Explainable Machine Learning in Deployment

Umang Bhatt[1,2,4,10], Alice Xiang[2], Shubham Sharma[3], Adrian Weller [4,5,10], Ankur Taly[6], Yunhan Jia[7], Joydeep Ghosh[3,8], Ruchir Puri[9], José M. F. Moura[1], Peter Eckersley[2]
[1]Carnegie Mellon University, [2]Partnership on AI, [3]University of Texas at Austin, [4]University of Cambridge, [5]The Alan Turing Institute, [6]Fiddler Labs, [7]Baidu, [8]CognitiveScale, [9]IBM Research, [10]Leverhulme CFI
umang@partnershiponai.org

ABSTRACT

Explainable machine learning seeks to provide various stakeholders with insights into model behavior via feature importance scores, counterfactual explanations, and influential samples, among other techniques. Recent advances in this line of work, however, have gone without surveys of how organizations are using these techniques in practice. This study explores how organizations view and use explainability for stakeholder consumption. We find that the majority of deployments are not for end users affected by the model but for machine learning engineers, who use explainability to debug the model itself. There is a gap between explainability in practice and the goal of public transparency, since explanations primarily serve internal stakeholders rather than external ones. Our study synthesizes the limitations with current explainability techniques that hamper their use for end users. To facilitate end user interaction, we develop a framework for establishing clear goals for explainability, including a focus on normative desiderata.

KEYWORDS

machine learning, explainability, transparency, deployed systems, qualitative study

1 INTRODUCTION

Machine learning (ML) systems are being increasingly embedded into many aspects of daily life, such as healthcare [18], finance [27], and social media [6]. In an effort to design ML systems worthy of human trust, research has proposed a variety of techniques for explaining ML models to stakeholders. Deemed "explainability," this body of previous work attempts to illuminate the reasoning used by ML models. "Explainability" loosely refers to any technique that helps the user of a ML model understand why the model behaves the way it does. Explanations can come in many forms: from telling patients which symptoms were indicative of a particular diagnosis [36] to helping factory workers analyze inefficiencies in a production pipeline [19].

Explainability has been touted as a way to enhance transparency of ML systems, particularly for end users. Often releasing (or forcing organizations to release) the data that models were trained on or the accompanying code is challenging due to user privacy issues and incentives to preserve trade secrecy. Moreover, end users are generally not equipped to be able to understand how raw data and code translate into benefits or harms that might affect them individually. By providing an explanation for how the model made a decision, explainability techniques seek to provide transparency directly targeted to human users, often with the goal of improving user trust. The importance of explainability as a concept has been reflected in legal and ethical guidelines for data and ML. In cases of automated decision-making, Articles 13-15 of the European General Data Protection Regulation (GDPR) require that data subjects have access to "meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject" [44]. In addition, technology companies have released artificial intelligence (AI) principles that include transparency as a core value, including notions of explainability, interpretability, or intelligibility [1, 2].

This growing interest in "peering under the hood" of ML models and being able to provide explanations to human users has made explainability an important subfield of ML. Despite this growing literature, there has been little work characterizing how explanations have been deployed by organizations in the real world.

In this paper, we attempt to understand how organizations have deployed local explainability techniques so that we can observe which techniques work best in deployment, report on shortcomings of particular techniques, and better guide future research. We focus specifically on local explainability techniques. These techniques explain individual predictions, which makes them more relevant for providing transparency for end users.

Our interview study synthesizes interviews of roughly fifty people from approximately thirty organizations to understand which explainability techniques are used and how. We report trends from two sets of interviews and provide suggestions for future research directions that combine explainability with privacy, fairness, and causality. To the best of our knowledge, we are the first to conduct a study of how explainability techniques are used by organizations that use ML models in their workflows.

1.1 Terms

For the sake of clarity, we define various terms based on the context in which they appear in the forthcoming prose.

- *Predictor* refers to a trained ML model.
- *Explainability* refers to attempts to provide insights into the predictor's behavior.
- *Stakeholders* are the people who either want the model to be "explainable", will consume the explanation, or are affected by the model itself.
- *Practice* refers to the real-world context in which the predictor has been deployed.
- *Local Explainability* aims to explain the predictor's behavior at a specific input.
- *Global Explainability* attempts to understand the high-level concepts and reasoning used by a predictor.

1.2 Format

The rest of this paper is organized as follows:

(1) We discuss the methodology of our survey, describing the interviews and introducing notation in Section 2.
(2) We summarize our overall findings in Section 3.
(3) We detail how local explainability techniques are used at various organizations and discuss technique-specific takeaways in Section 4.
(4) We develop a framework for establishing clear goals when deploying local explainability in Section 5.1.
(5) We discuss ethical desiderata for explainability in Section 5.2.
(6) We conclude in Section 6.

2 METHODOLOGY

2.1 Interview Structure

In the spirit of Holstein et al. [29], we study how industry practitioners look at and deploy explainable ML. Specifically, we study how particular organizations deploy explainability algorithms, including who consumes the explanation and how it is evaluated for the intended stakeholder. We conduct two set of interviews: Group 1 looked at how data scientists who are not currently using explainable machine learning hope to leverage various explainability tools, while Group 2, the crux of this paper, looked at how explainable machine learning has been deployed in practice.

For Group 1, Fiddler Labs led a set of around twenty interviews to assess explainability needs across various organizations in technology and financial services. We specifically focused on teams that do not currently employ explainability technology. These semi-structured, hour-long interviews included the following questions:

- What are your ML use cases?
- What is your current model development workflow?
- What explainability tools do you use?
- What are your pain points in deploying ML models?
- Do you think explainability will help address those points?

Group 2 spanned roughly thirty people across approximately twenty different organizations, both for-profit and non-profit. Most of these organizations are members of the Partnership on AI, which is a global multistakeholder non-profit established to study and formulate best practices on AI technologies for the benefit of society. With each individual, we held a thirty-minute to two-hour semi-structured interview to understand the state of explainability in their organization, their motivation for using explanations, and the benefits and shortcomings of the methods used. Some organizations asked to stay anonymous, not to be referred to explicitly in the prose, or not to be included in the acknowledgements.

Of the people we spoke with in Group 2, around one-third represented non-profit organizations (academics, civil society organizations, and think tanks), while the rest worked for for-profit organizations (corporations, industrial research labs, and start-ups). Broken down by organization, around half were for-profit and half were academic / non-profit. Around one-third of the interviewees were executives at their organization, around half were research scientists or engineers, and the remainder comprised professors at academic institutions, who commented on the consulting they have done with industry leaders to commercialize their research. The questions we asked Group 2 included, but were not limited to, the following:

- Does your organization use ML model explanations?
- What type of explanations have you used (e.g., feature-based, sample-based, counterfactual, or natural language)?
- Who is the audience for the model explanation (e.g., research scientists, product managers, domain experts, or users)?
- In what context have you deployed the explanations (e.g., informing the development process, informing human decision-makers about the model, or informing the end user on how actions were taken based on the model's output)?
- How does your organization decide when and where to use model explanations?

