UNDER REVIEW

RESCIENCE C

Replication / ML Reproducibility Challenge 2022

[Re] FOCUS: Flexible Optimizable Counterfactual Explanations for Tree Ensembles

Kyosuke Morita[1, ID]

[1]Heidelberg University, Heidelberg, Germany

Edited by
(Editor)

Reviewed by
Anonymous Reviewers

Received
–

Published
–

DOI
–

Reproducibility Summary

Scope of Reproducibility — This study aims to reproduce the results of the paper 'FOCUS: Flexible Optimizable Counterfactual Explanations for Tree Ensembles' by Lucic et al.[1]. The main claims of the original paper are that FOCUS is able to (i) generate counterfactual explanations for all the instances in a dataset; and (ii) find counterfactual explanations that are closer to the original input for tree-based algorithms than existing methods.

Methodology — This study replicates the original experiments using the code, data, and models provided by the authors. Additionally, this study re-implements code and retrains the models to evaluate the robustness and generality of FOCUS. All the experiments were conducted on a personal laptop with a quad-core CPU with 8GB of RAM and it approximately took 33 hours in total.

Results — This study was able to replicate the results of the original paper in terms of finding counterfactual explanations for all instances in datasets. Additional experiments were conducted to validate the robustness and generality of the conclusion. While there were slight deviations in terms of generating smaller mean distances, half of the models still outperformed the results of the existing method.

What was easy — The implementation of the original paper is publicly available on GitHub. The repository contains the models and data used in the original experiments. Also, the authors provided a technical appendix, which includes all hyperparameters that were used for the experiments for reproduction upon request.

What was difficult — Although the implementation code was available, it employs outdated packages and the code structure is complex. Also, the comments in the functions and the documentation of the code are sparse or nonexistent, which made it difficult to follow the code.

Copyright © 2023 K. Morita, released under a Creative Commons Attribution 4.0 International license.
Correspondence should be addressed to Kyosuke Morita (kyosuke1029@icloud.com)
The authors have declared that no competing interests exist.
Code is available at https://github.com/kyosek/focus-reproducibility – DOI 10.5281/zenodo.7931344. – SWH swh:1:dir:e096a518285f9ee2f9ee2c5943293ba30f7e17b0.
Data is available at https://github.com/a-lucic/focus.
Open peer review is available at https://openreview.net/forum?id=n1q-iz83S5&noteId=60kzDmcWau.

Communication with original authors — I reached out to the authors to obtain the hyperparameters used in the experiments. The authors responded promptly with a detailed technical appendix of the original paper.

1 Introduction

The importance of interpretability in machine learning models is growing as they are increasingly being applied in real-world scenarios. Understanding how models make decisions not only benefits the users of the model, but also those who are affected by the decisions made by the model. Counterfactual explanations have been developed to cope with this issue, as they allow individuals to understand how they would achieve a desirable outcome with minimal changes to their original data. Lucic et al.[1] proposed a method called FOCUS, which is designed to generate optimal distance counterfactual explanations to the original data for all the instances in tree-based machine learning models. This study aims to reproduce and evaluate their findings, as well as conduct additional experiments.

2 Scope of reproducibility

The generation of counterfactual explanations is a problem that has been addressed by several existing methods. Wachter, Mittelstadt, and Russell [2] formulated this problem into an optimisation framework, however, this approach is limited to differentiable models. The original paper aimed to extend the framework to non-differentiable models, specifically tree-based algorithms, by introducing a probabilistic model approximation. A crucial aspect of this method is the approximation of a pretrained tree-based model, represented as f, achieved by replacing each split in each tree with a sigmoid function with a parameter σ that is defined as:

$$sig(z) = (1 + exp(\sigma \cdot z))^{-1}, \tag{1}$$

where $\sigma \in \mathbb{R}_{>0}$. This sigmoid function is incorporated into the function $\tilde{t}_j(x)$ that approximates the node j activation $t_j(x)$ of the tree-based model f for a given input x. This function is defined as:

$$\tilde{t}_j(x) = \begin{cases} 1, & \text{if } j \text{ is the root,} \\ \tilde{t}_{p_j}(x) \cdot sig(\theta_j - x_{f_j}), & \text{if } j \text{ is left child,} \\ \tilde{t}_{p_j}(x) \cdot sig(x_{f_j} - \theta_j), & \text{if } j \text{ is right child,} \end{cases} \tag{2}$$

where θ_j is a threshold for activation of node j.
This method approximates a single decision tree $\mathcal{T}$. A tree approximation can be defined as:

$$\tilde{\mathcal{T}}(y|x) = \sum_{j \in \mathcal{T}_{leaf}} \tilde{t}_j(x) \cdot \mathcal{T}(y|j). \tag{3}$$

