DALE: Differential Accumulated Local Effects for efficient and accurate global explanations

Abstract

Accumulated Local Effect (ALE) is a method for accurately estimating feature effects, overcoming fundamental failure modes of previously-existed methods, such as Partial Dependence Plots. However, the *approximation* method for ALE faces two weaknesses. Firstly, it does not scale well in cases where the input has high dimensionality, and, secondly, it is vulnerable to out-of-distribution (OOD) sampling in cases with limited training examples. In this paper, we propose a novel ALE approximation, called Differential Accumulated Local Effects (DALE), a fast and accurate alternative for estimating feature effects in cases where the ML model is differentiable and an auto-differentiable framework is accessible. Our proposal has significant computational advantages. Specifically, the computational complexity for computing the feature effect for all attributes using DALE is similar to what ALE requires for a single attribute. DALE, therefore, makes feature effect estimation applicable to high-dimensional Machine Learning scenarios with near-zero computational overhead. Furthermore, unlike ALE, our proposal does not create artificial points for calculating the feature effect. The approximation is exclusively based on the training examples, resolving misleading estimations due to OOD sampling. Finally, we formally prove that, under some hypotheses, DALE is an unbiased estimator of ALE and we present a method for quantifying the uncertainty of the explanation. Experiments using both synthetic and real datasets demonstrate the value of the proposed approach. Code to reproduce these experiments is provided in the submission and will become publiclyavailable upon acceptance.

Keywords: Feature Effect; Explainable AI; Differentiable Models; Neural Networks

1. Introduction

Recently, Machine Learning (ML) models have flourished in critical, high-stakes application domains, such as healthcare and finance. These fields require methods with the ability to explain their predictions, i.e., justify why a specific outcome has emerged. However, several types of accurate and highly non-linear models like Deep Neural Networks do not meet this requirement. Therefore, there is a growing need for explainability methods for interpreting such "black-box" models. Feature effect forms a fundamental category of global explainability methods (i.e. characterizing the model as a whole, not a particular input). The goal of the feature effect is to isolate the average impact of a single feature on the output. This class of methods is attractive due to the simplicity of the explanation that is easily understandable by a non-expert.

There are three popular feature effect methods: (i) Partial Dependence Plots (PDPlots) (Friedman, 2001), (ii) Marginal Plots (MPlots) (Apley and Zhu, 2020) and (iii) Aggregated Local Effects (ALE) (Apley and Zhu, 2020). PDPlots and MPlots assume that input features are not correlated. When this does not hold, both methods lead to misestimation; PDPlots

quantify the effect by marginalizing over out-of-distribution (OOD) synthetic instances, and MPlots yield aggregated effects on single features. Therefore, both methods perform well only in independent or low-correlated features. ALE is the only feature effect method that succeeds in staying on distribution and isolating feature effects in situations where input features are highly correlated.[1] However, in most cases, it is impossible to compute ALE through its definition since this would require (a) solving a high-dimensional integral, which is infeasible, and (b) evaluating the data generating distribution, which is usually unknown. Therefore, Apley and Zhu (2020) proposed an estimating ALE with a Monte-Carlo approximation. This approximation faces two weaknesses. First, it becomes computationally inefficient in cases of datasets with numerous high-dimensional instances. Second, it is still vulnerable to OOD sampling in cases of wide bin sizes.

This paper proposes Differential Aggregated Local Effects (DALE), a novel approximation for ALE that resolves both weaknesses. DALE leverages auto-differentiation for computing the derivatives wrt each instance in a single pass. Therefore, it scales well in the case of high-dimensional inputs, large training sets and expensive black-box models. Furthermore, DALE estimates the feature effect using only the examples from the training set, securing that the estimation is not affected by OOD samples. The contributions of this work are:

- We introduce DALE, a novel approximation to efficiently create ALE plots on differentiable black-box models. DALE is more efficient than the traditional ALE approximation, scales much better to high-dimensional datasets, and avoids OOD sampling.
- We formally prove that DALE is an unbiased estimator of ALE and quantify the standard error of the approximation.
- We experiment with synthetic and real datasets, showing that DALE: (a) scales in all cases better than ALE, (b) provides a better approximation compared to ALE, especially in cases of wide bin sizes

2. Related Work

Explainable AI (XAI) is a fast-evolving field with a growing interest. In recent years, the domain has matured by establishing its terminology and objectives (Hoffman et al., 2018). Several surveys have been published (Barredo Arrieta et al., 2020), (Adadi and Berrada, 2018) classifying the different approaches and detecting future challenges on the field (Molnar et al., 2020a).

There are several criterias for grouping XAI methods. A very popular distinction is between local and global ones. Local interpretability methods explain why a model made a specific prediction given a specific input. For example, local surrogates such as LIME (Ribeiro et al., 2016) train an explainable-by-design model in data points generated from a local area around the input under examination. SHAP values (Lundberg and Lee, 2017) measures the contribution of each attribute in a specific prediction, formulating a game-theoretical framework based on Shapley Values. Counterfactuals (Wachter et al., 2017) search for a data point as close as possible to the examined input that flips the prediction. Anchors (Ribeiro

1. In Section 3, we provide a thorough analysis for clarifying the differences between these three approaches.

et al., 2018) provide a rule, i.e. a set of attribute values, that is enough to freeze the prediction, independently of the value of the rest of the attributes.

Global methods, which is the focus of this paper, explain the average model behaviour. For example, prototypes (Gurumoorthy et al., 2019) search for a data point that is a characteristic representative of a specific class. Criticisms (Kim et al., 2016), search for data points whose class is ambiguous. Global feature importance methods characterize each input feature by assigning to it an importance score. Permutation feature importance (Fisher et al., 2019) measures the change in the prediction score of a model, after permuting the value of each feature. Often, apart from knowing that a feature is important, it also valuable to know the type of the effect on the output (positive/negative). Feature effect methods take a step further and quantify the type of a each feature attribute influences the output on average. There are three popular feature effect techniques Partial Dependence Plots (Friedman, 2001), Marginal Plots and ALE (Apley and Zhu, 2020). Another class of global explanation techniques measures the interaction (Friedman and Popescu, 2008) between features. Feature interaction quantifies to what extent the effect of two variables on the output comes is because of their combination. Friedman and Popescu (2008) proposed a set of appropriate visualizations for such interactions. The generalization of feature effect and variable interactions is functional decomposition (Molnar et al., 2020b), that decomposes the black-box function into a set of simpler ones that may include more than two features.

3. Background

This section introduces the reader to three popular feature effect methods; PDPlots, MPlots and ALE.

