arXiv:2004.11488v1 [cs.LG] 23 Apr 2020

Adversarial Machine Learning: An Interpretation Perspective

Ninghao Liu, Mengnan Du, Xia Hu
Department of Computer Science and Engineering
Texas A&M University
College Station, Texas, USA
{nhliu43, dumengnan}@tamu.edu, hu@cse.tamu.edu

Abstract

Recent years have witnessed the significant advances of machine learning in a wide spectrum of applications. However, machine learning models, especially deep neural networks, have been recently found to be vulnerable to carefully-crafted input called adversarial samples. The difference between normal and adversarial samples is almost imperceptible to human. Many work have been proposed to study adversarial attack and defense in different scenarios. An intriguing and crucial aspect among those work is to understand the essential cause of model vulnerability, which requires in-depth exploration of another concept in machine learning models, i.e., interpretability. Interpretable machine learning tries to extract human-understandable terms for the working mechanism of models, which also receives a lot of attention from both academia and industry. Recently, an increasing number of work start to incorporate interpretation into the exploration of adversarial robustness. Furthermore, we observe that many previous work of adversarial attacking, although did not mention it explicitly, can be regarded as natural extension of interpretation. In this paper, we review recent work on adversarial attack and defense, particularly, from the perspective of machine learning interpretation. We categorize interpretation into two types, according to whether it focuses on raw features or model components. For each type of interpretation, we elaborate on how it could be used in attacks, or defense against adversaries. After that, we briefly illustrate other possible correlations between the two domains. Finally, we discuss the challenges and future directions along tackling adversary issues with interpretation.

1. Introduction

Machine learning (ML) techniques, especially recent deep learning models, are progressing rapidly and have been increasingly applied in various applications. Nevertheless, concerns have been posed about the security and reliability issues of ML models. In particular, many advanced machine learning models are susceptible to *adversarial attack* [94, 31]. That is, after adding certain well-designed but human imperceptible perturbation or transformation to a clean data instance, we are able to manipulate the prediction of the model. The data instances after being attacked are called *adversarial samples*. The phenomenon is intriguing since clean samples and adversarial samples are usually not distinguishable to human. Adversarial samples may be predicted dramatically differently from clean samples, but the predictions usually do not make sense to human.

The vulnerability of models to adversarial attack have been widely discovered in different tasks and under different constraints. For examples, approaches for crafting adversarial samples have been proposed in tasks such as classification (e.g., for image data, text data, tabular data, network data), object detection, fraud detection. Adversarial attack could be initiated under different constraints, such as assuming limited knowledge of attackers on target models [76, 75], assuming higher generalization level of attack [69, 70], posing different real-world constraints on attack [96, 47]. Given these progresses, several questions could be posted. First, are these progresses relatively independent of each other, or is there a underlying perspective from which we are able to discover the common rationale behind them? Second, should adversarial samples be seen as the negligent corner cases that could be fixed by putting patches to models, or are they deeply rooted to the working mechanism of models that we cannot get rid of unless new models are proposed?

Motivated by the idiom that "if you know yourself and your enemy, you will win every war", in this paper, we review the recent advances of adversarial attack and defense approaches from the perspective of interpretation. On one hand, if adversaries knows how the target model work, they may utilize it as a backdoor to the model and initiate attacks. On the other hand, if defenders knows how their models work, they could identify the model's vulnerability and try to mitigate the problem. Interpretation refers to the human-understandable information explaining what a model have learned or how a model makes prediction. Despite the recent advances, machine learning has been criticized due to the lack of transparency. To overcome the limitation, exploration of model interpretability has also attracted interest in recent years. Meanwhile, many recent work start to involve interpretability into the analysis of adversarial robustness. Also, although not being explicitly specified, a number of existing adversary-related work can be comprehended from another perspective as extension of model interpretation.

Before connecting the two domains, we first briefly introduce the subjects of interpretation to be covered in this paper. *Interpretability* is defined as "the ability to explain or to present in understandable terms to a human [20]". Although a formal definition of interpretation still remains elusive [20, 60, 42, 38], the overall goal is to obtain and transform information from models or their behaviors into a domain that human can make sense of [68]. For a more structured analysis, we categorize existing work on interpretable ML into two categories: feature-level interpretation and model-level interpretation, as shown in Figure 1. Feature-level interpretation targets to find the most important features in a data sample to its prediction. Model-level interpretation explores the functionality of model components, and their internal states after being fed with input. This categorization is based on whether the internal working mechanism of models involves in interpretation.

Following the above categorization, the overall structure of this article is organized as below. To begin with, we briefly introduce different types of adversarial attack and defense strategies in Section 2 and Section 3, respectively. Then, we introduce different categories of interpretation approaches, and demonstrate in detail how interpretation correlates to the attack and defense strategies. Specifically, we discuss feature-level interpretation in Section 4 and model-level interpretation in Section 5. After that, we extend the discussion to other relations between interpretation and adversarial aspects of model. Finally, we discuss some opening challenges for future work.

