eXplainable Random Forest

Guy Amit[1], Shlomit Gur[1]

[1]IBM Research, Haifa , Israel

Abstract

Advanced machine learning models have become widely adopted in various domains due to their exceptional performance. However, their complexity often renders them difficult to interpret, which can be a significant limitation in high-stakes decision-making scenarios where explainability is crucial. In this study we propose eXplainable Random Forest (XRF), an extension of the Random Forest model that takes into consideration, crucially, during training, explainability constraints stemming from the users' view of the problem and its feature space. While numerous methods have been suggested for explaining machine learning models, these methods often are suitable for use only after the model has been trained. Furthermore, the explanations provided by these methods may include features that are not human-understandable, which in turn may hinder the user's comprehension of the model's reasoning. Our proposed method addresses these two limitations. We apply our proposed method to six public benchmark datasets in a systematic way and demonstrate that XRF models manage to balance the trade-off between the models' performance and the users' explainability constraints.

Keywords

Explainability, Machine Learning, Random Forest,

1. Introduction

In recent years Artificial Intelligence (AI) has been widely adopted in many domains and aspects of lives [1], including healthcare [2], finance [3], games [4], and other complex cognitive tasks. This wide adoption can be attributed to improved performance of AI models, which has been made possible by increasingly complex, and therefore also increasingly difficult to explain, models [5, 6]. However, the goal of these AI models is often to aid or make decisions for humans, thus giving cause for concern, especially in domains with potentially severe individual or societal consequences. For example, the use of Machine Learning (ML) algorithms in healthcare to predict a medical outcome or a patient's diagnosis [7, 8, 9] may raise concerns regarding their reliability, trustworthiness, and ethics [10, 11], especially in terms of the patients' characteristics these models rely on.

These concerns have spurred public debates, which in turn gave rise to legislation (e.g., [12, 13, 14]) and high-volume research in the domain of eXplainable Artificial Intelligence (XAI) [15]. While many XAI approaches have been proposed, they are predominantly post-hoc. That is, they only deem the ML model explainable or not, after it is set (i.e., done training). If the ML model is deemed not explainable, there is often no known course of action to rectify it. One could try, for example, other ML models, modify hyper-parameters, or modify the training data,

HAII5.0: Embracing Human-Aware AI in Industry 5.0, at ECAI 2024, 19 October 2024, Santiago de Compostela, Spain.

guy.amit@ibm.com (G. Amit); shlomit.gur@ibm.com (S. Gur)
https://orcid.org/0000-0002-5987-4402 (G. Amit)

but none of these options is guided by any form of explainability. As such, these solutions could result in no more, if not less, explainable ML models. A few exceptions to this rule include data-type-specific (often unstructured data, such as images or text) methods for Deep Neural Networks (DNNs) [16] and Teaching Explanations for Decisions (TED) [17].

In this work we propose eXplainable Random Forest (XRF), a method for incorporating a form of user-defined explainability into the training stage of Random Forest (RF) models. The user-defined explainability in the current context is a global Feature Preference (FP) vector, which is provided as input to the model by the user. The preference is dependent on the user and can be based on, for example, human-understandability, political correctness, or actionability of the features. The proposed method can be viewed as a user-driven "soft" feature selection where, unlike traditional feature selection methods, all the features remain available for the ML model to use.

The objective of the proposed method is to foster a sound trade-off between the performance of the ML model and its explainability, as defined by the adherence of the model's Feature Importance (FI) values to the user-defined FP vector. Thus, we employed a systematic method that challenges RF's FI values. Our testing method encouraged the ML model to use k features with the lowest FI values and discouraged the use of k features with the greatest FI values. The rest of the features were treated with neutral preference. The generality of this method allowed us to test its applicability using multiple benchmark datasets in a way that is neither dataset- nor domain-specific. It is our belief that other studies could benefit from this type of systematic testing in the future.

Our results demonstrate that our method manages to balance the trade-off between the performance and explainability of the ML model. For example, if a feature does not hold great predictive power, a strong user-defined preference will not inflate its importance in the model at any cost to the model's performance. Alternatively, a high-importance feature with a negative user-defined preference can be substituted by another feature or a set of features, even if the model's performance is negatively impacted.

The main contributions of the current work are as follows:

1. We introduce XRF, an extension of RF that balances performance and user-defined explainability during training.
2. We propose an "eXplainability Score", a metric to quantify explainability and demonstrate on six benchmark datasets XRF's ability to balance the trade-off between the model's performance and its explainability, as measured by this metric.
3. We present a novel testing scheme to empirically evaluate the trade-off between performance and explainability in the absence of a human agent.

2. Background

2.1. eXplainable Artificial Intelligence

XAI refers to the ability to understand and interpret the decision-making process involving an AI model. It serves multiple purposes [18, 15], one of which is increasing trust and accountability in AI-based (thus also ML-based) systems. This is especially imperative in high-stakes applications with potentially severe individual or societal consequences (e.g., judicial, health-related, or financial decisions).

Some ML models are interpretable [18, 15], that is, their inner workings are inherently human-understandable (e.g., Decision Trees (DTs)). These models are often very simple and do not perform well on real-world complex tasks. Conversely, ML models that perform well on real-world tasks are usually complex and not interpretable. XAI methods can be used to make non-interpretable models explainable. Explainable ML models are models whose inner workings are not human-understandable, but can be explained using proxies (e.g., FI of surrogate interpretable models [19, 20]).

