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

Vasilis Gkolemis gkolemis@hua.gr, vgkolemis@athenarc.gr
Harokopio University of Athens, IMIS ATHENA Research Center

Theodore Dalamagas dalamag@athenarc.gr
IMIS ATHENA Research Center

Christos Diou cdiou@hua.gr
Harokopio University of Athens

Editors: Emtiyaz Khan and Mehmet Gönen

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, *ALE's approximation*, i.e. the method for estimating ALE from the limited samples of the training set, faces two weaknesses. First, it does not scale well in cases where the input has high dimensionality, and, second, it is vulnerable to out-of-distribution (OOD) sampling when the training set is relatively small. In this paper, we propose a novel ALE approximation, called Differential Accumulated Local Effects (DALE), which can be used in cases where the ML model is differentiable and an auto-differentiable framework is accessible. Our proposal has significant computational advantages, making feature effect estimation applicable to high-dimensional Machine Learning scenarios with near-zero computational overhead. Furthermore, DALE does not create artificial points for calculating the feature effect, 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 standard error of the explanation. Experiments using both synthetic and real datasets demonstrate the value of the proposed approach.

Keywords: Feature Effect; Explainable AI; Interpretability; Global Methods; Neural Networks

1. Introduction

Machine Learning (ML) models have been adopted to 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) Accumulated 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 Accumulated 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 show with synthetic and real datasets that DALE: (a) scales better than ALE, (b) provides a better approximation than ALE, especially in cases of wide bins. Code for reproducing all experiments is provided at https://github.com/givasile/DALE.

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 (Arrieta et al., 2020), (Adadi and Berrada, 2018) classifying the different approaches and detecting future challenges on the field (Molnar et al., 2020). There are several criteria 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

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

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 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 behavior. Global surrogates approximate the black-box model with a simple one, for example, decision trees (Nanfack et al., 2021). Prototypes (Gurumoorthy et al., 2019) search for data points that are representative for each class and criticisms (Kim et al., 2016) for ambiguous data points that representative of the boarder between classes. 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 outcome 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., 2019), 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 use uppercase and calligraphic font $\mathcal{X}$ for random variables (rv), plain lowercase x for scalar variables and bold $\mathbf{x}$ for vectors. Often, we partition the input vector $\mathbf{x} \in \mathbb{R}^D$ to the feature of interest $x_s \in \mathbb{R}$ and the rest of the features $\mathbf{x}_c \in \mathbb{R}^{D-1}$, and for convenience we notate it $\mathbf{x} = (x_s, \mathbf{x}_c)$. We clarify that $(x_s, \mathbf{x}_c)$ corresponds to the vector $(x_1, \cdots, x_s, \cdots x_D)$. Equivalently, we notate the corresponding rv to $\mathcal{X} = (\mathcal{X}_s, \mathcal{X}_c)$. The black-box function is $f : \mathbb{R}^D \to \mathbb{R}$ and the feature effect of the s-th feature is $f_{<\texttt{method}>}(x_s)$, where $< \texttt{method} >$ is the name of the feature effect method.

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 accumulation of the local effects. The local effect at point z is the expected change on the output over the conditional distribution $\mathcal{X}_c|\mathcal{X}_s = z$, i.e. $\mathbb{E}_{\mathcal{X}_c|\mathcal{X}_s=z}\left[\frac{\partial f(x_s, \mathcal{X}_c)}{\partial x_s}\right]$. 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 $x_1 \sim \mathcal{U}(0,1)$ 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: [2] [3]

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

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

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

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.

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.