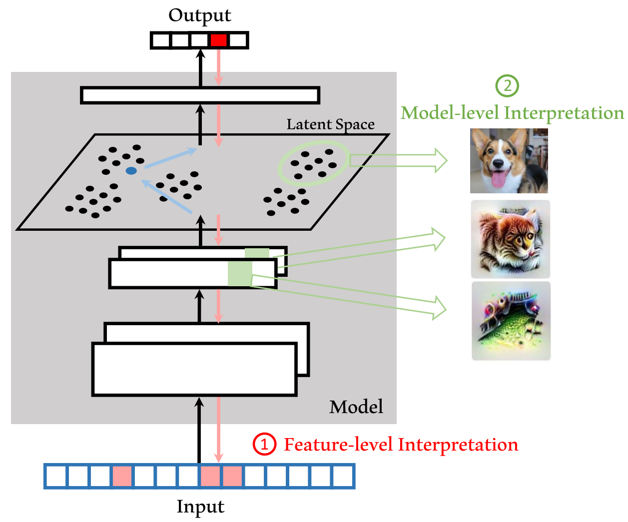

Figure 1. Illustration of Feature-level Interpretation and Model-level Interpretation for a deep model.

2. Taxonomies of Adversarial Threat Model

In this section, we systematically introduce different types of adversarial attacks. The overall adversarial threat models may be divided into different subgroups under diverse criterion. Based on different application scenarios, conditions, and adversary capabilities, specific attack strategies will be deployed.

2.1. Untargeted vs Targeted Attack

Based on the goal of attackers, the threat models can be classified into targeted and untargeted ones. For *targeted* attack, it attempts to mislead a model's prediction to a specific class given an instance. Let f denote the target model exposed to adversarial attack. A clean data instance is $\mathbf{x}_0 \in X$, and X is the input space. We consider classification tasks, so $f(\mathbf{x}_0) = c, c \in \{1, 2, ..., C\}$. One way of formulating the task of targeted attack is as below [94]:

$$\min_{\mathbf{x} \in X} \; d(\mathbf{x}, \mathbf{x}_0), \quad \text{s.t. } f(\mathbf{x}) = c' \tag{1}$$

where $c' \neq c$, and $d(\mathbf{x}, \mathbf{x}_0)$ measures the distance between the two instances. A typical choice of distance measure is to use l_p norms, where $d(\mathbf{x}, \mathbf{x}_0) = \|\mathbf{x} - \mathbf{x}_0\|_p$. The core idea is to add small perturbation to the original instance $\mathbf{x}_0$ to make it being classified as c'. However, in some cases, it is important to increase the confidence of perturbed samples being misclassified, so the task may also be formulated as:

$$\max_{\mathbf{x} \in X} \; f_{c'}(\mathbf{x}), \quad \text{s.t. } d(\mathbf{x}, \mathbf{x}_0) \leq \delta \tag{2}$$

where $f_{c'}(\mathbf{x})$ denotes the probability or confidence that $\mathbf{x}$ is classified as c' by f, and δ is a threshold limiting perturbation magnitude. For *untargeted* attack, its goal is to prevent a model from assigning a specific label to an instance. The objective of untargeted attack could be formulated in a similar way as targeted attack, where we just need to change the constraint as $f(\mathbf{x}) \neq c$ in Equation 1, or change the objective as $\min_{\mathbf{x} \in X} f_c(\mathbf{x})$ in Equation 2.

In some scenarios, the two types of attack above are also called *false positive* attack and *false negative* attack. The former aims to make models misclassify negative instances as positive, while the latter tries to mislead models to classify positive instances as negative. False positive attack and false negative attack sometimes are also called Type-I attack and Type-II attack.

2.2. One-Shot vs Iterative Attack

According to practical constraints, adversaries may initiate one-shot or iterative attack to target models. In *one-shot* attack, they have only one chance to generate adversarial samples, while *iterative attack* could take multiple steps to explore better direction. Iterative attack can generate more effective adversarial samples than one-shot attack. However, it also requires more queries to the target model and more computation to initiate each attack, which may limit its application in some computational-intensive tasks.

2.3. Data Dependent vs Universal Attack

According to information sources, adversarial attacks could be data dependent or independent. In *data dependent* attack, perturbations are customized based on the target instance. For example, in Equation 1, the adversarial sample $\mathbf{x}$ is crafted based on the original instance $\mathbf{x}_0$. However, it is also possible to generate adversarial samples without referring to the input instance, and it is also named as *universal* attack [69, 67]. The problem can be abstracted as looking for a perturbation vector $\mathbf{v}$ so that

$$f(\mathbf{x}+\mathbf{v}) \neq f(\mathbf{x}) \;\text{ for "most" }\; \mathbf{x} \in X. \tag{3}$$

We may need a number of training samples to obtain $\mathbf{v}$, but it does not rely on any specific input at test time. Adversarial attack can be implemented efficiently once the vector $\mathbf{v}$ is solved.