XAI methods for ML models can be categorized into local, global, and hybrid methods. Local explanation methods, such as Local Interpretable Model-Agnostic Explanations (LIME) [19] or SHapley Additive exPlanations (SHAP) [20], provide an explanation for a specific model prediction. These methods usually provide an explanation in the form of a list. This list can indicate, for example, the respective contribution of each of the features to the particular prediction or alternative values that would result in a different prediction (i.e., counterfactual explanation). In contrast, global XAI methods explain a ML model over the entire input space, as captured by the training data. Hybrid methods explain a ML model over sub-samples of the input space. In this paper we focus on adjusting the global explanation provided by XRF models to accommodate the user's FP.

2.2. Genetic Algorithms

Genetic Algorithms (GAs) [21, 22] are a family of gradient-free optimization algorithms inspired by the mechanics of natural selection and genetics. They are known for solving a variety of optimization and search problems [23, 24, 25, 26]. GAs encode the solution space of a problem as a set of individuals (i.e., a population) and then repeatedly apply genetic operators such as selection, crossover (recombination), and mutation in order to make the individuals (i.e., solutions) evolve with the end goal of finding a sufficient solution to the problem.

3. Related Work

Our work is another step in the ongoing quest to make AI more transparent and human-understandable. Some of the earliest XAI techniques include LIME [19] and SHAP [20]. These techniques are model-agnostic and propose feature-attribution-based explanations to a ML model using a surrogate ML model. While LIME uses a linear surrogate model, SHAP makes use of Shapley values [27], borrowed from the field of cooperative game theory. More recent papers introduced Bayesian aspects to these techniques [28, 29]. For example, the Local Interpretation-Driven Abstract Bayesian Network (LINDA-BN) method proposed using a Probabilistic Graphical Model to provide a local explanation. That said, these methods are applied post-hoc and, therefore, do not incorporate consideration of explainability in the training phase.

Several approaches have been proposed to improve a ML model's explainability during its training. One approach is to use the TED [17] framework, which predicts both a label and an explanation from a predefined set of explanations. This is done using a multi-label classifier, trained to predict the original label coupled with the explanation. TED is model-agnostic and can be applied to every model capable of performing multi-label classification, but it suffers from the following limitations: (1) the training set must include both labels and explanations, (2) no new explanations outside the predefined set can be "learned" from the data, and (3) there is nothing to prevent a predicted explanation from contradicting features in the explained data point.

An example of a model-specific approach [30] proposed a framework that trains an AdaBoost ensemble while enforcing the fairness of the model. In this approach the samples' weights, used for training each weak learner, are chosen while considering SHAP values. In each iteration, a surrogate model is fitted in addition to a weak learner and the SHAP values from the surrogate model are used together with the weak learner's error rate to update the training samples' weights. Another model-specific approach [31] comes from the domain of data privacy. In this field of study some of the dataset features are considered more sensitive and are encouraged to be used less, in order to protect against Membership Inference attacks [32]. This approach extends the DT's training process: in each feature split, a predefined weight is considered in addition to the entropy score. This training process yields a DT whose FI values are aligned with the privacy requirements that were defined by the user. Although this approach is meant for enforcing privacy requirements, it can also be used to improve a DT's explainability. However, the user has no way of balancing model performance against privacy requirements or explainability.

Finally, it is worth noting that there is a large body of work that focuses on DNNs [16, 33, 34]. In this work, however, we focus on DT ensemble models.

4. Explainable Random Forest

In this part of the paper we present our proposed model, the XRF. XRF is an adaptation of the RF model, which allows the user to influence the model's dependency on specific features in the training data. More accurately, given a FP vector, $FP \in \mathbb{R}^N$, the model is adjusted to use all the N features in the training data, while attempting to assign weights to features in accordance with FP.

Training an XRF model consists of two steps: (1) constructing the DTs (the FP vector does not affect this step) and (2) optimizing the Objective Function (OF) (equation 2) with respect to the FP vector. In the following subsections we will describe each of these steps.

4.1. Constructing the Decision Trees

As previously mentioned, XRF is an extension of RF. The RF model employs a bagging method, which means that each DT in the RF is constructed independently. During inference, all the DTs are used in a voting scheme to get a final prediction from the ensemble. In RF, each DT is given an equal weight and the prediction for a data point x is computed as follows:

$$\hat{y} = \underset{c \in C}{argmax}\{\sum_{i=1}^{k} w_i \cdot t_i(c|x)\} = \underset{c \in C}{argmax}\{\frac{1}{k} \cdot \sum_{i=1}^{k} t_i(c|x)\} \tag{1}$$

Where C is the set of possible classes, $\hat{y}$ is the predicted class, k is the number of DTs in the RF, t_i is the i^{th} DT, and w_i is its weight. Notice that RF assumes that $\sum_{i=1}^{k} w_i = 1$ and gives each

DT an equal vote. Therefore, in plain RF, $w_i = \frac{1}{k}$ for all $1 \ \leq i \ \leq k \in \mathbb{N}$.

4.2. Optimization Process

In this step the user-defined FP vector is used to influence the model's use of the features. During the optimization process the weights of the DTs, w_i for $1 \leq i \leq k \in \mathbb{N}$, are modified such that the FI values of the XRF adhere to the specifications in the FP vector as much as possible.