2.2 Technical Details

Let $\mathcal{F}$ be a family of black box predictors. Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input space and output space, respectively. A black box predictor $f \in \mathcal{F}$ maps an input $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ to an output $f(\boldsymbol{x}) \in \mathcal{Y}$, $f : \mathbb{R}^d \mapsto \mathcal{Y}$. When we assume f has a parametric form, we denote that parametric black box predictor as f_θ where $\theta \in \Theta$. We denote $\mathcal{D}$ to be a training dataset, where $(\boldsymbol{x}^{(i)} \in \mathcal{X}, y^{(i)} \in \mathcal{Y})$ is an input-output pair, $|\mathcal{D}| = N$, and $\mathcal{D}_x$ denotes all N inputs $\boldsymbol{x}$.

Each organization we spoke with has deployed an ML model f. They hope to explain a data point $\boldsymbol{x}$ using an explanation function $\boldsymbol{g}$. Local explainability refers to an explanation for why f predicted $f(\boldsymbol{x})$ for a fixed point $\boldsymbol{x}$. The local explanation methods we discuss come in one of the following forms: Which feature x_i of $\boldsymbol{x}$ was most important for prediction with f? Which training datapoint $z \in \mathcal{D}_{\boldsymbol{x}}$ was most important to $f(\boldsymbol{x})$? What is the minimal change to the input $\boldsymbol{x}$ required to change the output $f(\boldsymbol{x})$?

In this paper, we deliberately decide to focus on the more popularly deployed local explainability techniques instead of global explainability techniques. Global explainability refers to techniques that attempt to explain the model as a whole. These techniques attempt to characterize the concepts learned by the model [32], simpler models learned from the representation of complex models [19], prototypical samples from a particular model output [14], or the topology of the data itself [21]. None of our interviewees reported deploying global explainability techniques, though some studied these techniques in research settings.

3 SUMMARY OF FINDINGS

3.1 Explainability Needs

This subsection provides an overview of explainability needs that were uncovered with Group 1 - data scientists from organizations that do not currently deploy explainability techniques. These scientists were asked to describe their top "pain points" in building and deploying ML models, and how they hope to use explainability.

- **Model performance debugging**: Most data scientists struggle with debugging poor model performance. They wish to identify causes for why the model performs poorly on certain inputs, and also to identify regions of the input space with below average performance. In addition, they seek guidance on how to engineer new features, drop redundant features, and gather more data to improve model performance. For instance, one data scientist said: "If I have 60 features, maybe it's equally effective if I just have 5 features." Dealing with feature interactions was also a concern, as the data scientist continued, "Feature A will impact feature B, [since] feature A might negatively affect feature B - how do I attribute [importance in the presence of] correlations?" Others mentioned explainability as a debugging solution, helping to "narrow down where things are broken."

- **Model monitoring**: Several data scientists worry about drift in the feature and prediction distributions after deployment. Ideally, they would like to be alerted when there is a significant drift relative to the training distribution [7, 46]. One organization would like explanations for how drift in feature distributions would impact model outcomes and feature contribution to the model: "We can compute how much each feature is drifting, but we want to cross-reference with which features are impacting the model a lot."
- **Model transparency**: Organizations that deploy models to make decisions that directly affect end users seek explanations for model predictions. The explanations are meant to increase model transparency and comply with current or forthcoming regulation. In general, data scientists believe that explanations can also help communicate predictions to a broader external audience of other business teams and customers. One company stressed the need to "show your work" to provide reasons on underwriting decisions to customers, and another company needed explanations to respond to customer complaints.
- **Model audit**: In financial organizations, due to regulatory requirements, all deployed ML models must go through an internal audit. Data scientists building these models need to have them reviewed by internal risk and legal teams. One of the goals of the model audit is to conduct various kinds of tests provided by regulations like SR 11-7 [43]. An effective model validation framework should include: (1) evaluation of conceptual soundness of the model, (2) ongoing monitoring, including benchmarking, and (3) outcomes analysis, including back-testing. Explainability is viewed as a tool for evaluating the soundness of the model on various data points. Financial institutions would like to conduct sensitivity analyses, checking the impact of small changes to inputs on model outputs. Unexpectedly large changes in outputs can indicate an unstable model.

3.2 Explainability Usage

In Table 1, we aggregate some of the explainability use cases that we received from different organizations in Group 2. For each use case, we define the domain of use (i.e., the industry in which the model is deployed), the purpose of the model, the explainability technique used, the stakeholder consuming the explanation, and how the explanation is evaluated. Evaluation criteria denote how the organization compares the success of various explanation functions for the chosen technique (e.g., after selecting feature importance as the technique, an organization can compare LIME [48] and SHAP [35] explanations via the faithfulness criterion [66]).

In our study, feature importance was the most common explainability technique, and Shapley values were the most common type of feature importance explanation. The most common stakeholders were ML Engineers (or Research Scientists), followed by domain experts (Loan Officers and Content Moderators). Section 4 provides definitions for each technique and further details on how these techniques were used at Group 2 organizations.

3.3 Stakeholders

Most organizations in Group 2 deploy explainability atop their existing ML workflow for one of the following stakeholders:

(1) **Executives**: These individuals deem explainability necessary to align with the company's internal AI principles. One research scientist felt that "explainability was strongly advised and marketed by higher-ups," though sometimes explainability simply became a checkbox.

(2) **ML Engineers**: These individuals (including data scientists and researchers) train ML models at their organization and use explainability techniques to understand how the trained model works: do the most important features, most similar samples, and nearest training point(s) in the opposite class make sense? Using explainability to debug what the model has learned, this group of individuals were the most common explanation consumers in our study.

(3) **End Users**: This is the most intuitive consumer of an explanation. The person consuming the output of an ML model or making a decision based on the model output is the end user. Explainability shows the end user why the model behaved the way it did, which is important for showing the model is trustworthy and also providing greater transparency.

(4) **Other Stakeholders**: There are many other possible stakeholders for explainability. One such group is regulators, who may mandate that certain algorithmic decision-making systems provide explanations either for affected populations or for their own regulatory activities. It is important that this group understands how explanations are deployed based on existing research, what techniques are feasible, and how the techniques can align with the desired explanation from a model. Another group is domain experts, who are often tasked with auditing the model's behavior and ensuring it aligns with expert intuition. For many organizations, minimizing the divergence between the expert's intuition and the explanation used by the model is key to successfully implementing explainability.

Overwhelmingly, we found that local explainability techniques are mostly consumed by ML engineers and data scientists to audit models before deployment rather than to provide explanations to end users. Our interviews reveal factors that prevent organizations from showing explanations to end users or those affected by decisions made from ML model outputs.

3.4 Beyond Deep Learning

Though deep learning has gained popularity in recent years, many organizations in Group 2 still use classical ML techniques (e.g., logistic regression, support vector machines, and GP regression), likely due to a need for simpler, interpretable models [49].

Domain	Model Purpose	Explainability Technique	Stakeholders	Evaluation Criteria
Finance	Loan Repayment	Feature Importance	Loan Officers	Completeness [35]
Content Moderation	Malicious Reviews	Feature Importance	Content Moderators	Completeness [35]
Finance	Cash Distribution	Feature Importance	ML Engineers	Sensitivity [66]
Facial Recognition	Smile Detection	Feature Importance	ML Engineers	Faithfulness [8]
Content Moderation	Sentiment Analysis	Feature Importance	QA ML Engineers	ℓ_2 norm
Healthcare	Medicare access	Counterfactual Explanations	ML Engineers	normalized ℓ_1 norm
Content Moderation	Object Detection	Adversarial Perturbation	QA ML Engineers	ℓ_2 norm

Table 1: Summary of deployed local explainability use cases

A subset of the explainability community has focused on interpreting black-box deep learning models, even though practitioners overwhelmingly feel that there is a dearth of model-specific techniques to understand traditional ML models. For example, one research scientist noted that, "Many [financial institutions] use kernel-based methods on tabular data." As a result, there is a desire to translate explainability techniques for kernel support vector machines for genomics [54] to models trained on tabular data.