Additionally, this method replaces the maximum operation of f, which is an ensemble of M many trees with weights $\omega_m \in \mathbb{R}$ by a softmax function with temperature $\tau \in \mathbb{R}_{>0}$. Thus, the approximation $\tilde{f}$ can be expressed as:

$$\tilde{f}(y|x) = \frac{exp(\tau \cdot \sum_{m=1}^{M} \omega_m \cdot \tilde{\mathcal{T}}_m(y|x))}{\sum_{y'} exp(\tau \cdot \sum_{m=1}^{M} \omega \cdot \tilde{\mathcal{T}}_m(y'|x)} \tag{4}$$

It is important to note that this approximation method can be applied to any tree-based model.
The main claims of the original paper are that FOCUS is able to:

- generate counterfactual explanations for all instances in a dataset - *Reliability*.
- find counterfactual explanations that are closer to the original input for tree-based algorithms than existing frameworks - *Effectiveness*.

3 Methodology

This study uses the code, data, and models provided by the original authors to reproduce their original experiments. In addition, to evaluate the robustness and generality of FOCUS, several modifications were made to the original implementation. These modifications include: (i) updating the versions of Tensorflow from 1.14.0 to 2.11.0 and scikit-learn from 0.21.3 to 1.0.2, (ii) reorganising the code by removing redundant functions and simplifying the code structure and (iii) adding unit tests. Furthermore, this study conducts an additional experiments on "German credit" dataset [3].

3.1 Model descriptions

The pretrained models include Decision Tree (DT), Random Forest (RF), and Adaptive Boosting (AB) with DT as a base learner. In addition, this study retrained all models. The sets of employed hyperparameters are reported in Table 6 in Appendix A. In the cases where hyperparameters were not specified, the default values were used. The accuracy of the retrained models is reported in Table 8 in Appendix C.

3.2 Datasets

The four binary classification datasets used in the original experiments are:

- *Wine Quality* [4] - This dataset contains 4,898 data points with 11 features. The original dataset presents the wine quality on a scale of 0-10, but the original authors modified it into binary classification. The modified dataset adapts a "high quality" wine if the quality is higher than or equal to 7. There are 1,060 positive class data (22%).
- *HELOC* [5] - This dataset contains 10,459 data points with 23 features. There are 5,000 positive class data (48%).
- *COMPAS* [6] - This dataset contains 6,172 data points with 6 features. There are 2,990 positive class data(48%).
- *Shopping* [7] - This dataset contains 12,330 data points with 9 features. There are 1,908 positive class data (15%).

The original paper states that all features in the datasets were transformed into the range of 0 and 1, and all categorical features were removed. These datasets were pre-processed by the original authors. In addition to those datasets, this study employs the German credit dataset to test the generality of FOCUS. This German Credit dataset aims to classify individuals into two categories, those with good credit risk and those with bad credit risk. It contains 999 data points with 49 features, including 7 numerical and 42 categorical features. Instead of removing all the categorical features, this study used one-hot encoding for all categorical features. Furthermore, to run the experiments, this study normalised the numerical features, so that all the values are between 0 and 1. There are 300 bad credit risk data points (30%) in this dataset.
All models used in the experiments are trained on 70% of each dataset and the rest of 30% were used to find counterfactual examples.

3.3 Hyperparameters

There are four hyperparameters of FOCUS, specifically, sigma (Equation 1), temperature (Equation 4), distance weight, which is a trade-off parameter between distance loss and prediction loss and learning rate of Adam [8]. This study used the hyperparameters provided by the original authors to reproduce the original experiments.

Additionally, this study conducted a hyperparameter tuning using the Optuna package[9]'s Bayesian optimisation for the retrained models. The search spaces of hyperparameters can be found in Table 9 in the Appendix D. The search was conducted for 100 trials. It is worth noting that since DT models do not use the temperature parameter, the search for temperature was disabled when tuning DT models.
Due to resource and time constraints, this study was unable to run hyperparameter tuning for all models and dataset combinations, particularly for larger models such as RF and AB models. The used hyperparameters for all the retrained models are reported in Table 10, 11, 12 and 13 in Appendix E.

