IJCI v10 -01(2023) 54-64
International Journal of Computers and Information (IJCI)
Available online at *https://ijci.journals.ekb.eg/*

Improved version of explainable decision forest: Forest-Based Tree

Faten A. Khalifa[*,a], Asmaa H. Aly[*,b], Hatem M. Abdelkader[*,c]

[*]*Information Systems dept., Faculty of Computers and Information, Menoufia University, Shebin Elkom 32511, Menoufia, Egypt.*
[a]*faten_a_khalifa@ci.menofia.edu.eg,* [b]*asmaa.elsayed@ci.menofia.edu.eg,* [c]*hatem.abdelkader@ci.menofia.edu.eg*

Abstract

A Decision Forest is an ensemble learning method that seeks to enhance the predictivity of a single decision tree via training several trees and combining their decisions. However, it is not easy to explain the rationale behind the predictions of decision forests; as each prediction consists of an integration of many decisions. This defect of decision forest makes it a black box, missing interpretable capability, and it will be difficult for humans to understand its entire logic. In this article, we discuss the transformation of the decision forest into a single decision tree, Forest-Based Tree (FBT), without sacrificing accuracy. The proposed method combines the decision rules of individual trees and organizes them into a tree structure. We focus on how to optimize the algorithm; to build an intelligible and lightweight forest-based tree quickly. The open source software for FBT is also provided. Results on 30 UCI datasets show the objective to approximate the predictive performance of the decision forest through forming an explainable decision tree in less computational time.

Keywords: Decision forest; ensemble pruning; ensemble-derived models; explainable AI; random forest.

1. Introduction

Ensemble learning is the strategy of using numerous learning algorithms to get more accurate predictions than all of the constituent learning algorithms alone [1–4]. Thanks to their capabilities, ensembles received great attention in the applications related to data mining (e.g. remote sensing, banking and economical systems, recommender systems, emotion recognition, medical systems, human activity recognition, etc.) [5, 6]. From the literature review, the performance of ensemble classification is conditioned by generating a large pool of individual classifiers [7, 8]. However, a large-size ensemble comes with many defects. First, the processing and storage resources should be enough to cope with the complexity of the formed model. Second, the classification latency does not suit real-time applications; as all the individual models should be aggregated for generating each prediction. In addition, regardless of ensemble size, the ensemble methods cannot be interpreted.

Many researchers focused on solving the above vulnerabilities through two major directions: ensemble pruning and ensemble-derived models [9]. Ensemble pruning is the process of selecting some trained individual learners in order to reduce the storage and the computational costs, while reserving the original performance [5]. Theoretical and practical studies show the significant effect of ensemble pruning in terms of accuracy and model space complexity. Nevertheless, ensemble interpretation is still unresolved. Ensemble-derived models focus on transforming the original ensemble into a simple, light, and intelligible model without sacrificing the predictivity power of the original ensemble.

In Domingo's combined multiple models (CMM) [10], an enormous set of synthetic or unlabeled data is to be classified by a trained decision forest. Then a final decision tree, that simulates the decision forest, is to be

trained using the artificial labeled data. The usage of CMM is restricted by the availability of a vast amount of unlabeled data or data generation technique that fits the training set. GENESIM [11] is an algorithm that transforms a decision forest into a single decision tree by using a genetic algorithm. The complexity of GENESIM increases exponentially with the size of the inputted ensemble as well as the size of the data. Furthermore, the performance of GENESIM is sensitive and parameter-dependent. Recently, Sagi et al. [9] proposed a forest-based tree that combines conjunction sets of forest members and then reorganizes the aggregated conjunction set into a tree structure using the highest information gain for node splitting.

In this article, we reinvestigate the ensemble-derived model from [9], discuss its pitfalls, and propose an improvement. Particularly, this model consumes too much time to build and train forest-based tree. Its time complexity increases exponentially with both the decision forest size and the dataset size. This would make it extremely hard for this model to be applied to large and high-dimensional datasets. Our contribution is to optimize their work to tackle this defect without affecting the accuracy or the explanation of the formed tree. Our proposition increases the model's scalability to adapt to a decision forest of any size while reducing the training time. The organization of the remaining content is as follows: Section 2 introduces in short a literature overview and the related work. Our proposition is illustrated in section 3. Section 4 presents the experimental settings, evaluation, and discussion of the results. Finally, the conclusions and several possible future directions are reported in section 5.