Model agnostic techniques like Lundberg and Lee [35] can be used for traditional models, but are "likely overkill" for explaining kernel-based ML models, according to one research scientist, since model-agnostic methods can be computationally expensive and lead to poorly approximated explanations.

3.5 Key Takeaways

This subsection summarizes some key takeaways from Group 2 that shed light on the reasons for the limited deployment of explainability techniques and their use primarily as sanity checks for ML engineers. Organizations generally still consider the judgments of domain experts to be the implicit ground truth for explanations. Since explanations produced by current techniques often deviate from the understanding of domain experts, some organizations still use human experts to evaluate the explanation before it is presented to users. Part of this deviation stems from the potential for ML explanations to reflect spurious correlations, which result from models detecting patterns in the data that lack causal underpinnings. As a result, organizations find explainability techniques useful for helping their ML engineers identify inconsistencies between the model's explanations and their intuition or that of domain experts, rather than for directly providing explanations to end users.

In addition, there are technical limitations that make it difficult for organizations to show end users explanations in real-time. The non-convexity of certain models make certain explanations (e.g., providing the most influential datapoints) hard to compute quickly. Moreover, providing certain explanations can raise privacy concerns by running the risk of model inversion.

More broadly, organizations lack frameworks for deciding why they want an explanation, and current research fails to capture the objective of an explanation. For example, large gradients, representing the direction of maximal variation with respect to the output manifold, do not necessarily "explain" anything to end users. At best, gradient-based explanations provide an interpretation of how the model behaves upon an infinitesimal perturbation (not necessarily a feasible one [30]), but does not "explain" if the model captures the underlying causal mechanism in the data.

4 DEPLOYING LOCAL EXPLAINABILITY

In this section, we dive into how local explainability techniques are used at various organizations (Group 2) . We start by defining each local explainability technique, then discuss organizations' use cases, and finally report takeaways for the technique in question.

4.1 Feature Importance

Feature importance was by far the most popular technique we found across our study. It is used across domains (finance, healthcare, facial recognition, content moderation). Also known as feature-level interpretations, feature attributions, or saliency maps, this method is by far the most widely used and most well-studied explainability technique [12, 26].

4.1.1 Formulation. Feature importance defines an explanation functional $\boldsymbol{g} : \boldsymbol{f} \times \mathbb{R}^d \mapsto \mathbb{R}^d$ that takes in a predictor $\boldsymbol{f}$ and a point of interest $\boldsymbol{x}$ and returns importance scores $\boldsymbol{g}(\boldsymbol{f}, \boldsymbol{x}) = \boldsymbol{\phi_x} \in \mathbb{R}^d$ for all features, where $\boldsymbol{g}(\boldsymbol{f}, \boldsymbol{x})_i = \phi_{\boldsymbol{x}, i}$ (simplified to ϕ_i in context) is the importance of (or attribution for) feature x_i of $\boldsymbol{x}$.

These explanation functionals roughly fall into two categories: perturbation-based techniques [9, 16, 23, 35, 48, 58] and gradient-based techniques [8, 41, 53, 55, 56, 59]. Note that gradient-based techniques can be seen as a special case of a perturbation-based technique with an infinitesimal perturbation size. Heatmaps are also a type of feature-level explanation that denote how important a region or collection of features, is [5, 23]. A prominent class of perturbation based methods are based on Shapley values from cooperative game theory [50]. Shapley values are a fair way to distribute the gains from a cooperative game to its players. In applying the method to explaining a model prediction, a cooperative game is defined between the features with the model prediction as the gain. The highlight of Shapley values is that they enjoy axiomatic uniqueness guarantees. Additional details about Shapley value explanations can be found in Lundberg and Lee [35], Sundararajan et al. [59], and Aas et al. [3].

4.1.2 Shapley Values in Practice. Organization A works with financial institutions and helps explain models for credit risk analysis. To integrate into the existing ML workflow of these institutions, Organization A proceeds as follows. They let data scientists train a model to the desired accuracy. Note that Organization A focuses mostly on models trained on tabular data, though they are beginning to venture into unstructured data (i.e., language and images). During model validation, risk analysts conduct stress tests before deploying the model to loan officers and other decision-makers. After decision-makers vet the model outputs as a sanity check and decide whether or not to override the model output, Organization A generates Shapley value explanations.

Before launching the model, risk analysts are asked to review the Shapley value explanations to ensure that the model exhibits expected behavior (i.e., the model uses the same features that a human would for the same task). Notably, the customer support team at these institutions can also use these explanations to tell individuals who inquire as to what went into the decision-making process for their loan approval or cash distribution decision. They are shown the percentage contribution to the model output (the positive ℓ_1 norm of the Shapley value explanation along with the sign of contribution). This means that the explanation would be along the lines of, "55% of the decision was decided by your age, which positively correlated with the predicted outcome."

When comparing Shapley value explanations to other popular feature importance techniques, Organization A found that in practice LIME explanations [48] give unexpected explanations that do not align with human intuition. Recent work [68] shows that the fragility of LIME explanations can be traced to the sampling variance when explaining a singular data point and to the explanation sensitivity to sample size and sampling proximity.

Though decision-makers have access to the feature-importance explanations, end users are still not shown these explanations as reasoning for model output. Organization A aspires to eventually expose this "explanation" to end users.

For gradient-based language models, Organization A uses Integrated Gradients, a path integral variant of Shapley Values [35, 59], to flag malicious reviews and moderate content at the aforementioned institutions. This information can be highlighted to ensure the trustworthiness and transparency of the model to the decision-maker (the hired content moderator), since they can now see which word was most important to flag the content as malicious.

Going forward, Organization A intends to use a global variant of the Shapley value explanations by exposing how Shapley value explanations work on average for datapoints of a particular predicted class (e.g., on average someone who was denied a loan had their age matter most for the prediction). This global explanation would help risk analysts get a birds-eye view of how a model behaves and whether it aligns with their expectations.

4.1.3 Heatmaps in Transportation. Organization B looks to detect facial expressions from video feeds of users driving. They hope to use explainability to identify the actions a user is performing while the user drives. Organization B has tried feature visualization and activation visualization techniques that get attributions by back-propagating gradients to regions of interest [5, 67]. Specifically, they use these probabilistic Winner-Take-All techniques (variants of existing gradient-based feature importance techniques [53, 59]) to localize the region of importance in the input space for a particular classification task. For example, when detecting a smile, they expect the mouth of the driver to be important.

Though none of these desired techniques have been deployed for the end user (the driver in this case), ML engineers at Organization B found these techniques useful for qualitative review. On tiny datasets, engineers can figure out which scenarios have false positives (videos falsely detected to contain smiles) and why. They can also identify if true positives are paying attention to the right place or if there is a problem with spurious artifacts.