We optimize an OF that balances between the model's performance and a similarity measure between the model's FI values and the input FP vector. We refer to this similarity measure as "eXplainability Score" (XS) and denote it by XS. Formally, we define the OF of the XRF as follows:

$$OF(XRF, FP) = performance(XRF) + \alpha \cdot XS(FI, FP) \tag{2}$$

Where α is a hyper-parameter controlling the similarity metric's weight during the optimization process and FI is a vector of the FI values of the XRF model.

It is worth noting that the representation of the FP vector should match the choice of XS. For example: if XS cannot handle negative values or 0, then the FP vector should hold only positive values; and if XS expects probability distributions, then the values in the FP vector should sum up to 1.

Taking into consideration the FI values makes the OF a parametric non-differentiable function and, therefore, our method must use a gradient-free optimization algorithm, such as the GA used in our implementation [35]. The GA searches for the weights of the DTs, $\mathcal{W} = [w_1, \ldots, w_k]$, to maximize the value of the OF over a validation set. Formally, this can be written as:

$$\max_{\mathcal{W}} OF(XRF, FP) \tag{3}$$

To ensure that $\sum_{i=1}^{k} w_i = 1$, which is a requirement in plain RF, a normalization strategy is used in our proposed method. This normalization is applied each time the GA calculates the OF, as well as when the final weights for the DTs are returned by the optimization process.

5. Experimental Setting

Our evaluation consisted of compering the performance and the explainability of XRFs for different α values, using plain RFs as baselines. We performed the evaluation on six publicly available benchmark datasets of varying sizes (see Table 1) and domains: Yeast [1] [36], Adult Income [2] [37], German Credit [3] [38], Nursery [4] [39], Iris [5] [40], and Breast Cancer [6] [41]. Due

[1] https://archive.ics.uci.edu/ml/datasets/yeast

[2] http://archive.ics.uci.edu/ml/datasets/Adult

[3] http://archive.ics.uci.edu/ml/datasets/statlog+(german+credit +data)

[4] https://archive.ics.uci.edu/ml/datasets/nursery

[5] https://archive.ics.uci.edu/ml/datasets/iris

[6] https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin +diagnostic

Dataset	#Samples	#Features	#Labels
Yeast	1,299	8	4*
Adult Income	32,561	12	2
German Credit	1,000	20	2
Nursery	12,960	8	3*
Iris	150	4	3
Breast Cancer	569	30	2

Table 1
Description of the datasets used in the XRF evaluation. Due to the scarcity of some of the labels, we used only the most common labels for the German Credit and the Nursery datasets.

Dataset	#Trees	Max Depth	Max Samples	#Gens.	Mating Prob.	Mutation Prob.
Yeast	40	5	0.4	20	0.5	0.7
Adult Income	70	6	0.6	30	0.5	0.7
German Credit	70	6	0.8	120	0.5	0.4
Nursery	30	6	0.8	100	0.5	0.7
Iris	10	2	0.2	10	0.1	0.1
Breast Cancer	40	3	0.6	20	0.5	0.4

Table 2
Values of hyper-parameters per dataset based on grid searches. The number of DTs (#Trees), max depth, and max samples (per tree) were searched on RFs. Their results were set and the number of generations (#Gens), mating probability and mutation probability (prob.) were then searched on XRF with $\alpha = 0.0$.

to the scarcity of some of the labels in the German Credit and Nursery datasets, we used only their top four [7] and top three [8] labels, respectively. These datasets are popular, also in related work [42, 31, 43, 44]. Additionally, the sizes of their feature spaces allow us to study, as well as comprehensively visualize, the method's performance (e.g., the change in feature attribution as a function of performance-vs-explainability trade-off).

The XRF relies on the user to specify a FP vector. However, datasets with user-defined FP vectors, as well as expected explanations, are not readily available. Moreover, human preferences and feedback are subjective and could vary greatly, possibly adding complexity that would detract from the methodological focus of the current work. Therefore, we chose to not involve humans in the evaluation. In the absence of human involvement in the evaluation process, we developed an automated testing procedure to challenge the XRF's ability to affect the FI values. Inverting the FI values of a trained model can be a complicated task, since often the signal supporting the model prediction is embedded within the features with the greatest FI values. Thus, to test the effect of XRF's OF, we employed the following testing scheme:

1. We trained a RF model on a dataset $\mathcal{D}$
2. We extracted the FI values of $\mathcal{D}$'s features for that RF
3. Finally, we set the FP vector such that the k top-importance and k bottom-importance

[7] *CYT*, *NUC*, *MIT*, and *ME3*
[8] *not_recom*, *priority*, and *spec_prior*

features are penalized and rewarded, respectively. The rest of the features were assigned neutral preference (default). We used $k = 1$ if $|\mathcal{D}| \leq 5$ and $k = 2$ if $|\mathcal{D}| > 5$, where $|\mathcal{D}|$ is the number of features in $\mathcal{D}$.

For XS we used the cosine similarity measure:

$$XS\,(FI, FP) = \frac{FI \cdot FP}{\|FI\| \cdot \|FP\|} \tag{4}$$

As the performance of XRF is bounded between 0 and 1 (for both accuracy and F_1 score), it is more convenient if XS is bounded too (as is the case with cosine similarity: $-1 < XS < 1$) for ease of choice of the hyper-parameter α (see equation 2). As previously mentioned, the representation of the FP vector should match the choice of XS. For simplicity, we considered FP values, $FP_i \in \mathbb{R}$, such that (1) $FP_i > 0$ indicates that the i^{th} feature should be preferred by the model, (2) $FP_i < 0$ indicates that the model should avoid it as much as possible, and (3) $FP_i = 0$ indicates that the model may use it as needed (default; neutral preference).