2. Overview and Related work

Artificial Intelligence (AI), in general, and decision forest models, in particular, are superior for handling large and complex tasks. The explanation of these models has become a crucial area of study to understand their behavior. Specifically, in situations where decisions taken by systems may directly or indirectly affect humans (e.g., justice, insurance, health-care systems, etc.) [12, 13]. The interpretation of the model has high importance, equal to its performance. This is because interpretability is the core to achieving trustworthy and secure AI-based systems [14]. The following section reviews interpretable models, decision forests, and decision forest-based trees.

2.1. Overview

An interpretable model allows humans to understand its entire reasoning which leads to a particular outcome [15]. In other means, the model is interpretable if a human can explain its operation using a plain text or a simple visualization [16]. For example, linear models, decision rules, and decision trees are interpretable models [17, 18]; because they can easily explain themselves.

A Decision Tree (DT) is a highly predictive model. Lior Rokach [19] defined it as "the process of recursively partitioning the covariates space to subspaces that comprise a base for prediction". DT is a kind of indirect rule learning that uses structure branching decisions to model the relationships among the features and the predicted class value. DT is extensively used in the area of machine learning and can model any type of data. It is a human-readable model which is appropriate in applications where legal reasoning is required. Despite its interpretability, DT is vulnerable to overfitting, and its internal parameters should be tuned [20]. Furthermore, DT is extremely sensitive to tiny changes in the training data and is biased towards the splits on features. The reasoning from DT can be understood through visualized prediction paths linking the root of the tree to its leaves. Each prediction path represents a set of rules that maximize the splitting of samples into disjoint partitions. A complete example of classifying the Iris dataset using DT can be found in [19].

Decision rules have a general structure: IF the conditions are satisfied THEN produce a certain prediction. Each rule has two measurements: coverage and accuracy. Rule coverage measures the percentage of samples, from the training set, that satisfy that rule. Rule accuracy measures how accurate the rule is when it predicts the correct class for all samples covered by it [21]. Decision rules are called divide-conquer algorithms [22]. The data parts are divided according to one rule, then recursively conquer the divided parts. Those models are

transparent/ explainable to non-experts in the form of logical structures. Available features are analyzed to find homogeneous groups, and then an additional rule is built to drill down more. There are many algorithms for rule induction (e.g. OneR, Sequential covering, Bayesian Rule Lists, etc.). CORELS (Certifiable Optimal RulE ListS) is a custom discrete optimization methodology developed for constructing rule lists [23]. CORELS is used for criminal recidivism prediction to generate the interpretable model as in Fig. 1.

IF	age between 18-20 and gender is male	THEN predict arrest (within 2 years)
ELSE IF	age between 21-23 and 2-3 prior offences	THEN predict arrest
ELSE IF	more than three priors	THEN predict arrest
EISE	predict no arrest	

Fig. 1. Rule List generated by CORELS algorithm for criminal recidivism prediction

Linear Models make a prediction by summing the weights of the feature inputs [21]. The contribution of each feature to the target is estimated and combined in an additive manner. The linearity of the model supports interpretation on a modular level; through estimated weights. The weights represent the slope (gradient) of the linear model in each direction to determine how the target value can be influenced per feature change. That interpretation depends on the feature type (e.g. numerical, binary, and categorical). One of the advantages of the linear model is that it can be used for feature selection. On the other side, it is difficult to understand the interaction or joint distribution of multi features.

A Decision forest is an ensemble methodology that combines several decision trees. The key to obtaining an improved predictive model is that the base decision trees complement each other. If each tree in the forest has a preference that is more accurate than a random guess, then summing all trees outputs could lead to a better prediction [19]. The individual trees can be generated iteratively such as AdaBoost [24], or in a non-iterative manner like Bagging [7]. A comparison between boosting-based and bagging-based decision forests indicated that boosting is sometimes more precise than Bagging [25]. Furthermore, the iterative-based decision forests are more sensitive to noisy data. Decision forests have been utilized to overcome several learning challenges, such as imbalanced datasets, concept drift, curse of dimensionality, and hierarchical multilabel classification tasks [19]. In [26], random forest as a bagging-based decision forest proved its superiority over 179 classification algorithms according to 121 datasets. With inspiration from those results, random forest has been deployed with many real-world applications. Moreover, the popularity of random forest comes from its simplicity and availability as an off-the-shelf package within various software platforms, such as Weka, R, Matlab, sci-kit-learn, and C#.

2.2. Related work

As previously explained, random forest suffers from several drawbacks. First, it suffers from raising latency issues in real-time systems. The second drawback is that it is non-comprehensible or non-explainable by a human. That restricts the domains in which random forest can be used (e.g., medicine, insurance, justice, manufacturing industries, etc.). One solution is to derive an interpretable model from a given random forest [10, 11]. We define interpretation as providing a thorough grasp of the relationship between inputs and outputs to uncover data insights. By capturing the learning of the trained forest, the new model will preserve predictive performance while providing faster and more comprehensible decisions. In [27], the minimum requirements for an interpretable model are to be simple, stable, and predictive. Clément et al. [27] extracted rules from a tree ensemble depending on how frequently they appear. The most common rules, which represent the data's most robust and powerful patterns, are then linearly merged to generate forecasts.