2.4. Perturbation vs Replacement Attack

Adversarial attacks can also be categorized based on the way of input distortion. In *perturbation* attack, input features are shifted by specific noises so that the input is misclassified by the model. In this case, let $\mathbf{x}^*$ denote the final adversarial sample, then it can be obtained via

$$\mathbf{x}^* = \mathbf{x}_0 + \Delta\mathbf{x}, \tag{4}$$

and usually $\|\Delta\mathbf{x}\|_p$ is small.

In *replacement* attack, certain parts of input are replace by adversarial patterns. Replacement attack is more natural in physical scenarios. For examples, criminals may want to wear specifically designed glasses to prevent them from being recognized by computer vision systems [1]. Also, surveillance cameras may fail to detect persons wearing clothes attached with adversarial patches [96]. Suppose $\mathbf{v}$ denote the adversarial pattern, then replacement attack can be represented by using a mask $\mathbf{m} \in \{0,1\}^{|\mathbf{x}_0|}$, so that

$$\mathbf{x}^* = \mathbf{x}_0 \odot (\mathbf{1}-\mathbf{m}) + \mathbf{v} \odot \mathbf{m} \tag{5}$$

where "$\odot$" denotes element-wise multiplication.

[1] https://www.inovex.de/blog/machine-perception-face-recognition/

2.5. White-Box vs Black-Box Attack

In *white-box* attack, it is assumed that attackers know everything about the target model, which may include model architecture, weights, hyper-parameters and even training data. White-box attack helps discovering intrinsic vulnerabilities of the target model. It works in ideal cases representing the worst scenario that defenders have to confront. *Black-box* attack assumes that attackers are only accessible to the model output, just like normal end users. This is a more practical assumption in real-world scenarios. Although a lot of detailed information about models are occluded, black-box attack still poses significant threat to machine learning systems due to the transferability property of adversarial samples [75]. In this sense, attacker could build a new model f' to approximate the target model f, and adversarial samples created on f' could still be effective to f.

3. Taxonomies of Defense Strategies

In this section, we briefly introduce the basic idea of different defense strategies against adversaries.

- **Input Denoising.** As adversarial perturbation is a type of human imperceptible noise added to data, then a natural defense solution is to filter it out, or to use additional random transformation to offset adversarial noise. It is worth noting that f_m could be added prior to model input layer [101, 54, 103], or as an internal component inside the target model [102]. Formally, for the former case, given an instance $\mathbf{x}^*$ which is probably affected by adversaries, we hope to design a mapping f_m, so that $f(f_m(\mathbf{x}^*)) = f(\mathbf{x}_0)$. For the latter case, the idea is similar except that f is replace by certain intermediate layer output h.
- **Model Robustifying.** Refining the model to prepare itself against potential threat from adversaries is another widely applied strategy. The refinement of model could be achieved from two directions: changing the training objective, or modifying model structure. Some examples of the former include adversarial training [94, 31], and replacing empirical training loss with robust training loss [64]. The intuition behind is to consider in advance the threat of adversarial samples during model training, so that the resultant model gains robustness from training. Examples of model modification include model distillation [77], applying layer discretization [61], controlling neuron activations [95]. Formally, let f' denote the robust model, the goal is to make $f'(\mathbf{x}^*)) = f'(\mathbf{x}_0) = y$.
- **Adversarial Detection.** Different from the previous two strategies where we hope to discover the true label given an instance, adversarial detection tries to identify whether the given instance is polluted by adversarial perturbation. The general idea is to build another predictor f_d,

so that $f_d(\mathbf{x}) = 1$ if $\mathbf{x}$ has been polluted, and otherwise $f_d(\mathbf{x}) = 0$. The establishment process of f_d could follow the normal routine of building a binary classifier [30, 66, 33].

The first two strategies proactively recover the correction prediction from influences of adversarial attack, by fixing the input data and model architectures respectively. The third strategy reactively decide whether the model should make predictions against the input in order not to be fooled. Implementations of the proactive strategies are usually more challenging than the reactive one.

4. Feature-Level Interpretation

Attribution-based interpretation is a widely used post-hoc method to identify feature importance with respect to model prediction. Attribution methods focus on the end-to-end relation between input and output, instead of carefully examining the internal states of models. Some examples include measuring the importance of phrases of sentences in text classification [22], and pixels in image classification [112]. In this section we will discuss how attribution-based interpretation can be related with the attack and defense of adversaries, though the original work may not necessarily analyze adversaries from this perspective.

