Proceedings of Machine Learning Research 204:1–20, 2023 Conformal and Probabilistic Prediction with Applications

Approximating Score-based Explanation Techniques Using Conformal Regression

Amr Alkhatib ALKHAT@KTH.SE
Henrik Boström BOSTROMH@KTH.SE
Sofiane Ennadir ENNADIR@KTH.SE
School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden
Ulf Johansson ULF.JOHANSSON@JU.SE
Dept. of Computing, Jönköping University, Sweden

Editor: Harris Papadopoulos, Khuong An Nguyen, Henrik Boström and Lars Carlsson

Abstract

Score-based explainable machine-learning techniques are often used to understand the logic behind black-box models. However, such explanation techniques are often computationally expensive, which limits their application in time-critical contexts. Therefore, we propose and investigate the use of computationally less costly regression models for approximating the output of score-based explanation techniques, such as SHAP. Moreover, validity guarantees for the approximated values are provided by the employed inductive conformal prediction framework. We propose several non-conformity measures designed to take the difficulty of approximating the explanations into account while keeping the computational cost low. We present results from a large-scale empirical investigation, in which the approximate explanations generated by our proposed models are evaluated with respect to efficiency (interval size). The results indicate that the proposed method can significantly improve execution time compared to the fast version of SHAP, TreeSHAP. The results also suggest that the proposed method can produce tight intervals, while providing validity guarantees. Moreover, the proposed approach allows for comparing explanations of different approximation methods and selecting a method based on how informative (tight) are the predicted intervals.

Keywords: Inductive conformal prediction · Explainable machine learning · Multi-target regression

1. Introduction

Explainable machine learning has become increasingly important as machine learning algorithms are applied to real-world problems within various domains, including healthcare, finance, and criminal justice, among others (Lakkaraju et al., 2017). In many cases, it is essential that the output of such machine learning algorithms are transparent and can be understood by the users.

Explanation methods are classified based on the scope of the explanations they produce, which can be either local or global (Molnar, 2022). Local explanation methods provide instance-based explanations for a single prediction of a black-box model, while global explanation methods provide an overview of how a model behaves in general (Molnar, 2022). Explanation methods can also be classified into two main categories: model-agnostic and model-specific. Model-agnostic techniques are designed to explain any black-box model and

have been described in previous research, e.g., Ribeiro et al. (2016a). On the other hand, model-specific techniques leverage the properties of the underlying model to generate explanations, e.g., Boström et al. (2018). Explanation methods can produce explanations in the form of plots, such as the Partial Dependence Plot (PDP) and the Accumulated Local Effects (ALE) Plot (Apley and Zhu, 2020). Rule-based explanations, e.g., those produced by Anchors (Ribeiro et al., 2018), are another easy to interpret explanation form. Finally, additive feature importance scores, such as those generated by LIME (Local Interpretable Model-agnostic Explanations) (Ribeiro et al., 2016b) and SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017), are another form of explanation.

However, the deployment of explanation methods comes with its challenges. One of the most significant challenges is the computational cost, which can be for several reasons. One main reason is that many methods analyze the behavior of a black-box model by running multiple iterations on different input data points, which can be time-consuming (Molnar et al., 2020). Also, some explanation methods, e.g., LIME (Ribeiro et al., 2016b), involve creating multiple local (white-box) surrogate models. Furthermore, one of the most influential explanation methods, SHAP, involves approximating Shapley values to compute the marginal importance of the features (Lundberg and Lee, 2017). Computing Shapley values requires calculating the contributions of each feature in a given prediction, where the cost grows exponentially with the number of features.

In this work, we address the challenge of handling the computational cost associated with explainable machine learning methods. We propose to reduce the cost by approximating the output of any selected explanation method that produces feature importance scores using an algorithm of lower complexity at inference time and provide some validity guarantees on the produced explanations using the conformal prediction framework (Vovk et al., 2005). The proposed approach will be thoroughly evaluated to demonstrate its effectiveness in approximating the explanation method while reducing the computational cost.

The main contributions of this study are:

- an approach for approximating the explanations to a black box predictions in the form of feature importance scores accompanied with validity guarantees from the conformal prediction framework
- a set of non-conformity measures for the conformal prediction framework
- a large-scale empirical investigation on 30 publicly available datasets comparing the performance of two different algorithms in approximating the explanation method, as well as the performance using the proposed non-conformity measures

In the next section, we provide a background on the conformal prediction framework. In Section 3, we briefly review related work. In Section 4, we describe the proposed method and propose a set of non-conformity measures designed to reduce the computational cost. Section 5 presents and discusses the results of the large-scale empirical investigation. Finally, Section 6 recapitulates the main conclusions and outlines future work directions.

2. Background

Conformal Prediction (CP) has been introduced as an approach for providing guarantees on the prediction error. A user-predefined confidence score bounds the probability of producing wrong predictions (Johansson et al., 2014). CP was initially introduced as a transductive approach that learns a model for each data instance, which is computationally expensive (Gammerman et al., 1998; Saunders et al., 1999). Consequently, Vovk et al. (2005) introduced inductive conformal prediction (ICP), where only one model is induced from the provided data, which is then used for making predictions on new data instances. In the remaining part of this section, we briefly describe inductive conformal prediction for regression models, as well as the possible non-conformity functions.

2.1. Inductive Conformal Prediction for Regression Models

Let $\mathcal{X}$ be a feature space and $\mathcal{Y}$ be the target variable. Given a dataset $\mathcal{Z} = \{z_1; z_2; \ldots; z_n\}$, where $z_i = (x_i; y_i)$, $x_i \in \mathcal{X}$, and $y_i \in \mathbb{R}$. Assuming the provided data are independent and identically distributed (i.i.d), the inductive conformal regression consists of the following main steps:

1. Split the provided dataset $\mathcal{Z}$ into a proper training subset $\mathcal{Z}_t = \{z_1; z_2; \ldots; z_m\}$ and a calibration subset $\mathcal{Z}_c = \{z_{m+1}; z_{m+2}; \ldots; z_n\}$.