Notation. We refer to random variables (r.v.) using uppercase and calligraphic font $\mathcal{X}$, whereas to simple variables with plain lowercase x. Bold $\mathbf{x}$ denotes a vector variable, $\mathcal{X}_s$ the r.v. of the feature of interest and $\mathcal{X}_c$ the rest of the features so that $\mathcal{X} = (\mathcal{X}_s, \mathcal{X}_c)$ represents the input space. The black-box function is notated as f and the feature effect of the s-th feature as $f_{<\texttt{method}>}(x_s)$, where $<\texttt{method}>$ is the name of the feature effect method.[2]

Feature Effect Methods. PDPlots formulate the feature effect of the s-th attribute as an expectation over the marginal distribution $\mathcal{X}_c$, i.e., $f_{\texttt{PDP}}(x_s) = \mathbb{E}_{\mathcal{X}_c}[f(x_s, \mathcal{X}_c)]$. MPlots formulate it as an expectation over the conditional $\mathcal{X}_c|\mathcal{X}_s$, i.e., $f_{\texttt{MP}}(x_s) = \mathbb{E}_{\mathcal{X}_c|\mathcal{X}_s=x_s}[f(x_s, \mathcal{X}_c)]$. ALE computes the global effect at x_s as an integration of local effects. The local effects are measured as the expected change on the output $\frac{\partial f(x_s, \mathcal{X}_c)}{\partial x_s}$ over the conditional distribution $\mathcal{X}_c|\mathcal{X}_s$. The formula that defines ALE is presented below:

$$f_{\texttt{ALE}}(x_s) = c + \int_{-\infty}^{x_s} \mathbb{E}_{\mathcal{X}_c|\mathcal{X}_s=z}\left[\frac{\partial f(z, \mathcal{X}_c)}{\partial z}\right] \partial z \tag{1}$$

The constant c is used for centering the ALE plot. For illustrating the differences between the methods and the superiority of ALE, we provide a toy example. We select a bivariate black-box function f with correlated features; the first feature x_1 follows a uniform distribution

2. An extensive list of all symbols used in the paper is provided in the helping material.

$x_1 \sim \mathcal{U}(0,1)$ and the second feature gets the value of x_1 in a deterministic way, i.e., $x_2 = x_1$. The black-box function is the following piece-wise linear mapping:

$$f(x_1, x_2) = \begin{cases} 1 - x_1 - x_2 & x_1 + x_2 \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Due to the piece-wise linear form, it is easy to isolate the effect of x_1; Insider the region $0 \leq x_1 \leq 0.5$, the effect is linear, i.e., $-x_1$, and outside it is constant, i.e. the effect does not depend on x_1. The closed-form solution for each method is presented below: [3] [4]

$$f_{\texttt{PDP}}(x_1) = \mathbb{E}_{\mathcal{X}_2}[f(x_1, \mathcal{X}_2)] = \frac{(1 - x_1)^2}{2}, \; \forall x_1 \in [0,1] \tag{3}$$

$$f_{\texttt{MP}}(x_1) = \mathbb{E}_{\mathcal{X}_2|\mathcal{X}_1 = x_1}[f(x_1, \mathcal{X}_2)] = \begin{cases} 1 - 2x_1 & x_1 \leq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$f_{\texttt{ALE}}(x_1) = c + \int_{z_0}^{x_1} \mathbb{E}_{\mathcal{X}_2|\mathcal{X}_1 = z}[\frac{\partial f(z, \mathcal{X}_2)}{\partial z}]\partial z = \begin{cases} c - x_1 & 0 \leq x_1 \leq 0.5 \\ c - 0.5 & 0.5 \leq x_1 \leq 1 \end{cases} \tag{5}$$

The effect computed in Eqs. (3), (4) helps us understand that PDPlots and MPlots provide misleading results in cases of correlated features. PDPlots integrate over unrealistic instances due to the use of the marginal distribution $p(\mathcal{X}_1)$. Therefore, they incorrectly result in a quadratic effect in the region $x_1 \in [0,1]$. MPlots resolve this issue using the conditional distribution $\mathcal{X}_2|\mathcal{X}_1$ but suffer from computing combined effects. In the linear subregion, the effect is overestimated as $-2x_1$ which is the combined effect of both x_1 and x_2. As Eq. (5) shows, ALE resolves both issues and provides the correct effect.

In real scenarios, we cannot obtain a solution directly from Eq. (1). Therefore, Apley and Zhu (2020) proposed a solution by splitting the x_s axis into bins, computing the local effects inside each bin with a Monte Carlo approximation, and, finally, averaging the bin effects. As we discuss extensively in Sections 4.2 and 4.3, this approximation does not scale well to high-dimensional datasets and is vulnerable to OOD sampling. The following section presents DALE, a novel approximation for resolving both issues.

4. Differential Accumulated Local Effects (DALE)

In this section, we present DALE. First, we formulate the expression for the first and second-order DALE and, then, we explain its computational benefits and its robustness to OOD sampling. Finally, we quantify the standard error of the DALE estimation.

4.1. Method presentation

As briefly discussed in Section 3, in most cases it is infeasible to compute ALE analytically. Therefore, Apley and Zhu (2020) proposed the following approximation that is based on the instances of the training set:

3. Detailed derivations can be found in the hepling material.
4. Due to symmetry, for each method, the effect for x_2 is the same with the effect of x_1

$$\hat{f}_{\texttt{ALE}}(x_s) = \sum_{k=1}^{k_x} \frac{1}{|\mathcal{S}_k|} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k} [f(z_k, \mathbf{x}^i_{\mathbf{c}}) - f(z_{k-1}, \mathbf{x}^i_{\mathbf{c}})] \tag{6}$$

We denote as $\mathbf{x}^i$ the i-th example of the training set and as x^i_s its s-th feature. k_x is the index of the bin x_s belongs to, i.e. $k_x : z_{k_x-1} \leq x_s < z_{k_x}$ and $\mathcal{S}_k$ is the set of points that lie in the k-th bin, i.e. $\mathcal{S}_k = \{\mathbf{x}^i : z_{k-1} \leq x^i_s < z_k\}$. For understanding Eq. (6) better, we split it in three levels: Instance effect is the effect computed on the i-th example, i.e. $\Delta f_i = f(z_k, \mathbf{x}^i_{\mathbf{c}}) - f(z_{k-1}, \mathbf{x}^i_{\mathbf{c}})$, local effect is the effect computed on the k-th bin, i.e. $\frac{1}{|\mathcal{S}_k|} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k} \Delta f_i$, and global effect is ALE approximation $\hat{f}_{\texttt{ALE}}(x_s)$. The approximation splits the axis into K equally-sized bins and computes a single local effect in each bin. The local effect is estimated by averaging the instance effects of the instances that lie in each bin. Then, the global effect is the accumulation of the local effects. In ALE approximation, the number of bins (hyperparameter K), has two important consequences; (a) defines the resolution of the ALE plot and (b) affects the instance effects, which are computed at the bin limits.