For example: let $\mathcal{D}$ be a dataset with four features ($|\mathcal{D}| = 4$). Then for a RF model on $\mathcal{D}$ with FI values of $[0.15, 0.3, 0.2, 0.35]$, the first and last features are the least (0.15) and most (0.35) important features in the RF, respectively. Thus, for the computation of XS in the XRF's optimization process, the FP vector would be set to: $[1, 0, 0, -1]$.

For the optimization step, we employed in our implementation a simple version of the GA from the DEAP framework [9]. We used CxOnePoint crossover operator, a tournament selection, and an additive Gaussian noise as a mutation operator. We chose softmax to normalize the weights:

$$\hat{w}_i = \frac{e^{w_i}}{\sum_{j=1}^{k} e^{w_j}} \tag{5}$$

We performed in addition an ablation study, examining the effect of a different normalization strategy and a different XS.

In all of our experiments we performed a grid search over the hyper-parameters of the plain RF and the GA (searched per dataset [10]). To reduce complexity, we split the grid search into two steps: (1) RF-related features (i.e., number of trees $\in [10, 20, ..., 120]$, max depth $\in [1, 2, ..., 7]$, and max samples $\in [0.2, 0.4, ..., 1]$) were first searched using a plain RF and then (2) fixing these values, the GA-related features (i.e., number of generations $\in [10, 20, ..., 120]$, mating probability $\in [0.1, 0.3, 0.5]$, and mutation probability $\in [0.1, 0.4, 0.7]$) were searched using an XRF with $\alpha = 0$. This split also ensures that XRF results could be fairly compared to RF results. Table 2 summarizes the results of these grid searches. We then used these hyper-parameters to evaluate three XRF models for three different values of α, the hyper-parameter controlling the XS's weight in the XRF's OF. To measure the prediction performance of the model, we used the F_1 score and accuracy metrics, while to measure the adherence to the FP vector, we used the XS. Notice that performance-wise, the model was trained in respect to its accuracy. We ran these experiments ten times per dataset, using a different random seed in each run.

[9] https://deap.readthedocs.io/en/master/

[10] random seed $= 42$

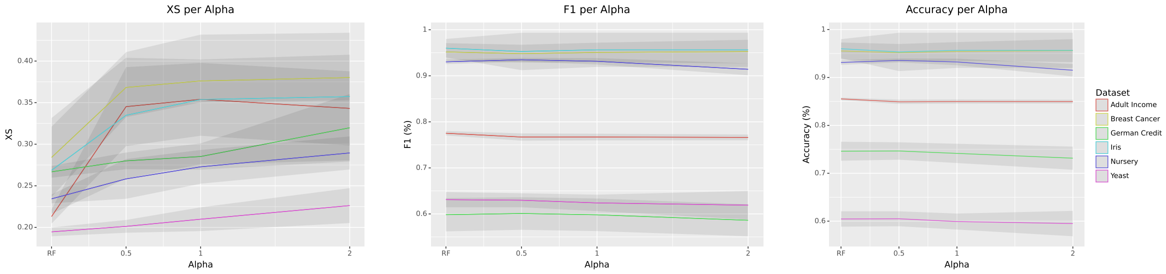

Figure 1: Performance comparison of models with different α vealues, per dataset.

6. Evaluation

In this section we present the results of our evaluations. The XRF optimization process balances between the task performance (e.g., accuracy or F_1 score) and the XS based on the choice of α. Therefore, we started out by testing how α affects the XRF model's performance and explainability. To this end we computed three metrics for the resulting XRF model: (1) accuracy, (2) F_1 score, and (3) XS. We performed this evaluation on six public benchmark datasets (see Table 1) for three different α values (0.5, 1.0, and 2.0; see Figure 1).

The classification performance metrics (i.e., accuracy and F_1) were compared to a baseline model: a plain RF classifier that was trained using comparable configuration (number of trees, max depth, and max samples) as the respective XRF models (see RF in Figure 1). Our results demonstrate that for most of the datasets the classification performance metrics decreased gradually, compered to the respective baselines models, as α increased. Intriguingly, for high-dimensional datasets, such as the Breast Cancer and German Credit datasets, this decrease was less pronounced. This can be attributed to the optimization process employed in the XRF algorithm, which is responsible for increasing the similarity between the FI values and FP vector. As the number of features in the dataset increases, the space of possible solutions widens, allowing for the discovery of solutions that strike a better balance between explainability and performance (i.e., reduce the performance less). Conversely, as desired, for most of the datasets XS gradually increased as α increased. For some datasets (e.g., Nursery) we observed an increase in the performance metrics of the XRF model with $\alpha = 0.5$, as compared to the baseline RF model. This might be attributed to the optimization step in XRF, which searches for weights to be assigned to the DTs in the ensemble such that the OF (equation 2) is maximal, as for $\alpha < 1$ the OF assigns greater importance to the performance term than to the XS term.