4.1. Feature-Level Interpretation for Understanding Adversarial Attack

4.1.1 Gradient-Based Techniques

Following the notations in previous discussion, we let $f_c(\mathbf{x}_0)$ denote the probability that model f classify the input instance $\mathbf{x}_0$ as class c. One of the intuitive ways to understand why such prediction is derived is to attribute prediction $f_c(\mathbf{x}_0)$ to feature dimensions of $\mathbf{x}_0$. A fundamental technique to obtain attribution scores is backpropagation. According to [89], $f_c(\mathbf{x}_0)$ can be approximated with a linear function surrounding $\mathbf{x}_0$ by computing its first-order Taylor expansion:

$$f_c(\mathbf{x}) \approx f_c(\mathbf{x}_0) + \mathbf{w}_c^T \cdot (\mathbf{x} - \mathbf{x}_0) \tag{6}$$

where $\mathbf{w}_c$ is the gradient of f_c with respect to input at $\mathbf{x}_0$, i.e., $\mathbf{w}_c = \nabla_{\mathbf{x}} f_c(\mathbf{x}_0)$. From the interpretation perspective, $\mathbf{w}_c$ entries of large magnitude correspond to the features that are important around the current output.

However, another perspective to comprehend the above equation is that, the interpretation coefficient vector $\mathbf{w}_c$ also indicates the most effective direction of locally changing the prediction result by perturbing input away from $\mathbf{x}_0$. If we let $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_0 = -\mathbf{w}_c$, we are attacking the model f with respect to the input-label pair $(c, \mathbf{x}_0)$. Such perturbation method is closely related to the Fast Gradient Sign (FGS) attacking method [31], where:

$$\Delta\mathbf{x} = \epsilon \cdot sign(\nabla_{\mathbf{x}} J(f, \mathbf{x}_0, c)), \tag{7}$$

except that (1) FGS computes the gradient of a certain cost function J nested outside f, and (2) it applies an additional $sign()$ operation on gradient for processing images. However, if we define J with cross entropy loss, and the true label of $\mathbf{x}_0$ is c, then

$$\nabla_{\mathbf{x}} J(f, \mathbf{x}_0, c) = -\nabla_{\mathbf{x}} \log f_c(\mathbf{x}_0) = -\frac{1}{f_c(\mathbf{x}_0)} \nabla_{\mathbf{x}} f_c(\mathbf{x}_0), \tag{8}$$

which points to the same perturbation direction as reversing the interpretation $\mathbf{w}_c$. Both gradient-based interpretation and FGS rely on the assumption that the targeted model can be locally approximated by linear models.

The traditional FGS method is proposed under the setting of untargeted attack, where the goal is to impede input from being correctly classified. For targeted attack, where the goal is to misguide the model prediction towards a specific class, a typical way is Box-constrained L-BFGS (L-BFGS-B) method [94]. Assume c' is the target label, the problem of L-BFGS-B is formulated as:

$$\underset{\mathbf{x} \in X}{\operatorname{argmin}} \quad \alpha \cdot d(\mathbf{x}, \mathbf{x}_0) + J(f, \mathbf{x}, c') \tag{9}$$

where d is considered to control perturbation degree, and X is the input domain (e.g., $[0, 255]$ for each channel of image input). The goal of attack is to make $f(\mathbf{x}) = c'$, while maintaining $d(\mathbf{x}, \mathbf{x}_0)$ to be small. Suppose we apply gradient descent to solve the problem, and $\mathbf{x}_0$ is the starting point. Similar to the previous discussion, if we define J as the cross entropy loss, then

$$-\nabla_{\mathbf{x}} J(f, \mathbf{x}_0, c')) = \nabla_{\mathbf{x}} \log f_{c'}(\mathbf{x}_0) \propto \mathbf{w}_{c'}\,. \tag{10}$$

On one hand, $\mathbf{w}_{c'}$ locally and linearly interprets $f_{c'}(\mathbf{x}_0)$, and it also serves the most effective direction to make $\mathbf{x}_0$ towards being classified as c'.

According to the taxonomy of adversarial attack, the two scenarios discussed above can also be categorized into: (1) one-shot attack, since we only perform interpretation once, (2) data-dependent attack, since the perturbation direction is related with $\mathbf{x}_0$, (3) white-box attack, since model gradients are available. Other types of attack could be crafted if different interpretation strategies are applied, which will be discussed in later sections.

Improved Gradient-Based Techniques. The interpretation methods based on raw gradients, as discussed above, are usually unstable and noisy [82, 71]. The possible reasons include: (1) the target model itself is not stable in terms of function surface or model establishment; (2) gradients only consider the local output-input relation so that its scope is too limited; (3) the prediction mechanism is too complex to be approximated by a linear substitute. Some approaches for improving interpretation (i.e., potential adversarial attack) are as below.

