From Random Forest to an interpretable decision tree - An evolutionary approach

Krzysztof Jurczuk
Bialystok University of Technology
Bialystok, Poland
k.jurczuk@pb.edu.pl

Marcin Czajkowski
Bialystok University of Technology
Bialystok, Poland
m.czajkowski@pb.edu.pl

Marek Kretowski
Bialystok University of Technology
Bialystok, Poland
m.kretowski@pb.edu.pl

ABSTRACT

Random Forest (RF) is one of the most popular and effective machine learning algorithms. It is known for its superior predictive performance, versatility, and stability, among other things. However, an ensemble of decision trees (DTs) represents a black-box classifier. On the other hand, interpretability and explainability are ones of the top artificial intelligence trends, to make predictors more trustworthy and reliable. In this paper, we propose an evolutionary algorithm to extract a single DT that mimics the original RF model in terms of predictive power. The initial population is composed of trees from RF. During evolution, the genetic operators modify individuals (DTs) and exploit the initial (genetic) material. e.g., splits/tests in the tree nodes or more expanded parts of the DTs. The results show that the classification accuracy of a single DT predictor is not worse than that of the original RF. At the same time, and probably most importantly, the resulting classifier is a single smaller-size DT that is almost self-explainable.

CCS CONCEPTS

• **Computing methodologies** → **Ensemble methods**; **Classification and regression trees**; **Supervised learning**.

KEYWORDS

evolutionary algorithms, decision tree, ensemble learning, explainable machine learning

ACM Reference Format:
Krzysztof Jurczuk, Marcin Czajkowski, and Marek Kretowski. 2023. From Random Forest to an interpretable decision tree - An evolutionary approach. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3583133.3590732

1 INTRODUCTION

Artificial intelligence (AI) has undergone significant and continuous progress in the last two decades [11]. However, with the high increase in the predictive power of models and solving many difficult problems, the fast growth of model complexity and using non-transparent block-box predictors could be observed. In many domains (like medicine, insurance or penology), it is not rare that a clear explanation for individual decisions, verifiability, and comprehension are very desired. Thus, recently, the explainability and interpretability of the models have gained the attention of both the scientific and business communities, becoming one of the top trends in AI.

Decision trees (DTs) are a supervised machine learning (ML) technique that can be used for classification and regression problems [7]. DTs are considered as an interpretable and self-explanatory ML algorithm that belongs to eXplainale AI (XAI). Despite more than 50 years of research on DTs, they are constantly being explored. In this paper, we show how an interpretable DT can be generated starting from a black-box strong classifier while maintaining the predictive performance. We chose Random Forest (RF) [6] which is well known for its superior predictive power. However, it is not easy to explain the rationale behind the RF-based classifications because each of them is derived from the results of multiple DTs.

We propose an evolutionary approach that is novel to previous attempts. In the most related work [12], a set of rule conjunctions representing the original RF model were created, filtered (using heuristics), and, then, hierarchically organized into DT (using the entropy measure). It was a robust solution, however, it limited the search space due to the computational complexity. For example, a linear increase in the forest size resulted in an exponential growth in the conjunction set size. Other works mainly focused on simplifying DT ensembles, e.g., pruning [5] or generating simple rules greedily combining and simplifying their base trees [9].

2 EVOLUTION OF AN INTERPRETABLE DECISION TREE FROM RANDOM FOREST

The general idea of extracting a single DT from an RF model is presented in Fig. 1. It is based on an evolutionary induction and extends an existing tree inducer from a well-founded Global Decision Tree (GDT) framework [4]. We call the proposed approach as RF-based GDT (RFbGDT). The evolution starts with generating an initial population. First, the RF classifier is trained and DTs from the ensemble are used as the initial individuals. By default, all generated DTs are employed, but it is not obligatory.

Concerning the RF training, in brief, each DT in the forest is induced using random subsets of the training dataset [6]. Moreover, the algorithm when creating subsequent tree nodes (splits/tests, leaves) checks a random subset of attributes and chooses the best split among them (in a greedy top-down manner). The forest of multiple weak DTs forms a robust classifier, in aggregate.