3.4 Experimental setup and code

Experiment 1 — This study aims to reproduce experiments from the original paper, with the exception of other papers' proposed methods. The experiments include (i) producing counterfactual explanations for all datasets by using pretrained models to examine the *reliability* claim and (ii) evaluating *effectiveness* claim by comparing the average distance of counterfactual explanations against the existing methods called DACE [10].
The same evaluation metric as the original paper will be utilised in this study. Let X be the set of N original data points and $\bar{X}$ be the set of N generated counterfactual explanations. The mean distance metric can be derived as:

$$d_{mean}(X, \overline{X}) = \frac{1}{N} \sum_{N=1}^{N} d(x^{(n)}, \overline{x}^{(n)}). \tag{5}$$

Four distance functions are used for evaluation: Euclidean, Cosine, Manhattan, and Mahalanobis. The results of these experiments can be found in Table 1 and 3.

Experiment 2 — This study conducts additional experiments to provide further support for the claims. These experiments aim to evaluate the robustness and generality of the FOCUS. Robustness is tested by updating the code implementation and models, and generality is tested by applying the updated FOCUS implementation on a different dataset. The results of these experiments can be found in Table 4 and 5.

3.5 Computational requirements

All the experiments in this study were conducted on a laptop with a 1.4 GHz Quad-core Intel Core i5 processor and 8 GB of RAM. The run time to rerun the experiments on the models was: Decision Tree (DT) models took under a minute, Random Forest (RF) models took approximately 20 minutes and Adaptive Boosting (AB) models took approximately 15 minutes on average. To run the retrained models, DT models took under a minute, RF models took approximately 30 minutes and AB models took 15 minutes on average. The study also conducted hyperparameter tuning on a few DT models, which took around 3 hours per model. In total, rerunning the experiments took around 8 hours, running the retrained models took around 9 hours, and hyperparameter tuning took around 16 hours.

4 Results

4.1 Results reproducing original paper

Experiment 1 evaluates the main claims, specifically *Reliability* and *Effectiveness*. As described in Section 3.4.1, experiment 1 reruns the published code by the authors and compares the results to the reported results in the original paper.

Reliability — Table 1 validates the *Reliability* claim of FOCUS for nearly all models, datasets and distance function combinations. There are two outcomes that failed to find counterfactual explanations for all instances - RF and AB models on COMPAS dataset using Manhattan distance. Based on the fact that the majority of outcomes align with the original results, it is conjectured that the two unsuccessful outcomes were caused by misreported hyperparameters. To evaluate this hypothesis, this study conducted hyperparameter tuning for those two models. Table 2 reports the mean Manhattan distance and found hyperparameters for those two cases. After the hyperparameter tuning, both experiments were able to find counterfactual explanations for all the instances and also the mean distance was closer to the original results. Although there are slight discrepancies in the rerun results in terms of the mean distances, this study was able to produce similar results to the original paper and draw the same conclusion - rerunning the original experiment was able to find a counterfactual explanation for all instances.
The results presented above demonstrate that the hyperparameters of FOCUS have a strong impact on the outcome of the experiments. To provide more insight on this point, section 4.2 discusses how the choice of hyperparameters affects the results and their tendencies.

Dataset	Distance function	DT	RF	AB
Wine	Euclidean	0.268 (0.268)	0.188 (0.188)	**0.268** **(0.188)**
	Cosine	0.003 (0.003)	0.009 (0.008)	**0.026** **(0.014)**
	Manhattan	0.268 (0.268)	0.312 (0.312)	**0.528** **(0.360)**
HELOC	Euclidean	0.133 (0.133)	0.186 (0.186)	0.136 (0.136)
	Cosine	0.001 (0.001)	0.002 (0.002)	0.001 (0.001)
	Manhattan	0.152 (0.152)	0.284 (0.284)	0.203 (0.203)
COMPAS	Euclidean	**0.015** **(0.092)**	0.079 (0.079)	0.076 (0,076)
	Cosine	0.008 (0.008)	0.011 (0.011)	0.007 (0.007)
	Manhattan	**0.102** **(0.093)**	**0.002*** **(0.085)**	**0.072*** **(0.090)**
Shopping	Euclidean	0.142 (0.142)	0.023 (0.025)	0.028 (0.028)
	Cosine	0.055 (0.055)	0.013 (0.013)	0.006 (0.006)
	Manhattan	0.128 (0.128)	0.026 (0.026)	0.047 (0.046)

Table 1. Mean Euclidean, Cosine and Manhattan distance for all the original datasets and model combinations. The numbers in the parentheses are the mean distance of the reported distance in the original paper. * denotes that it failed to produce counterfactual explanations for all instances.