- **Region-Based Smoothing:** To reduce random noises in interpretation, SmoothGrad is proposed in [91], where the final interpretation $\mathbf{w}_c$, as a sensitivity map, is obtained by averaging a number of sensitivity maps of instances sampled around the target instance $\mathbf{x}_0$, i.e., $\mathbf{w}_c = \sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x}_0)} \frac{1}{|\mathcal{N}(\mathbf{x}_0)|} \nabla f_c(\mathbf{x}')$. The averaged sensitivity map will be visually sharpened. A straightforward way to extend it for adversarial attack is to perturb input by reversing the averaged map. Furthermore, [97] designed a different strategy by adding a step of random perturbation before gradient computation in attack, to jump out of the non-smooth vicinity of the initial instance. Spatial averaging is a common technique to stabilize output. For examples, [11] applied it as a defense method to derive more stable model predictions. Also, [51] proposed a certifiable interpretation method that is robust to perturbation by revising SmoothGrad.

- **Path-Based Integration:** To improve interpretation, [93] proposes Integrated Gradient (InteGrad). After setting a baseline point $\mathbf{x}^b$, e.g., a black image in object recognition tasks, the interpretation is defined as:

$$\mathbf{w}_c = \frac{(\mathbf{x}_0 - \mathbf{x}^b)}{D} \circ \sum_{d=1}^{D} [\nabla f_c](\mathbf{x}^b + \frac{d}{D}(\mathbf{x}_0 - \mathbf{x}^b)), \quad (11)$$

which is the weighted sum of gradients along the straight-line path from $\mathbf{x}_0$ to the baseline point $\mathbf{x}^b$. A similar strategy in adversarial attack is iterative attack [48], where the sample is iteratively perturbed as:

$$\mathbf{x}'_0 = \mathbf{x}_0, \;\; \mathbf{x}'_{d+1} = Clip_{\mathbf{x}_0,\epsilon}\{\mathbf{x}'_d + \alpha \nabla_{\mathbf{x}} J(f, \mathbf{x}'_d, c)\}, \quad (12)$$

which gradually explore the perturbation along a path directed by a series of gradients. $Clip_{\mathbf{x}_0,\epsilon}(\mathbf{x})$ denotes element-wise clipping $\mathbf{x}$ so that $d(\mathbf{x} - \mathbf{x}_0) \leq \epsilon$.

Interestingly, although raw gradients can be shown to human as interpretation through straightforward visualization methods, it is no longer perceivable if used as perturbation added to input.

4.1.2 Distillation-Based Techniques

The interpretation techniques discussed so far require internal information from models, such as gradients and relevance scores. It is also possible to extract interpretation without querying a model f more than $f(\mathbf{x})$. Thus, this type of interpretation methods, here named as distillation-based methods, can be used for black-box attack.

The main idea of applying distillation for interpretation is to use an interpretable model g (e.g., a decision tree) to mimic the behavior of the target deep model f [12, 27]. Once we obtain g, existing white-box attack methods could be applied to craft adversarial samples [59]. In addition, given an instance $\mathbf{x}_0$, to guarantee that g more accurately mimics the nuanced behaviors of f, we could further require that g locally approximates f around the instance. The objective is thus as below:

$$\min_g \mathcal{L}(f, g, \mathbf{x}_0) + \alpha \cdot C(g), \quad (13)$$

where $\mathcal{L}$ denotes the approximation error around $\mathbf{x}_0$. For examples, in LIME [79]:

$$\mathcal{L}(f, g, \mathbf{x}_0) = \sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x}_0)} \exp(-d(\mathbf{x}_0, \mathbf{x}')) \|f(\mathbf{x}') - g(\mathbf{x}')\|^2, \quad (14)$$

and $\mathcal{N}(\mathbf{x}_0)$ denotes the local region around $\mathbf{x}_0$. In addition, LEMNA [35] adopts mixture regression models for g and fused lasso as regularization $C(g)$. After obtaining g, we can craft adversarial samples targeting g using attack methods by removing or reversing the interpretation result. According to the property of transferability [75], an adversarial sample that successfully fools g is also likely to fool f. The advantages are two-fold. First, the process is model agnostic and does not assume availability to gradients. It could be used for black-box attack or attacking certain types of models (such as tree-based models) that do not use gradient backpropagation in training. Second, one-shot attacks on g could be more effective thanks to the smoothness term $C(g)$ as well as extending the consideration to include the neighborhood of $\mathbf{x}_0$ [7]. Thus, it has the potential to cause defense methods that are based on obfuscated gradients [1] to be less robust. The disadvantage is that crafting each adversarial sample requires high computation cost.

