ± 3.19	9.91 ± 0.58	11.09 ± 0.71	4.27 ± 0.19
Credit	454.28 ± 2.39	296.1 ± 0.58	13.26 ± 1.65	12.59 ± 1.19	9.81 ± 4.84
Cryotherapy	313.0 ± 11.17	284.76 ± 1.57	12.56 ± 0.79	12.49 ± 0.81	11.56 ± 3.29
Divorce	162.14 ± 12.62	219.15 ± 3.61	11.45 ± 0.96	12.03 ± 0.54	12.03 ± 0.54
Ecoli	415.13 ± 6.14	245.37 ± 8.5	14.49 ± 0.62	14.48 ± 0.72	14.48 ± 0.72
Forest	440.8 ± 3.96	289.57 ± 2.99	13.02 ± 0.98	11.75 ± 1.03	11.75 ± 1.03
German	460.31 ± 1.78	298.1 ± 0.28	12.96 ± 0.72	12.59 ± 0.77	12.59 ± 0.77
Glass	432.84 ± 3.65	290.26 ± 3.92	12.94 ± 0.89	12.62 ± 0.61	12.62 ± 0.61
Haberman	465.16 ± 1.38	218.56 ± 7.24	11.96 ± 1.01	13.68 ± 0.44	12.64 ± 3.42
Internet	324.78 ± 5.56	267.87 ± 2.89	8.89 ± 0.67	9.3 ± 0.64	2.89 ± 0.06
Iris	255.01 ± 10.23	257.84 ± 3.87	13.62 ± 0.82	10.16 ± 1.12	7.75 ± 3.78
Kohkiloyeh	422.42 ± 13.48	295.98 ± 0.82	10.12 ± 0.52	10.64 ± 0.5	9.3 ± 2.99
Liver	447.77 ± 2.03	259.41 ± 5.55	15.29 ± 1.27	13.23 ± 0.84	11.22 ± 4.4
Magic	464.79 ± 1.63	261.08 ± 5.53	14.44 ± 1.12	14.7 ± 1.68	14.7 ± 1.68
Mamographic	408.61 ± 3.28	233.86 ± 8.65	9.1 ± 0.79	11.8 ± 0.67	10.01 ± 3.76
Nurse	444.67 ± 4.37	298.64 ± 1.29	13.31 ± 0.86	12.04 ± 0.9	2.92 ± 0.18
Occupancy	297.11 ± 8.5	264.19 ± 6.02	13.05 ± 1.26	11.29 ± 0.97	2.48 ± 0.09
Pima	458.42 ± 2.51	237.06 ± 6.21	14.1 ± 0.39	14.0 ± 0.93	9.69 ± 5.78
Seismic	441.86 ± 3.37	252.25 ± 6.19	13.08 ± 1.11	14.08 ± 0.61	3.0 ± 0.0
Spambase	451.48 ± 1.45	175.19 ± 8.8	18.27 ± 1.28	15.6 ± 0.7	15.6 ± 0.7
Tic-tac-toe	450.26 ± 1.37	299.99 ± 0.02	12.25 ± 0.83	12.03 ± 0.67	4.84 ± 0.06
Vegas	456.27 ± 3.01	294.19 ± 0.82	14.61 ± 0.56	12.11 ± 1.28	11.33 ± 2.29
Winery	316.26 ± 4.99	287.49 ± 2.34	13.16 ± 0.7	13.49 ± 0.73	12.37 ± 3.35
Zoo	335.98 ± 8.26	264.0 ± 2.67	12.36 ± 0.86	12.47 ± 0.6	2.89 ± 0.2
Average	393.0 ± 84.4	265.1 ± 31.2	12.5 ± 2.5	12.1 ± 1.8	8.2 ± 4.3

Marked with bold - The best score and its equivalent scores ($p\_value \geq 0.05$) *

