Explainable Machine Learning in Deployment

Umang Bhatt[1,2,3,4], Alice Xiang[2], Shubham Sharma[5], Adrian Weller [3,4,6], Ankur Taly[7], Yunhan Jia[8],
Joydeep Ghosh[5,9], Ruchir Puri[10], José M. F. Moura[1], Peter Eckersley[2]

[1]Carnegie Mellon University, [2]Partnership on AI, [3]University of Cambridge, [4]Leverhulme CFI, [5]University of Texas at Austin, [6]The Alan Turing Institute, [7]Fiddler Labs, [8]Baidu, [9]CognitiveScale, [10]IBM Research

umang@partnershiponai.org

ABSTRACT

Explainable machine learning offers the potential to provide stakeholders with insights into model behavior by using various methods such as feature importance scores, counterfactual explanations, or influential training data. Yet there is little understanding of how organizations use these methods in practice. This study explores how organizations view and use explainability for stakeholder consumption. We find that, currently, the majority of deployments are not for end users affected by the model but rather for machine learning engineers, who use explainability to debug the model itself. There is thus a gap between explainability in practice and the goal of transparency, since explanations primarily serve internal stakeholders rather than external ones. Our study synthesizes the limitations of 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. We end by discussing concerns raised regarding explainability.

KEYWORDS

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

1 INTRODUCTION

Machine learning (ML) models are being increasingly embedded into many aspects of daily life, such as healthcare [16], finance [26], and social media [5]. To build ML models worthy of human trust, researchers have 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 or developer of ML models understand why models behave the way they do. Explanations can come in many forms: from telling patients which symptoms were indicative of a particular diagnosis [35] to helping factory workers analyze inefficiencies in a production pipeline [17].

Explainability has been touted as a way to enhance transparency of ML models [33]. Transparency includes a wide variety of efforts to provide stakeholders, particularly end users, with relevant information about how a model works [67]. One form of this would be to publish an algorithmâĂŹs code, though this type of transparency would not provide an intelligible explanation to most users. Another form would be to disclose properties of the training procedure and datasets used [39]. Users, however, 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 aiming to increase trustworthiness [44]. The importance of explainability as a concept has been reflected in legal and ethical guidelines for data and ML [53]. 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" [45]. 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].

With growing interest in "peering under the hood" of ML models and in providing explanations to human users, explainability has become an important subfield of ML. Despite a burgeoning literature, there has been little work characterizing how explanations have been deployed by organizations in the real world.

In this paper, we explore how organizations have deployed local explainability techniques so that we can observe which techniques work best in practice, report on the shortcomings of existing techniques, and recommend paths for future research. We focus specifically on local explainability techniques since these techniques explain individual predictions, making them typically the most relevant form of model transparency for end users.

Our study synthesizes interviews with 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 recommendations to organizations deploying explainability. To the best of our knowledge, we are the first to conduct a study of how explainability techniques are used by organizations that deploy ML models in their workflows. Our main contributions are threefold:

- We interview around twenty data scientists, who are not currently using explainability tools, to understand their organization's needs for explainability.
- We interview around thirty different individuals on how their organizations have deployed explainability techniques, reporting case studies and takeaways for each technique.
- We suggest a framework for organizations to clarify their goals for deploying explainability.

The rest of this paper is organized as follows:

(1) We discuss the methodology of our survey 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 and discuss concerns of explainability in Section 5.2.

2 METHODOLOGY

In the spirit of Holstein et al. [28], 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 sets of interviews: Group 1 consisted 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, consisted 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 the technology and financial services sectors. We specifically focused on teams that do not currently employ explainability tools. These semi-structured, hour-long interviews included, but were not limited to, the following questions:

- What are your ML use cases?
- What is your current model development workflow?
- What are your pain points in deploying ML models?
- Would explainability help address those pain 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 for AI technologies to benefit 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 were professors at academic institutions, who commented on the consulting they had 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?

3 SUMMARY OF FINDINGS

Here we synthesize the results from both interview groups. For the sake of clarity, we define various terms based on the context in which they appear in the forthcoming prose.