This approach has some weaknesses. Firstly, it is computationally demanding since it evaluates f for $2 \cdot N \cdot D$ artificial samples, where N is the number of samples in the dataset and D is the number of features. Secondly, it is vulnerable to OOD sampling when the bin's length becomes large. Finally, artificial samples generation makes the whole computation usable only for a predefined bin length; altering the bin size for assessing the feature effect at a different resolution, requires all computations to be repeated from scratch.

4.1.1. First-order DALE.

To address these drawbacks, we propose Differential Accumulated Feature Effect (DALE) that exploits the partial derivatives without altering the data points. The following formula describes the first-order DALE approximation:

$$f_{\texttt{DALE}}(x_s) = \Delta x \sum_{k=1}^{k_x} \frac{1}{|\mathcal{S}_k|} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k} [f_s(\mathbf{x}^i)] = \Delta x \sum_{k=1}^{k_x} \hat{\mu}_k \tag{7}$$

where Δx is the bin size and f_s the partial derivative wrt x_s, i.e. $f_s(\mathbf{x}^i) = \frac{\partial f(\mathbf{x}^i)}{\partial x_s}$. We use $\hat{\mu}^s_k = \frac{1}{|\mathcal{S}_k|} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k} [f_s(\mathbf{x}^i)]$ to indicate the estimated local effect at the k-th bin.

DALE uses only the dataset samples and doesn't perturb any feature, securing that we estimate the local effect from on-distribution (observed) data points. In Eq. (7), the estimation of the local effect at each training sample is independent from the bin size. In contrast with ALE approximation, the number of the bins (hyperparameter K) affects only the resolution of the plot and **not** the instance effects. Finally, DALE enables computing the local effects $f_s(\mathbf{x}^i)$ for $s = \{1, \ldots, D\}$, and $i = \{1, \ldots, N\}$ once, and reusing them to produce ALE plots of different resolutions. Therefore, the user may experiment with feature effect plots at many different resolutions, with near-zero computational cost.

4.1.2. Second-order DALE.

ALE Apley and Zhu (2020) also provide a formula for approximating the combined effect of a pair of attributes x_l, x_m:[5]

$$\hat{f}_{\texttt{ALE}}(x_l, x_m) = \sum_{p=1}^{p_x}\sum_{q=1}^{q_x} \frac{1}{|\mathcal{S}_{p,q}|} \sum_{i:\mathbf{x}^i \in \mathcal{S}_{p,q}} \Delta^2 f_i \tag{8}$$

where $\Delta^2 f_i = [f(z_p, z_q, \mathbf{x}_c) - f(z_{p-1}, z_q, \mathbf{x}_c)] - [f(z_p, z_{q-1}, \mathbf{x}_c) - f(z_{p-1}, z_{q-1}, \mathbf{x}_c)]$. As before, instead of evaluating the second-order derivative at the limits of the grid, we propose accessing the second-order derivatives on the data points. The following formula describes the second-order DALE approximation:

$$f_{\texttt{DALE}}(x_l, x_m) = \Delta x_l \Delta x_m \sum_{p=1}^{p_x}\sum_{q=1}^{q_x} \frac{1}{|\mathcal{S}_{p,q}|} \sum_{i:\mathbf{x}^i \in \mathcal{S}_{p,q}} f_{l,m}(\mathbf{x}^i) = \Delta x_l \Delta x_m \sum_{p=1}^{p_x}\sum_{q=1}^{q_x} \hat{\mu}^s_{p,q} \tag{9}$$

where $f_{l,m}(\mathbf{x})$ is the second-order derivative evaluated at $\mathbf{x}^i$, i.e. $f_{l,m}(\mathbf{x}) = \dfrac{\partial^2 f(x)}{\partial x_l \partial x_m}$, and Δx_l, Δx_m correspond to the bin step for features x_l and x_m, respectively. As in the first-order description, we use $\hat{\mu}^s_{p,q} = \frac{1}{|\mathcal{S}_{p,q}|}\sum_{i:\mathbf{x}^i \in \mathcal{S}_{p,q}} f_{l,m}(\mathbf{x}^i)$ to express the local effect at the bin (p, q). DALE second-order approximation has the same advantages over ALE as in the first-order case; it is faster, protected from OOD sampling and permits multi-resolution plots, with near-zero additional cost.

4.2. Computational Benefit

DALE approximation has significant computational advantages, especially in cases of high-dimensional input space. For estimating the feature effect of all features, our approach processes N data points, namely, the examples of the training set. In contrast, ALE approximation generates and processes $2 \cdot N \cdot D$ artificial data points. The difference by a factor of D has major implications in the computational complexity and the memory requirements. Our method scales nicely in problems with high dimensionality, as is the case in most Deep Learning setups. Our approach is built on the computation of the Jacobian matrix,

$$\mathbf{J} = \begin{bmatrix} \nabla_{\mathbf{x}} f(\mathbf{x}^1) \\ \vdots \\ \nabla_{\mathbf{x}} f(\mathbf{x}^N) \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{x}^1) & \dots & f_D(\mathbf{x}^1) \\ \vdots & \ddots & \vdots \\ f_1(\mathbf{x}^N) & \dots & f_D(\mathbf{x}^N) \end{bmatrix} \tag{10}$$

where, as before, $f_s(\mathbf{x}^i)$ is the partial derivative of the s-th feature evaluated at the i-th training point. Automatic differentiation enables the computation of the gradients w.r.t. all features in a single pass. Computing the gradient vector for a training example $\mathbf{x}^i$ w.r.t all features $\nabla_{\mathbf{x}} f(\mathbf{x}^i) = [f_1(\mathbf{x}), \dots, f_D(\mathbf{x})]$ is computationally equivalent to evaluating $f(\mathbf{x}^i)$. Based on this observation, computing the whole Jacobian matrix costs $\mathcal{O}(N)$. In constrast, in

5. For completness, we provide the second-order ALE definition in the helping material.

ALE, the evaluation of f for $N \cdot D$ times costs $\mathcal{O}(N \cdot D)$. Our method, also, takes advantage of all existing automatic differentiation frameworks which are optimized for computing the gradients efficiently.[6] In Algorithm 1, we present DALE in an algorithm form. The algorithm needs as input: (a) the black-box function f, (b) the derivative of $\nabla_{\mathbf{x}} f$ and (c) the dataset $\mathbf{X}$.[7] The parameter K defines the resolution of the DALE plot. The algorithm returns a matrix $\mathbf{A}$, where the cell $\mathbf{A}_{s,j}$ contains the effect of the j-th bin of the s-th feature, i.e., $f^s_{\texttt{DALE}}(x) = \mathbf{A}_{s,k_x}$. Steps 3-5 iterate over each attribute, therefore these steps have complexity $\mathcal{O}(N \cdot D)$. However these steps involve relatively cheap operations (allocation, averaging and aggregation) in comparison with the computation of the Jacobian matrix. Finally, with matrix $\mathbf{A}$ computed, evaluating $f_{\texttt{DALE}}(x)$ requires only locating the bin k_x that x belongs to.

