AIDE: Antithetical, Intent-based, and Diverse Example-Based Explanations

Abstract

For many use-cases, it is often important to explain the prediction of a black-box model by identifying the most influential training data samples. Existing approaches lack customization for user intent and often provide a homogeneous set of explanation samples, failing to reveal the model's reasoning from different angles.

In this paper, we propose AIDE, an approach for providing antithetical (i.e., contrastive), intent-based, diverse explanations for opaque and complex models. AIDE distinguishes three types of explainability intents: interpreting a correct, investigating a wrong, and clarifying an ambiguous prediction. For each intent, AIDE selects an appropriate set of influential training samples that support or oppose the prediction either directly or by contrast. To provide a succinct summary, AIDE uses diversity-aware sampling to avoid redundancy and increase coverage of the training data.

We demonstrate the effectiveness of AIDE on image and text classification tasks, in three ways: qualitatively, comparing anecdotal evidence from AIDE and other example-based approaches; quantitatively, assessing correctness and faithfulness; and via a user study, evaluating multiple aspects of AIDE. The results show that AIDE addresses the limitations of existing methods and exhibits several desirable traits for an explainability method.

1 INTRODUCTION

Failure of ML-based systems in numerous cases, e.g., due to data errors, biases, misalignment [24, 32], has prompted researchers to work on explainability techniques. Different taxonomies for such methods exist, e.g., [12], but one common classification is on the type of explanation generated [22]. *Model-based* methods involve creating interpretable surrogate models, such as decision trees or linear models, which approximate the complex black box ML model [29, 31]. *Feature-based* methods focus on pinpointing important features of the input, such as words in text or parts in an image, which contribute the most to the prediction [28, 8, 4]. *Example-based* methods provide explanations for a specific target outcome by deriving the importance of training samples [15, 9, 18, 19, 10, 25], or provide a global overview of the model identifying representative examples [34, 27].

Example-based explainability offers several advantages. They are typically model-agnostic, and offer easy to understand explanations. More importantly, as they seek to discover a causal relationship between training examples and model behavior, they can assist in model debugging and data cleansing [14]. However, they have two key limitations.

First, they don't offer *contrastivity* [23], which is key aspect in how humans understand decisions [20]. While most methods can distinguish between *supporters* (aka proponents, helpful or excitatory examples), and *opposers* (aka opponents, harmful or inhibitory examples), they do not relate this information to ground truth labels (examples of class same as or different than predicted) or to the explanation intent (is the prediction correct/wrong, hard to tell).

More importantly, existing example-based methods are highly susceptible to *class outliers*. An outlier is a training instance that is mislabeled, or an instance (training or target) that is ambiguous and does not clearly belong to a class. Mislabeled or ambiguous training instances tend to be explanations for any target instance, as they play a significant role in forming the decision boundary. Ambiguous target instances confuse the classifier (low confidence) and make it hard to pick good explanations.

In this paper, we propose a novel *Antithetical, Intent-based, and Diverse Example-based explainability* (AIDE), that offers contrastivity and is robust to outliers. At its core, AIDE is based on the concept of *influence functions* [13, 18]. For a fixed target instance, the *influence* of a training sample

Submitted to the 40th Conference on Uncertainty in Artificial Intelligence (UAI 2024). **To be used for reviewing only**.

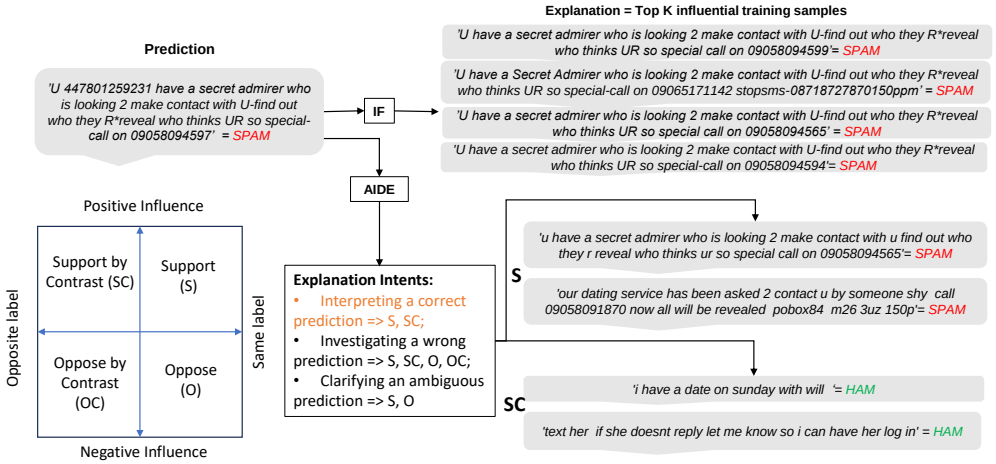

Figure 1: Explanations for a spam classification task, depicting a correctly classified spam message and its influence-based explanations generated by IF and AIDE.

is a score conveying its impact on the classifier's outcome. Ideally, the influence is the change observed in the loss value for the target if the training sample was excluded from the training data. While influence scores can be *estimated* by methods, such as TraceIn [9] and Datamodels [15], we use the framework of the influence function approach [18], termed IF, to efficiently compute influence scores.

To better understand AIDE's contribution, we first showcase the issues that plague example-based explainability methods, taking IF as the representative—extensive qualitative and quantitative comparison with other methods is presented in Section 4. Consider a classifier that predicts whether short text messages are spam. Figure 1 shows that for the depicted target message, the prediction is spam. This is a correct prediction, and IF identifies the four most influential training samples at the top of Figure 1. We observe that explanations lack *diversity*, as they are highly similar to each other. More importantly however, they lack *contrastivity*, as the user does not gain any insight about how the model decides what is spam and what not; all the user learns is that similar texts were labelled spam. The issue of susceptibility to outliers does not manifest in this example, mainly because the prediction is clearly correct. However, it manifests when for example what the correct prediction should be is not clear, as in in Figure 2.