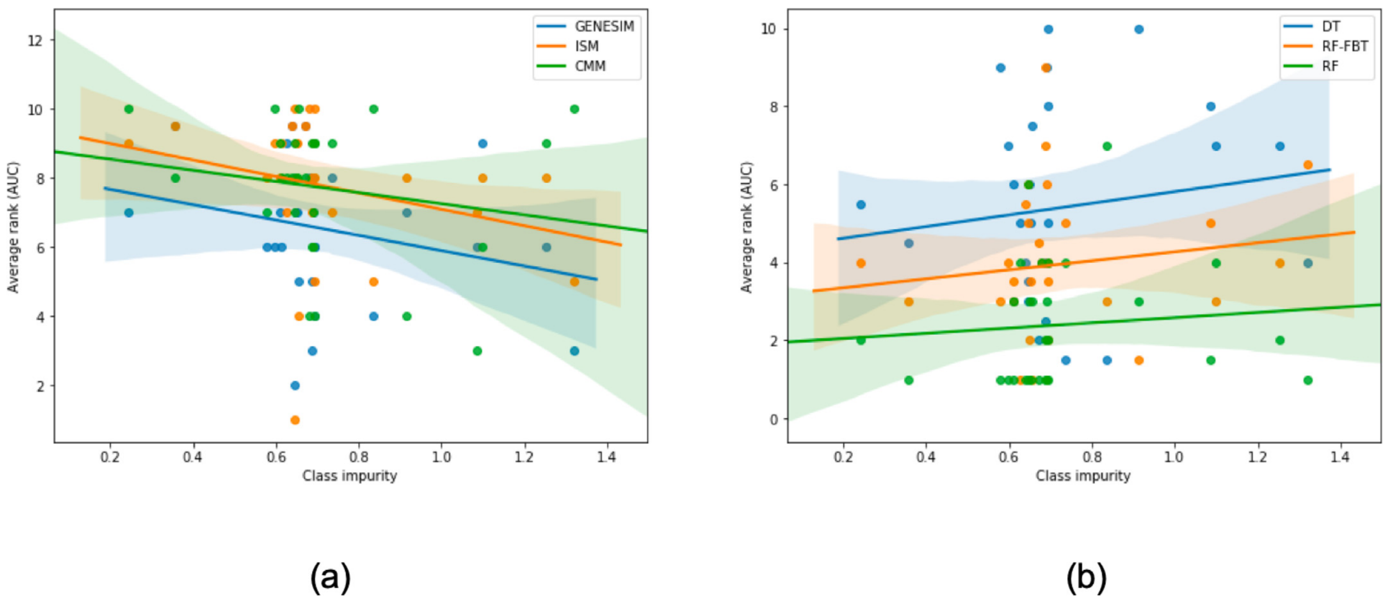


Fig. 10. The ROC AUC ranks of as a function of dataset's class impurity (calculated as entropy).

Table 9
Number of rules per sample comparison – Decision trees and rule classifier benchmarks.