However, while trying to understand why the model erred by analyzing similarities in false positives, they have struggled to scale this local technique across heatmaps in aggregate across multiple videos. They are able to qualitatively evaluate a sequence of heatmaps for one video, but doing so across 100M frames simultaneously is far more difficult. Paraphrasing the VP of AI at Organization B, aggregating saliency maps across videos is moot and contains little information. Note that an individual heatmap is an example of a local explainability technique, but an aggregate heatmap for 100M frames would be a global technique. Unlike aggregating Shapley values for tabular data as done at Organization A, taking an expectation over heatmaps (in the statistical sense) does not work, since aggregating pixel attributions is meaningless. One option Organization B discussed would be to clustering low dimensional representations of the heatmaps and then tagging each cluster based on what the model is focusing on; unfortunately, humans would still have to manually label the clusters of important regions.

4.1.4 Spurious Correlations. Related to model monitoring for feature drift detection discussed in Section 3.1, Organization B has encountered issues with spurious correlations in their smile detection models. Their Vice President of AI noted that "[ML engineers] must know to what extent you want ML to leverage highly correlated data to make classifications." Explainability can help identify models that focus on that correlation and can find ways to have models ignore it. For example, there may be a side effect of a correlated facial expression or co-occurrence: cheek raising, for example, co-occurs with smiling. In a cheek-raise detector trained on the same dataset as a smile detector but with different labels, the model still focused on the mouth instead of the cheeks. Both models were fixated on a prevalent co-occurrence. Attending to the mouth was undesirable in the cheek-raise detector but allowed in the smile detector.

One way Organization B combats this is by using simpler models on top of complex feature engineering. For example, they use black box deep learning models for building good descriptors that are robust across camera viewpoints and will detect different features that subject matter experts deem important for drowsiness. There is one model per important descriptor (i.e., one model for eyes closed, one for yawns, etc.). Then, they fit a simple model on the extracted descriptors such that the important descriptors are obvious for the final prediction of drowsiness. Ideally, if Organization B had guarantees about the disentanglement of data generating factors [4], they would be able to understand which factors (descriptors) play a role in downstream classification.

4.1.5 Feature Importance - Takeaways.

(1) Not only do Shapley values come with nice axiomatic guarantees, they are also simple to deploy for decision-makers to sanity check the models they have built.

(2) Feature importance is not used directly for end users, and instead explanations require looping in decision-makers who are acting based the original model outputs.
(3) Heatmaps are hard to aggregate over, which makes it hard to do false positive detection at scale.
(4) Spurious correlations can be detected with simple gradient-based techniques.

4.2 Counterfactual Explanations

Counterfactual explanations are techniques that explain individual predictions by providing a means for recourse. While some existing open source implementations for counterfactual explanations exist [63, 65], they either work for specific model-types or are not black-box in nature. In this section, we discuss the formulation for counterfactual explanations and describe one solution for each deployed technique.

4.2.1 Formulation. Counterfactual explanations are points close to the input for which the decision of the classifier changes. For example, for a person who was rejected for a loan by a ML model, a counterfactual explanation would possibly suggest: "Had your income been greater by $5000, the loan would have been granted."

Given an input $\boldsymbol{x}$ and a classifier $\boldsymbol{f}$, a counterfactual explanation $\boldsymbol{c}$ can be found by solving the optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{c}} d(\boldsymbol{x}, \boldsymbol{c}) \\ &\text{s.t.} \boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{f}(\boldsymbol{c}) \end{aligned} \tag{1}$$

While the term counterfactual has a well understood meaning in the causality literature [28, 45], counterfactual explanations for ML were introduced by Wachter et al. [64]. Sharma et al. [51] provide details on existing techniques.

4.2.2 Counterfactual Explanations in Healthcare. Organization C uses a faster version of the formulation in Sharma et al. [51] to find counterfactual explanations for projects in healthcare. When people apply for Medicare, Organization C hopes to flag if a user's application has errors and to provide explanations on how to correct the errors. Moreover, ML engineers can use the robustness score to compare different models trained using this data: this robustness score is effectively the distance between the counterfactual and original point in Euclidean space. The original formulation makes use of a slower genetic algorithm, so they optimized the counterfactual explanation generation process. They are currently developing a first-of-its-kind application that can directly take in any black-box model and data and return a robustness score, fairness measure, and counterfactual explanation, all from a single underlying algorithm.

The use of this approach has several advantages: it can be applied to black-box models, works for any input data type, and generates multiple explanations in a single run of the algorithm. However, there are some shortcomings that Organization C is trying to address. One challenge of counterfactual models is that the counterfactual might not be feasible. Organization C plans to address this by using the training data to guide the counterfactual generation process, ensuring that the counterfactuals are feasible given the training distribution. In addition, the flexibility of the counterfactual approach comes with a drawback that is common among explanations for black-box models: there is no guarantee of the optimality of the explanation since black-box techniques cannot guarantee optimality.

Through the creation of a deployed solution for this method, the organization realized that clients would ideally want an explainability score, along with the measure of fairness and robustness. They are currently developing an explainability score that seeks to measure how explainable different models are. However, since explanations are subjective, it is crucial to see how such a measure and the produced explanations are received by clients.

4.2.3 Counterfactual Explanations - Takeaways.

(1) Counterfactual explanation solutions yield client interest, since the underlying method is flexible and such explanations are easy for end users to understand.
(2) Since the method is heuristic, it is hard to say that the explanation produced is optimal. In general, counterfactual explanations are difficult to evaluate.

4.3 Adversarial Training

In order to ensure the predictor being deployed is robust to adversaries and behaves as intended, many organizations use adversarial training to ensure the predictor fits to the desired, robust, and human-interpretable features. Interestingly, this is a use case for explainability techniques that is not for enhancing transparency so much as protecting the integrity of the algorithmic decision-making process.

4.3.1 Formulation. Recent works have also explored the intersection between adversarial robustness and model interpretations [20, 22, 24, 55, 66]. In particular, adversarially trained models have been shown not only to be robust but also to provide sharper and clearer feature importance scores. The claim of one of these works is that the closest adversarial example should perturb the robust features (indicative of a particular class) and not fit to spurious non-robust features [31]. The robustness of a model to adversarial attacks depends on how well the feature importance (saliency) map aligns with the input. The setup of feature importance in Singla et al. [55] is as follows:

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{f}, \boldsymbol{x}) = \max_{\tilde{\boldsymbol{x}}} \quad & \mathcal{L}\left(f_{\theta^*}(\tilde{\boldsymbol{x}}), y\right) \\ & \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_0 \leq k \\ & \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \rho \end{aligned}$$

We let $|\tilde{\boldsymbol{x}} - \boldsymbol{x}|$ be the top-k feature importance scores of the input, $\boldsymbol{x}$. This is similar to the adversarial example setup which is usually written in the same manner as the above (without the ℓ_0 norm to limit the number of features that changed). It is also interesting to note that the formulation to find counterfactual explanations above matches the formulation for finding adversarial examples. Sharma et al. [51] use this connection to generate adversarial examples and define a black-box model robustness score.

4.3.2 Image Content Moderation. Organization D moderates user-generated content (UGC) on several public platforms. Specifically, the R&D team at Organization D developed several models to detect adult and violent content from users' uploaded images. Their quality assurance (QA) team measures model robustness to improve content detection accuracy under the threat of adversarial examples.