To further evaluate the effects of the OF on the FI values, we plotted, per dataset, the FI value of each feature as a function of α (Figure 2). Solid blue and magenta lines in the plots indicate features that were set (in the FP vectors) to be rewarded and penalized, respectively. In some datasets (e.g., Iris and Nursery) we saw a clear exchange in the FI values between penalized and rewarded features, while in others (e.g., Adult Income) we saw a clear exchange between penalized and neutral features. Finally, we observed in some datasets (e.g., Adult Income and Yeast) that the FI values of rewarded features remained fairly unchanged, suggesting that they were not found sufficiently useful by the model. Overall, these results lend support to our claim that the XRF's optimization process indeed shifts the FI values of rewarded and penalized

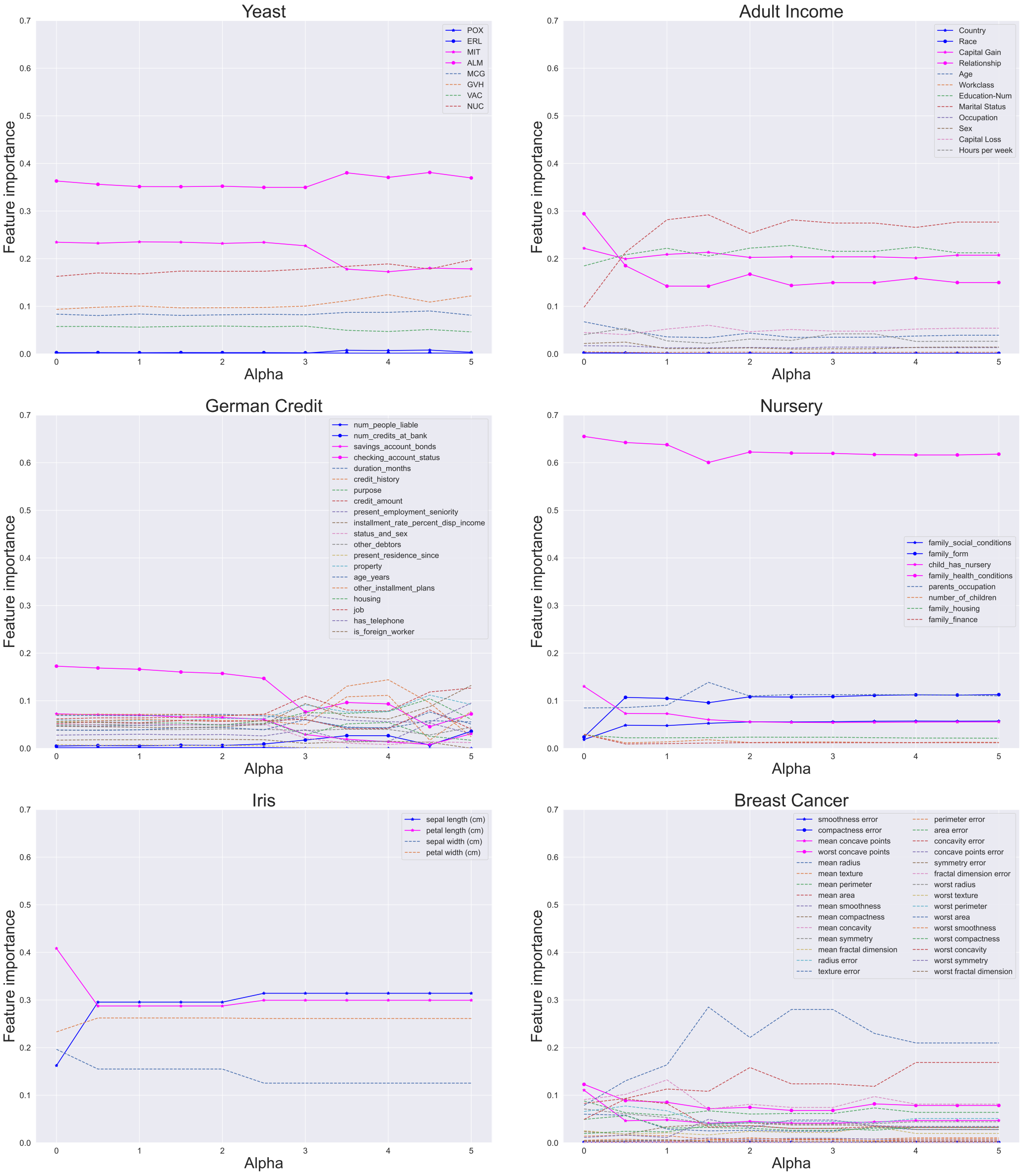

Figure 2: The effect of α on the FI values. In solid blue and magenta are the features to be rewarded and penalized, respectively. In dotted lines are the neutral features.

features in the intended direction when possible, but does not impose the FP at any cost to the XRF's performance.