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

4.1. Definition of DALE

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

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

We denote as $\mathbf{x}^i$ the i-th example of the training set and as x_s^i 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. $\mathcal{S}_k = \{\mathbf{x}^i : z_{k-1} \leq x_s^i < z_k\}$. For understanding Eq. (6) better, we split it in three levels: Instance effect is the effect computed on the i-th example, i.e., $\Delta f_i = f(z_k, \mathbf{x}^i_{\mathbf{c}}) - f(z_{k-1}, \mathbf{x}^i_{\mathbf{c}})$, bin effect is the effect computed by the samples that are in 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 $\hat{f}_{\mathtt{ALE}}(x_s)$. The approximation splits the axis into K equally-sized bins and computes each bin effect by averaging the instance effects of the samples that lie in each bin. The global effect is the accumulation of the bin effects. To make a connection with ALE definition (Eq. 1), the bin effect is an estimation of the accumulated local effects of the interval covered by the bin, i.e. $\int_{z_{k-1}}^{z_k} \mathbb{E}_{\mathcal{X}_c|\mathcal{X}_s = z} \left[\frac{\partial f(z, \mathcal{X}_c)}{\partial x_s} \right] \partial z$. The approach of Eq. (6) 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 bins length becomes large. This happens because the instance effects are estimated by generating artificial samples at the bin limits. Finally, 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:

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

where Δx is the bin length and f_s the partial derivative wrt x_s, i.e. $f_s = \frac{\partial f}{\partial x_s}$. We use $\hat{\mu}_k^s = \frac{1}{|\mathcal{S}_k|} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k} [f_s(\mathbf{x}^i)]$ to indicate the estimated k-th bin effect. DALE uses only the dataset samples and doesn't perturb any feature, securing that we estimate the bin effect from on-distribution (observed) data points. In Eq. (7), the estimation of the instance effect at each training sample is independent from the bin size. Unlike 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 $s = \{1, \dots, D\}$, $i = \{1, \dots, N\}$ once, and reusing them to create 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.

Apley and Zhu (2020) also provide a formula for approximating the combined effect of a pair of attributes x_l, x_m:[4]

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

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:

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

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, protects from OOD sampling and permits multi-resolution plots, with near-zero additional cost.

4.2. Computational Benefit

DALE approximation has significant computational advantages. For estimating the feature effect of all features, our approach processes the N data points of the training set. In contrast, ALE approximation generates and processes $2 \cdot N \cdot D$, weighting by a factor of D the computational complexity and the memory requirements. Therefore, DALE 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 wrt all features in a single pass. Computing the gradient vector for a training example $\mathbf{x}^i$ wrt all features $\nabla_{\mathbf{x}} f(\mathbf{x}^i) = [f_1(\mathbf{x}), \cdots, f_D(\mathbf{x})]$ is computationally equivalent to evaluating $f(\mathbf{x}^i)$. Based on this observation, computing the whole Jacobian matrix costs $\mathcal{O}(N)$. In contrast, in ALE, the evaluation of f for $2 \cdot N \cdot D$ times costs $\mathcal{O}(N \cdot D)$. Our method, also, takes advantage of all existing automatic differentiation frameworks which are optimized for

4. For completeness, we provide the second-order ALE definition in the helping material.

computing the gradients efficiently.[5] In Algorithm 1, we present DALE in an algorithmic form. The algorithm needs as input: (a) the black-box function f, (b) the derivative of $\nabla_{\mathbf{x}} f$ and (c) the dataset $\mathbf{X}$.[6] 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., $\hat{f}^s_{\texttt{DALE}}(x) = \mathbf{A}_{s,k_x}$. 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 $\hat{f}_{\texttt{DALE}}(x)$ requires only locating the bin k_x that x belongs to. The same computational advantage also hold for the second-order DALE. In the second-order we need to compute the Hessian Matrix instead of the Jacobian (Step 1) and to allocate the points in a 2D grid instead of the sequence of intervals (Step 3).