- *Trustworthiness* refers to the extent to which stakeholders can reasonably trust a model's outputs.
- *Transparency* refers to attempts to provide stakeholders (particularly external stakeholders) with relevant information about how the model works: this includes documentation of the training procedure, analysis of training data distribution, code releases, feature-level explanations, etc.
- *Explainability* refers to attempts to provide insights into a model's behavior.
- *Stakeholders* are the people who either want a model to be "explainable," will consume the model explanation, or are affected by decisions made based on model output.
- *Practice* refers to the real-world context in which the model has been deployed.
- *Local Explainability* aims to explain the model's behavior for a specific input.
- *Global Explainability* attempts to understand the high-level concepts and reasoning used by a model.

3.1 Explainability Needs

This subsection provides an overview of explainability needs that were uncovered with Group 1, comprised of data scientists from organizations that do not currently deploy explainability techniques. These data 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 [6, 47]. 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 [this] 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 regulations. 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 [50] and SHAP [34] explanations via the faithfulness criterion [69]).

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 (e.g., 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 achieve an organization's 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 that 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 to affected populations or the regulators themselves. 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 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 and reconcile 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.,

Domain	Model Purpose	Explainability Technique	Stakeholders	Evaluation Criteria
Finance	Loan Repayment	Feature Importance	Loan Officers	Completeness [34]
Insurance	Risk Assessment	Feature Importance	Risk Analysts	Completeness [34]
Content Moderation	Malicious Reviews	Feature Importance	Content Moderators	Completeness [34]
Finance	Cash Distribution	Feature Importance	ML Engineers	Sensitivity [69]
Facial Recognition	Smile Detection	Feature Importance	ML Engineers	Faithfulness [7]
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 select deployed local explainability use cases

providing the most influential datapoints) hard to compute quickly. Moreover, finding plausible counterfactual datapoints (that are feasible in the real world and on the input data manifold) is nontrivial, and many existing techniques currently make crude approximations or return the closest datapoint of the other class in the training set. Moreover, providing certain explanations can raise privacy concerns due to 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 stakeholders. At best, gradient-based explanations provide an interpretation of how the model behaves upon an infinitesimal perturbation (not necessarily a feasible one [29]), but does not "explain" if the model captures the underlying causal mechanism from the data.

4 DEPLOYING LOCAL EXPLAINABILITY

In this section, we dive into how local explainability techniques are used at various organizations (Group 2). After reviewing technical notation, we define local explainability techniques, discuss organizations' use cases, and then report takeaways for each technique.

4.1 Preliminaries

A black box model f maps an input $x \in X \subseteq \mathbb{R}^d$ to an output $f(x) \in \mathcal{Y}$, $f : \mathbb{R}^d \mapsto \mathcal{Y}$. When we assume f has a parametric form, we write f_θ. $\mathcal{L}(f(x), y)$ denotes the loss function used to train f on a dataset $\mathcal{D}$ of input-output pairs $(x^{(i)}, y^{(i)})$.

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

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 [31], simpler models learned from the representation of complex models [17], prototypical samples from a particular model output [10],

or the topology of the data itself [20]. None of our interviewees reported deploying global explainability techniques, though some studied these techniques in research settings.

4.2 Feature Importance

Feature importance was by far the most popular technique we found across our study. It is used across many different domains (finance, healthcare, facial recognition, and 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 [9, 24].

4.2.1 Formulation. Feature importance methods define an explanation function $g : f \times \mathbb{R}^d \mapsto \mathbb{R}^d$ that takes in a model f and a point of interest x and returns importance scores $g(f, x) \in \mathbb{R}^d$ for all features. is the importance of (or attribution for) feature x_i of x.

These explanation functions roughly fall into two categories: perturbation-based techniques [8, 14, 22, 34, 50, 61] and gradient-based techniques [7, 41, 57, 59, 60, 62]. 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 the importance of a region or collection of features [4, 22]. A prominent class of perturbation based methods is based on Shapley values from cooperative game theory [54]. Shapley values are a 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. Unfortunately, calculating the exact Shapley value is exponential in d, input dimensionality; however, the literature has proposed approximate methods using weighted linear regression [34], Monte Carlo approximation [61], centroid aggregation [11], and graph-structured factorization [14]. When we refer to Shapley-related methods hereafter, we mean such approximate methods.