6.1. Ablation Study

In the current work we report the results for a particular choice of XS (cosine similarity) and normalization strategy (softmax). However, the proposed XRF model does not limit the user

Config.	Metric	XRF$_{0.5}$	XRF$_{1.0}$	XRF$_{2.0}$
Abs. + CS	Acc.	0.6046 ± 0.0206	0.6015 ± 0.0180	0.5942 ± 0.0189
	F_1	0.6310 ± 0.0221	0.6273 ± 0.0205	0.6171 ± 0.0225
	XS	0.2062 ± 0.0095	0.2139 ± 0.0114	0.2230 ± 0.0135
Abs. + CE	Acc.	0.6062 ± 0.0137	0.6092 ± 0.0122	0.6042 ± 0.0177
	F_1	0.6319 ± 0.0152	0.6351 ± 0.0118	0.6311 ± 0.0188
	XS	0.1405 ± 0.0056	0.1431 ± 0.0062	0.1472 ± 0.0058
SM + CS	Acc.	0.6050 ± 0.0156	0.5992 ± 0.0167	0.5950 ± 0.0265
	F_1	0.6297 ± 0.0152	0.6239 ± 0.0179	0.6190 ± 0.0310
	XS	0.2012 ± 0.0077	0.2098 ± 0.0143	0.2262 ± 0.0210
SM + CE	Acc.	0.6042 ± 0.0121	0.6038 ± 0.0118	0.6035 ± 0.0164
	F_1	0.6314 ± 0.0142	0.6282 ± 0.0114	0.6286 ± 0.0173
	XS	0.1403 ± 0.0054	0.1430 ± 0.0063	0.1468 ± 0.0065

Table 3

Performance comparison of XRFs with different configurations for the modified Yeast dataset. Results are presented in Mean±SD form based on ten runs, each with a different random seed. Abs. = Absolute, CS = Cosine Similarity, SM = Softmax, Acc. = Accuracy, SD = Standard Deviation, XRF$_k$ = XRF with $\alpha = k$.

to this configuration and the user can choose to use other similarity metrics for XS and other normalization strategies for the DTs' weights. In this part of the paper we consider XRF models that use alternative configurations.

During these experiments we considered defining the input FP vector as a probability distribution ($FP_i \in [0, 1]$ and $\sum_{i=1}^{N} FP_i = 1$). Under this definition, the model's usage of features with high or low probabilities in the FP vector will be encouraged or discouraged, respectively, by the optimization process. Cross-Entropy (CE) is a well-defined distance metric between two probability distributions:

$$CE(P, Q) = -\sum_{i=1}^{m} P_i \cdot \log(Q_i) \tag{6}$$

Where Q and P are two probability distributions over the same m variables in corresponding order. Thus, we used for XS a CE -based similarity measure. Given the FP vector and the FI values, which satisfy the conditions for probability distributions, we define the following similarity measure:

$$XS\left(FI, FP\right) = \frac{1}{CE\left(FP, FI\right) + 1} \tag{7}$$

For the weights normalization strategy we used absolute value normalization:

$$\hat{w}_i = \frac{|w_i|}{\sum_{i=1}^{k} |w_i|} \tag{8}$$

Where w_i for $1 \leq i \leq k \in \mathbb{N}$ is the weight of the i^{th} DT in the XRF model.

Finally, in these experiments we considered only the modified Yeast dataset (with four labels after removing relatively scarce labels) and performed a grid search over the hyper-parameters for each of the combinations of normalization strategy and XS. As these choices (of normalization strategy and XS) do not affect the performance of RF, there was no need to repeat the grid search over RF-related hyper-parameters (i.e., number of trees, max depth, and max samples). Thus, we performed grid searches over the hyper-parameters that are related to the optimization process (the GA's hyper-parameters: number of generations, mating probability, and mutation probability) using XRFs with $\alpha = 0$. The results of these grid searches were identical to the results using the cosine similarity measure and the softmax normalization strategy (see top row in Table 2). Then, similarly to our main experimental setting, we evaluated the performance of three XRF models with $\alpha \in [0.5, 1.0, 2.0]$ based on ten runs, using a different random seed in each run. Overall, our results (see Table 3) suggest that the configuration we chose to use in the main experimental setting (cosine similarity for XS and the softmax normalization strategy) is either comparable to or better than the alternatives considered here, according to our metrics.

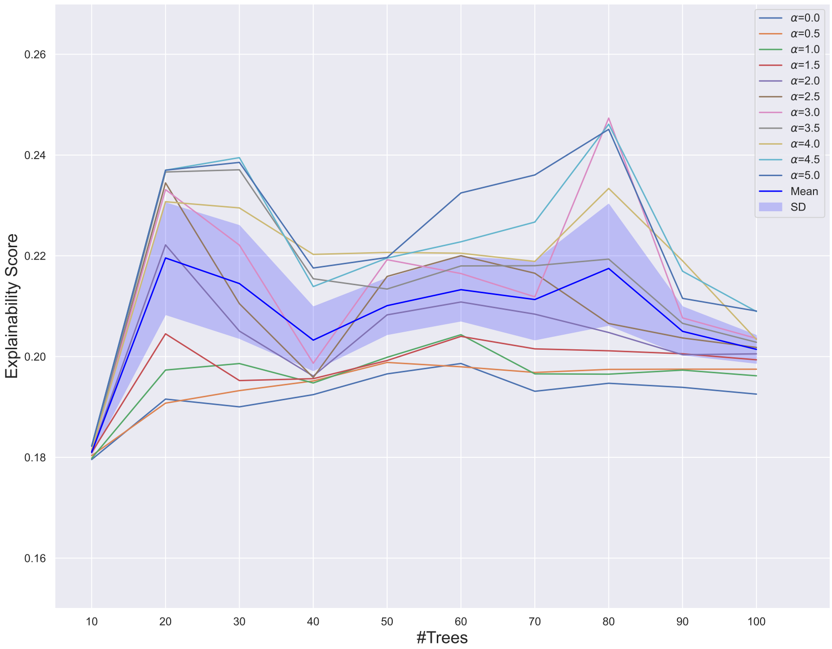

Figure 3: The effect of increasing the number of DTs in the XRF ensemble on the XS.