Contribution. *AIDE features contrastivity*. Given a target instance to be explained, AIDE computes the influence of each training sample. But to present an explainability summary, AIDE distinguishes samples along two key explainability dimensions. The first is the *influence polarity*: a sample with positive influence *supports* the prediction, while one with negative influence *opposes* the decision. The second dimension is the label of the training sample, which is either the same or opposite as the target instance. These two dimensions define the four AIDE quadrants, denoted as support (S), support by contrast (SC), oppose (O), and oppose by contrast (OC). Assuming a binary classifier and that the prediction is $y \in \{-1, 1\}$, intuitively, S explains "*why it's* y", SC explains "*why it's not* $-y$", O explains "*why it might be* $-y$", and OC explains "*why it might not be* y". These quadrants offer contrastivity, providing to the user answers to distinct counterfactual questions. Figure 1 depicts the quadrants at the bottom left.

AIDE is intent-aware. A user faced with a correct prediction, would more likely need additional evidence that the model has learned the correct patterns. A user recognizing a wrong prediction would want to narrow down the sources of the problem. A user looking at an ambiguous prediction, would want to learn more about how the model handles such cases. AIDE customizes its explanations by distinguishing three types of intents a user might have: interpreting a correct, investigating a wrong, or clarifying an ambiguous prediction. For a seemingly correct prediction, AIDE presents the most influential but diverse samples from the support and support by contrast (S and SC) quadrants. The intuition is that the user needs to better understand where the decision boundary lies. For an ambiguous prediction, AIDE presents samples from the support and oppose (S and O) quadrants. The intuition here is to contrast between two possible predictions and let the user decide whether one is better than the other. For a wrong prediction, AIDE presents samples from all quadrants to allow the user to investigate evidence for all alternatives. An example for interpreting a correct prediction is depicted at the bottom right of Figure 1, where the presented examples help the user increase their confidence that the model's prediction is correct.

AIDE outperforms state-of-the-art example-based methods. We perform an extensive qualitative and quantitative comparison against state-of-the-art methods for example-based explainability. Shapley-based approaches [10, 19] were excluded as (a) they primarily aim to capture the overall contribution of training samples to the trained model (data valua-

tion), and not geared for local explanations, and (b) they can be impractical for local explainability due to their high computational cost. The main conclusions drawn are as follows. Datamodels [15] approximate well the influence scores, but perform poorly for outlier targets. The principal reason is that Datamodels explain a class of models, and not a particular model. They thus fail to identify nuances picked up by a single model; while an unambiguous target will receive similar predictions by all models in the class, models will greatly differ in their predictions for an ambiguous target. TraceIn [9] is highly susceptible to outliers in the training data and performs poorly in tests of correctness and truthfulness. The reason lies in the way TraceIn estimates influence: it considers the difference in the total (training) loss when a training sample is included or not during checkpoints; outliers have high individual loss, contributing significantly to the total loss, and are thus awarded high importance. Regular Influence functions (IF) are similarly affected by outliers in the training data. RelatIF [5] seeks to address this problem, by penalizing samples that have high loss. However, these high-loss samples can at times be highly informative. For example to explain a target instance that is ambiguous, it is often insightful to present those outlier training examples that are similar, so as to potentially uncover interesting labelling rules or protocols. In contrast, AIDE considers outliers as long as they are relevant to the target instance.

2 RELATED WORK

Example-based Explanations. Beyond influence functions, Data Shapley [10] is one of the prominent methods in this line, which just like its feature-based version [21] uses the game theory and revises the contribution of a point in all possible subsets to uncover its marginal effect for the models' performance. Due to the computational exhaustiveness of possible sets, even the approximation based on sampling methods e.g. Monte Carlo (MC) or Truncated MC, is still computationally expensive. A more robust version of datashap, betashap proposed by [19] reduces noise in importance scores, however, still inherits the high cost of computation. Both datashap and betashap compute the contribution of a single point for the models predictive performance overall, and using them for providing local explanations per sample would make it completely impractical in terms of cost and thus are not chosen as baselines. Another very similar to the IF method and an obvious baseline, TraceIn [9] measures the influence of a training sample X on a specific test sample X_0 as the cumulative loss change on X_0 due to updates from mini-batches containing X. They practically approximate this with TraceInCP, which considers checkpoints during training and sums the dot product of gradients at X and X_0 at each checkpoint. Another interesting and unconventional work [15] fixes a test point to explain and samples a large number of subsets from the training set and trains models with each of these subsets. It then trains a linear model where the input will be $1_S i$ encoding of a subset and the output is the performance of the model trained on this subset for the test sample of interest. The weights of the linear model will represent the importance score of a training sample in the same position. To obtain a good result a huge number of intermediate models has to be trained on subsets, which is exhaustive, and thus a faster version of datamodels was proposed by [25] and claimed to preserve almost the same accuracy. However, since our focus is on the effectiveness of explanation we still use the original datamodels as a baseline.

Evaluating Explanations. A profound study of functionality-grounded strategies by [23], advocates twelve quantifiable properties that can be evaluated to assess the quality of explanations. They categorize the state-of-the-art metrics into twelve classes depending on which property the metric focuses on and what type of explanation is provided. The following properties are most relevant for local, example-based explanations: (1) Consistency and continuity, both describe how deterministic the explanation is concerning identical and similar samples, assuming that these samples should have identical and similar explanations, in many works they are also referred as the *faithfulness* [16, 1] of explanation and has gained popularity in explainability domain; (2) Contrastivity is the ability of an explanation to interpret classes different than the prediction class; (3) Compactness is encoded in the size of an explanation as well as calculating a redundancy in the explanation; (4) Context describes how relevant the explanation is to the user needs; (5) Controlled synthetic Data check - Controlled Experiment: a synthetic dataset is developed with a predetermined reasoning, ensuring that the predictive model aligns with this reasoning, as verified through metrics like accuracy. An assessment is done to check whether the explanation provided by the model corresponds to the same reasoning embedded in the data generation process, [2, 6].