Dataset	DT	CMM	ISM	GENESIM
Abalone	**3.0 ± 0.0**	11.05 ± 0.26	*NA*	*NA*
Acute-inflam	2.04 ± 0.0	2.66 ± 0.25	**1.25 ± 0.28**	1.97 ± 0.13
Acute-nephitis	1.53 ± 0.03	2.92 ± 0.14	**1.0 ± 0.0**	1.42 ± 0.24
Aust_credit	3.36 ± 1.14	4.01 ± 0.76	**2.86 ± 0.8**	4.43 ± 1.24
Balance_scale	5.96 ± 0.06	5.26 ± 0.21	**5.14 ± 0.54**	5.56 ± 0.86
Bank	3.0 ± 0.0	*NA*	*NA*	*NA*
Banknote	4.32 ± 0.37	4.12 ± 0.33	6.72 ± 0.64	**3.88 ± 0.99**
Biodeg	7.02 ± 0.27	**3.27 ± 1.39**	*NA*	*NA*
Breast cancer	**2.97 ± 0.06**	4.65 ± 0.48	3.13 ± 0.52	**2.84 ± 0.67**
Car	**4.27 ± 0.19**	4.77 ± 0.29	4.44 ± 0.29	**4.26 ± 0.19**
Credit	**2.98 ± 0.01**	**2.95 ± 0.74**	**2.91 ± 0.79**	**2.28 ± 2.01**
Cryotherapy	3.2 ± 0.54	4.32 ± 0.38	**2.26 ± 0.74**	**2.22 ± 0.65**
Divorce	1.64 ± 0.18	4.26 ± 0.76	**1.0 ± 0.0**	1.04 ± 0.14
Ecoli	**3.41 ± 0.95**	5.53 ± 0.3	5.54 ± 1.04	3.96 ± 0.77
Forest	**3.76 ± 1.59**	5.03 ± 0.31	5.23 ± 1.04	4.38 ± 1.06
German	6.54 ± 1.38	**4.12 ± 1.02**	4.55 ± 1.35	6.04 ± 1.27
Glass	5.57 ± 0.15	5.77 ± 0.29	7.45 ± 1.42	**4.96 ± 0.66**
Haberman	6.22 ± 2.23	4.63 ± 0.42	**2.66 ± 0.97**	3.34 ± 1.39
Internet	2.89 ± 0.06	4.6 ± 0.38	**2.25 ± 0.36**	3.02 ± 0.1
Iris	2.39 ± 0.09	4.08 ± 0.53	2.03 ± 0.32	**1.79 ± 0.22**
Kohkiloyeh	3.72 ± 1.11	5.84 ± 0.38	3.92 ± 0.93	**3.06 ± 1.46**
Liver	5.53 ± 1.33	**0.89 ± 0.49**	3.27 ± 0.87	4.15 ± 2.0
Magic	6.94 ± 0.02	7.06 ± 0.91	*NA*	*NA*
Mamographic	2.99 ± 0.0	**2.88 ± 0.49**	3.14 ± 1.2	4.28 ± 1.02
Nurse	2.92 ± 0.18	3.86 ± 0.85	2.41 ± 1.15	**1.53 ± 1.02**
Occupancy	2.48 ± 0.09	2.83 ± 0.58	2.45 ± 0.56	**2.28 ± 0.41**
Pima	**3.0 ± 0.0**	5.85 ± 0.32	5.62 ± 0.93	4.53 ± 1.61
Seismic	3.0 ± 0.0	**0.0 ± 0.0**	**0.0 ± 0.0**	2.5 ± 1.87
Spambase	8.14 ± 0.54	**3.79 ± 0.88**	*NA*	*NA*
Tic-tac-toe	4.84 ± 0.06	5.03 ± 0.15	5.27 ± 0.48	**4.68 ± 0.15**
Vegas	**3.51 ± 1.19**	6.56 ± 1.15	5.15 ± 2.1	4.6 ± 1.69
Winery	2.73 ± 0.1	6.01 ± 0.58	3.57 ± 0.65	**2.28 ± 0.47**
Zoo	2.89 ± 0.2	5.69 ± 0.37	2.68 ± 0.21	**2.55 ± 0.18**
Average	3.9 ± 1.7	4.4 ± 2.1	3.7 ± 2.1	3.5 ± 1.6

Marked with bold - The best score and its equivalent scores ($p\_value \geq 0.05$) NA is assigned when the given algorithm did not manage to train a classification model

Table 10
Nemenyi test for AUC comparison.

	DT	RF	FBT	CMM	AFBT	ET	ET_FBT	CN2	ISM
RF	0.00	–	–	–	–	–	–	–	–
FBT	0.26	0.75	–	–	–	–	–	–	–
CMM	0.26	0	0	–	–	–	–	–	–
AFBT	0.09	0.94	1	0	–	–	–	–	–
ET	0.68	0.32	1	0	0.99	–	–	–	–
ET_FBT	0.85	0.18	1	0	0.94	1	–	–	–
CN2	0.00	0	0	0.49	0	0	0	–	–
ISM	1.00	0	0.02	0.82	0.01	0.15	0.28	0.01	–
GENESIM	1.00	0	0.02	0.82	0.01	0.15	0.28	0.01	1

Table 11
Nemenyi test for generalized accuracy comparison.

	DT	RF	FBT	CMM	AFBT	ET	ET_FBT	CN2	ISM
RF	0.33	–	–	–	–	–	–	–	–
FBT	1.00	0.39	–	–	–	–	–	–	–
CMM	1.00	0.52	1	–	–	–	–	–	–
AFBT	0.71	1	0.76	0.87	–	–	–	–	–
ET	0.87	0	0.82	0.71	0.02	–	–	–	–
ET_FBT	1.00	0.07	1	0.99	0.26	1	–	–	–
CN2	0.00	0	0	0	0	0	0	–	–
ISM	1.00	0.05	1	0.99	0.22	1	1	0	–
GENESIM	1.00	0.05	1	0.99	0.22	1	1	0	1

Table 12
Nemenyi test for average depth comparison.