In certain scenarios, it is beneficial to make adversarial patterns understandable to humans as real-world simulation when identifying the vulnerability of models. For examples, in autonomous driving, it helps to consider physically-possible patterns that could cause misjudgement of autonomous vehicles [8]. One possible approach is to constrain adversarial instances to fall into the data distribution. For example, [17] achieves this through an additional regularization term $\|\mathbf{x}_0 + \Delta\mathbf{x} - AE(\mathbf{x}_0 + \Delta\mathbf{x})\|$, where $AE(\cdot)$ denotes an autoencoder. Another strategy is to predefine a dictionary, and then makes the adversarial perturbation to match one of the dictionary tokens [8], or a weighted combination of the tokens [85].

4.2. Feature-level Interpretation Against Adversaries

Feature-level interpretation could be used for defense against adversaries through adversarial training and detecting model vulnerability.

4.2.1 Adversarial Training

Adversarial training [31][48] is one of the most applied proactive countermeasures to improve the robustness of the model. Its core idea is to first generate adversarial samples to unveil the weakness of the model, and then inject the adversarial samples into training set for data augmentation. The overall loss function can be formulated as:

$$\min_{f} \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}}\,[\alpha J(f(\mathbf{x}),y) + (1-\alpha)J(f(\mathbf{x}^*),y)]. \quad (15)$$

In the scenario of adversarial training, feature-level interpretation helps in preparing adversarial samples $\mathbf{x}^*$, which may refer to any method discussed in Section 4.1. Although such an attack-and-then-debugging strategy has been successfully applied in many traditional cybersecurity scenarios, one key drawback is that it tends to overfit to the specific approach that is used to generate $\mathbf{x}^*$. It is untenable and ineffective [37] to exhaust a number of possible attacking methods for data preparation. Meanwhile, it is argued that naive adversarial training may actually perform gradient masking instead of moving the decision boundary [1][49].

To train more robust models, some optimization based methods have been proposed. [64] argued that traditional Empirical Risk Minimization (ERM) fails to yield models that are robust to adversarial instances, and proposed a min-max formulation to train robust models:

$$\min_{f} \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}}\,[\max_{\delta\in\Delta X} J(\mathbf{x}+\delta,y)], \quad (16)$$

where ΔX denotes the set of allowed perturbations. It formally defines adversarially robust classification as a learning problem to reduce adversarial expected risk. This min-max formulation provides another perspective on adversarial training, where the inner task aims to find adversarial samples, and the outer task retrains model parameters. [97] further improves its defense performance by crafting adversarial samples from multiple sources to augment training data. This strategy is also implicitly supported in [86] which shows training robust models requires much greater data complexity. [108] further identifies a trade-off between robust classification error [14, 86] and natural classification error, which provides a solution to reduce the negative effect on model accuracy after adversarial training.

4.2.2 Model Vulnerability Detection

In the scenario where a model is subject to adversarial attack, interpretation may serve as a new type of information for directly detecting adversarial patterns. A straightforward way is to train another classifier f_d as the detector

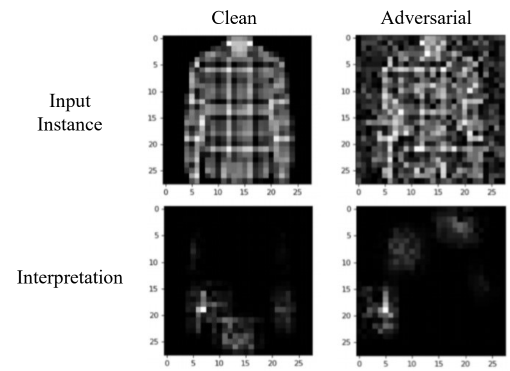

Figure 2. The interpretations between clean and adversarial inputs may differ.

trained with interpretations of both clean and adversarial instances, paired with labels indicating whether the sample is clean [26, 107, 25, 105]. Specifically, [107] directly uses gradient-based saliency map as interpretation, [105] adopts the distribution of Leave-One-Out (LOO) attribution scores, while [26] proposes a new interpretation method based on masks highlighting important regions.

In more scenarios, interpretation serves as a diagnosis tool to qualitatively identify model vulnerability. First, we could use interpretation to identify whether inputs are affected by adversarial attack. For example, if interpretation result shows that unreasonable evidences have been used for prediction [19], then it is possible that there exists suspicious but imperceptible input pattern. Second, interpretation may reflect whether a model is susceptible to adversarial attack. Even given a clean input instance, if interpretation of model prediction does not make much sense to human, then the model is under the risk of being attacked. For examples, in a social spammer detection system, if the model regards certain features as important but they are not strongly correlated with maliciousness, then attackers could easily manipulate these features without much cost to fool the system [59]. Also, in image classification, CNN models have been demonstrated to focus on local textures instead of object shapes, which could be easily utilized by attackers [2]. An interesting phenomenon in image classification is that, after refining a model with adversarial training, feature-level interpretation results indicate that the refined model will be less biases towards texture features [110].