Random Forest Optimal Counterfactual Set Extractor (RFOCSE) is presented in [28]. It is based on partially blending tree predictors from the random forest into only one decision tree. Counterfactual is an Explainable ML approach. It explains predictions by describing the adjustments that must be made on a sample to reverse the prediction's outcome.

In [9], Sagi et al. performed post-processing on a random forest to transform it into a set of conjunction rules with eliminating the contradicted ones. They measured the rule coverage in an iterated manner to select the most probable and significant rules. Finally, the conjunction set is organized in a tree structure to support fast and explainable predictions. They applied ensemble pruning to reduce the exponential complexity coming from the large-random forest. That could result in less training time and generation of relevant conjunctions via eliminating redundant trees. However, while analyzing their work, the ensemble pruning consumed exorbitant time (will present that in the experimentation section). In addition, the pruning time increases exponentially with the random forest size and the dataset size. They did not perform any vector-based computations. This makes it very hard to be applied to large-size datasets. The results from their article do not discuss any time complexity.

The authors of [29] proposed a post-hoc and model-specific algorithm (Non-Overlapping Tree Ensemble (NOTE)) to provide global explanations of a random forest. This algorithm converts a random forest into a decision tree ensemble. It utilizes non-overlapping competence regions to create an ensemble classifier composed of several decision trees. However, NOTE has some drawbacks; it is considered an interpretable alternative to the random forest. One of these drawbacks is that it is not always accurate as the random forest. Moreover, the algorithm cannot be applied to large datasets because its computational complexity is very high.

In their work, Zolanvari et al. [30] proposed an XAI model called TRUST (Transparency Relying Upon Statistical Theory) to build accurate AI-enabled cyber security systems [31]. It converts the input features into a new set of latent variables by using factor analysis. These variables are ranked to pick only the most influential ones which will represent the classes. Then, the classification process is accomplished via multi-modal Gaussian distributions. The main limitation of TRUST is that it suffers from overfitting the training data. In the next section, we present our proposition to mitigate those defects and develop a more computationally efficient decision tree from a random forest. Our proposed method depends on eliminating redundant trees quickly to reduce the training time.

3. **Proposition**

As we stated previously, our contribution is an improvement of what has been discussed in [9]. The forest-based tree in [9] follows three phases (as shown in Fig. 2): ensemble pruning, conjunction set generation, and tree building from the conjunction set. The pruning phase is a critical step to eliminating forest size, eliminating conjunction set size, and reducing model construction time. A greedy method is used to explore different subsets that maximize AUC (Area Under the Curve) by iteratively expanding empty subset until there is no improvement. For forest of size L, $\{\psi_1, \psi_2, \psi_3, \ldots, \psi_k, \psi_{k+1} \ldots, \psi_L\}$. If ψ_1 is the most accurate tree over the pruning set, then the AUC of all combined subsets with ψ_1 should be measured; $\{\psi_1, \psi_2\}$, $\{\psi_1, \psi_3\}$, $\{\psi_1, \psi_k\}$, $\{\psi_1, \psi_{k+1}\}$, $\{\psi_1, \psi_L\}$. Then the subset with the highest AUC is reserved for further expansion. The computational cost of this method increases monotonically with: pruning set size, ensemble size (L), and the subset size $\{\psi_1, \ldots, \ldots\}$. The predictions of ψ_1 are repeatedly obtained with each new combination (e.g. with $\{\psi_1, \psi_2\}$, $\{\psi_1, \psi_3\}$, . . .). That increases the computational time with no effective gain.

After the pruning step, a set of rule conjunctions characterizing the decision forest is built iteratively. Each conjunction is a logical conjunction that simulates a possible outcome of the decision forest. That outcome consists of a number of labels combined with the probability of the corresponding class. For each tree in the decision forest, all the conjunctions are merged by applying a Cartesian product. Then, all the conflicting conjunctions and the conjunctions with probability under a predefined threshold are removed.