Algorithm 1 DALE aproximation

Input: $f, \nabla_{\mathbf{x}} f, \mathbf{X}$
Parameter: K
Output: $\mathbf{A}$

1: Compute the Jacobian $\mathbf{J}$ of Eq. (10)
2: **for** $s = 1, \ldots, D$ **do**
3: Allocate points $\Rightarrow \mathcal{S}_k \forall k$
4: Estimate local effect $\Rightarrow \hat{\mu}^s_k \forall k$ of Eq. (7)
5: Aggregate $\Rightarrow \mathbf{A}_{s,j} = \Delta x \sum_{k=1}^{j} \hat{\mu}^s_k, j = 1, \ldots, K$
6: **end for**
7: **return A** || Note that $f_{\texttt{DALE}}(x) = \mathbf{A}_{s,k_x}$

4.3. Robustness to out-of-distribution sampling

OOD sampling is the source of failure in many explainability methods that perturb features (Hooker et al., 2021). ALE is vulnerable to OOD sampling when the bin length is relatively big, or, equivalently, when the number of bins (hyperparameter K) is relatively small. We use the word *relatively* to indicate that the threshold for characterizing a bin as big/small depends on the properties of the black-box function, i.e., how quickly it diverges outside of the data manifold. ML models learn to map $\mathbf{x} \to y$ only in the manifold of the data generating distribution $\mathcal{X}$. Therefore, the black-box function f can take any arbitrary form away from $\mathcal{X}$ without any increase in the training loss. On the other hand, when a limited number of samples is available, it maybe necessary to lower K to ensure a robust estimation of the mean effect. An end-to-end experimentation on the effect of OOD will be provided in Case 2 of Section 5.1.2.

In Figure 1 we illustrate a small example where the underlying black-box function f has different behavior on the data generating distribution and away from it. As can be seen in Figure 1(a), we set the black-box function to be $f = x_1 x_2$ inside $|x_1 - x_2| < 0.5$ and to rapidly diverge outside of the region. The first feature follows a uniform distribution, i.e.

6. For example, the computation of that Jacobian can be done in a single command using Tensorflow `tf.GradientTape.jacobian(predictions, X)` and Pytorch `torch.autograd.functional.jacobian(f, X)`
7. Technically, having access to $\nabla_{\mathbf{x}} f$ is not a prerequisite, since the partial derivative $\frac{\partial f}{\partial x}$ can be approximated numerically, with finite differences. However, in this case, the computational advantages are canceled.

$x_1 \sim U(0, 10)$, and for the second feature $x_2 = x_1$. The local effect of x_1 is $f_1(\mathbf{x}) = x_2$. Spliting in K bins, the first bin is in $[0, \frac{10}{K})$, therefore, the ground truth bin effect is $\int_0^{10/K} \mathbb{E}_{x_2|z}\left[f_1(\mathbf{x})\right] \partial z = \frac{5}{K}$. In Figure 1(b), we observe that as the bin-length becomes bigger (smaller K), DALE approximates the effect perfectly, whereas, ALE fails due to OOD sampling. This happens because in the ALE approximation of Eq. (6), the bin limits z_{k-1}, z_k fall outside of the region $|x_1 - x_2| < 0.5$.

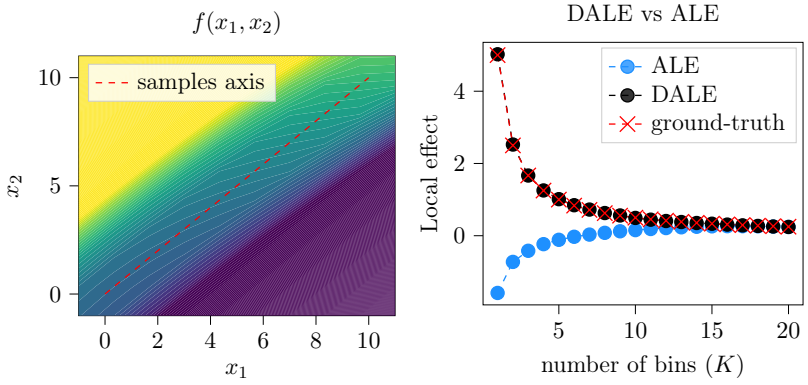

Figure 1: (Left) The black-box function f of Section 4.3. (Right) Estimation of the local effect of the first bin for DALE and ALE, for varying number of bins K.

4.4. Bias and variance

Let a finite dataset of samples $\mathcal{S}$, drawn independently and indentically distributed (i.i.d) from the data generating distribution of $\mathcal{X}$. DALE computes the accumulated local effect (Eq. (1)), using the approximation in (Eq. (7)). The expected value of the approximation across different datasets is

$$\mathbb{E}_{\mathcal{S}}[f_{\texttt{DALE}}(x)] = \Delta x \sum_{k=1}^{k_x} \mathbb{E}_{\mathcal{S}}[\frac{1}{|\mathcal{S}_k|} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k} f_s(\mathbf{x}^i)] \tag{11}$$

Notice also that for the values of x at the end of bin k_x, Eq. (1) can be rewritten as (after ommiting the constant c)

$$f_{\texttt{ALE}}(x) = \sum_{k=1}^{k_x} \int_{x_{k-1}}^{x_k} \mathbb{E}_{\mathcal{X}_c|\mathcal{X}_s=z}[f_s(\mathbf{x})] \partial z \tag{12}$$

where $x_0 = x_{s,min}$ and x_i, $i = 1, \ldots, k_x$ are the bin limits.
If we assume that each bin is sufficiently small such that $f_s(\mathbf{x})$ does not depend on x_s (i.e., $f(x)$ is linear wrt x_s) within the bin, then Eq. (12) becomes

$$f_{\texttt{ALE}}(x) = \sum_{k=1}^{k_x} \mathbb{E}_{\mathcal{X}_c|\mathcal{X}_s \in \mathcal{S}_k}[f_s(\mathbf{x})] \int_{x_{k-1}}^{x_k} \partial z = \Delta x \sum_{k=1}^{k_x} \mathbb{E}_{\mathcal{X} \in \mathcal{S}_k}[f_s(\mathbf{x})] \tag{13}$$