2. The underlying model h is trained using $\mathcal{Z}_t$.

3. For each example $z_i \in \mathcal{Z}_c$, use the non-conformity function to calculate the non-conformity score $\quad_i$ to get the sequence $\mathcal{S} = \{\quad_{m+1}; \quad_{m+2}; \ldots; \quad_n\}$. The non-conformity function can be simply the absolute error (Papadopoulos et al., 2002):

$$\quad_i = |y_i - \tilde{y}_i| \tag{1}$$

   where $\tilde{y}_i$ is the predicted outcome by the underlying model h.

4. Given a predefined significance level $\quad$ and the sequence of the non-conformity scores $\mathcal{S}$, the smallest $\quad \in \mathcal{S}$ such that:

$$\frac{|\{z_i \in \mathcal{Z}_c | \quad_i < \quad\}| + 1}{|\mathcal{Z}_c| + 1} \geq 1 - \tag{2}$$

   Consequently, with a probability of 1- , the non-conformity score of a new data instance x_{n+1} will be less than or equal to $\quad$, since $\quad$ provides a probabilistic bound for the non-conformity scores at the significance level $\quad$. Finally, an interval covering the true prediction with a probability of 1- is produced as follows:

$$\tilde{Y}_{n+1} = [\tilde{y}_{n+1} - \quad ; \tilde{y}_{n+1} + \quad] \tag{3}$$

2.2. Normalized Non-Conformity Functions

The previous description of the inductive conformal regression results in a xed interval size for all predictions. However, some predictions are expected to be more accurate than others, and a natural improvement is to predict smaller intervals for more accurate (easy) cases. Therefore, the non-conformity score $_i$ of the instance x_i can be normalized by a di culty estimate $_i$:

$$_i = \frac{jy_i \quad \hat{y}_i j}{_i} \tag{4}$$

For a prediction on a new data instance x_{n+1}, the predicted interval is given by:

$$\tilde{Y}_{n+1} = [\hat{y}_{n+1} \quad _{n+1}; \hat{y}_{n+1} + \quad _{n+1}] \tag{5}$$

Papadopoulos et al. (2011) proposed to estimate the di culty using the k-nearest neighbours (KNN). According to Papadopoulos et al. (2011), this can be done by computing the sum of the distances between x_i and its k-nearest neighbours:

$$d_i^k = \sum_{2N} \text{distance}(x_i; x \ ) \tag{6}$$

where N is the set of the k-nearest neighbours. Then d_i^k is normalized by the median value of the distances in the training data:

$$_i^k = \frac{d_i^k}{\text{median}(d_j : z_j \ 2 \ Z_t)} \tag{7}$$

Finally, the non-conformity score is computed as follows:

$$_i = \frac{jy_i \quad \hat{y}_i j}{+ \ _i^k}; \tag{8}$$

and

$$_i = \frac{jy_i \quad \hat{y}_i j}{\exp(\ _i^k)}: \tag{9}$$

where $\quad 0$ is a parameter that controls the measure's sensitivity to any changes in $_i^k$.

Another di culty estimate proposed by Papadopoulos et al. (2011) is based on the difference between the labels of the k-nearest neighbours measured in their standard deviation. The high agreement among the k-nearest neighbours typically means a more accurate prediction. The standard deviation of the labels of the k-nearest neighbours is computed as follows:

$$s_i^k = \sqrt{\frac{1}{k} \sum_{j=1}^{k} (y_{i_j} \quad \overline{y_{i_{1;2;:::;k}}})^2}; \tag{10}$$

where

$$\overline{y_{i_{1;2;\ldots;k}}} = \frac{1}{k}\sum_{j=1}^{k} y_{i_j} \tag{11}$$

Then s_i^k can also be normalized, similar to what has been proposed with d_i^k:

$$_i^k = \frac{s_i^k}{\text{median}(s_j : z_j \ 2 \ Z_t)} \tag{12}$$

Consequently, the non-conformity measure can be computed as follows:

$$_i = \frac{jy_i \quad \hat{y}_i j}{+ \ _i^k}; \tag{13}$$

and

$$_i = \frac{jy_i \quad \hat{y}_i j}{\exp(\ _i^k)}: \tag{14}$$

Finally, $_i^k$ and $_i^k$ can be combined to de ne the following non-conformity measures:

$$_i = \frac{jy_i \quad \hat{y}_i j}{+ \ _i^k + \ _i^k}; \tag{15}$$

and

$$_i = \frac{jy_i \quad \hat{y}_i j}{\exp(\ _i^k) + \exp(\ _i^k)}; \tag{16}$$

where controls the sensitivity of the measure to the changes in $_i^k$, similar to with $_i^k$

3. Related Work

In this section, we start by describing two popular explanation methods and clarify why they are computationally expensive. We also provide some pointers to the contributions to providing computationally more e cient machine learning explanation methods.

Explainable machine learning have recently caught signi cant attention as a research eld, especially since the introduction of the LIME technique for generating local explanations (for a speci c prediction) in 2016 (Ribeiro et al., 2016b). LIME ts a white-box model on the predictions of the underlying black box on perturbed data points, which are weighted by proximity to the being explained data point. The feature weights of the white box are used to generate an explanation for the prediction. Since LIME provides local explanations, its complexity does not depend on the size of the dataset but on the number of perturbed data instances, the complexity of the underlying black box, the complexity of the used white box, and the length of the produced explanation (number of features to explain a prediction). SHAP (Lundberg and Lee, 2017) is another prominent local explanation method that assigns importance scores to the features by approximating the Shapley values. Since the exact Shapely values computation requires all possible coalitions of the feature values,

SHAP (i.e., Kernel SHAP) approximates the exact Shapley values using sampled feature coalitions, and a linear model is tted to approximate the Shapley values.