3 THE AIDE FRAMEWORK

3.1 PRELIMINARIES

In what follows, we assume a classification task where a *model* $f_{\boldsymbol{\theta}}$, described by *parameters* $\boldsymbol{\theta}$, maps an input $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$ to a predicted class $f_\theta(\boldsymbol{x}) \in \mathcal{Y}$. We use the notation $\boldsymbol{z} = (\boldsymbol{x}, y)$ to refer to a pair of input and its actual class. Let $\mathcal{S} \subseteq \boldsymbol{\mathcal{X}} \times \mathcal{Y}$ denote a *training set* of size $n = |\mathcal{S}|$. Let $\ell(\boldsymbol{z}, \boldsymbol{\theta})$ be the *loss function* of the model for $\boldsymbol{z}$, and let $L(\mathcal{S}, \boldsymbol{\theta}) = \frac{1}{n} \sum_{\boldsymbol{z} \in \mathcal{S}} \ell(\boldsymbol{z}, \boldsymbol{\theta})$ denote the training *objective*, i.e., the mean loss for set $\mathcal{S}$.[1] We denote as $\boldsymbol{\theta}_0^*$ the parameters that minimize the objective: $\boldsymbol{\theta}_0^* = \arg\min_{\boldsymbol{\theta}} L(\mathcal{S}, \boldsymbol{\theta})$.

The goal is to explain the model's prediction for a spe-

[1] We assume regularization terms are folded in L.

cific *test instance* $\boldsymbol{z}_t = (\boldsymbol{x}_t, y_t)$, in terms of the influence each training example $\boldsymbol{z} \in \mathcal{S}$ makes on the model's prediction $f_{\boldsymbol{\theta}}(\boldsymbol{x}_t)$, and specifically on its prediction loss $\ell(\boldsymbol{z}_t, \boldsymbol{\theta}_0^*)$. Concretely, the *influence* of $\boldsymbol{z} \in \mathcal{S}$ on $\boldsymbol{z}_t$ is defined as the change in the prediction loss after removing example $\boldsymbol{z}$ from the training data [18]. The removal of a training example changes the objective and thus leads to a different model and parameters. Suppose that instead of removing $\boldsymbol{z}$ we change the weight of its contribution (i.e., its training loss) to the objective by some value ϵ. We can view the parameters that minimize this altered objective as a function of ϵ, i.e., $\boldsymbol{\theta}^*(\epsilon) = \arg\min_{\boldsymbol{\theta}}\{L(\mathcal{S}, \boldsymbol{\theta}) + \epsilon\ell(\boldsymbol{x}, y, \boldsymbol{\theta})\}$. Setting $\epsilon = 0$, we retrieve the optimal parameters for the original objective, i.e., $\boldsymbol{\theta}^*(0) = \boldsymbol{\theta}_0^*$. Moreover, observe that $\boldsymbol{\theta}^*(-\frac{1}{n})$ corresponds to the parameters that minimize the altered objective after removing training example $\boldsymbol{z}$. Based on this observation, the *exact influence* of $\boldsymbol{z}$ on the prediction for $\boldsymbol{z}_t$ is defined as:

$$I^{exact}(\boldsymbol{z}, \boldsymbol{z}_t) = \ell(\boldsymbol{z}_t, \boldsymbol{\theta}^*(-1/n)) - \ell(\boldsymbol{z}_t, \boldsymbol{\theta}^*(0)). \quad (1)$$

Computing the exact influence requires us to optimize the loss after removing a training point $\boldsymbol{z}$; repeating this for each training point is prohibitively costly. Instead, we approximate the exact influence. Specifically, we view the loss function as a function of ϵ, and make a linear approximation of the exact influence using the derivative of ℓ at point $\epsilon = 0$: $I^{exact}(\boldsymbol{z}, \boldsymbol{z}_t) \approx -\frac{1}{n} \left.\frac{\mathrm{d}\ell(\boldsymbol{z}_t, \boldsymbol{\theta}^*)}{\mathrm{d}\epsilon}\right|_{\epsilon=0}$ Since the term $\frac{1}{n}$ is the same for all $\boldsymbol{z}, \boldsymbol{z}_t$ pairs, we simply define (approximate) *influence* [18] as:

$$I(\boldsymbol{z}, \boldsymbol{z}_t) = -\left.\frac{\mathrm{d}\ell(\boldsymbol{z}_t, \boldsymbol{\theta}^*)}{\mathrm{d}\epsilon}\right|_{\epsilon=0}. \quad (2)$$

When the influence of $\boldsymbol{z}$ on $\boldsymbol{z}_t$ is *positive*, the loss tends to decrease, and we say that training example $\boldsymbol{x}$ *supports* the prediction for $\boldsymbol{z}_t$; otherwise, we say that the example *opposes* the prediction.

To compute the derivative of the loss, we use the chain rule to decompose it into the derivative of loss with respect to the parameters and the derivative of the parameters with respect to ϵ. Concretely, we have:

$$I(\boldsymbol{z}, \boldsymbol{z}_t) = -\left.\nabla_{\boldsymbol{\theta}^*}^{\intercal}\ell(\boldsymbol{z}_t, \boldsymbol{\theta}^*)\right|_{\boldsymbol{\theta}^*=\boldsymbol{\theta}_0^*} \left.\frac{\mathrm{d}\boldsymbol{\theta}^*}{\mathrm{d}\epsilon}\right|_{\epsilon=0}, \quad (3)$$

which is the dot product of two row vectors, the loss gradient $\nabla_{\boldsymbol{\theta}^*}\ell$ at $\boldsymbol{\theta}^* = \boldsymbol{\theta}^*(0)$ and the derivative of the optimal parameters for the altered objective $\frac{\mathrm{d}\boldsymbol{\theta}^*}{\mathrm{d}\epsilon}$ at $\epsilon = 0$.