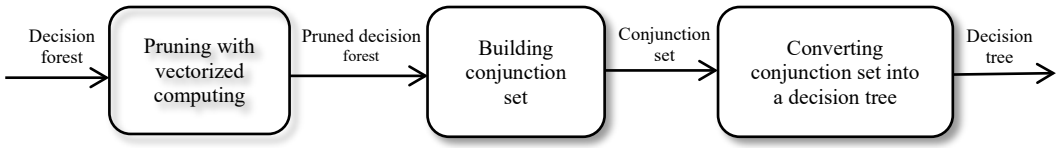


Fig. 2. Phases of building a forest-based tree

The last phase is transforming the conjunction set into a decision tree. First, it is described as a dataset where each row constitutes a single conjunction and also stores the class probabilities as well as the predicted class. All the conjunctions are defined as a root node. Then, an iterative splitting operation is performed by routing all the relevant conjunctions to the left node and the irrelevant ones to the right node.

Our proposition is to use vectorized computing [32] to do intricate calculations on the whole data without needing loops. This would result in a significant speed improvement. Vectorization is the process of transforming an algorithm from operating on a single value at a time to operating on a set of values (vector) at one time. The predictions of all classification trees over the pruning set are calculated once, in advance, not in a repeated manner with each subset as in [9]. Fig. 3 illustrates a specific case study for a pruning set size = 7, forest size = 5, number of labels = 2 {'P','N'}. First, the conditional probability vectors obtained from classification trees $\{\psi_k\}$ are combined to form an N-dimensional array (e.g. 7 × 10). Second, with information about forest size and the number of labels, the indices of columns are determined (e.g. P={0, 2, 4, 6, 8} and N={1, 3, 5, 7, 9}). This information speeds up the aggregation process. For the subset $\{\psi_1, \psi_3\}$, the P column is the sum of 0's and 4th columns, while the Nth column is the sum of 1st and 5th columns. Finally, the class with the highest value in the aggregated vector is selected and the AUC metric is measured for that subset. Only with indices and vectorized computing, the pruning time is dramatically decreased.

4. Experimentation

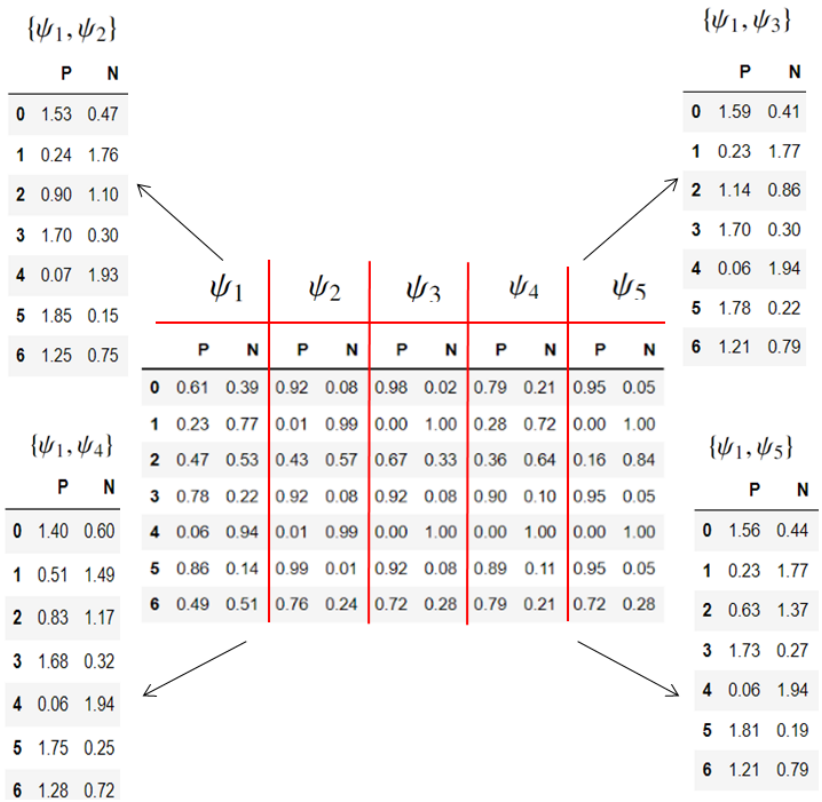
The effectiveness of the improved Forest-Based Tree (FBT) is evaluated and compared to the results of [9]. Two evaluation criteria are considered: the pruning time and the predictive performance. The pruning time is a part of the learning process and reflects how to generate FBT quickly. We measure the pruning time of our proposed method in seconds, and compare it with the pruning time of [9]. The predictive performance is assessed using ROC AUC measurement [33]. In addition, we report the generalized accuracy of FBT. The rest of all results are typically same as in [9], according to estimating the classification complexity, and relevant rule extraction.