	DT	RF	FBT	CMM	AFBT	ET	ET_FBT	ISM
RF	0.00	–	–	–	–	–	–	–
FBT	0.00	0.01	–	–	–	–	–	–
CMM	1.00	0	0	–	–	–	–	–
AFBT	0.22	0	0.42	0.76	–	–	–	–
ET	0.00	0.97	0.22	0	0	–	–	–
ET_FBT	0.00	0.01	1	0	0.4	0.24	–	–
ISM	0.97	0	0	0.58	0.01	0	0	–
GENESIM	0.97	0	0	0.58	0.01	0	0	1

5. Conclusion and future work

In this paper we presented a novel method for building an intelligible decision tree based on a given decision forest. The resulting tree often approximates the predictive performance obtained by the source forest while significantly reducing its prediction complexity. The new tree also provides a decision path as an explanatory mechanism for its classifications. As opposed to existing methods that aim to achieve the same objective, the proposed method does not require the availability of unlabeled data. In addition, it is scalable and can be applied to forests of any size. The main contribution of this work is enriching the available toolbox for supervised machine learning tasks, especially for domains where outputs should be justified by decision paths (e.g., insurance, healthcare, etc.) and in general cases where there is a trade-off between prediction accuracy and prediction interpretability. Despite its advantages, the proposed method has several limitations which can be addressed in future research. First, other approaches for filtering rule conjunctions can be tested in order to decrease the information loss caused by the current procedure. This may also enable applying the algorithm for forests of dependent trees (e.g., XGBoost). Another possible improvement is to enable early stopping criterion for constructing shallower and simpler trees without significantly deceasing the predictive performance. Finally, other pruning approaches might be considered in addition to the greedy approach used in this work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.inffus.2020.03.013.

CRediT authorship contribution statement

Omer Sagi: Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Visualization, Writing - original draft, Data curation, Validation. **Lior Rokach:** Conceptualization, Supervision, Writing - review & editing, Validation.

References

[1] L. Rokach, Decision forest: twenty years of research, Inf. Fusion 27 (2016) 111–125.
[2] T. Chen, C. Guestrin, Xgboost: a scalable tree boosting system, in: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 785–794.
[3] M. Belgiu, L. Drăguţ, Random forest in remote sensing: a review of applications and future directions, ISPRS J. Photogramm. Remote Sens. 114 (2016) 24–31.
[4] I. Partalas, G. Tsoumakas, I.P. Vlahavas, Focused ensemble selection: a diversity-based method for greedy ensemble selection., in: ECAI, 2008, pp. 117–121.
[5] L. Rokach, Collective-agreement-based pruning of ensembles, Comput. Stat. Data Anal. 53 (4) (2009) 1015–1026.
[6] A.A. Freitas, Comprehensible classification models: a position paper, ACM SIGKDD Explor. Newslett. 15 (1) (2014) 1–10.