Lundberg et al. (2018) introduced a faster variant of SHAP for tree-based models, e.g., random forests and gradient-boosted trees. The variant is TreeSHAP, a model-speci c alternative to the model-agnostic Kernel SHAP. The time complexity of SHAP can be reduced from $O(TL2^M)$ to $O(TLD^2)$ by using TreeSHAP for a tree-based model (Lundberg et al., 2018), where T is the number of trees, L is the maximum number of leaves in any tree, D is the maximum depth of any tree, and M is the number of features. Situ et al. (2021) proposed that any o -the-shelf explainer can be distilled into an explainer neural network (L2E, Learning to Explain). L2E focused mainly on imitating the explanations obtained on text classi cation tasks. Approximating the outcome of the explanation method using a neural network can reduce the time complexity to the level of the neural network at the inference time and can also help with producing stable explanations. However, at the inference time, there are no guarantees regarding the validity of the predicted explanation at the individual word level. FastSHAP was proposed by Jethani et al. (2022) to improve the run-time of the Shapley values approximation-based explanation methods. FastSHAP avoids conducting an optimization process for each data point by learning a parametric function to approximate the Shapley value explanations. For image classi cation explanations, h-Shap (Hierarchical Shap) (Teneggi et al., 2022) has been proposed as a fast and exact implementation of Shapley coe cients.

4. Method

This section rst describes a method to approximate any explanation method that produces additive feature importance scores. Afterwards, we discuss how to provide validity guarantees using the conformal prediction framework. Finally, we propose three possible non-conformity conformity measures for the conformal regression.

4.1. Explanation Method Approximation

The proposed method can be applied to any explanation method as long as the produced explanations take the form of additive feature importance scores. The explanation method (A) is considered a function $(A : f(x; t;) = y)$ that can be approximated using a machine learning model $(\tilde{A})$, where $x \in R^n$ is the data point with n features, t is the predicted outcome by the black box model, is some learnable parameters, and $y \in R^n$ is the vector containing the importance scores of the$n$ features. The approximation model $(\tilde{A})$ learns a mapping from $(x; t)$ to y. Since the target y is a vector with an importance score per feature, the problem can be formulated as a regression problem. There are two possible approaches to solving this regression problem: i) to handle it as a multi-target regression problem, using, for instance, a neural network, and ii) to handle it as a set of single-target regression problems.

A development dataset (X^{dev}) is provided for the regression model, where the underlying black box (B) produces predictions t^{dev} for the data points in X^{dev}, and the explanation method (A) generates explanations (Y^{dev}) for the predicted outcomes. For each data point in the development set, a feature vector (x) is augmented with its predicted outcome (t) to build the augmented development features $(x^0 = x [t)$. The aug-

mented development set X'^{dev} altogether with produced explanations Y^{dev} form $Z^{dev} = \{(x'^0_1, y_1), (x'^0_2, y_2), \ldots, (x'^0_n, y_n)\}$. $\tilde{A}$ is learned by fitting the regression model on Z^{dev}.

4.2. Validity Guarantees

The learned regression model $\tilde{A}$ predicts an importance score per feature, and we can simply control the error level of each feature independently from the others using the conformal prediction framework. In other words, each predicted importance score is considered a separate regression problem, and the conformal regression will be applied to control the error level. Alternatively, the problem can be handled as a conformal multi-target regression similar to the method proposed by Messoudi et al. (2020). However, we will leave the alternative approach for future work. Consequently, a calibration dataset X^{cal} is provided and augmented with the class label acquired from B to obtain X'^{cal}, which is provided to $\tilde{A}$ to generate importance scores for all the data points in the calibration set X^{cal} (predict $\tilde{Y}^{cal}$). Using the ground truth Y^{cal} obtained from A, a non-conformity score α^f_j is computed for each feature f for each example x_j in X^{cal}. Let δ^f be the score of feature f at a significance level ϵ. At prediction time, all values with a distance greater than δ^f are excluded: $\tilde{Y}^f_j = [\hat{y}^f_j - \delta^f, \hat{y}^f_j + \delta^f]$.

4.3. Non-Conformity Measures

As described in Section 2.2, the non-conformity score can be normalized using a difficulty estimate β_j, which can be computed using, for example, a trained model or the distance to the k-nearest neighbors. However, such difficulty estimates are computed at both the calibration and the inference time per instance and can be computationally expensive. Therefore we propose the following difficulty estimation functions:

1. Minimum distance to the distributions: we assume there is a mixture of distributions where each class C in the training data represents one distribution. Consequently, we fit a Gaussian mixture model (Rasmussen, 1999) on the training data. Then we use the mean μ_C and the covariance Σ_C of each distribution to compute the Mahalanobis distance (Mahalanobis, 1936) between a data point and each distribution:

$$d_{jC} = \sqrt{(x_j - \mu_C)^T \Sigma_C^{-1} (x_j - \mu_C)} \tag{17}$$

   Then the minimum distance is used as a difficulty estimate:

$$\beta_j = \log(\arg\min_C (d_{jC}) + 1) \tag{18}$$

2. Average distance to the distributions: we also assume a mixture of class distributions in the training data, but here we take the average of all distances from the data point to the distributions:

$$\beta_j = \log(\frac{1}{n} \sum_{C=1}^{n} d_{jC} + 1) \tag{19}$$

3. The prediction con dence: we assume that if the prediction towards one class has high algorithmic con dence, then it is an easy example for the black box model. Therefore, we propose the following di culty estimate:

$$_j = 1 \quad \max(P_C) \tag{20}$$

where P_C is the predicted score of class C, and for the binary classi cation we use the following:

$$_j = 1 \quad jP \quad 0{:}5j \tag{21}$$

where P is the predicted score.