Algorithm 1 DALE approximation

```
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 $\hat{f}_{\texttt{DALE}}(x) = \mathbf{A}_{s,k_x}$
```

4.3. Robustness to out-of-distribution sampling

OOD sampling is the source of failure in many explainability methods that perturb features(Baniecki et al. (2021), 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 $\mathbf{x} \rightarrow y$ 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. 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

5. 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)`
6. 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.

this region. The first feature follows a uniform distribution, i.e., $x_1 \sim U(0, 10)$, and for the second feature $x_2 = x_1$. The local effect of x_1 is $\mathbb{E}_{x_2|x_1}[f_1(\mathbf{x})] = x_1$. Splitting in K bins, the first bin covers the region $[0, \frac{10}{K})$, therefore, as discussed in 4.1, the ground truth bin effect is $\int_0^{10/K} \mathbb{E}_{x_2|z}[f_1(\mathbf{x})]\,\partial z = \frac{5}{K}$. 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 $|x_1 - x_2| < 0.5$.

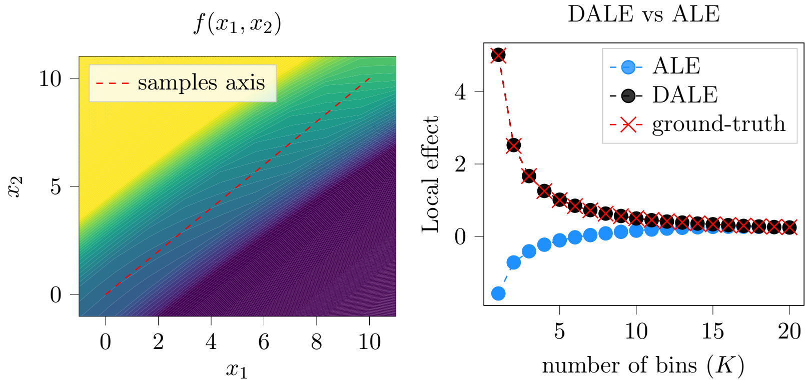

Figure 1: (Left) The black-box function f of Section 4.3. (Right) Estimation of the bin effect of the first bin for DALE and ALE, for varying number of bins K.

4.4. Bias and variance

Let a finite dataset of samples $\mathcal{S}$, drawn independently and identically 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

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

Notice also that for the values of x at the end of bin k_x, Eq. (1) can be rewritten as (after omitting 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, $i = 1, \dots, k_x$ 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

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

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[7] 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 $k = 1, \dots, k_x$ are also independent. The variance of the estimation can then be approximated as

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

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 ($\hat{f}_{\texttt{DALE}}$) with ALE approximation ($\hat{f}_{\texttt{ALE}}$) from the perspectives of both efficiency and accuracy.

Metrics. For evaluating the efficiency of the approximations we measure the execution times (in seconds). For evaluating the accuracy we use: (a) qualitative comparison of the feature effect plots and (b) the Normalized Mean Squared Error which is defined as $\text{NMSE}_{\texttt{<approx>}} = \frac{\mathbb{E}[(f_{\texttt{ALE}} - f_{\texttt{<approx>}})^2]}{\text{Var}[f_{\texttt{ALE}}]}$.

Synthetic Datasets. The first synthetic dataset (Case 1) is designed to compare the approximations in terms of efficiency. For this reason, we generate design matrices $\mathbf{X}$ of varying dimensionality D and number of instances N. The second synthetic dataset (Case 2) is designed to compare the approximations in terms of accuracy. We define a data generating distribution $\mathcal{X}$ and a black box function f with known forms for being able to directly compute the ground-truth ALE from Eq. (1). Both $\mathcal{X}$ and f are designed to amplify the effect of OOD sampling.

7. We show that in the supporting material.

Real Dataset We choose the Bike-Sharing dataset for two reasons. Firstly, it is the dataset utilized in the original ALE paper, so it is a proper choice for unbiased comparisons. Secondly, we wanted a dataset with enough training points to approximate the feature effect accurately, since the ground-truth is not available. Therefore, we want to check that $\hat{f}_{\texttt{DALE}}$ and $\hat{f}_{\texttt{ALE}}$ provide similar effects using dense bins. We also evaluate the accuracy of both methods behave when using larger bins.