Another important hyper-parameter of the XRF model, with the potential to considerably affect the model's performance, is the number of DTs in the ensemble. In the main experimental setting we selected the number of DTs based on a grid search on a plain RF. That is, the number of DTs was determined based on performance only and without optimization. As we aim to create a robust method, we examined how the number of DTs in the XRF ensemble affects the XS. To this end we trained multiple XRF models on the modified Yeast dataset (same variation as above), each with a different number of DTs. We set the rest of the hyper-parameters, as well as the XS and normalization strategy, in accordance with the main experimental setting (see top row in Table 2). We repeated this experiment with 11 α values for every number of DTs (see Figure 3). Our results did not reveal a clear trend in XS when plotted against the number of DTs, thereby suggesting that the XS is relatively unaffected by the number of DTs in the XRF.

Thus, we conclude that the number of DTs in the XRF is a hyper-parameter that can be left to the user to set, either manually or by using an automatic grid search.

It is worth noting that these results could be due to a limitation in our experimental setting. As we chose to use low-dimensional datasets (30 features at most), the number of significantly unique DTs for them is capped. In turn, it is possible that in higher-dimensional datasets the number of DTs will affect the XS.

7. Conclusion

In this paper we introduce XRF, a novel family of DT ensemble models that optimize a balance between a performance metric (e.g., accuracy or F_1 score) and a user-defined form of explainability during training. In this context, the explainability is defined as adherence of the model's feature attribution (i.e., FI values) to a user-defined FP vector. Through this vector the user may define which features they prefer the model to use, which features they prefer the model to avoid, and which features they are neutral about. Additionally, our method allows the user to determine the desired balance between the model's explainability and its predictive performance. That is, the user can determine to what degree the model should prioritize explainability over performance (or performance over explainability). In contrast to our proposed method, existing XAI methodologies for structured data are primarily either post-hoc [19, 20] or limited to a predefined set [17], thereby limiting the insight one could gain from the ML model.

The results of our experiments demonstrate that the FI values in the XRF model are affected as expected by the FP vector, and in a controlled manner. For example, the FI of a rewarded feature is increased only if either (1) the feature provides information that could be used for prediction by the model or (2) explainability is highly prioritized over performance ($1 << \alpha$). Similarly, the FI of a penalized feature is decreased only if either (1) the predictive signal it provides to the model can be replaced by another feature or features, or (2) explainability is highly prioritized over performance ($1 << \alpha$). Finally, our results also suggest that the XS of the XRF model is relatively unaffected by the number of DTs in the model. Although this should be reexamined in higher-dimensional settings.

We hope that this work serves as a building block towards the development of more ML models that balance the trade-off between performance and explainability during training in an informed manner. As such, we recognize some calculated limitations of the current study, including the low-dimensionality of the datasets and the time complexity introduced by the GA algorithm. These will need to be addressed in future work. Additional future research directions may include (1) extending models that are more complex than RF, (2) developing other forms of human-defined or metric-derived explainability, and (3) developing other explainability quantification metrics.

Acknowledgments

This project has received funding from the European Union's Horizon research and innovation programme under grant agreements no 101094905 (AI4GOV) and 101092639 (FAME).

References

[1] S. Laato, M. Tiainen, A. Najmul Islam, M. Mäntymäki, How to explain ai systems to end users: a systematic literature review and research agenda, Internet Research 32 (2022) 1–31.

[2] K. Shailaja, B. Seetharamulu, M. Jabbar, Machine learning in healthcare: A review, in: 2018 Second international conference on electronics, communication and aerospace technology (ICECA), IEEE, 2018, pp. 910–914.

[3] A. Mashrur, W. Luo, N. A. Zaidi, A. Robles-Kelly, Machine learning for financial risk management: a survey, IEEE Access 8 (2020) 203203–203223.

[4] S. Singh, A. Okun, A. Jackson, Learning to play go from scratch, Nature 550 (2017) 336–337.

[5] F. K. Došilović, M. Brčić, N. Hlupić, Explainable artificial intelligence: A survey, in: 2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO), IEEE, 2018, pp. 0210–0215.

[6] W. J. von Eschenbach, Transparency and the black box problem: Why we do not trust ai, Philosophy & Technology 34 (2021) 1607–1622.

[7] O. Dogan, S. Tiwari, M. Jabbar, S. Guggari, A systematic review on ai/ml approaches against covid-19 outbreak, Complex & Intelligent Systems 7 (2021) 2655–2678.

[8] A. Wong, E. Otles, J. P. Donnelly, A. Krumm, J. McCullough, O. DeTroyer-Cooley, J. Pestrue, M. Phillips, J. Konye, C. Penoza, et al., External validation of a widely implemented proprietary sepsis prediction model in hospitalized patients, JAMA Internal Medicine 181 (2021) 1065–1070.

[9] K. Zhang, X. Liu, J. Shen, Z. Li, Y. Sang, X. Wu, Y. Zha, W. Liang, C. Wang, K. Wang, et al., Clinically applicable ai system for accurate diagnosis, quantitative measurements, and prognosis of covid-19 pneumonia using computed tomography, Cell 181 (2020) 1423–1433.

[10] I. Y. Chen, E. Pierson, S. Rose, S. Joshi, K. Ferryman, M. Ghassemi, Ethical machine learning in healthcare, Annual review of biomedical data science 4 (2021) 123–144.