It can be shown [7] that under certain conditions (second order differentiability and convexity of the loss function) the derivative of $\boldsymbol{\theta}^*$ can be expressed as:

$$\left.\frac{\mathrm{d}\boldsymbol{\theta}^*}{\mathrm{d}\epsilon}\right|_{\epsilon=0} = -\boldsymbol{H}_{\boldsymbol{\theta}^*}^{-1} \left.\nabla_{\boldsymbol{\theta}^*}\ell(\boldsymbol{z}, \boldsymbol{\theta}^*)\right|_{\boldsymbol{\theta}^*=\boldsymbol{\theta}_0^*}, \quad (4)$$

where $\boldsymbol{H}_{\boldsymbol{\theta}^*}$ is the Hessian matrix (containing the second order partial derivatives) of the objective $L(\mathcal{S}, \boldsymbol{\theta}^*)$ calculated at $\boldsymbol{\theta}^* = \boldsymbol{\theta}_0^*$.

Defining the vector function $\boldsymbol{g}(\boldsymbol{z})$ as the gradient of the loss of the example $\boldsymbol{z}$ calculated at $\boldsymbol{\theta}^* = \boldsymbol{\theta}_0^*$, and substituting it in Equations 3 and 4, we get:

$$I(\boldsymbol{z}, \boldsymbol{z}_t) = \boldsymbol{g}^{\intercal}(\boldsymbol{z}_t)\boldsymbol{H}_{\boldsymbol{\theta}^*}^{-1}\boldsymbol{g}(\boldsymbol{z}). \quad (5)$$

To explain the prediction for $\boldsymbol{z}_t$, we use Equation 5 to compute the influence of each training example $\boldsymbol{z}$, which can be done efficiently as suggested in [18]. The IF explanation [18] for the prediction for $\boldsymbol{z}_t$ consists of the top-k training examples with the highest influence.

3.2 AIDE INGREDIENTS

Existing approaches for influence-based explainability [18, 5] compile an explanation as a set of highly influential training examples. We claim that other aspects, besides high influence, are also important. Specifically, AIDE creates explanations that contain training examples with *negative influence*, considers their *labels*, their *proximity* to the test instance, and their *diversity*.

Negative Influence Recall that negative influence means that removing the training example decreases the loss, thus opposing the prediction. Let us investigate closely when an example can have high-magnitude negative influence.

For the following discussion, assume a binary classification task, i.e., $\mathcal{Y} = \{0, 1\}$, where the model predicts the probability $p_{\boldsymbol{\theta}}^*(\boldsymbol{x})$ of an input $\boldsymbol{z} = (\boldsymbol{x}, y)$ belonging to the positive class. Further assume that the loss function is the logistic loss (binary cross entropy):

$$\ell(\boldsymbol{z}, \boldsymbol{\theta}^*) = -\left(y \log(p_{\boldsymbol{\theta}}^*(\boldsymbol{x})) + (1-y)\log(1 - p_{\boldsymbol{\theta}}^*(\boldsymbol{x}))\right)$$

Consider a test instance $\boldsymbol{z}_t = (\boldsymbol{x}_t, y_t)$ and let $\boldsymbol{z}_t' = (\boldsymbol{x}_t, 1 - y_t)$ be a counterfactual instance with the opposite label. Then, for some training point $\boldsymbol{z}$ the following lemma associates its influence for the predictions for $\boldsymbol{z}_t$ and $\boldsymbol{z}_t'$.

Lemma 1. *In binary classification with logistic loss, the influence of a training point $\boldsymbol{z}$ to the predictions of $\boldsymbol{z}_t = (\boldsymbol{x}_t, y_t)$ and $\boldsymbol{z}_t' = (\boldsymbol{x}_t, 1 - y_t)$ is related as follows:*

$$I(\boldsymbol{z}, \boldsymbol{z}_t) = -\left(\frac{1 - p_{\boldsymbol{\theta}}^*(\boldsymbol{x}_t)}{p_{\boldsymbol{\theta}}^*(\boldsymbol{x}_t)}\right)^{2y_t - 1} I(\boldsymbol{z}, \boldsymbol{z}_t')$$

Suppose that $\boldsymbol{z}$ is a strong opposer to the prediction for $\boldsymbol{z}_t$, i.e., $I(\boldsymbol{z}, \boldsymbol{z}_t) < 0$ with high magnitude. Lemma 1 explains how this may occur. This can happen if $\boldsymbol{z}$ is a strong supporter for the prediction of the opposite label, i.e., $I(\boldsymbol{z}, \boldsymbol{z}_t') > 0$ with high magnitude.

Another way is when $\left(\frac{1-p^*_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_t)}{p^*_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_t)}\right)^{2y_t-1}$ is high. Let us examine what this term means. Suppose that the true class is the positive, i.e., $y_t = 1$. Then, the term equals the *predicted odds* of the model for the negative class. Conversely, when $y_t = 0$, the term equals the predicted odds for the positive class. That is, the term equals the predicted odds for the *opposite* class. So, the term is high when the model is confident about the wrong prediction for $\boldsymbol{z}_t$.

Therefore, if a training example $\boldsymbol{z}$ is a strong opposer (i.e., has a high-magnitude negative influence), then it would be a strong supporter if the opposite class was true (supporting the counterfactual $\boldsymbol{z}'_t$), or the model is confident about the wrong prediction, or some combination of both. Such training examples are important to understand the model's decision for $\boldsymbol{z}_t$, particularly when the true class of $\boldsymbol{z}_t$ is not apparent.