5.1. Synthetic Datasets

5.1.1. Case 1 - Efficiency comparison

In this example, we evaluate the efficiency of the two approximations, $\hat{f}_{\texttt{DALE}}$ 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 L. In each experiment we generate a design-matrix $\mathbf{X}$, 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.

In Figure 2, we directly compare $\hat{f}_{\texttt{DALE}}$ 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 from a relatively small dimensionality. 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(c), we confirm that both N and L have crucial impact in ALE's execution times. Therefore, for a big dataset and a heavy model f, ALE's execution time quickly reaches prohibitive levels. In contrast, in Figures 3(b) 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 any dimensionality 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 - Accuracy Comparison

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

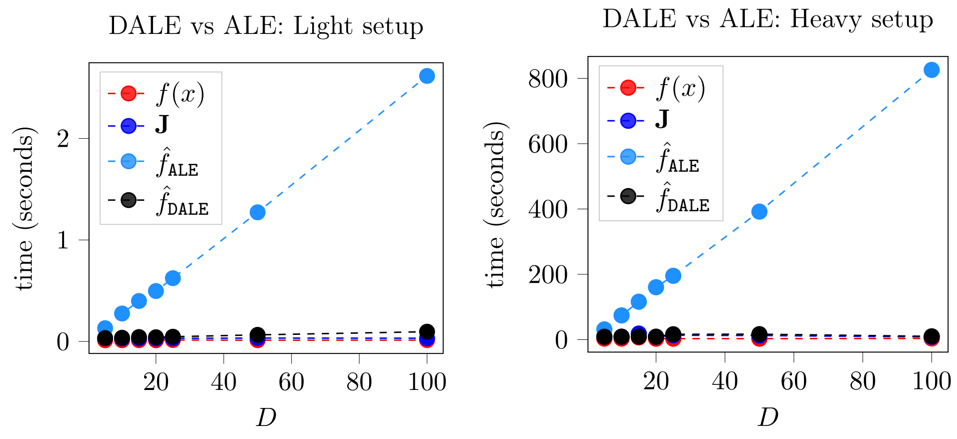

Figure 2: Case 1. Comparison of the execution time of DALE and ALE wrt dimensionality in two setups: (Left) Light setup; $N = 10^3, L = 2$. (Right) Heavy setup; $N = 10^5, L = 6$

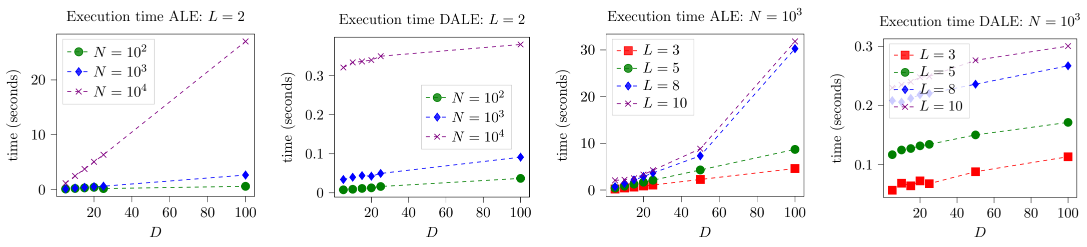

Figure 3: Case 1. Measurements of the execution time wrt dimensionality D. From left to right: (a) $\hat{f}_{\texttt{ALE}}$ for $L = 2$ and many dataset sizes N (b) $\hat{f}_{\texttt{DALE}}$ for $L = 2$ and many dataset sizes N (c) $\hat{f}_{\texttt{ALE}}$ for $N = 10^3$ and many model sizes L (d) $\hat{f}_{\texttt{DALE}}$ for $N = 10^3$ and many model sizes L

and the data generating distribution. The black-box function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ is split in three parts to amplify the effect of OOD sampling. It includes a mild term $f_0(x) = x_1 x_2 + x_1 x_3$ 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. 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

Table 1: Case 2. 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	$\hat{f}_{\texttt{ALE}}$	100.42	22.09	4.97	2.81	0.78	1.49	0.34	0.39
	$\hat{f}_{\texttt{DALE}}$	0.10	0.03	0.09	0.02	0.02	0.82	0.24	0.38

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.