From Eqs. (11) and (13) we have

$$\mathbb{E}_{\mathcal{S}}[f_{\texttt{DALE}}] - f_{\texttt{ALE}}(x) = \Delta x \sum_{k=1}^{k_x} \mathbb{E}_{\mathcal{S}}[\frac{1}{|\mathcal{S}_k|} \sum_{k:\mathbf{x}^k \in \mathcal{S}_k} f_s(\mathbf{x}^k)] - \Delta x \sum_{k=1}^{k_x} \mathbb{E}_{\mathcal{X} \in \mathcal{S}_k}[f_s(\mathbf{x})] = \Delta x \sum_{k=1}^{k_x} (\mathbb{E}_{\mathcal{S}}[\hat{\mu}_k^s] - \mu_k^s) = 0 \quad (14)$$

since the expected value of the sample mean is an unbiased estimator of μ_k^s. As a result, under the condition of linearity wrt x_s within the bin, DALE is an unbiased estimator of the feature effect. If this assumption is violated (e.g., large bin size or highly nonlinear function), then this approach may introduce bias. The variance of the estimator is given[8] by $\text{Var}[\hat{\mu}_k^s] = \frac{(\sigma_k^s)^2}{|\mathcal{S}_k|}$, where $(\sigma_k^s)^2$ is the variance of f_s within the bin. Furthermore, since the samples $\mathbf{x}^i$ are independent, $\hat{\mu}_k^s$ for $k = 1, \ldots, k_x$ are also independent. The variance of the estimation can then be approximated as

$$\text{Var}[f_{\texttt{DALE}}(x)] = (\Delta x)^2 \sum_{k}^{k_x} \text{Var}[\hat{\mu}_k^s] = (\Delta x)^2 \sum_{k}^{k_x} \frac{(\sigma_k^s)^2}{|\mathcal{S}_k|} \approx (\Delta x)^2 \sum_{k}^{k_x} \frac{(\hat{\sigma}_k^s)^2}{|\mathcal{S}_k|} \quad (15)$$

where $(\hat{\sigma}_k^s)^2$ is the sample variance within bin k. Equation (15) allows the calculation of the standard error for the DALE approximation.

5. Experiments

This section presents the experimental evaluation of DALE using two synthetic and one real dataset. The experiments aim to compare DALE ($f_{\texttt{DALE}}$) with ALE approximation ($\hat{f}_{\texttt{ALE}}$) from the perspectives of both efficiency and accuracy.

The first synthetic example (Case 1) is designed to compare the approximations in terms of efficiency. For this reason, we generate synthetic datasets of varying dimensionality to illustrate the substantial computational benefits of DALE. The efficiency is evaluated by measuring the execution time.

The second synthetic example (Case 2) is designed to compare the approximations in terms of accuracy. More specifically, the example shows that DALE succeeds in estimating the correct effect when ALE approximation fails due to OOD sampling. Firstly, we compute the ground-truth ALE ($f_{\texttt{ALE}}$) directly from Eq. (1)[9] and, then, we evaluate the divergence of the two approximations from the ground-truth using the Mean Squared Error, i.e., $\text{MSE}_{<\texttt{approx}>} = \mathbb{E}[(f_{\texttt{ALE}} - f_{<\texttt{approx}>})^2]$, and its normalized version, i.e., $\text{NMSE}_{<\texttt{approx}>} = \frac{\mathbb{E}[(f_{\texttt{ALE}} - f_{<\texttt{approx}>})^2]}{\text{Var}[f_{\texttt{ALE}}]}$.

In the last part, we choose the Bike-Sharing dataset to compare DALE and ALE approximation. We choose this dataset for two reasons. Firstly, it is the dataset utilised in the original ALE paper, so it is a proper choice for unbiased comparisons. Secondly, it

8. We show that in the supporting material.
9. This is feasible in the synthetic example, since we generate artificial data points from a known generating distribution $p(\mathcal{X})$ and we define a mapping $f : \mathbf{x} \to y$ which has a known closed-form.

contains enough training points to approximate the feature effect accurately. Therefore, we want to check that $f_{\texttt{DALE}}$ and $\hat{f}_{\texttt{ALE}}$ provide similar effects using dense bins. We also experiment with how both methods behave when using fewer samples and larger bins. Code to reproduce the experiments is provided in the supporting material, and it will become publicly available upon acceptance of the paper.

5.1. Synthetic Datasets

5.1.1. Case 1

In this example, we evaluate the efficiency of the two approximations, $f_{\texttt{DALE}}$ and $\hat{f}_{\texttt{ALE}}$, through the execution times. We want to compare how both approximations perform in terms of the dimensionality of the problem (D), the dataset size (N) and the size of the model f.

In each experiment we generate a dataset X, by drawing $N \cdot D$ samples from a standard normal distribution. The black-box function f is a fully-connected neural network with L hidden layers of 1024 units each. All experiments are done using $K = 100$. We want to clarify that the value of K plays almost no role in the execution times. Since we are interested in solely comparing the methods efficiency, we do not provide further details for the modelling part.

In Figure 2, we directly compare $f_{\texttt{DALE}}$ and $\hat{f}_{\texttt{ALE}}$ in two different setups: in Figure 2(Left), we use a light setup of $N = 10^3$ examples and a model of $L = 2$ layers, whereas in Figure 2(Right), a heavier setup with $N = 10^5$ and $L = 6$. We observe that in both cases, DALE executes in near-constant time independently of D, while ALE scales linearly with wrt D, confirming our claims of Section 4.2. The difference in the execution time reaches significant levels for relatively small dimensionalities. In the heavy setup, ALE needs almost a minute for $D = 20$, three minutes for $D = 50$, and 15 minutes for $D = 100$. In all these cases, DALE executes in a few seconds.

Another critical remark is that DALE's execution time is almost identical to the computation of the Jacobian $\mathbf{J}$, which is benefited by automatic differentiation. Hence, we confirm that the overhead of performing steps 3-5 of Algorithm 1 is a small fraction of the total execution time. Another consequence of this remark is that we can test many different bin sizes with near-zero computational cost.