Label The influence of a training example does not carry any information about the class of the training example. It is thus possible that a positive and a negative example have both high influence for the test instance. While both may support (in case they have positive influence) or oppose (in case they have negative influence) the model's decision, they do so in different ways as they stand on opposite sides of the decision boundary. One presents an analogous example, while the other presents a contrasting example to the test instance. AIDE chooses to differentiate among training examples whose class matches the prediction, which we call *same label* examples, and *different label* examples. The comparison between same and different label examples supports *contrastivity* [23].

Proximity Influence is agnostic to the similarity of the training examples to the test instance. As noted in [5], there may exist outliers and mislabeled training examples that can exhibit high magnitude influence scores. Such examples are often *globally influential*, i.e., they are influential for many test instances, just because they are unusual. These examples are rarely useful as an explanation, and [5] proposes to normalize the influence of an example with their global influence. Nonetheless, in some cases these outliers are extremely useful, e.g., when explaining another outlier.

To enhance the *interpretability* of the explanation and to avoid hiding useful outliers, AIDE takes a different approach and considers the proximity $P(\boldsymbol{z}, \boldsymbol{z}_t)$ of a training example $\boldsymbol{z}$ to the instance to be explained $\boldsymbol{z}_t$. Proximity should be appropriately defined for the domain and data type. A general approach is to consider the cosine similarity between the model's internal representations (e.g., embeddings) for $\boldsymbol{z}$ and $\boldsymbol{z}_t$, i.e., $P(\boldsymbol{z}, \boldsymbol{z}_t) = \mathrm{sim}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}_t)$, where $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{x}}_t$ are the representations of the training example and test instance, respectively, and sim is the cosine similarity, which for positive coordinates takes values in $[0, 1]$.

Diversity Example-based explainability methods, like IF, RelatIF, and AIDE, return to the user a small set of training examples, aiming for explanation *compactness* [23]. It is thus important that the set of examples avoids *redundancy*. AIDE, in contrast to prior work [18, 5], considers the diversity of the explanation set. Assuming an internal representation of training examples and an appropriate similarity measure sim, we define diversity of a set $\mathcal{E}$ of training examples as $D(\mathcal{E}) = 1 - \frac{1}{|\mathcal{E}|(|\mathcal{E}|-1)} \sum_{\boldsymbol{z},\boldsymbol{z}' \in \mathcal{E}} \mathrm{sim}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}'})$.

3.3 AIDE QUADRANTS

AIDE constructs four distinct explanation lists for a specific test instance $\boldsymbol{z}_t$ to be explained. These lists contain training examples that (1) have influence of high magnitude, (2) have high proximity to $\boldsymbol{z}_t$, (3) are diverse, and (4) lie in the four quadrants formed by two dimensions, *influence* (positive or negative), and *label* (same as or different from the test instance). We name these quadrants as follows.

Support. It comprises examples with *positive influence* and with the *same label* as the test instance. They play a positive role in the prediction and resemble the test instance in terms of their characteristics: "*You get the same outcome with these*".

Support by Contrast. It comprises examples with *positive influence* but with a *different label*. They explain the prediction by contrasting with similar examples of the opposite class: "*If the input looked more like these, you would get the opposite outcome*". They act similar to *nearest counterfactual explanations* [33, 17], but with the benefit that they represent *actual*, and not synthesized, examples.

Oppose. It comprises examples with *negative influence* and *different labels*. These examples are analogous to the test instance if it had the opposite label, and persuade the model that the test instance should belong to their class instead: "*The outcome is incorrect, because the input looks more like these*".

Oppose by Contrast. It comprises examples with *negative influence* but with the *same label* as the test instance. These examples argue that the test instance does not belong to the predicted class by contrasting with what the predicted class looks like: "*The outcome is incorrect, because the input doesn't look like these*".

To select the appropriate examples for each quadrant, we perform a series of steps. After partitioning the training examples in the four quadrants, we select only examples with high magnitude. We use the Interquartile Range (IQR) method, [3], to keep examples with positive influence above $Q_3 + \lambda IQR$, and to keep examples with negative influence below $Q_1 - \lambda IQR$, where Q_1 and Q_3 are the first and the third quartiles of the influence distribution, $IQR = Q_3 - Q_1$, and λ is a coefficient that controls the number of high-magnitude influential points, and is empirically determined.

After this filtering, we end up with a candidate set $\mathcal{S}_q$ of training examples for each quadrant $q \in \{1, 2, 3, 4\}$.

Among the training examples left in each quadrant, we select a small set of k examples that has high magnitude influence, high proximity to the test instance, and is diverse. Specifically, we aim for a balance among the three measures:

$$\mathcal{E}_q = \arg \max_{\mathcal{E} \subseteq \mathcal{S}_q, |\mathcal{E}|=k} \sum_{\boldsymbol{z} \in \mathcal{E}} (\alpha |I(\boldsymbol{z}, \boldsymbol{z}_t)| + \beta P(\boldsymbol{z}, \boldsymbol{z}_t)) + \gamma D(\mathcal{E}),$$

where α, β, γ are weighs empirically determined. Similar to other submodular maximization problems [11], we construct $\mathcal{E}_q$ in a incremental way, each time greedily selecting the example that maximizes the objective.

4 EXPERIMENTS

4.1 DATASETS AND MODELS

In our experiments, we used two datasets: the SMS Spam dataset[2], which comprises a collection of text messages labeled as either spam or non-spam (ham), commonly used for text classification and a derivative dataset with pictures of dogs and fish extracted from Imagenet[3]. For the spam classification task, we employed the BERT-base pre-trained word embedding model and incorporated two sequential layers to capture the specific characteristics of our data. Regarding the image classification task, we utilized a pre-trained InceptionV3 model removing the output layer and appending sequential layers to learn the peculiarity of our task. All the baselines were implemented with instructions given in their papers and GitHub repositories.