5. Empirical Evaluation

This section compares the generated explanations using two distinct regression models. The quality of the generated explanations is assessed with respect to the execution time as well as the produced interval sizes using the conformal prediction framework, where a tighter interval size implies a better approximation. We conduct two sets of experiments. We also evaluate the explanations using the proposed non-conformity measures.

5.1. Experimental Setup

In the following experiments, the proposed explanation algorithms are evaluated using 30 publicly available datasets[1]. Each dataset is split into a training set, calibration set, and test set. The calibration set is used to compute the non-conformity scores and nd the con dence percentile score, the training split is used to train the black-box model, and the test split is used to evaluate the generated explanations. The conformal regressors were generated using the crepes[2] Python package (Bostr om, 2022). Two algorithms are used for the explanation technique approximation: XGBoost and multi-layer perceptron (MLP). In the XGBoost experiments, one regressor is trained per feature, while in the case of MLP, one regression model is trained to predict all the feature importance scores. The XGBoost model employs a learning rate of 0.1, 600 estimators, and 0.01 for the regularization parameter. The MLP model has two layers, each of 1024 units and a Relu activation function. The MLP is trained with early stopping and 0.1 validation fraction.

The underlying black-box model is learned through an XGBoost algorithm. The hyper-parameters of XGBoost are tuned through grid search. The hyperparameters include the learning rate, the number of estimators, and the regularization parameter. The categorical features are binarized in the data preprocessing phase using one-hot encoding. The model is trained with each combination of hyperparameters on the training set and evaluated on the calibration set. The XGBoost model is trained using the best-performing set of hyper-parameters. TreeSHAP is used as an explanation technique to the underlying black-box model.

1. All the datasets were obtained from https://www.openml.org
2. https://github.com/henrikbostrom/crepes

Figure 1: The average rank of the compared regression models and TreeSHAP on the 30 datasets with respect to the execution time (in seconds), where the critical difference (CD) represents the largest difference that is not statistically significant.

5.2. Experimental Results

5.2.1. Execution Time

The execution time is measured on the test set of each dataset and recorded in seconds. All experiments have been performed in a Python environment on an Intel(R) Core(TM) i9-10885H CPU @ 2.40GHz system. The baseline for comparing execution times is set by TreeSHAP, a faster variant of SHAP created for tree-based models. In this experiment we compare TreeSHAP to MLP and XGBoost regressors with the proposed non-conformity measures in subsection 4.3: the minimum distance to the distributions (Min. Dist.), the average distance to the distributions (Avg. Dist.), the prediction confidence (Pred. Conf.), the k-nearest neighbours (KNN), and without any difficulty estimate (Baseline MLP and Baseline XGBoost).

The Friedman test (Friedman, 1939) is applied to assess the null hypothesis that there is no significant difference in the time needed to generate explanations using TreeSHAP explainer or any of the XGBoost and MLP regression models with the different non-conformity measures. The result of the Friedman test allows for rejecting the null hypothesis at the 0.05 level, thereby confirming a significant difference in the execution time. Subsequently, the post-hoc Nemenyi test (Nemenyi, 1963) is employed to determine significant pairwise differences, which are presented in Figure 1. The results show that XGBoost and MLP without a difficulty estimate and MLP with Pred. Conf.) significantly outperform TreeSHAP. However, no significant difference in execution time has been observed between TreeSHAP and MLP (with Min. Dist. or Avg. Dist.) or XGBoost (with Min. Dist., Avg. Dist., or Pred. Conf.). On the other hand, MLP and XGBoost with KNN are significantly outperformed by TreeSHAP. Furthermore, it has been observed that the execution time of XGBoost is affected by the number of features since we train a regressor per feature, while MLP is affected by the number of instances in a dataset. The detailed results are shown in Table 1.

5.2.2. Interval Size

Since the conformal regression provides the needed validity guarantees and the correct feature importance scores are ensured to be covered by the produced intervals with a specified

Table 1: The time needed to generate explanations for all data points in the test set. The execution time is measured in seconds. The compared methods are TreeSHAP and approximated explanations using XGBoost and MLP with di erent non-conformity measures. The best-performing model is colored in blue, and the second best-performing is colored in light blue.