In the next step, the population of DTs is modified and evaluated in the evolutionary loop (see Fig. 1). The solution follows the typical evolutionary algorithm schema with an unstructured, fixed-size population and a generational selection. The individuals in the

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0120-7/23/07.
https://doi.org/10.1145/3583133.3590732

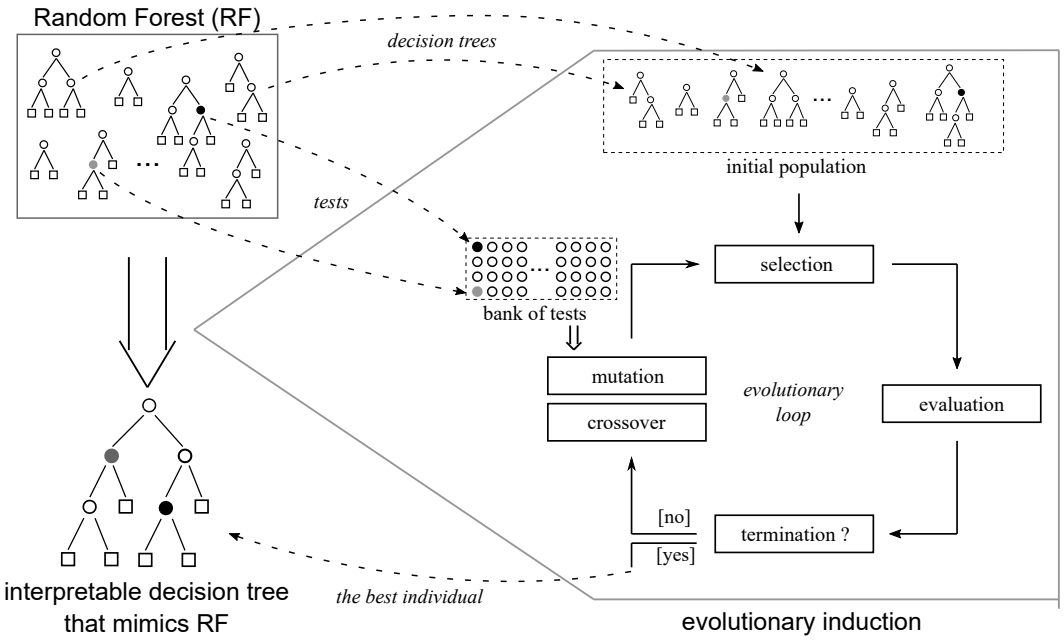


Figure 1: General idea of the proposed algorithm for the induction of an interpretable decision tree from the RF classifier.

population are represented and processed in their actual form as binary classification trees with univariate tests in the internal nodes [4]. In each generation (iteration), genetic operators (mutation and crossover) alter DTs and force the search for the best single tree. The linear ranking selection is applied, and, the best individual in the current population is copied to the next one (elitist strategy) [8]. The evolution ends when the maximum number of generations is reached or the best DT in the population does not improve during the fixed number of generations. Finally, the best DT in the population is taken as an (interpretable) representation of the initial RF model.

The genetic operators modify DTs, and, if needed, they use the DTs from the initial forest or their parts, e.g., tests, subtrees, branches (i.e., RF-based genetic material). Tests from the RF are extracted (e.g., pairs of attribute and threshold, for numeric features), and, they feed the bank of tests (see Fig. 1). It includes all tests from the RF trees, by default. Currently, information about the position in the original tree or prediction error of tests is not used.

The mutation starts with selecting a tree node type (leaf or internal node) and, then, one of the nodes to be affected. The procedure is based on the ranking linear selection [8] considering the position (tree level) and reclassification error of nodes [4]. Different specialized variants of mutations are applied, e.g.:

- *new test* - replacement the current test by a random one from the bank of tests;
- *prune* - changing an internal node into a leaf;
- *expand* - changing a leaf into an internal node using a random test from the bank of tests;
- *exchange* - exchanging tests/subtrees between parent and son nodes.

The crossover combines elements of two existing DTs, and two new individuals are created. Because the initial DTs come from the RF, we can simplify that its role is to exchange information between the tree of the forest (i.e., exchange genetic material) and, finally, to aggregate the most relevant in a single tree. The crossover begins by randomly selecting pairs of DTs. Then, in each DT, the position (node) is chosen, in a similar way as during the mutation. The exchange of information is carried out using specialized operator variants, e.g.:

- *exchange tests* - exchanging tests between nodes of two affected DTs;
- *exchange subtrees/branches* - exchanging subtrees/branches between nodes of two affected DTs.
- *asymmetric* crossover - transfer subtrees asymmetrically where the subtree of the first/second individual is replaced by a new one that was duplicated from the second/first individual; it is preferred that the receiver node has a high reclassification error, while the donor node a small error value as it is duplicated [4].