Model	Mean distance	sigma	temperature	distance weight	learning rate
RF	0.116	6	12	0.01	0.002
AB	0.090	4	1	0.05	0.001

Table 2. Found new hyperparameters and Manhattan mean distances.

Dataset	Model	Reproduction	Original	Original DACE
Wine	DT	2.354	0.542	1.325
HELOC	DT	1.128	0.810	1.427
COMPAS	DT	0.938	0.776	0.814
	AB	0.756	0.636	1.570
Shopping	DT	1.424	0.023	0.050
	AB	0.148	0.303	3.230

Table 3. Mean Mahalanobis distance for all the original datasets and model combinations.

Effectiveness — The results that support the *Effectiveness* claim are presented in Table 3. This table provides the mean Mahalanobis distance of the rerun models, the original models, and the existing framework, DACE. The rerun models' results slightly deviate from the original results. Several mean Mahalanobis distances of the rerun models were found to be larger than the reported results of DACE. This study attempted to replicate the results through hyperparameter tuning, however, no set of hyperparameters was discovered that would produce the results as originally reported. Another potential explanation for the deviation of results could be related to the calculation of the Mahalanobis distance, yet thorough unit tests of the relevant functions did not reveal any problematic areas. Further investigation and experimentation may be necessary to fully comprehend the source of the discrepancy observed in this experiment. Despite this, the study still provides evidence that half of the rerun models exhibited better results than those produced by the original DACE framework, lending partial support to the claim of *effectiveness*.

4.2 Results beyond original paper

As described in 3.4, this study conducts additional experiments to test the robustness and generality of FOCUS in terms of the *Reliability* claim. This is examined by retraining models on the updated code implementation and applying FOCUS on those models on all datasets including the German credit dataset.

Robustness and Generality — The robustness and generality of FOCUS are presented through the outcomes of the experiment, as illustrated in Tables 4 and 5. The findings reveal that all DT models are capable of generating counterfactual explanations for all instances, while a limited number of RF and AB models were able to do so. Additionally, a significant proportion of the RF models encountered difficulties running due to limited computational resources, which have impacted the ability to perform hyperparameter tuning for most of the RF and AB models. This limitation is further explored in following section.
Overall, the experiment results provide additional evidence of the robustness and generality of FOCUS's reliability claims. Although the conclusions drawn from the experiment are limited to DT models, they demonstrate that FOCUS can draw the same conclusions as the original study, even when models are retrained on updated codebases and applied to a different dataset. However, further research could extend these findings to other model types.

Impact of hyperparameters on results — During the experiments, this study learned that hyperparameters affect results strongly. Theoretically, the hyperparameters of FOCUS (sigma and temperature) influence the quality of the model approximation $\tilde{f}$, of f. As sigma increases, the probabilistic approximation of the node activation becomes an exact approximation of the indicator functions (as per Equation 1), and increasing temperature leads the maximum operation of f to a unimodal softmax distribution (per Equation 4).

Dataset	Distance function	DT	RF	AB
Wine	Euclidean	0.358 (0)	-	0.197 (954)
	Cosine	0.006 (0)	-	1.458 (1)
	Manhattan	0.358 (0)	-	0.578 (431)
	Mahalanobis	4.069 (0)	-	4.436 (435)
HELOC	Euclidean	0.122 (0)	-	0.110 (794)
	Cosine	0.001 (0)	1.248 (0)	1.213 (0)
	Manhattan	0.139 (0)	0.327 (0)	0.366 (207)
	Mahalanobis	0.876 (0)	-	0.913 (719)
COMPAS	Euclidean	0.083 (0)	0.099 (8)	0.054 (37)
	Cosine	0.012 (0)	1.273 (13)	1.088 (14)
	Manhattan	0.118 (0)	-	0.053 (1330)
	Mahalanobis	1.158 (0)	0.470 (181)	0.479 (37)
Shopping	Euclidean	0.0352 (0)	-	0.041 (280)
	Cosine	0.013 (0)	-	1.161 (40)
	Manhattan	0.043 (0)	-	0.067 (305)
	Mahalanobis	0.460 (0)	-	0.734 (317)

Table 4. Mean Euclidean, Cosine and Manhattan distance for all the original datasets and model combinations. - denotes that failed to run. The numbers in the parentheses indicate the number of instances that are unable to find a counterfactual explanation.