Dataset	Test Set Size	Num. Features	TreeSHAP	XGBoost					Multi-Layer Perceptron				
				Min. Dist.	Avg. Dist.	KNN	Pred. Conf.	Baseline	Min. Dist.	Avg. Dist.	KNN	Pred. Conf.	Baseline
Abalone	835	10	0.129	0.08	0.076	0.401	0.042	0.04	0.056	0.059	0.326	0.048	0.042
Ada Prior	912	100	0.106	0.519	0.442	4.615	0.245	0.238	0.407	0.402	4.626	0.064	0.064
Adult	2442	105	0.469	1.061	1.03	16.023	0.385	0.381	1.074	1.075	20.143	0.155	0.138
Bank 32 nh	1229	32	0.286	0.152	0.158	1.773	0.128	0.127	0.104	0.105	2.041	0.08	0.08
Breast Cancer	3937	9	0.719	0.158	0.156	0.913	0.079	0.075	0.29	0.279	1.245	0.228	0.22
Churn	1000	32	0.192	0.112	0.11	1.357	0.077	0.08	0.088	0.081	1.725	0.069	0.067
Credit Card Fraud	7120	30	0.195	0.574	0.428	36.168	0.315	0.319	3.384	3.333	47.278	3.379	3.339
Delta Ailerons	1782	5	0.407	0.08	0.069	0.218	0.033	0.027	0.096	0.097	0.23	0.078	0.078
Delta Elevators	1903	6	0.109	0.088	0.082	0.315	0.03	0.031	0.098	0.108	0.31	0.084	0.083
Electricity	4531	14	0.899	0.242	0.268	1.818	0.118	0.11	0.804	0.803	2.051	0.741	0.722
Elevators	2490	18	0.878	0.151	0.136	3.015	0.095	0.093	0.135	0.138	2.805	0.109	0.1
Higgs	4902	28	2.104	0.408	0.311	16.223	0.229	0.283	1.257	1.24	16.784	1.201	1.163
JM1	1088	21	0.143	0.105	0.094	0.941	0.081	0.073	0.065	0.068	0.854	0.037	0.034
Madelon	520	500	0.144	1.484	1.439	16.78	1.305	1.335	0.202	0.197	14.767	0.035	0.031
Magic Telescope	1902	10	0.263	0.088	0.092	0.476	0.044	0.041	0.111	0.102	0.576	0.082	0.083
Mozilla4	947	38	0.086	0.145	0.135	1.534	0.095	0.092	0.064	0.058	1.452	0.044	0.036
MC1	994	38	0.023	0.111	0.102	1.605	0.089	0.09	0.065	0.064	1.489	0.052	0.042
Numerai28.6	4816	21	2.551	0.368	0.274	11.788	0.198	0.151	0.528	0.531	11.827	0.463	0.453
PC2	1118	36	0.026	0.11	0.104	1.562	0.086	0.082	0.068	0.069	1.409	0.053	0.048
Phishing	1105	68	0.082	0.235	0.231	3.35	0.207	0.211	0.08	0.072	2.6	0.056	0.056
Phonemes	811	5	0.123	0.031	0.03	0.127	0.023	0.017	0.053	0.049	0.126	0.038	0.036
Pollen	770	5	0.047	0.032	0.022	0.116	0.015	0.013	0.057	0.056	0.159	0.047	0.04
Satellite	1147	36	0.049	0.115	0.118	1.814	0.082	0.081	0.09	0.082	1.855	0.075	0.072
Scene	602	304	0.048	0.825	0.801	9.984	0.731	0.699	0.361	0.339	12.919	0.058	0.056
Spambase	690	57	0.16	0.162	0.14	1.707	0.144	0.126	0.074	0.07	2.751	0.056	0.054
Speed Dating	1257	500	0.231	2.285	2.335	39.266	2.047	2.018	0.945	0.851	46.444	0.116	0.114
Telco Customer Churn	1056	45	0.386	0.117	0.121	1.891	0.102	0.1	0.088	0.082	2.501	0.064	0.062
Tic Tac Toe	4921	27	1.126	0.385	0.347	14.936	0.259	0.198	0.363	0.352	20.077	0.29	0.289
Vehicle sensIT	9853	100	5.994	4.558	4.098	262.276	1.729	1.78	9.137	9.03	323.104	5.767	5.735
Waveform-5000	1000	40	0.194	0.542	0.172	3.286	0.159	0.156	0.096	0.084	2.503	0.07	0.068
Average rank	{	{	6.517	6.9	5.967	10.33	4	3.467	6.1	5.65	10.6	3.65	2.82

con dence level c, then the size of the generated intervals becomes the metric of how useful the predictions are, where tighter intervals are more informative. The scale of the importance scores generated di ers between datasets. Therefore, the interval sizes displayed in Table 2, as well as Tables 3 and 4 in Appendix A, are normalized by the di erence between each feature's maximum and minimum importance values, as generated by the TreeSHAP explainer.

We display the results averaged across all importance scores within each dataset in Table 2, where the models are obtained through XGBoost and MLP using the proposed non-conformity measures in subsection 4.3.

The Friedman test is applied to test the null hypothesis that there is no signi cant difference in the resulting interval sizes produced using XGBoost and MLP with the di erent non-conformity measures. The Friedman test rejects the null hypothesis at the 0.05 leve, meaning there is a signi cant di erence in the produced interval sizes. The post-hoc Nemenyi test is applied to determine the signi cant pairwise di erences, which are summarized in Figure 2. The results show a signi cant di erence between XGBoost and MLP. However, no signi cant di erence in interval sizes is observed between the non-conformity measures applied to any of the two algorithms.

Since, in many cases, the importance scores are not evenly distributed over features, and the importance scores of a few top features are remarkably higher than the remaining ones, it can be su cient for the user to use the top features in order to explain the prediction. Therefore, we also display the detailed results using the top 10 and 5 features. The importance scores of each feature are averaged over the test set, and the top 10 and top 5 features are selected using the average values. The average size of the intervals of the top 10 and 5 features are reported in Table 3 and Table 4 in Appendix A, respectively. Friedman's and Nemenyi's pairwise signi cance tests are also applied, and the results are similar to those reported using Table 2. The pairwise tests are summarized in Figure 13 and Figure 14 in Appendix A, respectively.

The gures from 3 to 12 illustrate the predicted intervals using both XGBoost and MLP with the proposed di culty estimates on one data instance from the Elevators dataset. The bars in the gures are the importance scores generated by the underlying explainer, and the predicted intervals are added to each importance score bar. We plot only the top 10 important features for ease of presentation.

6. Concluding Remarks

We have proposed a method to approximate any explanation technique that produces explanations in the form of additive feature importance scores in order to reduce the computational cost of such techniques. We also applied the conformal prediction framework to provide validity guarantees on the approximated explanations. Moreover, a set of non-conformity measures is also proposed to keep the computational cost needed to estimate the sample di culty low using, for example, KNN. We have presented results from a large-scale empirical investigation comparing two di erent algorithms (XGBoost and multi-layer perceptron) using the proposed non-conformity measures. The performance of proposed non-conformity measures is compared with the results when KNN is used and when no di culty estimate is applied. The results show no signi cant di erence between any of the

compared non-conformity measures with respect to the interval size. However, the proposed difficulty estimates significantly outperform the ones using KNN with respect to the execution time. The results also show that using an XGBoost regressor per feature can produce a more accurate approximation than using one multi-target MLP regressor for all features.