In Figure 3, we rigorously quantify to what extent the dataset size N and the model size L affect both methods. In Figures 3(a) and 3(b), we confirm that both N and L impact how ALE complexity increases wrt D. Therefore, for a big dataset and a heavy model f, ALE's execution time quickly reaches prohibitive levels. In contrast, in Figures 3(c) and Figures 3(d), DALE is negligibly affected by these parameters. In the figures, we restrict the experiment to cases up to 100-dimensional input for illustration purposes. The same trend continues for an arbitrary number of dimensions. DALE can scale efficiently to an arbitrary number of dimensions as long as we have enough resources to store the dataset, evaluate the prediction model f and apply the gradients $\nabla_{\mathbf{x}} f$.

5.1.2. Case 2

In this example, we evaluate the accuracy of the two approximations, $f_{\texttt{DALE}}$ and $\hat{f}_{\texttt{ALE}}$, in a synthetic dataset where the ground truth ALE is accesible. As discussed in Section 4.3, ALE approximation is vulnerable to OOD sampling when the bins are wide, or equivalently, the

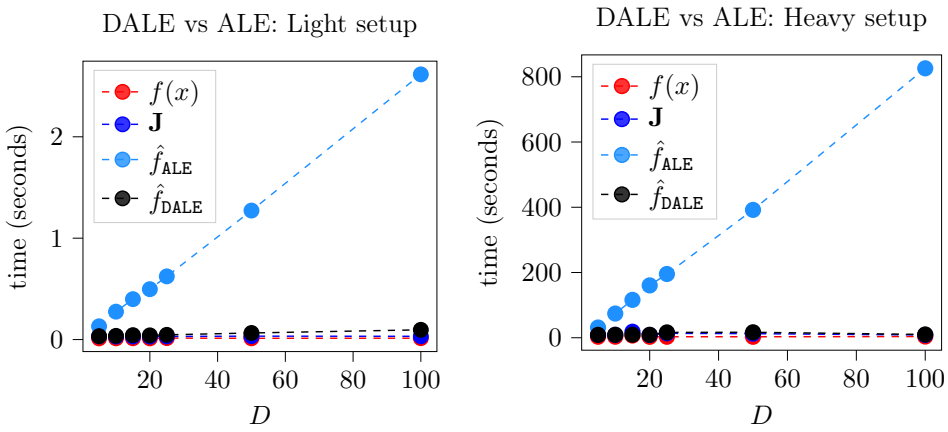

Figure 2: Comparison of the execution time of DALE and ALE in two setups: (Left) Light setup; $N = 10^3, L = 2$. (Right) Light setup; $N = 10^5, L = 6$

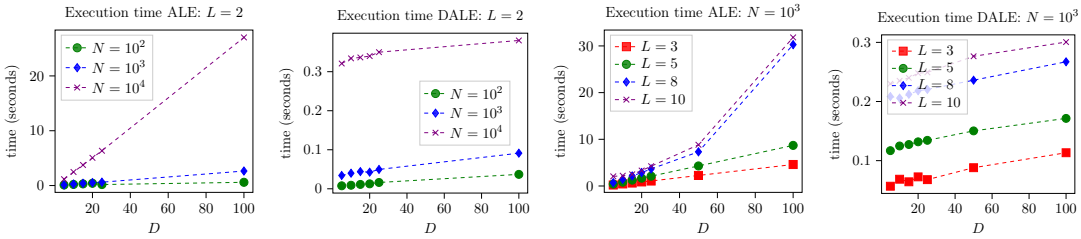

Figure 3: Measurements of the execution time wrt dimensionality D: (a) $\hat{f}_{\texttt{ALE}}$ for $L = 2$, and many dataset sizes N (b) $\hat{f}_{\texttt{ALE}}$ for $N = 10^3$, and many model sizes L (c) $f_{\texttt{DALE}}$ for $L = 2$, and many dataset sizes N (d) $f_{\texttt{DALE}}$ for $N = 10^3$, and many model sizes L

number of bins K is small. We want to compare how both approximations behave in a case where the local effect is noisy.

We design an experiment where we know the black-box function and the data generating distribution. The black-box function $f : \mathbb{R}^3 \to \mathbb{R}$ is split in three parts to amplify the effect of OOD sampling. It includes a mild term $f_0(x) = x_1x_2 + x_1x_3$ in the region $0 \leq |x_1 - x_2| < \tau$ and then a quadratic term $g(x) = \alpha((x_1 - x_2)^2 - \tau^2)$ is added(subtracted) over(under) the region, i.e.:

$$f(\mathbf{x}) = \begin{cases} f_0(x) & , 0 \leq |x_1 - x_2| < \tau \\ f_0(x) - g(x) & , \tau \leq |x_1 - x_2| \\ f_0(x) + g(x) & , \tau \leq -|x_1 - x_2| \end{cases} \tag{16}$$

The data points $X^i = (x_1^i, x_2^i, x_3^i)$ are generated as follows; x_1^i are clustered around the points $\{1.5, 3, 5, 7, 8.5\}$, $x_2^i \sim \mathcal{N}(\mu = x_1, \sigma_2 = 0.1)$ and $x_3^i \sim \mathcal{N}(\mu = 0, \sigma_3^2 = 10)$. In Figure 4(a), we illustrate $f(\mathbf{x})$ for $x_3 = 0$, as well as the generated data points.

Table 1: Evaluation of the NMSE between the approximations and the ground truth. Blue color indicates the values that are below 0.1.

Accuracy on the Synthetic Dataset (Case 2)

		Number of bins							
		1	2	3	4	5	10	20	40
NMSE	$f_{\texttt{DALE}}$	0.10	0.03	0.09	0.02	0.02	0.82	0.24	0.38
	$\hat{f}_{\texttt{ALE}}$	100.42	22.09	4.97	2.81	0.78	1.49	0.34	0.39

In this example, the local effect of x_1 is $\frac{\partial f}{\partial x_1} = x_2 + x_3$. Due to the noisy nature of x_3, both ALE and DALE need a large number of sample for robust estimation. Therefore, we need to lower the number of bins K. As will be shown below, both ALE and DALE fail to approximate the feature effect for high K. On the other hand, when using a lower K, ALE approximation fails due to OOD sampling, while DALE manages to accurately approximate the feature effect.

In Figure 4(b) and Figure 4(c), we observe the estimated effects for $K = 50$ and $K = 5$. In Figure 4(b), ($K = 50$) the approximations converge to the same estimated effect which is inaccurate due to many noisy artifacts. In Figure 4(c), ($K = 5$) we observe that for small K, DALE approximates the ground-truth effect well, whereas ALE fails due to OOD sampling. Table 1 provides the NMSE of both approximation for varying number of bins K. We observe that DALE consistently provides accurate estimations (NMSE ≤ 0.1) for all small K values.