Distance function	DT	RF	AB
Euclidean	0.003 (0)	0.112 (0)	0.003 (63)
Cosine	1.001 (0)	1.424 (0)	1.502 (6)
Manhattan	0.003 (0)	0.082 (9)	0.006 (40)
Mahalanobis	62.074 (0)	-	1.852 (47)

Table 5. Mean Euclidean, Cosine, Manhattan and Mahalanobis distance of each model on the German credit dataset. - denotes that failed to run. The numbers in the parentheses indicate the number of instances that are unable to find a counterfactual explanation.

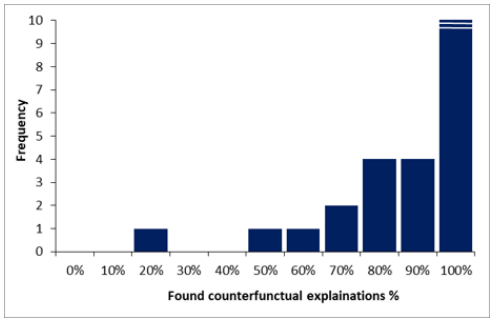

Figure 1. Found counterfactual explanations % on COMPAS dataset. This data was collected when hyperparameter tuning was run for 100 trials on the DT model by using Mahalanobis distance. The Hyperparameter tuning algorithm found optimal solutions for over 90% of instances in most cases (86 instances), therefore, the figure has been scaled for improved visualisation.

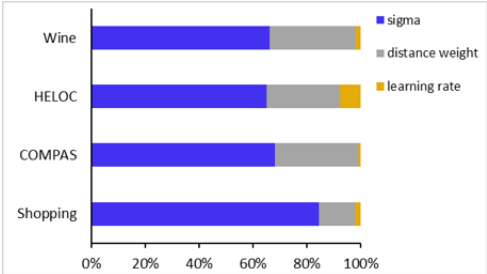

Figure 2. Hyperparameter importance for each dataset. This data was collected when hyperparameter tuning for DT models by using Mahalanobis distance was run. Note that DT models do not use temperature hyperparameters, thus there are only three hyperparameters tuned for those models.

Empirically, this study found that the quality of the approximation of the original model f has a significant effect on the results. For instance, the number of counterfactual explanations found can range from 20% to 100% based on the chosen hyperparameters as demonstrated in Figure 1. The analysis of the hyperparameter importance for DT models, as presented in Figure 2, indicates that the approximation of node activation (sigma) has a strong effect on both the mean Mahalanobis distance and the number of counterfactual explanations found on all datasets. Conversely, changes to the prediction loss-distance loss trade-off parameter (distance weight) and the learning rate of Adam did not exhibit a significant impact on the results. These findings are limited to DT models, and future studies could extend these findings to other model types.

Model size consideration — This study encountered difficulties in running RF models for more than half of the experiments. Initially, it was suspected that this difficulty was caused mainly due to limited computational resources. Also, the original paper's experiments were conducted on a machine with a 48-core CPU and 256GB of RAM, while this study's experiments were conducted on a computer with a quad-core CPU and 8GB of RAM.
However, Table 7 in Appendix B shows that the majority of the retrained models are smaller in size on the disk than the original ones. Despite this, the study was unable to run the retrained models but was able to run the original ones. This suggests that the inability to execute the retrained models may not be solely attributed to their size, and other factors may be contributing.

5 Discussion

This study aimed to assess the reliability and effectiveness claims of FOCUS and has drawn several conclusions based on the results of two experiments.
Firstly, in regards to the *reliability* claim, the experiments' results validate the original paper's results. Also, the additional experiment demonstrated that FOCUS is robust and generalisable. The additional experiment was limited to DT models, however, future studies could expand the investigation to other tree-based models such as XGBoost [11] and LightGBM [12].
Moreover, this study sheds light on the impact of hyperparameters on the results of

FOCUS. It was demonstrated that the selection of hyperparameters can significantly influence the ability of FOCUS to generate counterfactual explanations, thus emphasising the importance of hyperparameter tuning in future studies.
Additionally, the study also highlighted the issue of running larger models as described in Section 4.2. This study suggests that this difficulty may not be solely due to model size, but other factors may also be contributing. Further research is needed to investigate these factors and find ways to overcome these challenges, to enable the application of FOCUS on larger models.
The effectiveness claim is partially supported by this study. While FOCUS was able to generate the counterfactual explanations for all instances, the mean Mahalanobis distances were not consistent with the results reported in the original paper. This deviation raises questions about the reproducibility of the results and highlights the need for further investigation to determine the cause.