4.2.2 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 provide individuals information about 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 [50] give unexpected explanations that do not align with human intuition. Recent work [71] 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 provide this "explanation" to end users.

For gradient-based language models, Organization A uses Integrated Gradients (related to Shapley Values by Sundararajan et al. [62]) 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 here), since they can now see which word was most important to flag the content.

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.

4.2.3   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 [4, 70]. Specifically, they use these probabilistic Winner-Take-All techniques (variants of existing gradient-based feature importance techniques [57, 62]) 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.

4.2.4   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 [3], they would be able to understand which factors (descriptors) play a role in downstream classification.

4.2.5   Feature Importance - Takeaways.

(1) Shapley values are rigorously motivated, and approximate methods are simple to deploy for decision makers to sanity check the models they have built.

(2) Feature importance is not shown to end users, but is used by machine learning engineers as a sanity check. Looping other stakeholders (who make decisions based on model outputs) into the model development process is essential to understanding what type of explanations the model delivers.

(3) Heatmaps (and feature importance scores, in general) are hard to aggregate, which makes it hard to do false positive detection at scale.

(4) Spurious correlations can be detected with simple gradient-based techniques.

4.3 Counterfactual Explanations

Counterfactual explanations are techniques that explain individual predictions by providing a means for recourse. Contrastive explanations that highlight contextually relevant information to the model output are most similar to human explanations[40]; however, in their current form, finding the relevant set of plausible counterfactual point is no clear. Moreover, while some existing open source implementations for counterfactual explanations exist [65, 68], 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.

4.3.1 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 x, a classifier f, and a distance metric d, we find a counterfactual explanation c by solving the optimization problem:

$$\min_{c} d(x, c)$$
$$\text{s.t.} f(x) \neq f(c) \tag{1}$$

The method can be tailored to allow only certain relevant features to be changed. Note that the term counterfactual has a different meaning in the causality literature [27, 46]. Counterfactual explanations for ML were introduced by Wachter et al. [66]. Sharma et al. [55] provide details on existing techniques.

4.3.2 Counterfactual Explanations in Healthcare. Organization C uses a faster version of the formulation in Sharma et al. [55] 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 a suitably normalized and averaged 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 explanations, 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 addressing. One challenge of counterfactual models is that the counterfactual might not be feasible. Organization C addresses this by using the training data to guide the counterfactual generation process and by providing a user interface that allows domain experts to specify constraints. 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 a measure of fairness and robustness; as such, they have developed an explainability score that can be used to compare the explainability of different models.

4.3.3 Counterfactual Explanations - Takeaways.

(1) Organizations are interested in counterfactual explanation solutions since the underlying method is flexible and such explanations are easy for end users to understand.

(2) It is not clear exactly what should be optimized for when generating a counterfactual or how to do it efficiently. Still, approximate solutions may suffice in practical applications.

4.4 Adversarial Training

In order to ensure the model being deployed is robust to adversaries and behaves as intended, many organizations we interviewed use adversarial training to improve performance. It has recently been shown that in fact, this also can lead to more human interpretable features [30].

4.4.1 Formulation. Other works have also explored the intersection between adversarial robustness and model interpretations [18, 21, 23, 59, 69]. The claim of one of these works is that the closest adversarial example should perturb 'fragile' features, enabling the model to fit to robust features (indicative of a particular class) [30]. The setup of feature importance in the adversarial training setting from Singla et al. [59] is as follows:

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

We let $|\tilde{x} - x|$ be the top-k feature importance scores of the input, 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. [55] use this connection to generate adversarial examples and define a black-box model robustness score.

4.4.2 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 $f : \mathbb{R}^d \rightarrow \{1, \ldots, K\}$, and we assume $f(x) = \mathrm{argmax}_i (Z(x)_i)$ where $Z(x) \in \mathbb{R}^K$ is the final (logit) layer output, and $Z(x)_i$ is the prediction score for the i-th class. The objective can be formulated as the following optimization problem to find the minimum perturbation:

$$\underset{x}{\mathrm{argmin}} \{d(x, x_0) + c\mathcal{L}(f(x), y)\} \qquad (2)$$

$d(\cdot, \cdot)$ 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 $\mathcal{S} \subseteq \mathbb{R}^d$ [36]. The search process can be formulated as

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

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 [21, 30, 64].