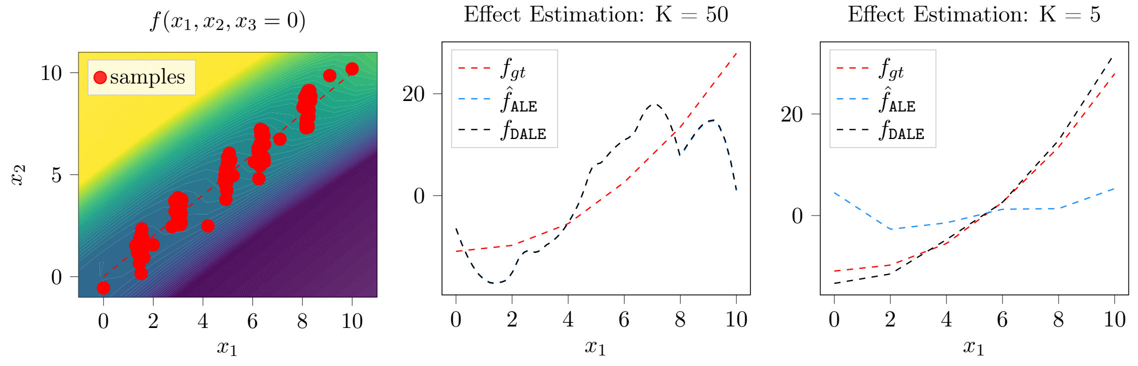

Figure 4: Case 2 experiment. (a) The black-box function f of Section 4.3. (b) Estimation of the feature effect for a large number of bins $K = 50$. (c) Estimation of the feature effect for a small number of bins $K = 5$.

Table 2: Bike-Sharing Dataset. Measurements of the execution time.
Efficiency on Bike-Sharing Dataset (Execution Times in seconds)

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

5.2. Real dataset

In this section, we test our approximation on the Bike-Sharing Dataset Fanaee-T and Gama (2013). [8] 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 almost identical approximations for all features for large K as we confirm in Figure 5. We also notice the 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 $\hat{f}_{\texttt{DALE}}$ 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 both approximations on $X_{\texttt{hour}}$, for different number of bins K. We set the ground-truth effect to be the approximation for $K = 200$. We observe that NMSE remains low 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.

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

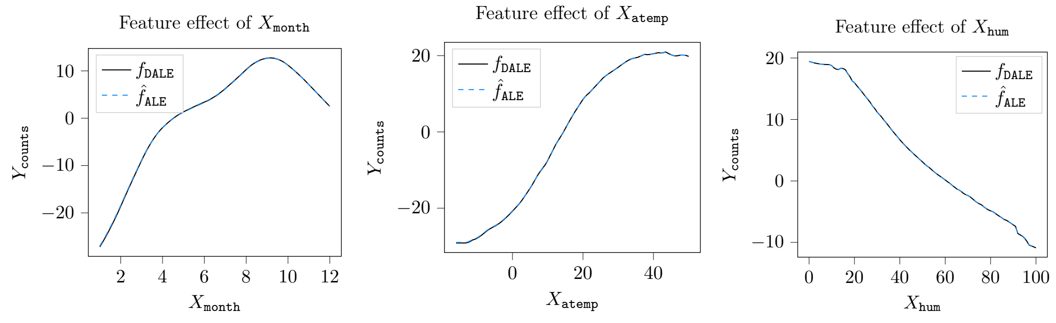

Figure 5: Bike-Sharing Dataset. DALE and ALE feature effect plots with $K = 200$ for: $X_{\texttt{month}}$, $X_{\texttt{atemp}}$,$X_{\texttt{hum}}$.