The experiments helps us confirm that when K increases, both approximations are based on a limited number of samples, and are vulnerable to noise. When K decreases, DALE lowers the resolution but provides more robust estimations. In contrast, ALE is vulnerable to OOD sampling.

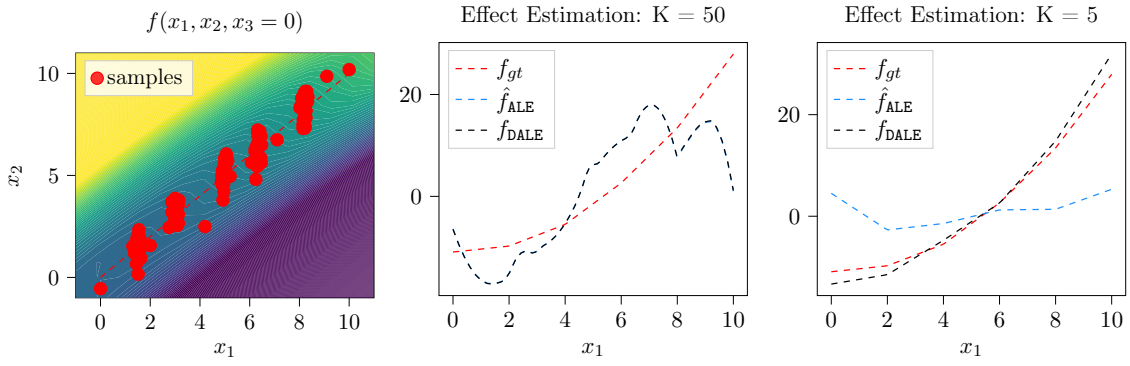

Figure 4: Example 2: Samples and (a) actual black box function and (b) black box function estimated via a neural network

Table 2: Measurements of the execution time in seconds, for DALE and ALE approximation on the Bike Sharing dataset.

Efficiency on Bike-Sharing Dataset

	Number of Features										
	1	2	3	4	5	6	7	8	9	10	11
$f_{\texttt{DALE}}$	1.17	**1.19**	**1.22**	**1.24**	**1.27**	**1.30**	**1.36**	**1.32**	**1.33**	**1.37**	**1.39**
$\hat{f}_{\texttt{ALE}}$	**0.85**	1.78	2.69	3.66	4.64	5.64	6.85	7.73	8.86	9.9	10.9

5.2. Real dataset

In this section, we test our approximation on the Bike-Sharing Dataset Fanaee-T and Gama (2013). [10] The Bike-Sharing Dataset is chosen as the main illustration example in the original ALE paper, therefore it was considered appropriate for comparisons. The dataset contains the bike rentals for almost every hour over the period 2011 and 2012. The dataset contains 14 features, which we denote as $X_{<\texttt{feature\_name}>}$. We select the 11 features that are relevant to the prediction task. Most of the features are measurements of the environmental conditions, e.g. $X_{\texttt{month}}$, $X_{\texttt{hour}}$, $X_{\texttt{temperature}}$, $X_{\texttt{humidity}}$, $X_{\texttt{windspeed}}$, while some others inform us about the day-type, e.g. whether we refer to a working-day $X_{\texttt{workingday}}$. The target value $Y_{\texttt{count}}$ is the bike rentals per hour, which has mean value $\mu_{\texttt{count}} = 189$ and standard deviation $\sigma_{\texttt{count}} = 181$. We train a deep fully-connected Neural Network with 6 hidden layers and 711681 parameters. We train the model for 20 epochs, using the Adam optimizer with learning rate 0.01. The model achieves a mean absolute error on the test of about 38 counts.

Efficiency For comparing the efficiency, we measure the execution time of DALE and ALE for a variable number of features. We present the results in Table 2. We confirm that DALE can compute the feature effect for all features in almost constant time wrt D. In contrast, ALE scales linearly wrt D which leads to an execution time of over 10 seconds.

Accuracy In the case of the Bike-Sharing, it is infeasible compare to compute the ground-truth ALE. We have lack of knowledge about the data-generating distribution and the dimensionality of the problem $D = 11$ is prohibitive for applying numerical integration on eq. (1). However, given the fact that the dataset has a large number of instances, DALE and ALE provide very similar approximations for large K, e.g. ($K = 200$), and we can treat these approximations as the ground-truth effect. In Figure 5, we illustrate the feature effect for three features.

For all feature features, except $X_{\texttt{hour}}$, lowering the number of bins K does not significantly impacts the approximation, since these features change slowly wrt the feature value. An exception is feature $X_{\texttt{hour}}$. In this case, the $f_{\texttt{DALE}}$ approximation remains accurate when lowering the number of bins K (Fig. 6(b)), while $\hat{f}_{\texttt{ALE}}$ deteriorates significantly (Fig. 6(c)). In Table 3 we evaluate the feature effect of $X_{\texttt{hour}}$ for different number of bins K. The ground-truth effect has been computed for $K = 200$. We observe that NMSE remains low

10. It is dataset drawn from the Capital Bikeshare system (Washington D.C., USA) over the period 2011-2012. The dataset can be found here

Table 3: Evaluation of DALE and ALE approximation when lowering the number of bins K. The ground-truth effect has been computed for $K = 200$.

Accuracy on Bike-Sharing Dataset - Feature $X_{\texttt{hour}}$

		Number of bins			
		100	50	25	15
NMSE	$f_{\texttt{DALE}}$	**0.007**	**0.01**	**0.03**	**0.09**
	$\hat{f}_{\texttt{ALE}}$	0.04	0.43	0.79	0.83

in DALE for all K, while for ALE it rapidly increases. This is due to OOD sampling that occurs when the bin size becomes large.