A possible direction for future work is to apply the conformal prediction framework to provide validity guarantees for entire explanations instead of providing validity guarantees per feature. Accordingly, an approach similar to conformal multi-target regression (Messoudi et al., 2020) can be applied.

Figure 2: The average rank of the compared regression models on the 30 datasets with respect to the interval size of all features (a lower rank is better), where the critical difference (CD) represents the largest difference that is not statistically significant.

Acknowledgments

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

Table 2: The average con dence interval sizes using all features . The generated intervals cover the true importance scores, as predicted by the underlying explainer, with 0.95 con dence. The compared algorithms are XGBoost and MLP, with di erent non-conformity measures. The best-performing model is colored in blue, and the second best-performing is colored in light blue.

Dataset	XGBoost					Multi-Layer Perceptron				
	Min. Dist.	Avg. Dist.	KNN	Pred. Conf.	Baseline	Min. Dist.	Avg. Dist.	KNN	Pred. Conf.	Baseline
Abalone	0.219	0.116	0.075	0.089	0.094	0.528	0.304	0.174	0.223	0.235
Ada Prior	0.027	0.027	0.023	0.028	0.027	12.766	12.593	12.384	13.845	12.543
Adult	0.018	0.018	0.018	0.02	0.019	3.886	3.824	3.613	4.459	3.8
Bank 32 nh	0.191	0.192	0.216	0.194	0.193	0.32	0.321	0.364	0.333	0.319
Breast Cancer	0.054	0.07	0.05	0.065	0.066	0.147	0.195	0.097	0.126	0.128
Churn	0.102	0.1	0.1	0.096	0.1	24.668	24.683	22.931	25.098	25.484
Credit Card Fraud	0.053	0.053	0.057	0.053	0.053	0.306	0.303	0.234	0.207	0.207
Delta Ailerons	0.153	0.067	0.054	0.061	0.066	0.175	0.144	0.137	0.133	0.141
Delta Elevators	0.082	0.08	0.073	0.079	0.08	0.154	0.151	0.152	0.158	0.154
Electricity	0.103	0.103	0.102	0.104	0.103	0.181	0.181	0.191	0.185	0.181
Elevators	0.064	0.063	0.063	0.064	0.063	6.109	6.107	5.043	5.776	5.346
Higgs	0.084	0.084	0.092	0.09	0.086	0.133	0.133	0.143	0.141	0.132
JM1	0.119	0.12	0.099	0.115	0.12	0.39	0.256	0.198	0.215	0.218
Madelon	0.128	0.128	0.127	0.127	0.128	2.775	2.77	2.752	2.967	2.792
Magic Telescope	0.1	0.1	0.107	0.105	0.101	0.271	0.209	0.195	0.182	0.172
Mozilla4	0.026	0.024	0.015	0.024	0.024	0.995	0.998	0.961	1.155	1.158
MC1	0.024	0.026	0.02	0.025	0.026	4.837	4.998	3.746	5.034	5.116
Numerai28.6	0.065	0.065	0.064	0.067	0.067	0.121	0.122	0.116	0.123	0.122
PC2	0.012	0.012	0.01	0.012	0.012	7.168	6.066	4.376	5.411	5.475
Phishing	0.03	0.029	0.025	0.03	0.029	20.439	20.449	20.024	20.969	20.484
Phonemes	0.176	0.172	0.182	0.16	0.166	0.296	0.258	0.281	0.252	0.245
Pollen	0.11	0.111	0.096	0.116	0.114	0.198	0.2	0.168	0.221	0.212
Satellite	0.031	0.031	0.022	0.03	0.03	2.655	2.265	1.404	1.763	1.777
Scene	0.096	0.095	0.087	0.091	0.091	10.161	9.831	9.224	9.765	9.708
Spambase	0.065	0.064	0.054	0.063	0.063	5.169	5.474	3.996	4.109	4.102
Speed Dating	0.028	0.028	0.028	0.028	0.028	4.104	4.107	4.047	4.414	4.095
Telco Customer Churn	0.033	0.033	0.029	0.033	0.033	5.689	5.68	5.358	5.718	5.607
Tic Tac Toe	0.037	0.036	0.027	0.032	0.032	0.032	0.032	0.027	0.033	0.031
Vehicle sensIT	0.096	0.098	0.098	0.109	0.1	0.199	0.205	0.202	0.221	0.201
Waveform-5000	0.185	0.178	0.188	0.176	0.18	0.399	0.392	0.396	0.371	0.364
Average rank	3.75	3.333	2.083	3.25	3.3	8.483	8.25	6.65	8.45	7.45

Figure 3: XGBoost with Min. Dist.

Figure 4: XGBoost with Avg. Dist.

Figure 5: XGBoost with KNN

Figure 6: MLP with Min. Dist.

Figure 7: MLP with Avg. Dist.

Figure 8: MLP with KNN

Figure 9: XGBoost with Prob. Conf.

Figure 10: XGBoost without di culty estimate

Figure 11: MLP with Prob. Conf.

Figure 12: MLP without di culty estimate

References

Daniel W. Apley and Jingyu Zhu. Visualizing the e ects of predictor variables in black box supervised learning models. Journal of the Royal Statistical Society. Series B: Statistical Methodology, 82(4):1059{1086, September 2020. ISSN 1369-7412.

Henrik Bostr om. crepes: a python package for generating conformal regressors and predictive systems. In Proceedings of the Eleventh Symposium on Conformal and Probabilistic Prediction and Applications, volume 179 of Proceedings of Machine Learning Research PMLR, 2022.

Henrik Bostr om, Ram B. Gurung, Tony Lindgren, and Ulf Johansson. Explaining random forest predictions with association rules. Archives of Data Science, Series A (Online First) , 5(1):A05, 20 S. online, 2018. ISSN 2363-9881.