4.4.3 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 $f : \mathcal{X} \rightarrow \mathcal{Y}$, which maps from feature space $\mathcal{X}$ to a set of class $\mathcal{Y}$, an adversary aims to generate an adversarial post x_{adv} from the original post $x \in \mathcal{X}$ whose ground truth label is $f(x) = y \in \mathcal{Y}$ so that $f(x_{adv}) \neq y$. The QA team tries to minimize $d(x, x_{adv})$ 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 $x = (x_1, x_2, \cdots, x_N)$ where x_i is the i-th word. The Jacobian matrix is as follows:

$$J_f(x) = \frac{\partial f(x)}{\partial x} = \left[\frac{\partial f_j(x)}{\partial x_i} \right]_{i \in 1 \ldots N, j \in 1 \ldots K} \qquad (3)$$

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

$$C_{x_i} = J_{f, i, y} = \frac{\partial f_y(x)}{\partial x_i} \qquad (4)$$

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.

4.4.4 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 [21].
  (2) Feature importance helps find minimal adversarial perturbations for language models in practice.

4.5 Influential Samples

This technique asks the question: Which data point in the training dataset $x \in \mathcal{D}_x$ is most influential to the model's output $f(x_{\text{test}})$ for a test point x_{test}? Statisticians have used measures like Cook's distance [15] 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.

4.5.1 Formulation. For over half of the organizations, influence functions has been the tool of choice for explaining which training points are influential to the model's output for a point x [32] (though only one organization actually deployed the technique). We let $\mathcal{L}(f_\theta, x)$ be the model's loss for point x. The empirical risk minimizer is given by $\hat{f}_\theta = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f_\theta, y_{x^{(i)}})$. Note that $y_x = \hat{f}_\theta(x)$ is the predicted output at x with the trained risk minimizer. Koh and Liang [32] define the most influential data point z to a fixed point x as that which maximizes the following:

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

This quantity measures the effect of upweighting on datapoint (z) on the loss at 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.

4.5.2 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 [32] to understand which past drivers had the greatest influence on the prediction for the observed driver. Unfortunately, Organization F has struggled to provide 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 in practice.

4.5.3   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 organizations based on the key takeaways in Section 3.4 and the technique-specific takeaways in Section 4. 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 and then include concerns associated with explainability.

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 attempt to devise and deploy explainability techniques accordingly. We propose the following three 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 stakeholders.** Who are your desired explanation consumers? Typically this will be those affected by or shown model outputs. Preece et al. [49] describe how stakeholders have different needs for explainability. Distinctions between these groups can help design better explanation techniques.
(2) **Engage with each stakeholder.** Ask the stakeholder some variant of "What would you need the model to explain to you in order to understand, trust, or contest the model prediction?" and "What type of explanation do you want from your model?" Doshi-Velez and Kim [19] highlight how the

task being modeled dictates what type of explanation the human will need from the model.

(3) **Understand the purpose of the explanation.** Once the context and helpfulness of the explanation are established, understand what the stakeholder wants to do with the explanation [24].
- *Static Consumption*: Will the explanation be used as a one-off sanity check for some stakeholders or shown to other stakeholders as reasoning for a particular prediction [48]?
- *Dynamic Model Updates*: Will the explanation be used to garner feedback from the stakeholder as to how the model ought to be updated to better align with their intuition? That is, how does the stakeholder interact with the model after viewing the explanation? Ross et al. [51] attempt to develop a technique for dynamic explanations, wherein the human can guide the model towards learning the correct explanation.

Once the desiderata are clarified, stakeholders should be consulted again.

5.2   Concerns of Explainability

While there are many positive reasons to encourage explainability of machine learning models, we note come concerns which were raised in our interviews.

5.2.1   On 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, with a few exceptions [13]. Though non-causal explanations can still provide valid and useful interpretations of how the model works [37], many organizations said that they would be keen to use causal explanations if they were available.