Nevertheless, there are several challenges that impede the intuitions above from being formulated to formal defense approaches. First, interpretation itself is also fragile in neural networks. Attackers could control prediction and interpretation simultaneously via indistinguishable perturbation [28, 92]. Second, it is difficult to quantify the robustness of model through interpretation [82]. Manual inspection of interpretation helps discover defects in model, but visually acceptable interpretation does not guarantee model robustness. That is, defects in feature-level interpretation indicate the presence but not the absence of vulnerability.

5. Model-level Interpretation

In this review, model-level interpretation is defined with two aspects. First, model-level interpretation aims to figure out what has been learned by intermediate components in a trained model [109, 89], or what is the meaning of different locations in latent space [45, 113, 56]. Second, given an input instance, model-level interpretation unveils how the input is encoded by those components as latent representation [45, 113, 54, 102]. In our discussion, the former does not rely on input instances, while the later is the opposite. Therefore, we name the two aspects as *Static Model Interpretation* and *Representation Interpretation* respectively to further distinguish them. Representation interpretation could rely on static model interpretation.

5.1. Static Model Interpretation for Understanding Adversarial Attack

For deep models, one of the most widely explored strategies is to explore the visual or semantic meaning of each neuron. A popular strategy for solve this problem is to recover the patterns that activate the neuron of interests at a specific layer [23, 89]. Following the previous notations, let $h(\mathbf{x})$ denote the activation of neuron h given input, the perceived pattern of the neuron can be visualized via solving the problem below:

$$\underset{\mathbf{x}'}{\operatorname{argmax}}\, h(\mathbf{x}') - \alpha \cdot C(\mathbf{x}'), \tag{17}$$

where $C(\cdot)$ such as $\|\cdot\|_1$ or $\|\cdot\|_2$ acts as regularization. Conceptually, the result contains patterns that neuron h is sensitive to. If we choose h to be f_c, then the resultant $\mathbf{x}'$ illustrates class appearances learned by the target model. Another discussion about different choices of h, such as neurons, channels, layers, logits and class probabilities, is provided in [73]. Similarly, we could also formulate another minimization problem

$$\underset{\mathbf{x}'}{\operatorname{argmin}}\, h(\mathbf{x}') + \alpha \cdot C(\mathbf{x}'), \tag{18}$$

to produce patterns that prohibit activation of certain model components or prediction towards certain classes.

The interpretation result $\mathbf{x}'$ is highly related with several types of adversarial attack, with some examples shown in Figure 3:

- **Targeted-Universal-Perturbation Attack:** If we set h to be class relevant mapping such as f_c, then $\mathbf{x}'$ can be directly added to target input instance as targeted perturbation attack. That is, given a clean input $\mathbf{x}_0$, the adversarial sample $\mathbf{x}^*$ is crafted simply as $\mathbf{x}^* = \mathbf{x}_0 + \lambda \cdot \mathbf{x}'$ to make $f(\mathbf{x}^*) = c$. It belongs to universal attack, because the interpretation process in Eq.17 does not utilize any information of the clean input.

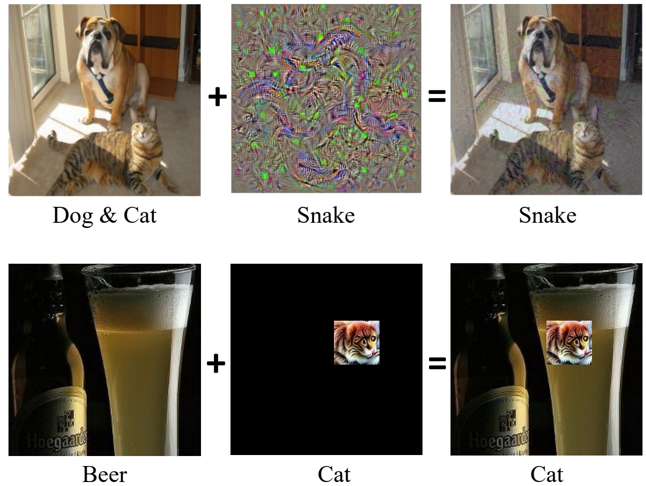

Figure 3. Example of adversarial attack after applying model-level interpretation. Upper: Targeted universal perturbation. Lower: Universal replacement attack.

- **Untargeted-Universal-Perturbation Attack:** If we set h to be the aggregation of a number of middle-level layer mappings, such as $h(\mathbf{x}') = \sum_l \log(h^l(\mathbf{x}'))$ where h^l denotes the feature map tensor at layer l, the resultant $\mathbf{x}'$ is expected to produce spurious activation to confuse the prediction of CNN models given any input, which implies $f(\mathbf{x}_0 + \lambda \cdot \mathbf{x}') \neq f(\mathbf{x}_0)$ with high probability [70].