Milton Friedman. A correction: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association, 34 (205):109{109, 1939.

A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In Proceedings of the Fourteenth Conference on Uncertainty in Arti cial Intelligence , UAI'98, page 148{155, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 155860555X.

Neil Jethani, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath. FastSHAP: Real-time shapley value estimation. In International Conference on Learning Representations, 2022.

Ulf Johansson, Henrik Bostr om, Tuve Lo stro m, and Henrik Linusson. Regression conformal prediction with random forests. Mach. Learn., 97(1{2):155{176, oct 2014.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Interpretable & explorable approximations of black box models. CoRR, abs/1707.01154, 2017.

Scott M Lundberg and Su-In Lee. A uni ed approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 4765{ 4774. Curran Associates, Inc., 2017.

Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. CoRR, 2018.

Prasanta Chandra Mahalanobis. On the generalised distance in statistics. Proceedings of the National Institute of Sciences of India, 2(1):49{55, 1936.

Soundouss Messoudi, Sebastien Destercke, and Sylvain Rousseau. Conformal multi-target regression using neural networks. In Alexander Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov, and Giovanni Cherubin, editors, Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications, volume 128 of Proceedings of Machine Learning Research, pages 65{83. PMLR, 09{11 Sep 2020.

Christoph Molnar. Interpretable Machine Learning. 2022.

Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable machine learning { a brief history, state-of-the-art and challenges. In ECML PKDD 2020 Workshops, pages 417{431, Cham, 2020. Springer International Publishing. ISBN 978-3-030-65965-3.

Peter Bjorn Nemenyi. Distribution-free multiple comparisons. PhD thesis, Princeton University, 1963.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alexander Gammerman. Inductive con dence machines for regression. In Proceedings of the 13th European Conference on Machine Learning, ECML '02, page 345{356, Berlin, Heidelberg, 2002. Springer-Verlag. ISBN 3540440364.

Harris Papadopoulos, Vladimir Vovk, and Alex Gammerman. Regression conformal prediction with nearest neighbours. J. Artif. Int. Res. , 40(1):815{840, jan 2011. ISSN 1076-9757.

Carl Rasmussen. The in nite gaussian mixture model. In S. Solla, T. Leen, and K. Muller, editors, Advances in Neural Information Processing Systems, volume 12. MIT Press, 1999.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. In ICML Workshop on Human Interpretability in Machine Learning (WHI) , 2016a.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classi er. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, pages 1135{1144, 2016b.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In AAAI Conference on Arti cial Intelligence (AAAI) , 2018.

Craig Saunders, Alexander Gammerman, and Volodya Vovk. Transduction with con dence and credibility. In Proceedings of the Sixteenth International Joint Conference on Arti cial Intelligence , IJCAI '99, page 722{726, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606130.

Xuelin Situ, Ingrid Zukerman, Cecile Paris, Sameen Maruf, and Gholamreza Ha ari. Learning to explain: Generating stable explanations fast. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 5340{5355, Online, August 2021. Association for Computational Linguistics.

Jacopo Teneggi, Alexandre Luster, and Jeremias Sulam. Fast hierarchical games for image explanations. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. Algorithmic learning in a random world, volume 29. Springer, 2005.

Appendix A. First Appendix

Table 3: The average con dence interval sizes using the top 10 important features. The generated intervals cover the true importance scores, as predicted by the underlying explainer, with 0.95 con dence. The compared algorithms are XGBoost and MLP, with di erent non-conformity measures. The best-performing model is colored in blue, and the second best-performing is colored in light blue.

Dataset	XGBoost					Multi-Layer Perceptron				
	Min. Dist.	Avg. Dist.	KNN	Pred. Conf.	Baseline	Min. Dist.	Avg. Dist.	KNN	Pred. Conf.	Baseline
Abalone	0.219	0.116	0.075	0.089	0.094	0.528	0.304	0.174	0.223	0.235
Ada Prior	0.087	0.088	0.087	0.094	0.088	0.232	0.227	0.243	0.256	0.228
Adult	0.04	0.04	0.039	0.043	0.04	0.101	0.099	0.1	0.112	0.098
Bank 32 nh	0.17	0.17	0.197	0.169	0.17	0.265	0.265	0.307	0.281	0.264
Breast Cancer	0.054	0.07	0.05	0.065	0.066	0.147	0.195	0.097	0.126	0.128
Churn	0.097	0.096	0.102	0.09	0.095	0.196	0.196	0.225	0.197	0.197
Credit Card Fraud	0.061	0.061	0.066	0.06	0.06	0.27	0.266	0.204	0.18	0.18
Delta Ailerons	0.153	0.067	0.054	0.061	0.066	0.175	0.144	0.137	0.133	0.141
Delta Elevators	0.082	0.08	0.073	0.079	0.08	0.154	0.151	0.152	0.158	0.154
Electricity	0.12	0.12	0.125	0.121	0.12	0.203	0.203	0.22	0.208	0.203
Elevators	0.07	0.069	0.073	0.07	0.069	0.156	0.143	0.144	0.139	0.13
Higgs	0.065	0.065	0.073	0.068	0.065	0.101	0.101	0.114	0.107	0.1
JM1	0.14	0.141	0.12	0.134	0.141	0.424	0.265	0.219	0.227	0.228
Madelon	0.247	0.245	0.266	0.23	0.242	0.602	0.604	0.588	0.592	0.588
Magic Telescope	0.1	0.1	0.107	0.105	0.101	0.271	0.209	0.195	0.182	0.172
Mozilla4	0.03	0.036	0.02	0.036	0.035	0.093	0.098	0.068	0.117	0.118
MC1	0.035	0.03	0.02	0.029	0.03	0.123	0.13	0.091	0.142	0.142
Numerai28.6	0.071	0.072	0.069	0.074	0.073	0.129	0.131	0.124	0.13	0.129
PC2	0.021	0.021	0.015	0.02	0.021	0.314	0.291	0.165	0.293	0.295
Phishing	0.049	0.048	0.043	0.047	0.046	0.059	0.059	0.057	0.061	0.059
Phonemes	0.176	0.172	0.182	0.16	0.166	0.296	0.258	0.281	0.252	0.245
Pollen	0.11	0.111	0.096	0.116	0.114	0.198	0.2	0.168	0.221	0.212
Satellite	0.026	0.025	0.017	0.024	0.024	0.459	0.374	0.209	0.292	0.293
Scene	0.182	0.177	0.149	0.16	0.165	0.478	0.462	0.422	0.432	0.433
Spambase	0.104	0.103	0.1	0.101	0.102	0.439	0.44	0.349	0.325	0.325
Speed Dating	0.113	0.113	0.127	0.113	0.113	0.319	0.318	0.366	0.322	0.316
Telco Customer Churn	0.066	0.066	0.057	0.066	0.066	0.16	0.162	0.139	0.159	0.159
Tic Tac Toe	0.04	0.039	0.029	0.034	0.035	0.028	0.028	0.024	0.03	0.028
Vehicle sensIT	0.09	0.091	0.093	0.104	0.093	0.159	0.168	0.168	0.187	0.165
Waveform-5000	0.136	0.129	0.14	0.126	0.129	0.218	0.214	0.215	0.2	0.195
Average rank	3.77	3.45	2.7	2.97	3.08	8.45	8.03	7.27	8.08	7.2