[7] Y. Zhang, S. Burer, W.N. Street, Ensemble pruning via semi-definite programming, J. Mach. Learn. Res. 7 (Jul) (2006) 1315–1338.
[8] P. Domingos, Knowledge discovery via multiple models, Intell. Data Anal. 2 (1–4) (1998) 187–202.
[9] C. Bucilu, R. Caruana, A. Niculescu-Mizil, Model compression, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 535–541.
[10] A. Van Assche, H. Blockeel, Seeing the forest through the trees: Learning a comprehensible model from an ensemble, in: European Conference on Machine Learning, Springer, 2007, pp. 418–429.
[11] G. Vandewiele, O. Janssens, F. Ongenae, F. De Turck, S. Van Hoecke, Genesim: genetic extraction of a single, interpretable model, in: NIPS2016, the 30th Conference on Neural Information Processing Systems, 2016, pp. 1–6.
[12] Z.C. Lipton, The mythos of model interpretability, arXiv:1606.03490(2016).
[13] T.G. Dietterich, Ensemble methods in machine learning, in: International Workshop on Multiple Classifier Systems, Springer, 2000, pp. 1–15.
[14] M. Bohanec, M.K. Borštnar, M. Robnik-Šikonja, Explaining machine learning models in sales predictions, Expert Syst. Appl. 71 (2017) 416–428.
[15] B. Letham, C. Rudin, T.H. McCormick, D. Madigan, et al., Interpretable classifiers using rules and Bayesian analysis: building a better stroke prediction model, Ann. Appl. Stat. 9 (3) (2015) 1350–1371.
[16] M.A. Ahmad, C. Eckert, A. Teredesai, Interpretable machine learning in healthcare, in: Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, ACM, 2018, pp. 559–560.
[17] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (5) (2018) 93.
[18] M.T. Ribeiro, S. Singh, C. Guestrin, Why should i trust you?: Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1135–1144.
[19] I. Bratko, Machine learning: between accuracy and interpretability, in: Learning, Networks and Statistics, Springer, 1997, pp. 163–177.
[20] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (5) (2019) 93.
[21] D. Gunning, Explainable artificial intelligence (xai), Defense Advanced Research Projects Agency (DARPA), nd Web, 2, 2017.
[22] A. Adadi, M. Berrada, Peeking inside the black-box: a survey on explainable artificial intelligence (xai), IEEE Access 6 (2018) 52138–52160.
[23] T.Z. Zarsky, Incompatible: the GDPR in the age of big data, Seton Hall Law Rev. 47 (2016) 995.
[24] B. Goodman, S. Flaxman, European union regulations on algorithmic decision-making and a ǣright to explanationg, AI Mag. 38 (3) (2017) 50–57.
[25] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, 2017, pp. 4765–4774.
[26] J. Bien, R. Tibshirani, et al., Prototype selection for interpretable classification, Ann. Appl. Stat. 5 (4) (2011) 2403–2424.
[27] E. Štrumbelj, I. Kononenko, Explaining prediction models and individual predictions with feature contributions, Knowl. Inf. Syst. 41 (3) (2014) 647–665.
[28] F. Wang, C. Rudin, Falling rule lists, in: Artificial Intelligence and Statistics, 2015, pp. 1013–1022.
[29] H. Lakkaraju, E. Kamar, R. Caruana, J. Leskovec, Interpretable & explorable approximations of black box models, arXiv:1707.01154(2017).
[30] M. Craven, J.W. Shavlik, Extracting tree-structured representations of trained networks, in: Advances in Neural Information Processing Systems, 1996, pp. 24–30.
[31] J.J. Thiagarajan, B. Kailkhura, P. Sattigeri, K.N. Ramamurthy, Treeview: peeking into deep neural networks via feature-space partitioning, arXiv:1611.07429(2016).
[32] N. Frosst, G. Hinton, Distilling a neural network into a soft decision tree, arXiv:1711.09784(2017).
[33] Y. Zhou, G. Hooker, Interpreting models via single tree approximation, arXiv:1610.09036(2016).
[34] S.B. Kotsiantis, Decision trees: a recent overview, Artif. Intell. Rev. 39 (4) (2013) 261–283.
[35] J.R. Quinlan, Generating production rules from decision trees., in: iJCAI, 87, Citeseer, 1987, pp. 304–307.
[36] C. Apté, S. Weiss, Data mining with decision trees and decision rules, Future Gener. Comput. Syst. 13 (2–3) (1997) 197–210.
[37] C. Yang, A. Rangarajan, S. Ranka, Global model interpretation via recursive partitioning, in: 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE, 2018, pp. 1563–1570.
[38] J. Huysmans, K. Dejaeger, C. Mues, Jan, An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models, Decis. Support Syst. 51 (1) (2011) 141–154.
[39] J.R. Quinlan, C4. 5: Programs for Machine Learning, Elsevier, 2014.
[40] L. Breiman, Classification and Regression Trees, Routledge, 2017.
[41] R. Pandya, J. Pandya, C5. 0 Algorithm to improved decision tree with feature selection and reduced error pruning, Int. J. Comput. Appl. 117 (16) (2015) 18–21.