In each evolutionary loop, all new DTs are evaluated. The original GDT system provides various multi-objective optimization strategies, including weight formula, lexicographic analysis and Pareto-dominance [4]. In this paper, a simple weighted form of the fitness function is used:

$$Fitness(T) = Accuracy(T) - \alpha \cdot (Comp(T) - 1.0). \quad (1)$$

It maximizes accuracy ($Accuracy(T)$) estimated on the training dataset and minimizes the complexity term ($Comp(T)$) of the tree T. The complexity term equals the tree size which is usually the number of tree nodes. α is a user-defined parameter that reflects the relative importance of the complexity term and, thus, may be used to steer the tree size and interpretability (default value is 0.001).

3 EXPERIMENTAL VALIDATION

3.1 Setup

Both real-life and artificial datasets have been used in the preliminary validation. The details of each one are presented in Table 1. We have chosen four real-life datasets (with a different number of samples and attributes) from the UCI Machine Learning Repository [1]. They were also tested in the related previous work [12]. Concerning the artificial datasets, we generated two two-dimensional problems to easily show the distribution of samples and decision boundaries. The first one was called *Slope* (see Fig. 3(a) top), while the second one *House* (see Fig. 3(a) bottom).

The RF algorithm settings were chosen as in the related work [12], e.g., 100 of estimators (DTs) in a forest, the maximum depth of each DT = 5, and the minimum number of samples in a leaf = 5. The main parameters of the evolutionary induction were as follows: population size = 100 individuals, crossover probability = 20% assigned to the tree, mutation probability = 80% assigned to the tree, elitism rate = 1 individual per generation, number of iterations = 1 000, and the probabilities of different variants of crossover/mutation were uniformly distributed. 10-fold cross-validation was repeated 10 times and the average results are presented.

3.2 Results and Discussion

The results are presented in Table 2. We show both the classification accuracy and the tree size (number of nodes). In the case of RF, it is the average size of a tree in the forest (so the actual size of an ensemble is 100 times bigger). We also present the results for GDT [4] where the search is more global, i.e., not limited to the RF genetic material. In most of the cases, the RFbGDT algorithm gives a comparable accuracy to the RF classifier. For two datasets (*Pima* and *Wine*), the accuracy is slightly lower. However, bearing in mind that RFbGDT generates smaller DTs and the final classifier consists of only a single DT, the results seem to be very promising.

Comparing GDT and RFbGDT, we see that the global search can give smaller DTs but sometimes at the cost of an accuracy drop. The comparison with the previous related work (called RF-FBT) [12] is also in favor of RFbGDT. The obtained RFbGDT accuracy is at least not worse but in most of the cases is a few percent better. We allow the evolution to work on all the genetic material coming from the RF model and do not limit the search space like in RF-FBT. We have left more in-depth validation and comparison for future research (statistical analysis, more datasets, etc.). However, the direct comparison between these two approaches is not simple. While it is quite easy to provide the same RF model for the initialization, these two solutions are based on totally various approaches and can be tuned in different ways. For example, the independent parameters can influence the final accuracy and/or DT size, like the value of α parameter in RFbGDT or, on the other hand, the filtering of conjunction set in RF-FBT.

The evolutionary induction of DTs is known to be time-demanding in comparison to the greedy top-down algorithms [2]. Concerning the presented results, the time needed to induce a single DT was at most in seconds or a few minutes using only a sequential CPU code. The execution time is going to increase fast with the increase in the number of samples. Fortunately, the evolutionary (global) DT induction was successfully parallelized both on computer clusters and GPUs [10], which would also allow us to deal with large-scale data mining. In RF-FBT [12], also the time of computations was one of the challenges. It has led to a reduction of the search space, e.g., the number of considered conjunctions was limited.