4.1. Experimental settings

The experimental settings are as follows:

- Python is the language of implementation, using sci-kit-learn's random forest, with CARET trees as training inducers.
- Ensemble size is set to be 100, and the maximum depth of random forest's base trees is 5.
- RF-FBT is a Forest-based tree extracted from a random forest with a maximum allowed number of conjunctions per iteration as 3000, and the minimum size of the pruned forest is 10.
- The datasets are randomly divided as: 0.8 of the datasets is for the training stage and remaining sets are for the testing stage. This procedure is repeated 10 times and the average is reported for the test set.
- The results from [9] serve as the base benchmark to compare against.
- The pruning of random forest is performed on the same training data as the pruning set.
- Table 1 summarizes the characteristics of the selected datasets from UCI repository.
- The source code with the reported results is publicly available online.
  https://github.com/FatenKhalifa/Improved-version-of-explainable-decision-forest

- Finally, the aim from those experiments is to reduce the time of deriving a decision tree from a random forest without sacrificing the accuracy.



$\{\psi_1, \psi_2\}$

	P	N
0	1.53	0.47
1	0.24	1.76
2	0.90	1.10
3	1.70	0.30
4	0.07	1.93
5	1.85	0.15
6	1.25	0.75

$\{\psi_1, \psi_3\}$

	P	N
0	1.59	0.41
1	0.23	1.77
2	1.14	0.86
3	1.70	0.30
4	0.06	1.94
5	1.78	0.22
6	1.21	0.79

$\{\psi_1, \psi_4\}$

	P	N
0	1.40	0.60
1	0.51	1.49
2	0.83	1.17
3	1.68	0.32
4	0.06	1.94
5	1.75	0.25
6	1.28	0.72

	ψ_1		ψ_2		ψ_3		ψ_4		ψ_5	
	P	N	P	N	P	N	P	N	P	N
0	0.61	0.39	0.92	0.08	0.98	0.02	0.79	0.21	0.95	0.05
1	0.23	0.77	0.01	0.99	0.00	1.00	0.28	0.72	0.00	1.00
2	0.47	0.53	0.43	0.57	0.67	0.33	0.36	0.64	0.16	0.84
3	0.78	0.22	0.92	0.08	0.92	0.08	0.90	0.10	0.95	0.05
4	0.06	0.94	0.01	0.99	0.00	1.00	0.00	1.00	0.00	1.00
5	0.86	0.14	0.99	0.01	0.92	0.08	0.89	0.11	0.95	0.05
6	0.49	0.51	0.76	0.24	0.72	0.28	0.79	0.21	0.72	0.28

$\{\psi_1, \psi_5\}$

	P	N
0	1.56	0.44
1	0.23	1.77
2	0.63	1.37
3	1.73	0.27
4	0.06	1.94
5	1.81	0.19
6	1.21	0.79

Fig. 3. Using vectorized computing for fast and easy indexing

4.2. *Results*

The first part of our experiments is to show whether the implementation of the proposed method is accurate. The selected trees by our pruning method should be similar to those of the pruning method from [9]. Furthermore, the selected trees from both methods must be identical. In Fig. 4, we justify how the output of the proposed method, represented on the x-axis, is correlated with the output of [9] which is represented on the y-axis. According to several datasets: {AustCredit, Breast cancer, Car, Iris, Mamographic, Nurse}, the selected trees via our pruning method are linearly correlated with the previous work [9]. The slope of all lines is 1, the intercept is 0, and the correlation coefficient r = 1, which confirms the exact matching of the selected subset of trees to the previous work of [9]. Note that the minimum size of selected trees is 10 as we determined in experimental settings.

Next, we measure the rank correlation to determine how the outputs of both pruning methods are matching. If the orderings are similar, then the correlated rank is strong, positive, and high. Table 2 shows the average of Spearman correlation coefficient and Kendall correlation coefficient for five iterations per dataset. Results from Table 2 show a perfect positive rank correlation where larger x values always correspond to larger y values and those values appear in the same order.

The second part of our experiments is to measure the average pruning time, AUC ROC metric, and accuracy by both methods. Table 3 shows the significant improvement, for all datasets, by our proposed

method in terms of training time. For example, with the Iris dataset, the pruned forest consumed 0.54 seconds by our method in comparison with 40.28 seconds reported by [9]. Furthermore, for the Spambase dataset, we reduce the pruning time by 192% as a significant improvement over [9]. Finally, the reported AUC ROC measurement and the accuracy show equal predictive performance. Our conclusion from this part is that we are able to generate an explainable decision tree from a random forest within a diminished time without affecting the predictive power.