4.2 QUALITATIVE EVALUATION

The baseline methods that we will compare AIDE to are IF [18], RelatIF [5], Datamodels [15], and TraceIn [9]. We provide the some anecdotes to compare the informativeness and understandability qualitatively. Apart from the examples given in 1, we selected one text and one image sample both corresponding to an ambiguous prediction. This diverse set of test cases allowed us to evaluate the performance and capabilities of AIDE in explaining predictions across different scenarios and levels of prediction certainty. The similarity between training examples, which are used for both proximity and diversity, is based on generating embeddings for images and text and using cosine similarity between the embeddings.

In the following, we present findings for the intent of clarifying an ambiguous prediction. Results for the other intents are found in the supplementary material.

[2] https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

[3] https://www.image-net.org/

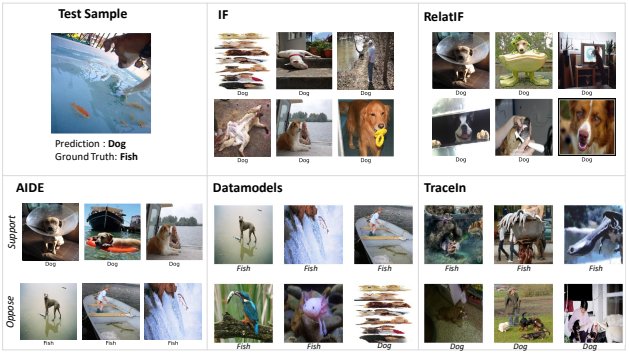

Figure 2: Explanations to clarify an ambiguous prediction for the test image by baselines and AIDE.

When faced with an ambiguous sample as in Figure 2, where the image contains both a dog and a fish, understanding why the model chose a specific class (in this case, a dog) despite the ground truth being a fish becomes crucial. AIDE's explanation unveils the underlying logic or rules employed during the labeling process that the model failed to generalize effectively. By examining the supporters, we observe that the model learns from both dog-related features and water-related features, which aligns with common sense. However, the opposers suggest the potential existence of a labeling rule that associates images containing both dogs and fish with the "fish" label. This rule may not have been strongly represented in the training data, leading to the model's inefficient learning of this specific rule. Unlike other methods such as ReleatIf and TraceIn, which lack comprehensive explanations, or IF, which is sensitive to outliers, Datamodels is in stark contrast to AIDE. We observed that when confronted with mislabeled or ambiguous samples, Datamodels may explain the opposite label prediction rather than the model's actual prediction. This happens due to a discordance between the model being explained and the intermediary models (of the same class) used to compute the importance of individual training examples; in fact, about 20% of the intermediary models predict a different class that the actual model.

Table 1 shows another ambiguous test sample from of spam classification. Determining whether this message is spam or not is challenging since it does not exhibit the typical form of either a "ham" message or a common spam message. Instead, it takes the form of an aphorism, which falls into an ambiguous category of messages. AIDE's supporters shed light on the presence of numerous aphorisms in the training set that are labeled as spam, indicating the existence of a labeling rule for categorizing such messages as spam during labeling. Thus, the model can correctly classify this message despite its ambiguity. The supporting samples provided by AIDE emphasize a specific rule that was likely injected during the labeling process, indicating that aphorisms were considered spam. These supporting samples contributed to the correct classification decision by reinforcing this rule. The opposing examples suggest that classifying the message

Table 1: AIDE for an ambiguous test message.

Test prediction of interest	**Label**
'Do you realize that in about 40 years, we'll have thousands of old ladies running around with tattoos?'	Spam
Supporters	
'Do you ever notice that when you're driving, anyone going slower than you is an idiot and everyone driving faster than you is a maniac?'	Spam
'How come it takes so little time for a child who is afraid of the dark to become a teenager who wants to stay out all night? '	Spam
'LIFE has never been this much fun and great until you came in. You made it truly special for me. I won't forget you!'	Spam
Opposers	
'You are always putting your business out there. You put pictures of your ass on facebook. Why would i think a picture of your room would hurt you, make you feel violated.'	Ham
'Yo you guys ever figure out how much we need for alcohol? Jay and I are trying to figure out how much we can spend on weed'	Ham
'Any chance you might have had with me evaporated as soon as you violated my privacy by stealing my phone number from your employer's paperwork.'	Ham

as non-spam could be a plausible interpretation.

However, the model's ability to correctly classify the message indicates that the rule regarding aphorisms being classified as spam is supported by an adequate number of training samples. This indicates that the model has learned and generalized this rule effectively.

4.3 QUANTITATIVE EVALUATION

Correctness. In this set of experiments, we follow the controlled synthetic data check protocol of [23]. AIDE possesses the capability to detect rules employed during the labeling process while providing explanations for corresponding test samples. For instance, if a rule dictates labeling messages shorter than 30 characters with a question mark as "spam" in the training set, AIDE can identify similar instances while explaining a test sample with analogous characteristics. To enhance the robustness of this detection, we introduce ambiguity by labeling a subset of training samples adhering to the rule with an opposite label, anticipating these instances in the "Oppose by" category. Subsequently, we evaluate the precision of AIDE by counting the retrieved samples conforming to the rule.

In this experimental setup, three rules were employed. ***Rule 1***: *All French messages are "spam"*. Initially, there were no French messages, 110 French messages were added in the following ratio 88 spam and 22 ham.
Rule 2: *if the message is shorter than 30 and it contains "?", it's labeled "spam"*. Initially, all 197 such messages were ham and intervention resulted in 157 spam and 43 ham.
Rule 3: *If a message contains a sequence of 4 consecutive digits, it's labeled "ham"*. Initially, 504 of 512 such samples were spam and intervention resulted in 398 ham.