5.2.2   On Privacy. Three organizations mentioned data privacy in the context of explainability, since in some cases explanations can be used to learn about the model [38, 63] or the training data [56]. Methods to counter these concerns have been developed. For example, Harder et al. [25] develop a methodology for training a differentially private model that generates local and global explanations using locally linear maps.

5.2.3   On Improving Performance. One purpose of explanations is to improve ML engineers' understanding of their models, in order to help them refine and improve performance. Since machine learning models are "dual use" [12], we should be aware that in some settings, explanations or other tools could enable malicious users to increase capabilities and performance of undesirable systems. For example, several organizations we talked with use explanation methods to improve their natural language processing and image recognition models for content moderation in ways that may concern some stakeholders.

5.2.4   Beyond Deep Learning. Though deep learning has gained popularity in recent years, many organizations still use classical ML techniques (e.g., logistic regression, support vector machines), likely due to a need for simpler, more interpretable models [52].

Many in the explainability community have focused on interpreting black-box deep learning models, even though practitioners

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 in genomics [58] to models trained on tabular data.

Model agnostic techniques like Lundberg and Lee [34] 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.

6 CONCLUSION

In this study, we critically examine how explanation techniques are used in practice. 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 be cognizant of the concerns associated with explainability. Through this analysis, we take a step towards describing explainability deployment and hope that future research builds trustworthy explainability solutions.

7 ACKNOWLEDGMENTS

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] 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.

[4] 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.

[5] 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.

[6] 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).

[7] 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)*.

[8] 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.

[9] 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.

[10] 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*.

[11] Umang Bhatt, Pradeep Ravikumar, and José M. F. Moura. 2019. Towards Aggregating Weighted Feature Attributions. abs/1901.10040 (2019).

[12] Miles Brundage, Shahar Avin, Jack Clark, Helen Toner, Peter Eckersley, Ben Garfinkel, Allan Dafoe, Paul Scharre, Thomas Zeitzoff, Bobby Filar, et al. 2018. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. *arXiv preprint arXiv:1802.07228* (2018).

[13] 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.

[14] 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.]).

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

[16] 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.

[17] 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.

[18] 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).

[19] Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning. (2017).

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

[21] 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.

[22] 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

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

[24] 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.

[25] Frederik Harder, Matthias Bauer, and Mijung Park. 2019. Interpretable and Differentially Private Predictions. *arXiv preprint arXiv:1906.02004* (2019).

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

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

[28] 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.

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

[30] 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.

[31] 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).

[32] 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.

[33] Bruno Lepri, Nuria Oliver, Emmanuel Letouzé, Alex Pentland, and Patrick Vinck. 2018. Fair, transparent, and accountable algorithmic decision-making processes. *Philosophy & Technology* 31, 4 (2018), 611–627.

[34] 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.

[35] 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.

[36] 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).

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

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

[39] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, 220–229.

[40] Brent Mittelstadt, Chris Russell, and Sandra Wachter. 2019. Explaining explanations in AI. In *Proceedings of the conference on fairness, accountability, and transparency*. ACM, 279–288.

[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] Onora OâĂŹNeill. 2018. Linking trust to trustworthiness. *International Journal of Philosophical Studies* 26, 2 (2018), 293–300.

[45] 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).

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

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

[48] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. 2018. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810* (2018).

[49] Alun Preece, Dan Harborne, Dave Braines, Richard Tomsett, and Supriyo Chakraborty. 2018. Stakeholders in explainable AI. *arXiv preprint arXiv:1810.00184* (2018).

[50] 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.

[51] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: training differentiable models by constraining their explanations. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2662–2670.

[52] 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.

[53] Andrew D Selbst and Solon Barocas. 2018. The intuitive appeal of explainable machines. *Fordham L. Rev.* 87 (2018), 1085.

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

[55] 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).

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

[57] 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.

[58] 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.

[59] 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.

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

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

[62] 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.

[63] 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.

[64] 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

[65] 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.

[66] 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.

[67] Adrian Weller. 2019. Transparency: motivations and challenges. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 23–40.

[68] 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).

[69] 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).

[70] 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.

[71] 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