The robustness of a content moderation model is measured by the minimum perturbations required for an image to evade detection. Given a gradient-based image classification model $\boldsymbol{f} : \mathbb{R}^d \to \{1, \ldots, K\}$, and we assume $\boldsymbol{f}(\boldsymbol{x}) = \mathrm{argmax}_i\,(Z(\boldsymbol{x})_i)$ where $Z(\boldsymbol{x}) \in \mathbb{R}^K$ is the final (logit) layer output, and $Z(\boldsymbol{x})_i$ is the prediction score for the i-th class. The objective can be formulated as the following optimization problem to find the minimum perturbation:

$$\underset{\boldsymbol{x}}{\mathrm{argmin}}\,\{d\,(\boldsymbol{x}, \boldsymbol{x}_0) + c\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}), y)\} \tag{2}$$

$d(\cdot, \cdot)$ is some distance measure that Organization D chooses to be the ℓ_2 distance in Euclidean space; $\mathcal{L}(\cdot)$ is the cross-entropy loss function and c is a balancing factor.

As is common in the adversarial literature, Organization D applies Projected Gradient Descent (PGD) to search for the minimum perturbation from the set of allowable perturbations $\mathcal{S} \subseteq \mathbb{R}^d$ [37]. The search process can be formulated as

$$\boldsymbol{x}^{t+1} = \Pi_{\boldsymbol{x}+\mathcal{S}}\left(\boldsymbol{x}^t + \alpha\,\mathrm{sgn}\,(\nabla_{\boldsymbol{x}}\mathcal{L}\,(f_{\theta^*}(\boldsymbol{x}), y))\right)$$

until x^t is misclassified by the detection model. ML engineers on the QA team are shown a ℓ_2-norm perturbation distance averaged over n test images randomly sampled from the test dataset. The larger the average perturbation, the more robust the model is, as it takes greater effort for an attacker to evade detection. The average perturbation required is also widely used as a metric when comparing different candidate models and different versions of a given model.

Organization D finds that more robust models have more convincing gradient-based explanations, i.e., the gradient of the output with respect to the input shows that the model is focusing on relevant portions of the images, confirming recent research [22, 31, 62].

4.3.3 Text Content Moderation. Organization E uses text content moderation algorithms on its UGC platforms, such as forums. Its QA team is responsible for the reliability and robustness of a sentiment analysis model, which labels posts as positive or negative, trained on UGC. The QA team seeks to find the minimum perturbation required to change the classification of a post. In particular, they want to know how to take misclassified posts (e.g., negative ones classified as positive) and change them to the correct class.

Given a sentiment analysis model $\boldsymbol{f} : \mathcal{X} \to \mathcal{Y}$, which maps from feature space $\mathcal{X}$ to a set of class $\mathcal{Y}$, an adversary aims to generate an adversarial post $\boldsymbol{x}_{adv}$ from the original post $\boldsymbol{x} \in \mathcal{X}$ whose ground truth label is $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{y} \in \mathcal{Y}$ so that $\boldsymbol{f}\left(\boldsymbol{x}_{\boldsymbol{adv}}\right) \neq y$. The QA team tries to minimize $d(\boldsymbol{x}, \boldsymbol{x}_{adv})$ for a domain-specific distance function. Organization E uses the ℓ_2 distance in the embedding space, but it is equally valid to use the editing distance [42]. Note that perturbation technique changes accordingly.

In practice, to find the minimum distance in the embedding space, Organization E chooses to iteratively modify the words in the original post, starting from the words with the highest importance. Here importance is defined as the gradient of the model output with respect to a particular word. ML engineers compute the Jacobian matrix of the given posts $\boldsymbol{x} = (x_1, x_2, \cdots, x_N)$ where x_i is the i-th word. The Jacobian matrix is as follows:

$$J_{\boldsymbol{f}}(\boldsymbol{x}) = \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \left[\frac{\partial \boldsymbol{f}_j(\boldsymbol{x})}{\partial x_i}\right]_{i \in 1 \ldots N, j \in 1 \ldots K} \tag{3}$$

where K represents the number of classes (in this case $K = 2$), and $\boldsymbol{f}_j(\cdot)$ represents the confidence value of the jth class. The importance of word x_i is defined as

$$C_{x_i} = J_{\boldsymbol{f}, i, y} = \frac{\partial \boldsymbol{f}_y(\boldsymbol{x})}{\partial x_i} \tag{4}$$

i.e., the partial derivative of the confidence value based on the predicted class y regarding to the input word x_i. This procedure ranks the words by their impact on the sentiment analysis results. The QA team then applies a set of transformations/perturbations to the most important words to find the minimum number of important words that must be perturbed in order to flip an sentiment analysis API result.

4.3.4 Adversarial Training - Takeaways.

1. There is a relation between model robustness and explainability. Model robustness improves the quality of feature importances (specifically saliency maps), confirming recent research findings [22].
2. Feature importance helps find minimal adversarial perturbations for language models in practice.

4.4 Influential Samples

This technique asks the question: Which data point in the training dataset $\boldsymbol{x} \in \mathcal{D}_X$ is most influential to the predictor's output $\boldsymbol{f}(\boldsymbol{x}_{\text{test}})$ for a test point $\boldsymbol{x}_{\text{test}}$? Statisticians have used measures like Cook's distance [17] which measure the effect of deleting a data point on the model output. However, such measures require an exhaustive search and hence do not scale well for larger datasets.

4.4.1 Formulation. For over half of the organizations, influence functions has been the tool of choice for explaining which training points are influential to the predictor's output for a point $\boldsymbol{x}$ [34] (though only one organization actually deployed the technique). We let $\mathcal{L}(\boldsymbol{f}_\theta, \boldsymbol{x})$ be the predictor's loss for point $\boldsymbol{x}$, so the empirical risk minimizer is given by $\hat{\boldsymbol{f}}_\theta = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\boldsymbol{f}_\theta, \boldsymbol{x}^{(i)})$. Note that $y_{\boldsymbol{x}} = \hat{\boldsymbol{f}}_\theta(\boldsymbol{x})$ is the predicted output at $\boldsymbol{x}$ with the trained risk minimizer. [34] defines the most influential data point $\boldsymbol{z}$ to a fixed point $\boldsymbol{x}$ as that which maximizes the following:

$$\mathcal{I}_{\text{up,loss}}\,(\boldsymbol{z}, \boldsymbol{x}) = -\nabla_\theta \mathcal{L}\left(\hat{\boldsymbol{f}}_\theta(\boldsymbol{x}), y_{\boldsymbol{x}}\right)^\top H^{-1}_{\hat{\boldsymbol{f}}_\theta} \nabla_\theta \mathcal{L}\left(\hat{\boldsymbol{f}}_\theta(\boldsymbol{z}), y_{\boldsymbol{x}}\right)$$

This quantity measures the effect of upweighting on datapoint $(\boldsymbol{z})$ on the loss at $\boldsymbol{x}$. The goal of sample importance is to uncover which training examples, when perturbed, would have the largest effect (positive or negative) on the loss of a test point.

4.4.2 Influence Functions in Insurance. Organization F uses influence functions to explain risk models in the insurance industry. They hope to identify which customers might see an increase in their premiums based on their driving history in the past. The organization hopes to divulge to the end user how the premiums for drivers similar to them are priced. In other words, they hope to identify the influential training data points [34] to understand which past drivers had the greatest influence on the prediction for the observed driver. Unfortunately, Organization F has struggled to expose this information to end users since the Hessian computation has made doing so impractical since the latency is high.