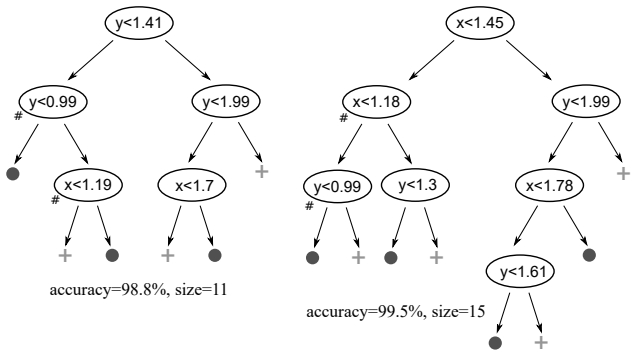


Figure 2: Examples of DTs trained by RFbGDT on the *Slope* dataset.

We also show an example visualization of the DT boundaries in Figure 3 (for two-dimensional datasets *Slope* (top) and *House* (bottom)). On the left (a), the analytical boundaries (used to generate datasets) are presented. In the middle (b), the boundaries of all DTs in one of the forests are drawn. On the right (c), we see the boundaries for the single corresponding DT. It is easily visible that single DT models are more transparent and much easier to interpret than their RF counterparts. Concerning the *House* dataset, interestingly, we can see that the split in $x = 1.5$ was not found by the RF classifier. This resulted in the lack of it also in the single DT predictor (see arrow in Fig. 3(c) bottom). The split would be included in RF when the maximum tree depth was increased.

However, we have to be conscious that the classification accuracy may impair a little when a single DT tries to mimic an RF predictor. In the case of *Slope* and *House* datasets, the drop is only about 0.5%. Obviously, this accuracy impairment can depend on the DT size, as it is shown in Figure 2. We see two examples of DTs training by RFbGDT, and the bigger one provides better accuracy. If comparing them more deeply we can observe that they consist of similar building blocks (e.g., $y < 0.99$ and $x < 1.18$ or $x < 1.19$, the nodes are marked by #), coming from the RF genetic material.

4 CONCLUSION

We propose an evolutionary approach to extract an interpretable DT from the RF classifier. The preliminary validation shows it is a robust and fast solution that can compete with the previous one [12]. It can be extended to other types of DTs, like regression ones [3], as well as to other ensemble techniques (e.g., XGBoosting [13]) or neural networks.

ACKNOWLEDGMENTS

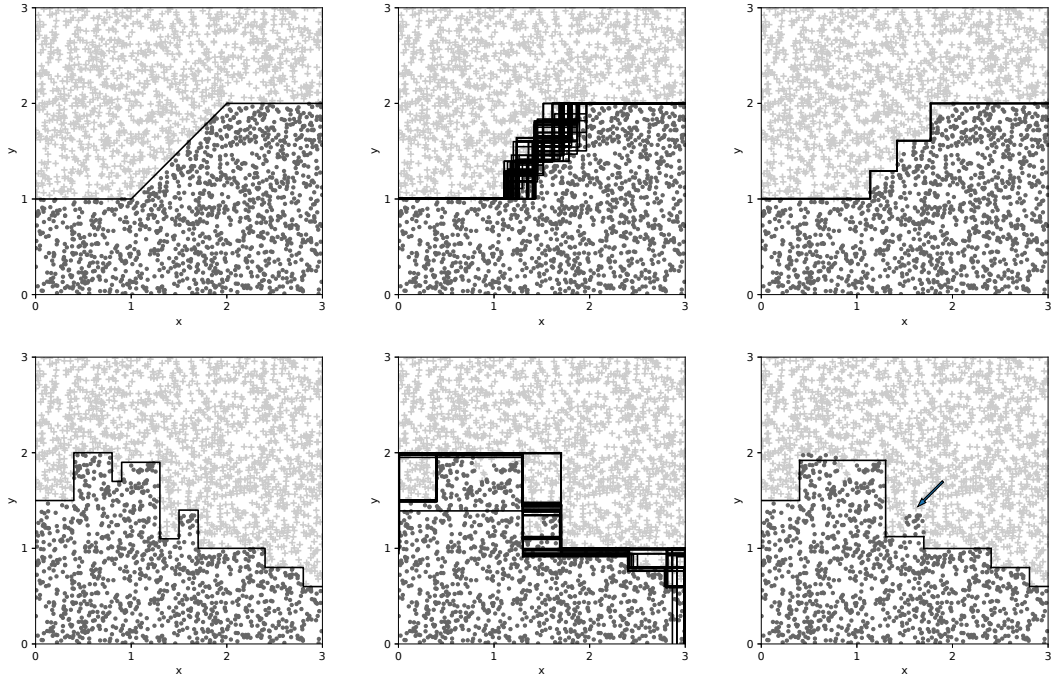
This work was supported by Bialystok University of Technology under the Grant WZ/WI-IIT/4/2023 founded by Ministry of Science and Higher Education