5.1 What was easy

The original paper's implementation is accessible on GitHub. The repository includes the models and data utilised in the experiments. The authors have also made available a technical appendix, which can be requested and provides all the necessary information, including hyperparameters to reproduce the experiments.

5.2 What was difficult

The code for the implementation was available, however, it utilises outdated packages and the code structure is complex, making it difficult to follow the code. Additionally, the comments and documentation within the code are minimal or absent. Adding unit tests to the codebase helped me to improve my understanding of the structure. Furthermore, for stronger support on the claims made in the paper, it would have been beneficial to run the previously developed framework, DACE, however, due to time constraints and the complexity of using the CPLEX Optimizer [1], this study was unable to do so.

5.3 Communication with original authors

I contacted the authors to obtain the hyperparameters used in the experiments, and they responded promptly with a detailed technical appendix of the original paper.

References

1. A. Lucic, H. Oosterhuis, H. Haned, and M. de Rijke. "FOCUS: Flexible optimizable counterfactual explanations for tree ensembles." In: **Proceedings of the AAAI Conference on Artificial Intelligence**. Vol. 36. 5. 2022, pp. 5313–5322.
2. S. Wachter, B. Mittelstadt, and C. Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." In: **Harv. JL & Tech.** 31 (2017), p. 841.
3. D. Dua and C. Graff. **UCI Machine Learning Repository**. 2017. URL: http://archive.ics.uci.edu/ml.
4. P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. "Modeling wine preferences by data mining from physicochemical properties." In: **Decision support systems** 47.4 (2009), pp. 547–553.
5. FICO2017. **HELOC Dataset**. 2017. URL: https://community.fico.com/s/explainable-machine-learning-challenge?tabset-158d9=3 (visited on 01/08/2023).
6. D. Ofer. "COMPAS Dataset." In: **Kaggle: https://www. kaggle. com/danofer/compass** (2017), p. 19.
7. C. O. Sakar, S. O. Polat, M. Katircioglu, and Y. Kastro. "Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks." In: **Neural Computing and Applications** 31.10 (2019), pp. 6893–6908.

[1] http://www.ibm.com/analytics/cplex-optimizer

8. D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization." In: **arXiv preprint arXiv:1412.6980** (2014).
9. T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. "Optuna: A next-generation hyperparameter optimization framework." In: **Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining**. 2019, pp. 2623–2631.
10. K. Kanamori, T. Takagi, K. Kobayashi, and H. Arimura. "DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization." In: **IJCAI**. 2020, pp. 2855–2862.
11. T. Chen and C. Guestrin. "XGBoost: A Scalable Tree Boosting System." In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. KDD '16. San Francisco, California, USA: ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785. URL: http://doi.acm.org/10.1145/2939672.2939785.
12. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. "Lightgbm: A highly efficient gradient boosting decision tree." In: **Advances in neural information processing systems** 30 (2017).

A Hyperparameters for retrained models

Table 6 reports the hyperparameters that were used to retrain each model for each dataset. Retrained DT models still employ the same hyperparameters as the original models, but the other models, most of them have a smaller structure than the original models.

Dataset	Hyperparameter	DT	RF	AB
Wine	Max Depth	2 (2)	4 (4)	2 (4)
	Num Trees	1 (1)	100 (500)	100 (100)
HELOC	Max Depth	4 (4)	2 (4)	1 (8)
	Num Trees	1 (1)	100 (500)	100 (100)
COMPAS	Max Depth	4 (4)	2 (4)	1 (2)
	Num Trees	1 (1)	100 (500)	100 (100)
Shopping	Max Depth	4 (4)	4 (8)	1 (2)
	Num Trees	1 (1)	100 (500)	100 (100)
German	Max Depth	2 (-)	3 (-)	2 (-)
	Num Trees	1 (-)	100 (-)	100 (-)

Table 6. Hyperparameters of retrained models. Numbers in the parentheses are the hyperparameters of the original models.