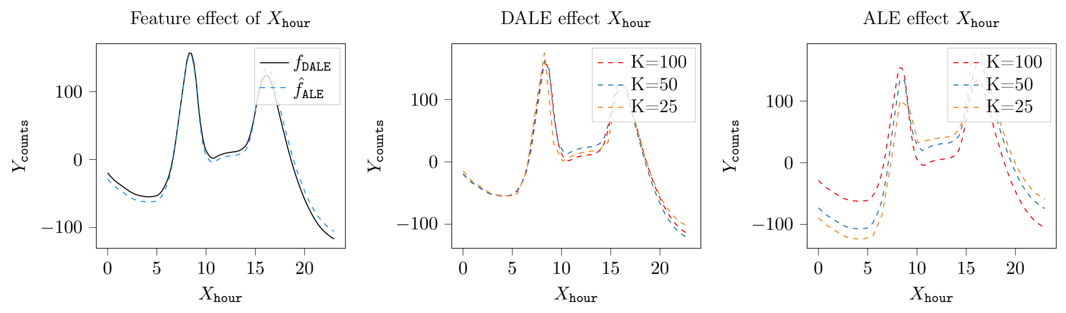

Figure 6: Bike-Sharing Dataset. Feature effect plots on $X_{\texttt{hour}}$: (Left) DALE vs ALE for $K = 200$. (Center) DALE plots for $K = \{25, 50, 100\}$. (Right) ALE plots for $K = \{25, 50, 100\}$

6. Conclusion and Future Work

This paper introduced DALE, an efficient and robust to OOD approximation for ALE. Although ALE models the feature effect correctly in cases of correlated features, ALE's approximation scales poorly in high-dimensional datasets and suffers from OOD sampling. As discussed in the paper, DALE addresses these limitations providing a fast and on-distribution

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	$\hat{f}_{\texttt{ALE}}$	0.04	0.43	0.79	0.83
	$\hat{f}_{\texttt{DALE}}$	**0.007**	**0.01**	**0.03**	**0.09**

alternative. We proved that under some hypotheses, our proposal is an unbiased estimator of ALE and we presented a method for quantifying the standard error of the approximation. The experiments verified our 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 multi-resolution feature effect plots framework.

Acknowledgments

We thank Eirini Ntoutsi for her insightful comments in the draft versions of the paper and Giorgos Giannopoulos for the valuable discussions while forming the initial idea.
This work was partially supported by projects i4metal (grant T2EDK- 03010) and XMANAI (grant agreement No 957362), which have received funding by the European Regional Development Fund of the EU and Greek national funds (through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call Research-Create-Innovate) and the EU 2020 Programme, ICT-38-2020 - Artificial intelligence for manufacturing, respectively.

References

Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

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.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

Hubert Baniecki, Wojciech Kretowicz, and Przemyslaw Biecek. Fooling partial dependence via data poisoning. *arXiv preprint arXiv:2105.12837*, 2021.

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.

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. *J. Mach. Learn. Res.*, 20(177):1–81, 2019.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The annals of applied statistics*, pages 916–954, 2008.

Karthik S Gurumoorthy, Amit Dhurandhar, Guillermo Cecchi, and Charu Aggarwal. Efficient data representation by selecting prototypes with importance weights. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 260–269. IEEE, 2019.

Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*, 2018.

Giles Hooker, Lucas Mentch, and Siyu 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):1–16, 2021.

Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29, 2016.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Quantifying model complexity via functional decomposition for better post-hoc interpretability. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 193–204. Springer, 2019.

Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable machine learning–a brief history, state-of-the-art and challenges. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 417–431. Springer, 2020.

Géraldin Nanfack, Paul Temple, and Benoît Frénay. Global explanations with decision rules: a co-learning approach. In *Uncertainty in Artificial Intelligence*, pages 589–599. PMLR, 2021.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.