[11] S. T. Jan, V. Ishakian, V. Muthusamy, Ai trust in business processes: The need for process-aware explanations, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020) 13403–13404.

[12] GDPR, General data protection regulation, 2018. https://gdpr-info.eu/.

[13] CCPA, The california consumer privacy act, 2018. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375.

[14] E. Commission, et al., White paper on artificial intelligence. a european approach to excellence and trust, COM (2020) 65 (2020).

[15] A. Adadi, M. Berrada, Peeking inside the black-box: a survey on explainable artificial intelligence (xai), IEEE access 6 (2018) 52138–52160.

[16] G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg, S.-I. Lee, Improving performance of deep learning models with axiomatic attribution priors and expected gradients, Nature machine intelligence 3 (2021) 620–631.

[17] M. Hind, D. Wei, M. Campbell, N. C. Codella, A. Dhurandhar, A. Mojsilović, K. Natesan Ramamurthy, K. R. Varshney, Ted: Teaching ai to explain its decisions, in: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 123–129.

[18] C. Meske, E. Bunde, J. Schneider, M. Gersch, Explainable artificial intelligence: objectives,

stakeholders, and future research opportunities, Information Systems Management 39 (2022) 53–63.

[19] 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, 2016, pp. 1135–1144.

[20] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems 30 (2017).

[21] M. Mitchell, An introduction to genetic algorithms, MIT press, 1998.

[22] K. De Jong, Learning with genetic algorithms: An overview, Machine learning 3 (1988) 121–138.

[23] R. Leardi, Genetic algorithms in chemistry, Journal of Chromatography A 1158 (2007) 226–233.

[24] A. E. Drake, R. E. Marks, Genetic algorithms in economics and finance: Forecasting stock market prices and foreign exchange—a review, Genetic algorithms and genetic programming in computational finance (2002) 29–54.

[25] S. H. Mousavi-Avval, S. Rafiee, M. Sharifi, S. Hosseinpour, B. Notarnicola, G. Tassielli, P. A. Renzulli, Application of multi-objective genetic algorithms for optimization of energy, economics and environmental life cycle assessment in oilseed production, Journal of Cleaner Production 140 (2017) 804–815.

[26] H. A. Saleh, R. Chelouah, The design of the global navigation satellite system surveying networks using genetic algorithms, Engineering Applications of Artificial Intelligence 17 (2004) 111–122.

[27] L. S. Shapley, Notes on the n-person game—ii: The value of an n-person game.(1951), Lloyd S Shapley 7 (1951).

[28] C. Moreira, Y.-L. Chou, M. Velmurugan, C. Ouyang, R. Sindhgatta, P. Bruza, Linda-bn: An interpretable probabilistic approach for demystifying black-box predictive models, Decision Support Systems 150 (2021) 113561.

[29] X. Zhao, W. Huang, X. Huang, V. Robu, D. Flynn, Baylime: Bayesian local interpretable model-agnostic explanations, in: Uncertainty in Artificial Intelligence, PMLR, 2021, pp. 887–896.

[30] J. M. Hickey, P. G. Di Stefano, V. Vasileiou, Fairness by explicability and adversarial shap learning, in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III, Springer, 2021, pp. 174–190.

[31] A. Goldsteen, G. Ezov, A. Farkash, Reducing risk of model inversion using privacy-guided training, arXiv preprint arXiv:2006.15877 (2020).

[32] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, in: 2017 IEEE symposium on security and privacy (SP), IEEE, 2017, pp. 3–18.

[33] P. Angelov, E. Soares, Towards explainable deep neural networks (xdnn), Neural Networks 130 (2020) 185–194.

[34] X. Bai, X. Wang, X. Liu, Q. Liu, J. Song, N. Sebe, B. Kim, Explainable deep learning for efficient and robust pattern recognition: A survey of recent developments, Pattern Recognition 120 (2021) 108102.

[35] F. Busetti, Genetic algorithms overview, Retrieved on December 1 (2007).

[36] K. Nakai, Yeast, UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5KG68.

[37] B. Becker, R. Kohavi, Adult, UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[38] H. Hofmann, Statlog (German Credit Data), UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[39] V. Rajkovic, Nursery, UCI Machine Learning Repository, 1997. DOI: https://doi.org/10.24432/C5P88W.

[40] R. A. Fisher, Iris, UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C56C76.

[41] W. Wolberg, O. Mangasarian, N. Street, W. Street, Breast Cancer Wisconsin (Diagnostic), UCI Machine Learning Repository, 1995. DOI: https://doi.org/10.24432/C5DW2B.

[42] C. Agarwal, S. Krishna, E. Saxena, M. Pawelczyk, N. Johnson, I. Puri, M. Zitnik, H. Lakkaraju, OpenXAI: Towards a transparent evaluation of model explanations, in: Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2022. URL: https://openreview.net/forum?id=MU2495w47rz.

[43] M. T. Keane, B. Smyth, Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable ai (xai), in: Case-Based Reasoning Research and Development: 28th International Conference, ICCBR 2020, Salamanca, Spain, June 8–12, 2020, Proceedings 28, Springer, 2020, pp. 163–178.

[44] O. Sagi, L. Rokach, Explainable decision forest: Transforming a decision forest into an interpretable tree, Information Fusion 61 (2020) 124–138.