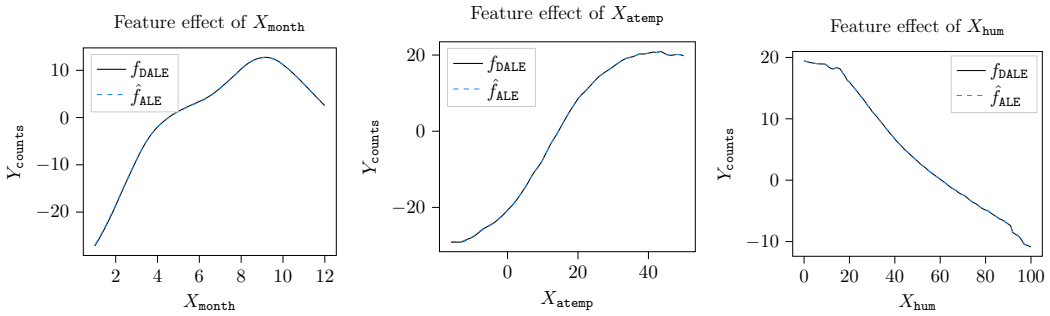

Figure 5: DALE and ALE feature effect plots with $K = 200$ for: $X_{\texttt{month}}$, $X_{\texttt{atemp}}$,$X_{\texttt{hum}}$.

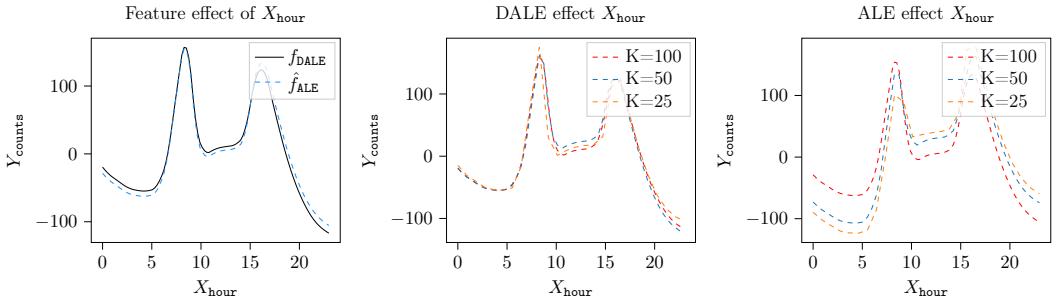

Figure 6: Feature effect plots on $X_{\texttt{hour}}$: (Left) DALE vs ALE for $K = 200$. (Center) DALE plots for $K = \{25, 50, 100\}$. (Right) ALE plots for $K = \{25, 50, 100\}$

6. Conclusion and Future Work

This paper introduced DALE, an efficient and robust-to-OOD approximation for ALE, the state-of-the-art method for feature effect analysis. First, we explored the advantages of ALE over the other two renowned feature effect methods, PDP and MPlots. However, ALE's

approximation scales poorly in big and high-dimensional datasets and suffers from OOD sampling in cases with limited samples. For addressing these deficiencies, we proposed DALE, a fast and on-distribution alternative. We presented the method and discussed the advantages over the typical ALE approximation. We proved that under some hypotheses, our proposal is an unbiased estimator of ALE and we presented a method for quantifying the uncertainty of the explanation, i.e. the standard error of the approximation. The experiments verify the aforementioned claims. DALE significantly improves the efficiency of ALE's approximation by orders of magnitude and secures that local effect estimations come from on-distribution samples. The latter leads to more accurate feature effect plots when the bins are wide and the black-box function changes away from the data generating distribution.

The computational efficiency of DALE delivers a substantial margin for future extensions. A significant advantage of our proposal is that effects are computed once on the training set points and can be reused in different-size bins. The decision for the bin density, i.e. the resolution of the plot, can be taken afterwards. Therefore, DALE permits creating feature effect plots at different resolutions with near-zero computational overhead, which can be embedded into a multiresolution feature effect plots framework.

In conclusion, we propose DALE as a fast and accurate alternative for estimating feature effect in cases where the ML model is differentiable and an auto-differentiable framework is accessible.

References

Amina Adadi and Mohammed Berrada. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, sep 2018. ISSN 21693536. doi: 10.1109/ACCESS.2018.2870052.

Daniel W. Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 82(4):1059–1086, 2020. ISSN 14679868. doi: 10.1111/rssb.12377.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020. ISSN 15662535. doi: 10.1016/j.inffus.2019.12.012.

Hadi Fanaee-T and Joao Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, pages 1–15, 2013. ISSN 2192-6352. doi: 10.1007/s13748-013-0040-3. URL [WebLink].

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20:1–81, 2019. ISSN 15337928. URL http://jmlr.org/papers/v20/18-760.html.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, oct 2001. ISSN 00905364. doi: 10.1214/aos/1013203451.

Jerome H. Friedman and Bogdan E. Popescu. Predictive learning via rule ensembles. *Annals of Applied Statistics*, 2(3):916–954, nov 2008. ISSN 19326157. doi: 10.1214/07-AOAS148. URL http://arxiv.org/abs/0811.1679http://dx.doi.org/10.1214/07-AOAS148.

Karthik S. Gurumoorthy, Amit Dhurandhar, Guillermo Cecchi, and Charu Aggarwal. Efficient data representation by selecting prototypes with importance weights. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2019-November:260–269, jul 2019. ISSN 15504786. doi: 10.1109/ICDM.2019.00036. URL https://arxiv.org/abs/1707.01212v4.

Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. Metrics for Explainable AI: Challenges and Prospects. 2018. ISSN 2331-8422. URL http://arxiv.org/abs/1812.04608.

G. Hooker, L. Mentch, and S. Zhou. Unrestricted permutation forces extrapolation: variable importance requires at least one more model, or there is no free variable importance. *Statistics and Computing*, 31(6), 2021. doi: 10.1007/s11222-021-10057-z. cited By 0.

Been Kim, Rajiv Khanna, and Oluwasanmi Koyejo. Examples are not enough, learn to criticize! Criticism for interpretability. In *Advances in Neural Information Processing Systems*, volume 29, pages 2288–2296, 2016.

Scott M. Lundberg and Su In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 2017-December:4766–4775, 2017. ISSN 10495258. URL https://github.com/slundberg/shap.

Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges. *Communications in Computer and Information Science*, 1323:417–431, 2020a. ISSN 18650937. doi: 10.1007/978-3-030-65965-3_28.

Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Quantifying model complexity via functional decomposition for better post-hoc interpretability. In *Communications in Computer and Information Science*, volume 1167 CCIS, pages 193–204, 2020b. ISBN 9783030438227. doi: 10.1007/978-3-030-43823-4_17.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. *NAACL-HLT 2016 - 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, pages 97–101, feb 2016. doi: 10.18653/v1/n16-3020. URL https://arxiv.org/abs/1602.04938v3.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-Precision Model-Agnostic Explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32 (1), apr 2018. ISSN 2374-3468. URL https://ojs.aaai.org/index.php/AAAI/article/view/11491.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. *SSRN Electronic Journal*, nov 2017. ISSN 1556-5068. doi: 10.2139/ssrn.3063289. URL https://arxiv.org/abs/1711.00399v3.