Table 1. Datasets characteristics. #catg - Number of categorical features, #Real - Number of continuous features

DataSet	#Catg	#Real	#Instances	#Labels	Majority label proportion
Abalone	0	8	4177	28	0.16
AustCredit	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
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

Finally, we analyze the performance of the proposed method with different ensemble sizes. Table 4 shows the dramatically reduced time by our method. For instance, with the car dataset, a random forest of size 400 can be pruned in 21.37 seconds via our method, while it consumes 4243 seconds by [9]. For all datasets and any ensemble size, our method shows a significant improvement over the previous work to build FBT quickly. This part proves that our proposed method is more scalable to prune forests of any size within less time.

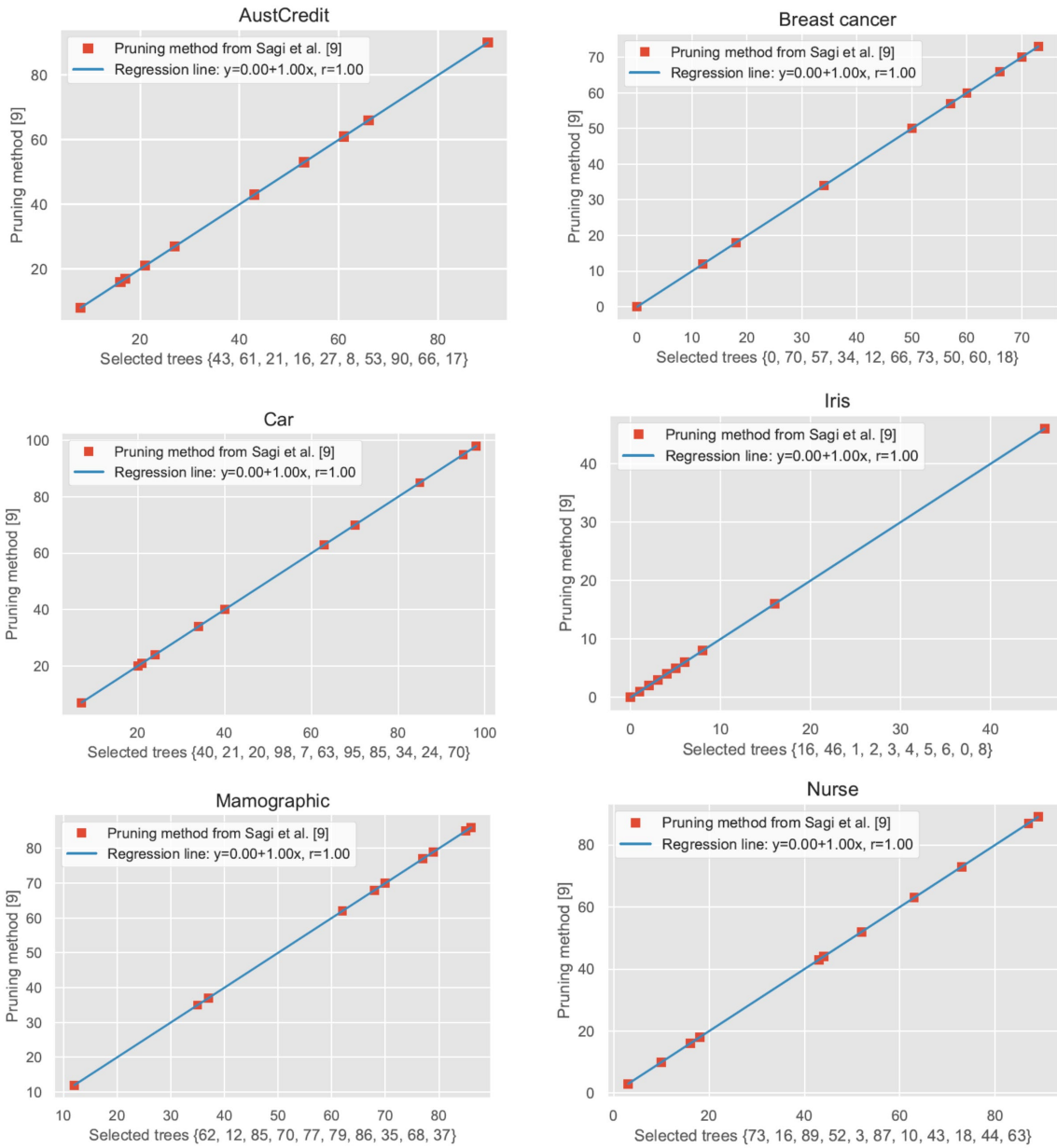


Fig. 4. The correlation between the outputs of proposed method and pruning method [9]

Table 2. Average of Spearman correlation coefficient and Kendall correlation coefficient, number of iterations= 5