More pressingly, even when Organization F lets the influence function procedure run, they find that many influential data points are simply outliers that are important for all drivers since those anomalous drivers are far out of distribution. As a result, instead of identifying which drivers are most similar to a given driver, the influential sample explanation identifies drivers that are very different from any driver (i.e., outliers). While this is could in theory be useful for outlier detection, it prevents the explanations from being used at deployment.

4.4.3 *Influential Samples - Takeaways.*

1. Influence functions can be intractable for large datasets; as such, a significant effort is needed to improve these methods to make them easy to deploy in practice.
2. Influence functions can be sensitive to outliers in the data, such that they might be more useful for outlier detection than for providing end users explanations.

5 RECOMMENDATIONS

This section provides recommendations for future work, based on the technique-specific takeaways in Section 4 and the key takeaways in Section 3.5. In order to address the challenges organizations face when striving to provide explanations to end users, we recommend a framework for establishing clear desiderata in explainability, including how to approach normative concerns.

5.1 Establish Clear Desiderata

Most organizations we spoke to solely deploy explainability techniques for internal engineers and scientists, as a debugging mechanism or as a sanity check. At the same time, these organizations also affirmed the importance of understanding the stakeholder, and hope to be able to explain a model prediction to the end user. Once the target population of the explanation is understood, organizations can devise and deploy explainability techniques accordingly. We propose the following 3 steps for establishing clear desiderata and improving decision making around explainability. These include: clearly identifying the target population, understanding their needs, and clarifying the intention of the explanation.

1. **Identify your target population (aka your stakeholder).** That is, who is your desired explanation consumer? Ideally this person is also affected by or is shown output based on the model.
2. **Engage with the stakeholder.** Ask them some variant of “What would you need the model to explain to you in order to understand, trust, or contest the model prediction?” and “How would an explanation help you?”
   - *If the explanation would not be helpful*: Better understand the use case of the model and how it is being deployed.
   - *If the explanation would be helpful*: Follow up with understanding how the explanation will better inform the target population.
3. **Understand the intention of the explanation.** Once the context of the explanation and the helpfulness of the explanation are established, examine what will be done with the explanation.
   - *Static Consumption*: Will the explanation be used as a sanity check for a data scientist or shown to an end user as reasoning for a particular prediction?
   - *Dynamic Model Updates*: Will the explanation be used to garner feedback from the end user as to how the model ought to be updated to better align with their intuition? That is, how does the user interact with the model after viewing the explanation?

Once the desiderata are clarified, domain experts can be shown the explanations to ensure that they exhibit expected behavior. Having clearer desiderata is vital since the current literature lacks a clear direction for why explanations are desired and how explanations would be helpful in practice.

5.2 Important Normative Desiderata

In this subsection, we discuss a few normative desiderata that companies should consider when deploying explainability techniques. These desiderata (with the exception of causality) were not explicitly mentioned in our interviews, and have been consciously included here in order highlight important AI ethics concerns.

5.2.1 *Fairness Guided By Explainability.* As organizations consider deploying explainability techniques, it is important to reflect on fairness as a key desideratum. Explanations can help expose fairness violations by providing insights into possible biases in a model. For example, work on counterfactual explanations [51] and [63] has demonstrated how explanations can be used to examine predictor fairness. We now define how bias can potentially be detected using two explainability approaches.

Approach 1: Given a binary predictor $\boldsymbol{f}$, an input $\boldsymbol{x}$, and a feature importance explanation function $\boldsymbol{g} : \boldsymbol{f} \times \mathbb{R}^d \mapsto \mathbb{R}^d$ that returns importance scores $\boldsymbol{g}(\boldsymbol{f}, \boldsymbol{x}) = \boldsymbol{\phi_x} \in \mathbb{R}^d$ for all features, where $\boldsymbol{g}(\boldsymbol{f}, \boldsymbol{x})_i = \phi_{\boldsymbol{x},i}$ (simplified to ϕ_i in context) is the importance of (or attribution for) feature x_i of $\boldsymbol{x}$. Let A be a protected attribute. Then, if $\phi_{\boldsymbol{x},A} > \epsilon$ the predictor indicates potential discrimination at a local level, where ϵ is a fairness-sensitive parameter decided by the ML engineer.

Approach 1 implies that if a protected attribute, such as race or gender, has an importance beyond a certain level for determining the prediction for an individual, this is indicative of potential bias, and the ML engineer should take steps to examine such decision-making. This approach can be extended to a global level by taking an expectation of the importance of the protected attribute over the different groups to find the importance of the protected attribute for every group. Note that the applicability of this approach for fairness is based on the assumption that features are independent, and other methods should be considered if this is not the case.

In addition, it is important to note that this is not meant to be a definition of "bias" itself but a potential indicator for bias. Given that much of the bias literature in ML relies on the use of protected attributes to mitigate or correct for bias [13], the fact that a protected class variable is considered important for a prediction does not necessarily imply the model exhibits algorithmic bias. Instead, the goal of this approach is to provide a sanity check. If a protected class variable plays a key role in the algorithm's decision, the ML engineer should ensure that the use of this variable is appropriate.

Approach 2: Consider a set of points Z that represent the most influential samples responsible for the prediction of input $\boldsymbol{x}$. Let A be a protected attribute. If a higher percentage of points in Z have the same value for A as the input $\boldsymbol{x}$, this is potentially indicative of prejudice towards or against the group having that value of A.

Approach 2 implies that a model is might be biased when individuals having the same protected attribute value are most responsible towards a new individual that belongs to the same protected attribute group. This might be indicative of bias arising due to a group being treated similarly (positive or negative) historically (which is reflected in the data and hence the model), or a lack of sample diversity for particular groups. This might also be reflective of an reliance of the protected attribute towards a model's decision, which, as discussed above, can be undesirable if not done to specifically correct for bias. Again, this is not meant to be a definition of bias, but rather a sanity check for ML engineers to check for bias.

The two approaches above intuitively follow from the respective explanation regimes (i.e., feature importance and influence functions), and we suggest how both approaches generate explanations that might be useful for detecting biases.

5.2.2 *Explainability and Privacy.*

Another important desideratum for organizations to consider when deploying explainability techniques is privacy, since model explanations can be used to reconstruct the model [40] or to do transparency-based membership inference [52]. Tramèr et al. [61] shows how models can be replicated with access to only the model APIs. Providing black-box explanations could potentially mitigate this issue. Preece [47] and Sokol and Flach [57] discuss how explainability could compromise on privacy. We describe an example case and then provide possible suggestions to practitioners on how to address these issues.

Consider the use of counterfactual explanations to explain a prediction to an end user. An end user could not only use such tools to change their data maliciously to fool a model, but by querying the model multiple times using a random set of inputs, a user can possibly learn the approximate decision boundary that is being used for classification. Moreover, methods such as the what-if tool for producing counterfactual explanations pose a risk to data privacy since they provide points that actually belong to the training dataset [65]. Providing the most relevant data points using sample importance unintentionally provides data that may belong to prior users, which infringes privacy. Where those risks are present, industry practitioners may need to develop methods to avoid the harmful use of explainability tools. Below is one possible way to approach such issues, applying a framework analogous to differential privacy.