[42] B.P. Roe, H.-J. Yang, J. Zhu, Y. Liu, I. Stancu, G. McGregor, Boosted decision trees as an alternative to artificial neural networks for particle identification, Nucl. Instrum. Methods Phys. Res., Sect. A 543 (2) (2005) 577–584.
[43] L. Rokach, O. Maimon, Data Mining With Decision Trees: Theory and Applications, World Scientific, 2014.
[44] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems, J. Mach. Learn. Res. 15 (1) (2014) 3133–3181.
[45] C. Zhang, C. Liu, X. Zhang, G. Almpanidis, An up-to-date comparison of state-of-the-art classification algorithms, Expert Syst. Appl. 82 (2017) 128–150.
[46] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT press, 2016.
[47] S. Paisitkriangkrai, C. Shen, A. van den Hengel, Pedestrian detection with spatially pooled features and structured ensemble learning, IEEE Trans. Pattern Anal. Mach. Intell. 38 (6) (2016) 1243–1257.
[48] L.F.S. Pereira, S. Barbon, N.A. Valous, D.F. Barbin, Predicting the ripening of papaya fruit with digital imaging and random forests, Comput. Electron. Agric. 145 (2018) 76–82.
[49] F. Idrees, M. Rajarajan, M. Conti, T.M. Chen, Y. Rahulamathavan, Pindroid: a novel android malware detection system using ensemble learning methods, Comput. Secur. 68 (2017) 36–46.
[50] N.C. Oza, K. Tumer, Classifier ensembles: select real-world applications, Inf. Fusion 9 (1) (2008) 4–20.
[51] Y. Freund, R.E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in: European Conference on Computational Learning Theory, Springer, 1995, pp. 23–37.
[52] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[53] T.K. Ho, Random decision forests, in: Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on, 1, IEEE, 1995, pp. 278–282.
[54] Y. Amit, D. Geman, Randomized Inquiries About Shape: an Application to Handwritten Digit Recognition., Technical Report, DTIC Document, 1994.
[55] O. Sagi, L. Rokach, Ensemble learning: a survey, Wiley Interdiscip. Rev. 8 (4) (2018) e1249.
[56] T. Han, D. Jiang, Q. Zhao, L. Wang, K. Yin, Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery, Transactions of the Institute of Measurement and Control (2017). 0142331217708242
[57] G. Tsoumakas, I. Partalas, I. Vlahavas, A taxonomy and short review of ensemble selection, in: ECAI 2008, Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications, 2008, pp. 41–46.
[58] R. Caruana, A. Niculescu-Mizil, G. Crew, A. Ksikes, Ensemble selection from libraries of models, in: Proceedings of the Twenty-First International Conference on Machine Learning, ACM, 2004, p. 18.
[59] Q. Hu, D. Yu, Z. Xie, X. Li, Eros: ensemble rough subspaces, Pattern Recognit. 40 (12) (2007) 3728–3739.
[60] Z.-H. Zhou, W. Tang, Selective ensemble of decision trees, Rough sets, fuzzy sets, data mining, and granular computing, 2003. 589–589
[61] C. Qian, Y. Yu, Z.-H. Zhou, Pareto ensemble pruning., in: AAAI, 2015, pp. 2935–2941.
[62] X. Jiang, C.-a. Wu, H. Guo, Forest pruning based on branch importance, Comput. Intell. Neurosci. 2017 (2017).
[63] L. Breiman, N. Shang, Born Again Trees, Technical Report, University of California, Berkeley, Berkeley, CA, 1996.
[64] Y. Akiba, S. Kaneda, H. Almuallim, Turning majority voting classifiers into a single decision tree, in: Tools with Artificial Intelligence, 1998. Proceedings. Tenth IEEE International Conference on, IEEE, 1998, pp. 224–230.
[65] W. Fan, F. Chu, H. Wang, P.S. Yu, Pruning and dynamic scheduling of cost-sensitive ensembles, in: AAAI/IAAI, 2002, pp. 146–151.
[66] D.J. Hand, R.J. Till, A simple generalisation of the area under the ROCcurve for multiple class classification problems, Mach. Learn. 45 (2) (2001) 171–186.
[67] A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognit. 30 (7) (1997) 1145–1159.
[68] F.E. Otero, A.A. Freitas, Improving the interpretability of classification rules discovered by an ant colony algorithm, in: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, ACM, 2013, pp. 73–80.
[69] R.C. Fong, A. Vedaldi, Interpretable explanations of black boxes by meaningful perturbation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3429–3437.
[70] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (1) (2006) 3–42.
[71] H. Ping, J. Stoyanovich, B. Howe, Datasynthesizer: privacy-preserving synthetic datasets, in: Proceedings of the 29th International Conference on Scientific and Statistical Database Management, 2017, pp. 1–5.
[72] P. Clark, T. Niblett, The CN2 induction algorithm, Mach. Learn. 3 (4) (1989) 261–283.
[73] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, et al., Orange: data mining toolbox in python, J. Mach. Learn. Res. 14 (1) (2013) 2349–2353.
[74] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.
[75] Y. Jin, B. Sendhoff, Pareto-based multiobjective machine learning: an overview and case studies, IEEE Trans. Syst. Man Cybern. Part C) 38 (3) (2008) 397–415.