DataSet	Spearman Correlation Coefficient	Kendall Correlation Coefficient
AustCredit	0.999975	0.99996
Breast cancer	0.99994	0.99994
Car	1	1
Iris	0.9999	0.9999
Mamographic	0.99998	0.99998
Nurse	0.99998	0.99998

Table 3. Average of consumed pruning time in seconds (s), ROC AUC, and Accuracy for FBT, number of iterations=10

DataSets	Pruning Time		AUC ROC		Accuracy	
	[9]	Proposed	From [9]	Proposed	From [9]	Proposed
Abalone	2075.66	**35.07**	89.07	89.07	23.12	23.12
AustCredit	295.65	**1.50**	92.46	92.46	86.96	86.96
Balance_scale	415.70	**1.87**	94.05	94.05	85.12	85.12
Bank	2472.52	**12.76**	93.65	93.65	88.65	88.65
Banknote	368.07	**2.08**	98.95	98.95	96.84	96.84
Biodeg	545.82	**2.34**	79.21	79.21	73.46	73.46
Breast cancer	191.34	**1.29**	99.18	99.18	95.62	95.62
Car	505.51	**2.84**	98.30	98.30	88.27	88.27
Credit	414.19	**1.66**	80.96	80.96	78.53	78.53
Cryotherapy	24.35	**0.47**	95.82	95.82	90.00	90.00
Ecoli	112.69	**1.11**	97.26	97.26	81.47	81.47
Forest	238.06	**1.41**	92.06	92.06	76.95	76.95
German	365.36	**1.92**	79.69	79.69	70.55	70.55
Glass	104.07	**1.01**	89.78	89.78	60.93	60.93
Haberman	98.07	**0.78**	81.16	81.16	73.55	73.55
Internet	84.43	**0.68**	100.0	100.0	100.0	100.0
Iris	40.28	**0.54**	98.89	98.89	94.67	94.67
Kohkiloyeh	27.87	**0.49**	93.69	93.69	85.00	85.00
Liver	352.25	**1.59**	77.25	77.25	69.91	69.91
Magic	864.90	**3.56**	89.66	89.66	83.80	83.80
Mamographic	277.94	**1.47**	90.44	90.44	83.07	83.07
Nurse	31.71	**0.57**	82.65	82.65	61.67	61.67
Occupancy	944.93	**4.17**	99.78	99.78	98.09	98.09
Pima	480.86	**1.94**	77.76	77.76	71.69	71.69
Seismic	1104.51	**5.24**	96.79	96.79	93.66	93.66
Spambase	2869.58	**14.89**	90.13	90.13	80.61	80.61
Tic-tac-toe	438.66	**2.40**	87.25	87.25	78.44	78.44
Vegas	366.38	**2.59**	79.63	79.63	46.73	46.73
Winery	61.31	**0.85**	98.25	98.25	92.78	92.78
Zoo	35.33	**0.77**	98.95	98.95	93.33	93.33

Table 4. Average pruning time, in seconds, with different ensemble sizes {100, 200, 400}, number of iterations= 10

DataSets	**Ensemble Size=100**		**Ensemble Size=200**		**Ensemble Size=400**	
	[9]	Proposed	[9]	Proposed	[9]	Proposed
AustCredit	295.65	**1.50**	1132.37	**5.34**	2535.49	**11.30**
Breast cancer	191.34	**1.29**	582.88	**3.59**	1453.66	**7.81**
Car	505.51	**2.84**	1739.72	**11.05**	4243.96	**21.37**
Iris	40.28	**0.54**	111.13	**1.56**	241.72	**3.25**
Mamographic	277.94	**1.47**	972.49	**5.66**	2542.21	**11.71**
Nurse	31.71	**0.57**	87.67	**1.90**	184.97	**2.80**

5. Conclusions and future work

This paper is an improvement for building a decision tree from a given decision forest. The formed tree reserves the predictive power of the original forest while being explainable. Regarding that, the derived ensemble tree can reveal intriguing relationships that have never been thought of before. This can assist in scientific discovery (e.g., medicine, insurance, justice, manufacturing industries, etc.). In literature, to cope with complex forests, ensemble pruning is employed to facilitate computations by reducing forest size. However, the pruning step could consume exorbitant time if not properly implemented. In this article, we propose an intelligent and fast pruning method based on indexing and vectorized computations. A greedy search method is applied to explore the efficiency of different tree subsets. The results show a noticeable reduction in time that reaches 200% of previous work calculated time. The proposed method is more scalable to prune forests of any size. Furthermore, the proposed method does not deviate from previous work in terms of accuracy. For future improvement, different ensemble pruning methods could be investigated to select a more representative sub-forest. In addition, other approaches for rule filtering can be analyzed to select more relevant and informative rules.