Table 4: The average con dence interval sizes using the top 5 important features. The generated intervals cover the true importance scores, as predicted by the underlying explainer, with 0.95 con dence. The compared algorithms are XGBoost and MLP, with di erent non-conformity measures. The best-performing model is colored in blue, and the second best-performing is colored in light blue.

Dataset	XGBoost					Multi-Layer Perceptron				
	Min. Dist.	Avg. Dist.	KNN	Pred. Conf.	Baseline	Min. Dist.	Avg. Dist.	KNN	Pred. Conf.	Baseline
Abalone	0.201	0.115	0.069	0.085	0.09	0.491	0.281	0.142	0.174	0.184
Ada Prior	0.08	0.08	0.085	0.083	0.08	0.22	0.217	0.244	0.238	0.214
Adult	0.028	0.028	0.027	0.03	0.028	0.077	0.076	0.077	0.086	0.075
Bank 32 nh	0.111	0.115	0.136	0.113	0.115	0.161	0.161	0.198	0.172	0.16
Breast Cancer	0.051	0.063	0.048	0.057	0.059	0.149	0.171	0.091	0.11	0.112
Churn	0.105	0.104	0.104	0.098	0.101	0.188	0.188	0.206	0.19	0.188
Credit Card Fraud	0.062	0.062	0.067	0.061	0.061	0.252	0.248	0.195	0.17	0.17
Delta Ailerons	0.153	0.067	0.054	0.061	0.066	0.175	0.144	0.137	0.133	0.141
Delta Elevators	0.079	0.076	0.074	0.075	0.076	0.148	0.141	0.151	0.145	0.144
Electricity	0.15	0.15	0.175	0.15	0.15	0.253	0.253	0.293	0.26	0.253
Elevators	0.052	0.051	0.056	0.052	0.051	0.098	0.093	0.091	0.088	0.083
Higgs	0.062	0.062	0.07	0.066	0.063	0.103	0.102	0.117	0.108	0.101
JM1	0.14	0.14	0.126	0.136	0.14	0.409	0.243	0.213	0.211	0.211
Madelon	0.226	0.224	0.242	0.213	0.222	0.615	0.617	0.593	0.608	0.603
Magic Telescope	0.073	0.073	0.077	0.076	0.073	0.159	0.13	0.119	0.112	0.106
Mozilla4	0.035	0.034	0.019	0.034	0.034	0.097	0.101	0.071	0.121	0.122
MC1	0.034	0.035	0.023	0.035	0.035	0.118	0.125	0.089	0.136	0.137
Numerai28.6	0.07	0.071	0.068	0.073	0.072	0.126	0.128	0.121	0.127	0.126
PC2	0.021	0.02	0.015	0.02	0.021	0.266	0.259	0.157	0.252	0.252
Phishing	0.056	0.055	0.05	0.054	0.053	0.057	0.058	0.056	0.059	0.058
Phonemes	0.176	0.172	0.182	0.16	0.166	0.296	0.258	0.281	0.252	0.245
Pollen	0.11	0.111	0.096	0.116	0.114	0.198	0.2	0.168	0.221	0.212
Satellite	0.027	0.027	0.019	0.025	0.026	0.364	0.323	0.181	0.264	0.264
Scene	0.15	0.143	0.102	0.131	0.134	0.277	0.277	0.22	0.261	0.256
Spambase	0.09	0.09	0.075	0.09	0.09	0.32	0.297	0.249	0.257	0.25
Speed Dating	0.11	0.11	0.127	0.111	0.109	0.312	0.311	0.358	0.315	0.308
Telco Customer Churn	0.062	0.062	0.054	0.063	0.063	0.155	0.156	0.131	0.158	0.155
Tic Tac Toe	0.037	0.037	0.027	0.031	0.032	0.027	0.027	0.022	0.029	0.027
Vehicle sensIT	0.071	0.072	0.073	0.082	0.074	0.119	0.126	0.123	0.14	0.122
Waveform-5000	0.142	0.136	0.151	0.133	0.136	0.225	0.221	0.218	0.2	0.191
Average rank	3.78	3.33	2.7	3.07	3.12	8.43	8.2	7.3	8.1	6.97