Approach: Given a model $\boldsymbol{f}$ and an input to the model $\boldsymbol{x}$, consider an explanation represented by $\boldsymbol{g}(\boldsymbol{f}, \boldsymbol{x})$, where $\boldsymbol{g}(\boldsymbol{f}, \boldsymbol{x})$ could be in the form of a counterfactual explanation point or a sample important to the inputs prediction. Then, the private explanation $\boldsymbol{e}$ should be such that $\boldsymbol{e} = \boldsymbol{g}(\boldsymbol{f}, \boldsymbol{x}) + \epsilon$ where $\epsilon \sim \text{Laplace}(0, b)$ is a zero-centered Laplace noise with scale b.

The approach above implies that publicly released training data points or counterfactual explanations need to have noise added to them so that data rights or model privacy are not compromised. Global explainability methods need to investigate ways to provide explanations about the model without providing details on model weights (directly or via global level feature importance scores) [40].

5.2.3 *Explainability and Causality.*

One chief scientist told us that "Figuring out causal factors is the holy grail of explainability." However, causal explanations are largely lacking in the literature, other than preliminary work on causal attribution for deep learning models [15]. Though non-causal explanations can still provide valid and useful interpretations of how the model works [39], many organizations posit that the lack of use of local explanation techniques for end users stem from a lack of causal interpretations.

The relationship between causality and explainability in ML relates to a broader dichotomy. In distinguishing ML techniques from those in econometrics or statistics, a key component is prediction vs. inference [10, 11, 33]. ML focuses on prediction, such that understanding the underlying process is generally considered less important than predictive accuracy. In fact, the power of ML tools is often presented as a trade-off between accuracy and explainability, with simpler more interpretable models often performing worse on accuracy measures [49]. That said, there has been a growing demand from end users, civil society groups, and law makers for ML engineers not only to justify the quality of their predictions through accuracy metrics but also to provide explanations for how the predictions were arrived at. Explanations, however, are inherently inferential, so these trends imply a growing need for an inferential approach to ML. While econometrics provides many tools for causal inference on linear models [11], more work needs to be done to connect statistical inference techniques to the context of more complex ML models.

5.2.4 *Unintended Consequences from Explainability Research.*

As ML is increasingly being deployed in high-stakes situations, including in finance, criminal justice, and content moderation, the ethical implications of how explainability techniques are used are an important concern. In order for explainability techniques to facilitate greater accountability for end users, ethical desiderata must include broader societal considerations for how the ML is being used.

For example, several organizations we talked with have begun to make extensive use of natural language processing and image recognition models for content moderation in response to business incentives, regulatory requirements, and sociopolitical pressure. In some instances, explainability techniques have become part of the workflows and development of those content moderation processes and are making them more effective.

Though there are aspects of these use cases that are clearly in users' interest, there are others where that is much less clear, with potential for adverse effects both when these systems work correctly and when they err [25, 38, 60]. ML researchers may not always be in a position to set the objectives and criteria for how their technology is applied, which makes it difficult to propose best practices for ethical approaches to such work. The ML research community should continue to be mindful about the potential for both constructive and unintended consequences of its work in this and other sensitive domains.

6 CONCLUSION

In this study, we critically examine how explanation techniques are used in practice and illuminate the gaps between current techniques and normative desiderata. We are the first, to our knowledge, to interview various organizations on how they deploy explainability in their ML workflows, concluding with salient directions for future research. We found that while ML engineers are increasingly using explainability techniques as sanity checks during the development process, there are still significant limitations to current techniques that prevent their use to directly inform end users. These limitations include the need for domain experts to evaluate explanations, the risk of spurious correlations reflected in model explanations, the lack of causal intuition, and the latency in computing and showing explanations in real-time. Future research should seek to address these limitations.

We also highlighted the need for organizations to establish clear desiderata for their explanation techniques and to incorporate ethics-related desiderata, taking into account issues such as fairness, privacy, causality, and the potential for adverse unintended consequences. Through this analysis, we take a step towards explaining explainability deployment and hope that future research builds trustworthy explainability solutions.

7 ACKNOWLEDGMENTS

The authors would like to thank the following individuals for their advice, contributions, and/or support: Karina Alexanyan (Partnership on AI), Gagan Bansal (University of Washington), Rich Caruana (Microsoft), Amit Dhurandhar (IBM), Krishna Gade (Fiddler Labs), Jette Henderson (CognitiveScale), Yannis Kalantidis (Facebook), Bahador Kaleghi (Element AI), Been Kim (Google), Hima Lakkaraju (Harvard University), Katherine Lewis (Partnership on AI), Peter Lo (Partnership on AI), Terah Lyons (Partnership on AI), Saayeli Mukherji (Partnership on AI), Erik Pazos (QuantumBlack), Inioluwa Deborah Raji (Partnership on AI), Francesca Rossi (IBM), Jay Turcot (Affectiva), Kush Varshney (IBM), Dennis Wei (IBM), Edward Zhong (Baidu), Gabi Zijderveld (Affectiva), and around ten other anonymous individuals.

REFERENCES

[1] 2019. IBM'S Principles for Data Trust and Transparency. https://www.ibm.com/blogs/policy/trust-principles/

[2] 2019. Our approach: Microsoft AI principles. https://www.microsoft.com/en-us/ai/our-approach-to-ai

[3] Kjersti Aas, Martin Jullum, and Anders Løland. 2019. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *arXiv preprint arXiv:1903.10464* (2019).

[4] Tameem Adel, Zoubin Ghahramani, and Adrian Weller. 2018. Discovering interpretable representations for both deep generative and discriminative models. In *International Conference on Machine Learning*. 50–59.

[5] Sarah Adel Bargal, Andrea Zunino, Donghyun Kim, Jianming Zhang, Vittorio Murino, and Stan Sclaroff. 2018. Excitation backprop for RNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1440–1449.

[6] Oscar Alvarado and Annika Waern. 2018. Towards algorithmic experience: Initial efforts for social media contexts. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 286.

[7] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565* (2016).

[8] Marco Ancona, Enea Ceolini, Cengiz Oztireli, and Markus Gross. 2018. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *6th International Conference on Learning Representations (ICLR 2018)*.

[9] Marco Ancona, Cengiz Oztireli, and Markus Gross. 2019. Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Value Approximation. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.), Vol. 97. PMLR, Long Beach, California, USA, 272–281.

[10] Susan Athey. 2017. Beyond prediction: Using big data for policy problems. 355, 6324 (2017), 483–485.

[11] Susan Athey and Guido W Imbens. 2017. The state of applied econometrics: Causality and policy evaluation. *Journal of Economic Perspectives* 31, 2 (2017), 3–32.

[12] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research* 11, Jun (2010), 1803–1831.

[13] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2018. *Fairness and Machine Learning*. fairmlbook.org. http://www.fairmlbook.org.

[14] Rajiv Khanna Been Kim and Sanmi Koyejo. 2016. Examples are not Enough, Learn to Criticize! Criticism for Interpretability. In *Advances in Neural Information Processing Systems*.

[15] Aditya Chattopadhyay, Piyushi Manupriya, Anirban Sarkar, and Vineeth N Balasubramanian. 2019. Neural Network Attributions: A Causal Perspective. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.), Vol. 97. PMLR, Long Beach, California, USA, 981–990.