References

[1] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8*, e1249, 2018.

[2] B. Krawczyk, et al., "Ensemble learning for data stream analysis: A survey," *Information Fusion*, 37, pp. 132–156, 2017.

[3] Z.-H. Zhou, "Ensemble learning," *Machine learning, Springer*, Singapore, pp. 181–210, 2021.

[4] X. Dong, et al., "A survey on ensemble learning," *Frontiers of Computer Science*, 14, pp. 241–258, 2020.

[5] A. M. Mohammed, et al., "An analysis of heuristic metrics for classifier ensemble pruning based on ordered aggregation," *Pattern Recognition*, 124, p. 108493, 2022.

[6] M. Woz´niak, et al., "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, 16, pp. 3–17, 2014.

[7] L. Breiman, "Bagging predictors," *Machine learning*, 24, pp. 123–140, 1996.

[8] L. Breiman, "Random forests," *Machine learning*, 45, pp. 5–32, 2001.

[9] O. Sagi and L. Rokach, "Explainable decision forest: Transforming a decision forest into an interpretable tree," *Information Fusion*, 61, pp. 124–138, 2020.

[10] P. Domingos, "Knowledge discovery via multiple models," *Intelligent Data Analysis*, 2, pp. 187–202, 1998.

[11] G. Vandewiele, et al., "Genesim: genetic extraction of a single, interpretable model," *arXiv preprint arXiv*:1611.05722, 2016.

[12] S. M. Lundberg, et al., "From local explanations to global understanding with explainable AI for trees," *Nature machine intelligence*, 2, pp. 56–67, 2020.

[13] C. Meske, et al., "Explainable artificial intelligence: objectives, stakeholders, and future research opportunities," *Information Systems Management*, 39, pp. 53–63, 2022.

[14] A. B. Arrieta, et al., "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information fusion*, 58, pp. 82–115, 2020.

[15] R. Guidotti, et al., "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, 51, pp. 1–42, 2018.

[16] M. T. Ribeiro, S. Singh and C. Guestrin, ""Why should I trust you?" Explaining the predictions of any classifier," *Proc. 22nd ACM SIGKDD International conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[17] H. Lakkaraju, et al., "Interpretable & explorable approximations of black box models," *arXiv preprint arXiv*:1707.01154, 2017.

[18] F. Wang and C. Rudin, "Falling rule lists," *Artificial intelligence and statistics*, PMLR, pp. 1013–1022, 2015.

[19] L. Rokach, "Decision forest: Twenty years of research," *Information Fusion*, 27, pp. 111–125, 2016.

[20] A. Géron, "Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems," *O'Reilly Media, Inc*., 2019.

[21] C. Molnar, "Interpretable machine learning," *Lulu. com*, 2020.

[22] J. Fürnkranz, D. Gamberger and N. Lavracˇ, "Foundations of rule learning," *Springer Science & Business Media*, 2012.

[23] E. Angelino, et al., "Learning certifiably optimal rule lists for categorical data," *arXiv preprint arXiv*:1704.01701, 2017.

[24] Y. Freund, et al., "Experiments with a new boosting algorithm," *icml*, vol. 96, pp. 148–156, 1996.

[25] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of artificial intelligence research*, 11, pp. 169–198, 1999.

[26] M. Fernández-Delgado, et al., "Do we need hundreds of classifiers to solve real world classification problems?," *The journal of machine learning research*, 15, pp. 3133–3181, 2014.

[27] C. Bénard, et al., "Interpretable random forests via rule extraction," *International Conference on Artificial Intelligence and Statistics*, PMLR, pp. 937–945, 2021.

[28] R. R. Fernández, et al., "Random forest explainability using counterfactual sets," *Information Fusion*, 63, pp. 196–207, 2020.

[29] Gulowaty, Bogdan, and Michał Woźniak, "Extracting Interpretable Decision Tree Ensemble from Random Forest," *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2021.

[30] M. Zolanvari, et al., "Trust xai: Model-agnostic explanations for ai with a case study on iiot security," *IEEE Internet of Things Journal*, 2021.

[31] G. Srivastava, et al., "XAI for Cybersecurity: State of the Art, Challenges, Open Issues and Future Directions," *arXiv preprint arXiv*:2206.03585, 2022.

[32] S. Van Der Walt, et al., "The numpy array: a structure for efficient numerical computation," *Computing in science & engineering*, 13, pp. 22–30, 2011.

[33] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, 30, pp. 1145–1159, 1997.