Before gauging the precision of the explanation, it is imperative to ensure that the model has effectively learned the rules. Three metrics are employed for this assessment: 1) Accuracy of Learning the Rule: Evaluating the model's performance on test samples corresponding to a rule. 2) Log-Likelihood: Expecting a substantial change in the log-likelihood of intervened points (LL_i) after the introduction of the rule, while the log-likelihood of untouched points (LL_u) is anticipated to remain relatively stable. 3) Probability Scores: Anticipating a notable alteration in the probability scores of intervened (Ps_i), compared to untouched point (Ps_u). Table 2 illustrates the results of these metrics. In all cases, the model demonstrates high accuracy in learning the rules without impacting its decisions for untouched points.

	Acc	LL_i		LL_u	Ps_i	Ps_u
Rule 1	0.83	Before	-5.87	-9.4	100	15
		After	-0.42	-9.2		
Rule 2	0.85	Before	-12	-9.3	100	24.5
		After	-3.4	-7.2		
Rule 3	0.92	Before	-0.07	-10.6	98	12
		After	-1.83	-9.5		

Table 2: Model's assessment in learning the rules

We expect to find rule followers and breakers in the support and oppose quadrants of AIDE, respectively, which is the case with high (around 0.9) precision for all rules. We repeat this experiment, for other baselines, and expect to find rule followers (resp. breakers) when we look at the training data with high positive (resp. low negative) influence. Fig. 3 shows that IF and Datamodels perform well but are not consistent. RelatIF performs poorly in uncovering followers and breakers, because of its loss-based outlier elimination. RelatIF treats training data with high loss as outliers, and excludes them from explanation lists—the rationale is that such data are global influencers and would appear in all explanations, thus have little utility. But in this case, it is precisely the rule followers and particularly the minority of rule breakers that have high losses due to the ambiguity in the labeling rule. TraceIn also fails to uncover the rule due to its low efficiency of identifying truly important samples, which is also demonstrated by [25].

	AIDE		IF		RelatIF		DM		TraceIn	
	P_{fol}	P_{br}	P_{fol}	P_{br}	P_{fol}	P_{br}	P_{fol}	P_{br}	P_{fol}	P_{br}
Rule 1	**0.99**	0.9	0.93	**0.91**	0.59	0.25	0.9	0.8	0.22	0.3
Rule 2	**0.88**	**0.8**	0.52	0.74	0.22	0.1	0.83	0.48	0.29	0.38
Rule 3	**0.9**	**0.87**	0.85	0.86	0.31	0.15	0.76	0.73	0.37	0.31

Table 3: Precision in uncovering rule followers P_{fol} and breakers P_{br}.

Faithfulness. To assess the quantitative effectiveness of AIDE, we employ the faithfulness metric, correlating sample similarity with the concordance of their explanations. Sample similarity is computed using cosine similarity of embeddings, and explanation similarity is computed using

Intent	**ML knowledge**	**Q1 (%)**	**Q2 (%)**	**Q3 (%)**	**Q4 (%)**	**Q5 (%)**	**Q6 (%)**	**Q7 (%)**
Int. correct	Advanced	88	94	100	75	-	-	-
	Intermediate	87	87	93	67	-	-	-
Inv. wrong	Advanced	88	81	88	81	63	63	88
	Intermediate	67	66	8	66	80	66	87
Cl. ambiguous	Advanced	69	81	88	-	60	-	69
	Intermediate	73	73	80	-	73	-	67

Table 4: Results from the user study per intent

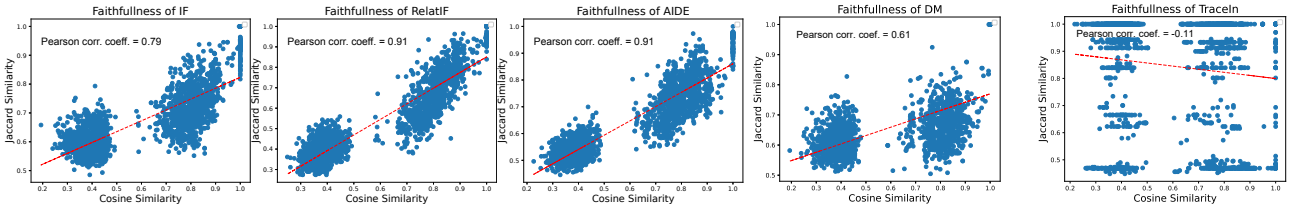

Figure 3: Faithfulness in terms of explanation similarity vs. instance pair similarity.

	Q8 (%)	Q9 (%)	Q10 (%)
Advanced	88	100	100
Intermediate	80	73	100

Table 5: Overall evaluation of AIDE

Fuzzy Jaccard [26] metric. For each sample prediction, a set is formed with the indices of training samples returned in the explanation. Fuzzy Jaccard involves solving a maximum bipartite matching problem. In spam classification, 100 random test samples are chosen. For each, the 10 most similar and dissimilar samples are identified, resulting in 2000 pairs. The same procedure is replicated with the image dataset, commencing with 50 random samples instead of 100, as this dataset is smaller in scale. The cosine similarity is plotted against Fuzzy Jaccard along with a linear regression line in red, and the Pearson correlation coefficient (PCC) for the spam datasets in Figure 3, the figures for the image dataset exhibiting the same trend can be found in the supplementary materials. RelatIF and AIDE perform similarly. In contrast, IF and Datamodels have a lower PCC and do not exhibit a clear separation between instance pairs of low and high similarity. This is because their explanations tend to include training data outliers that appear in all explanations (globally influential), and which inflate the explanation similarity even for dissimilar pairs. Finally, TraceIn performs poorly and provides identical explanations for dissimilar points, which is due to its extremely high susceptibility to outliers. RelatIF and AIDE are more robust because they seek to eliminate outliers, albeit in different ways (based on loss and proximity, respectively).