[16] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. [n. d.]. L-shapley and c-shapley: Efficient model interpretation for structured data. *7th International Conference on Learning Representations (ICLR 2019)* ([n. d.]).

[17] R Dennis Cook. 1977. Detection of influential observation in linear regression. *Technometrics* 19, 1 (1977), 15–18.

[18] Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan OâĂŹDonoghue, Daniel Visentin, et al. 2018. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine* 24, 9 (2018), 1342.

[19] Amit Dhurandhar, Karthikeyan Shanmugam, Ronny Luss, and Peder A Olsen. 2018. Improving simple models with confidence profiles. In *Advances in Neural Information Processing Systems*. 10296–10306.

[20] Ann-Kathrin Dombrowski, Maximilian Alber, Christopher J Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. 2019. Explanations can be manipulated and geometry is to blame. *arXiv preprint arXiv:1906.07983* (2019).

[21] William DuMouchel. 2002. Data squashing: constructing summary data sets. In *Handbook of Massive Data Sets*. Springer, 579–591.

[22] Christian Etmann, Sebastian Lunz, Peter Maass, and Carola Schoenlieb. 2019. On the Connection Between Adversarial Robustness and Saliency Map Interpretability. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.), Vol. 97. PMLR, Long Beach, California, USA, 1823–1832.

[23] Ruth Fong and Andrea Vedaldi. 2017. Interpretable Explanations of Black Boxes by Meaningful Perturbation. Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV). (2017). https://doi.org/10.1109/ICCV.2017.371 arXiv:arXiv:1704.03296

[24] Amirata Ghorbani, Abubakar Abid, and James Zou. 2019. Interpretation of neural networks is fragile. *AAAI* (2019).

[25] Tarleton Gillespie. 2018. *Custodians of the Internet: Platforms, content moderation, and the hidden decisions that shape social media.* Yale University Press.

[26] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE, 80–89.

[27] JB Heaton, Nicholas G Polson, and Jan Hendrik Witte. 2016. Deep learning in finance. *arXiv preprint arXiv:1602.06561* (2016).

[28] Paul W. Holland. 1986. Statistics and Causal Inference. *J. Amer. Statist. Assoc.* 81, 396 (1986), 945–960.

[29] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. 2019. Improving fairness in machine learning systems: What do industry practitioners need?. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 600.

[30] Giles Hooker and Lucas Mentch. 2019. Please Stop Permuting Features: An Explanation and Alternatives. *arXiv preprint arXiv:1905.03151* (2019).

[31] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial Examples Are Not Bugs, They Are Features. http://arxiv.org/abs/1905.02175 cite arxiv:1905.02175.

[32] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. 2017. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279* (2017).

[33] Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ziad Obermeyer. 2015. Prediction Policy Problems. *American Economic Review* 105, 5 (May 2015), 491–95.

[34] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70 (ICML 2017)*. Journal of Machine Learning Research, 1885–1894.

[35] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4765–4774.

[36] Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. 2018. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering* 2, 10 (2018), 749.

[37] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[38] Alice E Marwick, Lindsay Blackwell, and Katherine Lo. 2016. Best practices for conducting risky research and protecting yourself from online harassment (Data & Society Guide). (2016).

[39] Tim Miller. 2018. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* (2018).

[40] Smitha Milli, Ludwig Schmidt, Anca Dragan, and Moritz Hardt. 2019. Model Reconstruction from Model Explanations. *In Proceedings of ACM FAT* 2019* (2019).

[41] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. 2017. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition* 65 (2017), 211–222.

[42] Yilin Niu, Chao Qiao, Hang Li, and Minlie Huang. 2018. Word Embedding based Edit Distance. *arXiv preprint arXiv:1810.10752* (2018).

[43] Board of Governors of the Federal Reserve System. 2011. Supervisory Guidance on Model Risk Management. *https://www.federalreserve.gov/supervisionreg/srletters/sr1107a1.pdf* (2011).

[44] European Parliament and Council of European Union. 2018. European Union General Data Protection Regulation, Articles 13-15. *http://www.privacy-regulation.eu/en/13.htm* (2018).

[45] Judea Pearl. 2000. *Causality: models, reasoning and inference*. Vol. 29. Springer.

[46] Fábio Pinto, Marco OP Sampaio, and Pedro Bizarro. 2019. Automatic Model Monitoring for Data Streams. *arXiv preprint arXiv:1908.04240* (2019).

[47] Alun Preece. 2018. Asking âĂŸWhyâĂŹin AI: Explainability of intelligent systems–perspectives and challenges. *Intelligent Systems in Accounting, Finance and Management* 25, 2 (2018), 63–72.

[48] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. 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*. ACM, 1135–1144.

[49] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206.

[50] Lloyd S Shapley. 1953. A Value for n-Person Games. In *Contributions to the Theory of Games II*. 307–317.

[51] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. 2019. CERTIFAI: Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models. *arXiv preprint arXiv:1905.07857* (2019).

[52] Reza Shokri, Martin Strobel, and Yair Zick. 2019. Privacy Risks of Explaining Machine Learning Models. *arXiv preprint arXiv:1907.00164* (2019).

[53] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70 (ICML 2017)*. Journal of Machine Learning Research, 3145–3153.

[54] Avanti Shrikumar, Eva Prakash, and Anshul Kundaje. 2018. Gkmexplain: Fast and Accurate Interpretation of Nonlinear Gapped k-mer Support Vector Machines Using Integrated Gradients. *BioRxiv* (2018), 457606.

[55] Sahil Singla, Eric Wallace, Shi Feng, and Soheil Feizi. 2019. Understanding Impacts of High-Order Loss Approximations and Features in Deep Learning Interpretation. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.), Vol. 97. PMLR, Long Beach, California, USA, 5848–5856.

[56] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825* (2017).

[57] Kacper Sokol and Peter A Flach. 2019. Counterfactual Explanations of Machine Learning Predictions: Opportunities and Challenges for AI Safety.. In *SafeAI@ AAAI*.

[58] Erik Štrumbelj and Igor Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems* 41, 3 (2014), 647–665.

[59] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70 (ICML 2017)*. Journal of Machine Learning Research, 3319–3328.

[60] Nicolas Suzor, Sarah West, Andrew Quodling, and Jillian York. 2019. What Do We Mean When We Talk About Transparency? Toward Meaningful Transparency in Commercial Content Moderation. *International Journal of Communication* 13, 0 (2019).

[61] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*. 601–618.

[62] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness May Be at Odds with Accuracy. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SyxAb30cY7

[63] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, 10–19.

[64] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GPDR. *Harv. JL & Tech.* 31 (2017), 841.

[65] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viegas, and Jimbo Wilson. 2019. The What-If Tool: Interactive Probing of Machine Learning Models. *arXiv preprint arXiv:1907.04135* (2019).

[66] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David Inouye, and Pradeep Ravikumar. 2019. How Sensitive are Sensitivity-Based Explanations? *arXiv preprint arXiv:1901.09392* (2019).

[67] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. 2018. Top-down neural attention by excitation backprop. *International Journal of Computer Vision* 126, 10 (2018), 1084–1102.

[68] Yujia Zhang, Kuangyan Song, Yiming Sun, Sarah Tan, and Madeleine Udell. 2019. "Why Should You Trust My Explanation?" Understanding Uncertainty in LIME Explanations. arXiv:arXiv:1904.12991