B Model size comparison

Table 7 reports the model sizes of retrained and original models on the disk. Most retrained models have a smaller size as smaller hyperparameters were used compared to the original models.

	DT		RF		AB	
Dataset	Retrained	Original	Retrained	Original	Retrained	Original
Wine	3	2	263	711	48	131
HELOC	4	2	94	703	34	148
COMPAS	2	2	94	467	34	85
Shopping	4	2	265	143	34	89
German	2	-	144	-	48	-

Table 7. Size of models on the disk. The unit of this table is KB.

C Accuracy of retrained models

This study retrained models with new hyperparameters in order to conduct further experiments. The train/test split method used in this study follows the original paper, where 70% of the dataset was used for training and 30% was used for test. This study employs the accuracy score as a metric. The accuracy score can be derived as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{6}$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative.

Dataset	DT	RF	AB
Wine	0.796	0.788	0.771
HELOC	0.679	0.692	0.701
COMPAS	0.651	0.677	0.675
Shopping	0.890	0.893	0.892
German	0.700	0.713	0.723

Table 8. Accuracy of all the models

D Hyperparameter tuning

In this study, hyperparameter tuning was performed on a few pretrained models and retrained DT models by using Optuna's Bayesian optimisation. Table 9 illustrates the search spaces of hyperparameters. It is worth noting that since DT models do not use the temperature parameter, the search for temperature was disabled when tuning DT models to save some computational costs.

	Search space		
Hyperparameter	Min	Max	Step
sigma	1	20	1
temperature	1	20	1
distance weight	0.01	0.1	0.01
learning rate	0.001	0.01	0.001

Table 9. Hyperparameters and their search spaces

E FOCUS hyperparameters

Table 10, 11, 12 and 13 report used hyperparameters for retrained models. As DT models do not use temperature, it is not reported.

Dataset	Model	sigma	temperature	weight distance	learning rate
Wine	DT	1	-	0.05	0.001
	AB	5	1	0.05	0.005
HELOC	DT	2	-	0.05	0.001
	AB	10	1	0.05	0.001
COMPAS	DT	4	-	0.01	0.009
	RF	7	3	0.01	0.001
	AB	10	1	0.01	0.005
Shopping	DT	2	-	0.05	0.005
	AB	10	1	0.05	0.001
German	DT	7	-	0.01	0.001
	RF	7	3	0.01	0.001
	AB	7	3	0.01	0.001

Table 10. FOCUS hyperparameters for using Euclidean distance

Dataset	Model	sigma	temperature	weight distance	learning rate
Wine	DT	1	-	0.05	0.005
	AB	1	1	0.01	0.005
HELOC	DT	2	-	0.05	0.005
	RF	5	5	0.05	0.005
	AB	1	1	0.05	0.005
COMPAS	DT	10	-	0.05	0.005
	RF	10	6	0.01	0.005
	AB	10	1	0.05	0.005
Shopping	DT	10	-	0.05	0.001
	AB	10	5	0.05	0.001
German	DT	7	-	0.01	0.001
	RF	7	3	0.01	0.001
	AB	7	3	0.01	0.001

Table 11. FOCUS hyperparameters for using Cosine distance

Dataset	Model	sigma	temperature	weight distance	learning rate
Wine	DT	1	-	0.05	0.001
	AB	6	1	0.01	0.005
HELOC	DT	2	-	0.05	0.001
	RF	5	5	0.01	0.005
	AB	4	1	0.05	0.001
COMPAS	DT	6	-	0.01	0.005
	AB	5	10	0.05	0.005
Shopping	DT	2	-	0.05	0.005
	AB	10	1	0.05	0.001
German	DT	7	-	0.01	0.001
	RF	7	3	0.01	0.001
	AB	7	3	0.01	0.001

Table 12. FOCUS hyperparameters for using Manhattan distance

Dataset	Model	sigma	temperature	weight distance	learning rate
Wine	DT	4	-	0.01	0.003
	AB	10	1	0.01	0.005
HELOC	DT	7	-	0.01	0.002
	AB	10	1	0.01	0.005
COMPAS	DT	4	-	0.01	0.008
	RF	10	1	0.01	0.005
	AB	4	2	0.05	0.001
Shopping	DT	20	-	0.02	0.003
	AB	10	1	0.01	0.001
German	DT	18	-	0.01	0.003
	AB	7	3	0.01	0.001

Table 13. FOCUS hyperparameters for using Mahalanobis distance