Table 1: Characteristics of the real-life and artificial datasets: name, number of samples, number of attributes

Dataset	Number of samples	Number of attributes
Banknote	1372	4
Magic	2000	10
Pima	768	8
Wine	178	13
Slope	10 000	2
House	10 000	2

Table 2: Comparison of GDT, RFbGDT and RF inducers. The size equals the number of tree nodes, while, in the case of RF, it is the average tree size in the forest (average number of nodes). Accuracy of a classical DT can be found in [12].

Dataset	GDT		RFbGDT		RF	
	accuracy	size	accuracy	size	accuracy	size
Banknote	97.86±1.52	15±3.1	98.37±1.18	17±2.06	98.16±1.34	27±0.55
Magic	83.01±1.03	17±1.93	82.82±0.81	17±2.13	82.99±0.82	54±0.52
Pima	74.48±4.57	25±5.89	74.49±4.54	25±6.08	76.48±4.34	34±0.75
Wine	93.99±5.35	13±2.47	92.80±5.23	12±2.04	97.63±3.23	10±0.31
Slope	98.49±0.34	11±0.0	99.01±0.34	14±1.39	99.48±0.26	32±0.72
House	97.05±0.97	14±2.07	98.52±0.52	22±2.97	98.98±0.40	40±0.59



(a) analytical (b) RF (c) RFbGDT

Figure 3: An example visualization of decision boundaries: (a) analytical as well as of (b) RF (c) RFbGDT classifiers, for artificial datasets *Slope* (top) and *House* (bottom).

REFERENCES

[1] Dheeru Dua and Efi Karra Taniskidou. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[2] Krzysztof Jurczuk, Marcin Czajkowski, and Marek Kretowski. 2017. Evolutionary induction of a decision tree for large-scale data: a GPU-based approach. *Soft Computing* 21, 24 (2017), 7363–7379.

[3] Krzysztof Jurczuk, Marcin Czajkowski, and Marek Kretowski. 2022. GPU-based acceleration of evolutionary induction of model trees. *Applied Soft Computing* 119 (2022), 108503.

[4] Marek Kretowski. 2019. *Evolutionary Decision Trees in Large-Scale Data Mining.* Springer. https://doi.org/10.1007/978-3-030-21851-5

[5] Vrushali Y Kulkarni and Pradeep K Sinha. 2012. Pruning of Random Forest classifiers: A survey and future directions. In *2012 International Conference on Data Science & Engineering (ICDSE).* 64–68. https://doi.org/10.1109/ICDSE.2012.6282329

[6] Breiman Leo. 2001. Random Forests. *Machine Learning* 45 (2001), 5–32.

[7] Wei-Yin Loh. 2014. Fifty Years of Classification and Regression Trees. *International Statistical Review* 82, 3 (2014), 329–348.

[8] Zbigniew Michalewicz. 1996. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer.

[9] Josue Obregon and Jae-Yoon Jung. 2023. RuleCOSI+: Rule extraction for interpreting classification tree ensembles. *Information Fusion* 89 (2023), 355–381.

[10] Rafael Rivera-Lopez, Juana Canul-Reich, Efrén Mezura-Montes, and Marco Antonio Cruz-Chávez. 2022. Induction of decision trees as classification models through metaheuristics. *Swarm and Evolutionary Computation* 69 (2022), 101006.

[11] Waddah Saeed and Christian Omlin. 2023. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems* 263 (2023), 110273.

[12] Omer Sagi and Lior Rokach. 2020. Explainable decision forest: Transforming a decision forest into an interpretable tree. *Information Fusion* 61 (2020), 124–138.

[13] Omer Sagi and Lior Rokach. 2021. Approximating XGBoost with an interpretable decision tree. *Information Sciences* 572 (2021), 522–542.