4.4 USER STUDY

Following the recommendations of [30], we invited 33 participants with diverse levels of machine learning knowledge to assess AIDE explanations based on the following criteria:

Mental Model: Q1. The explanation provided helped to understand the model's prediction. To what extent do you agree?
Clarity: Q2. The explanation is clear and easy to comprehend. To what extent do you agree?
Usefulness of AIDE Quadrants: Q3, Q4, Q5, Q6. The group "Support", "Support by Contrast", "Oppose", "Oppose by Contrast" enhances the understanding of the model's prediction. To what extent do you agree?
Human-AI Collaboration: Q7. Did the explanation help understand how the model's performance can be improved?
Effectiveness: Q8. How would you rate the overall effectiveness of AIDE in helping to understand model predictions?
Helpfulness: Q9. To what extent did you find the provided samples relevant to the specific intent you encountered?
Contrastivity: Q10. Do you believe that the use of contrast in the groups of images shown enhanced your understanding of the model predictions?

All questions where accompanied by a 5-point Likert scale. All positive (i.e., strongly agree, somewhat agree) answers are considered to be in agreement. The metrics collectively provide a comprehensive qualitative assessment of AIDE's performance from the user's perspective, taking into account various aspects of interpretability and usability. In Table 4, the percentage of participants who agreed on the high quality of specific aspects of AIDEs' explanation for particular intents is presented. Whereas, in Table 5, the percentages of users who overall highly assessed AIDE's effectiveness, the utility of contrast in explanation, and AIDE's capability to tailor explanations according to user intent are depicted. A noteworthy observation is that participants with more advanced expertise tend to rate highly more frequently across various aspects of AIDE's explanation.

In our user study we did not explicitly compare with IF and RelatIF to prevent potential bias in favor of our novel method. However, the user study implicitly compares AIDE to IF and RelatIF. Observe that the support quadrant of AIDE contains explanations very similar to those IF and RelatIF produce. Specifically, we implicitly draw conclusions on the added value of AIDE, through the targeted questions that assess: (1) the *significance of the other three quadrants* (Q4, Q5, Q6), where 63%–81% of participants agree; and (2) the *importance of contrastivity* (Q10), where 100% of the participants agree.

5 CONCLUSION

In this paper, we introduce AIDE, a novel example-based explainability method that generates diverse and contrastive explanations tailored to user's needs and intentions. Through experiments on text and image datasets, we demonstrate AIDE's effectiveness in interpreting model decisions, uncovering the reasons behind errors, and identifying whether the model has learned complex and unconventional patterns present in the training data. Quantitative and qualitative analysis affirms that AIDE outperforms existing approaches.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/294a8ed24b1ad22ec2e7efea049b8737-Paper.pdf.

[2] Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 700–712. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/075b051ec3d22dac7b33f788da631fd4-Paper.pdf.

[3] Alan Agresti and Christine Franklin. Statistics: The art and science of learning from data. 2005.

[4] Ancona. A unified view of gradient-based attribution methods for deep neural networks. In *Workshop on Interpreting, Explaining and Visualizing Deep Learning*. NIPS, 2017.

[5] Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. Relatif: Identifying explanatory training examples via relative influence. *PMLR*, 2020.

[6] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 883–892. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/chen18j.html.

[7] R Dennis Cook and Sanford Weisberg. *Residuals and influence in regression*. New York: Chapman and Hall, 1982.

[8] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2950–2958, 2019.

[9] Garima, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[10] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019.

[11] Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *WWW*, pages 381–390. ACM, 2009.

[12] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)*, 51:1 – 42, 2018.

[13] Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the American statistical association*, 69(346):383–393, 1974.

[14] Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. Data cleansing for models trained with SGD. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4215–4224, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5f14615696649541a025d3d0f8e0447f-Abstract.html.

[15] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9525–9587. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ilyas22a.html.

[16] Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.386. URL https://aclanthology.org/2020.acl-main.386.

[17] Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Computing Surveys*, 55(5): 1–29, 2022.

[18] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

[19] Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning, 2022.

[20] Peter Lipton. Contrastive explanation. *Royal Institute of Philosophy Supplement*, 27:247–266, 1990. doi: 10.1017/s1358246100005130.

[21] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[22] Christoph Molnar. *Interpretable Machine Learning*. 2 edition, 2022. URL https://christophm.github.io/interpretable-ml-book.

[23] Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Comput. Surv.*, feb 2023. ISSN 0360-0300. doi: 10.1145/3583558. URL https://doi.org/10.1145/3583558.

[24] Osonde A Osoba and William Welser IV. *An intelligence in our image: The risks of bias and errors in artificial intelligence*. Rand Corporation, 2017.

[25] Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[26] Matej Petković, Blaž Škrlj, Dragi Kocev, and Nikola Simidjievski. Fuzzy jaccard index: A robust comparison of ordered lists. *Applied Soft Computing*, 113:107849, 2021. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2021.107849. URL https://www.sciencedirect.com/science/article/pii/S1568494621007717.

[27] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930, 2020.

[28] 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*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

[29] 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.

[30] Yao Rong, Tobias Leemann, Thai trang Nguyen, Lisa Fiedler, Peizhu Qian, Vaibhav Unhelkar, Tina Seidel, Gjergji Kasneci, and Enkelejda Kasneci. Towards human-centered explainable ai: A survey of user studies for model explanations, 2023.

[31] João Marques Silva, Thomas Gerspacher, Martin Cooper, Alexey Ignatiev, and Nina Narodytska. Explaining naive bayes and other linear classifiers with polynomial time and delay. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, volume 33, pages 1–11, 2020.

[32] Jason Tashea. Courts are using ai to sentence criminals. that must stop now, Apr 2017. URL https://www.wired.com/2017/04/courts-using-ai-sentence-criminals-must-stop-

[33] 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.

[34] Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

URL https://proceedings.neurips.cc/paper_files/paper/2018/file/8a7129b8f3edd95b7d969dfc2c8e9d9d-Paper.pdf.