- **Universal-Replacement Attack:** Adversarial patches, which completely replace part of input, represent a visually different attack from perturbation attack. Based on Eq.17, more parameters such as masks, shape, location and rotation could be considered in the optimization to control $\mathbf{x}'$ [10]. The patch is obtained as $\mathbf{x}' \odot \mathbf{m}$, and the adversarial sample $\mathbf{x}^* = \mathbf{x}_0 \odot (\mathbf{1} - \mathbf{m}) + \mathbf{x}' \odot \mathbf{m}$, where $\mathbf{m}$ is a binary mask that defines patch shape. Besides, based on Eq.18, after defining h as the objectness score function in person detectors [96] or as the logit corresponding to human class [88], it produces real-world patches attachable to human bodies to avoid them being detected by surveillance camera.

5.2. Representation Interpretation for Initiating Adversarial Attack

Representation learning plays a crucial role in recent advances of machine learning, with applications in vision [5], natural language processing [106] and network analysis [36]. However, the opacity of representation space also becomes the bottleneck for understanding complex models. A commonly used strategy toward understanding representation is to define a set of explainable basis, and then decompose representation points along the basis. Formally, let $\mathbf{z}_i \in \mathbb{R}^D$ denote a representation vector, and $\{\mathbf{b}_k \in \mathbb{R}^D\}_{k=1}^K$ denote the basis set, where D denotes the representation dimension and K is the number of base vectors. Then, through decomposition

$$\mathbf{z}_i = \sum_{k=1}^{K} p_{i,k} \cdot \mathbf{b}_k, \tag{19}$$

we can explain the meaning of $\mathbf{z}_i$ through referencing base vectors whose semantics are known, where $p_{i,k}$ measures the affiliation degree between instance $\mathbf{z}_i$ and $\mathbf{b}_k$. The work of providing representation interpretation following this scheme can be divided into several groups:

- **Dimension-wise Interpretation**: A straightforward way to achieve interpretability is to require each dimension to have a concrete meaning [40, 74], so that the basis can be seen as non-overlapping one-hot vectors. A natural extension to this would be to allow several dimensions (i.e., a segment) to jointly encode one meaning [58, 63].
- **Concept-wise Interpretation**: A set of high-level and intuitive concepts could first be defined, so that each $\mathbf{b}_k$ encodes one concept. Some examples include visual concepts [113, 45, 29], antonym words [65], and network communities [56].
- **Example-wise Interpretation**: Each base vector can be designed to match one data instance [44, 46, 99] or part of the instance [13]. Those instances are also called prototypes. For examples, a prototype could be an image region [13] or a node in networks [99].

The extra knowledge obtained from representation interpretation could be used to guide the direction of adversarial perturbation. However, the motivation of this type of work usually is to initiate more meaningful adversaries and then use adversarial training to improve model generalization, but not for the pure purpose of undermining model performance. For examples, in text mining, [85] restricts perturbation direction of each word embedding to be a linear combination of vocabulary word embeddings, which improves model performance in text classification with adversarial training. In network embedding, [16] restricts perturbation of a node's embedding towards the embeddings of the node's neighbors in the network, which benefits node classification and link prediction.

5.3. Model-level Interpretation Against Adversaries

5.3.1 Model Robustifying

Some high-level features learned by deep models are not robust, which is insufficient to train robust models. A novel algorithm is proposed in [41] to build datasets of robust features. Let $h : X \rightarrow \mathbb{R}$ denote a transformation function that maps input to a representation neuron. Each instance in the robust dataset $\mathcal{D}_r$ is constructed from the original dataset $\mathcal{D}$ through solving a optimization problem:

$$\mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_r}[h(\mathbf{x}) \cdot y] = \begin{cases} \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}}[h(\mathbf{x}) \cdot y], & \text{if } h \in \mathcal{H}_r \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

where $\mathcal{H}_r$ denotes the set of features utilized by robust models. In this way, input information that corresponds to non-robust representations are suppressed.

Despite not being directly incorporated in the procedure of model training, inspection of model-level interpretation, especially latent representation, has motivated several defense approaches. Through visualizing feature maps of latent representation layers, the noise led by adversarial perturbation can be easily observed [102, 54, 105]. With this observation, [102] proposes adding denoising blocks between intermediate layers of deep models, where the core function of the denoising blocks are chosen as low-pass filters. [54] observed that adversarial perturbation is magnified through feedforward propagation in deep models, and proposed a U-net model structure as denoiser.

5.3.2 Adversarial Detection

Instead of training another large model as detector using raw data, we can also leverage model-level interpretation to detect adversarial instances more efficiently. By regarding neurons as high-level features, readily available interpretation methods such as SHAP [62] could be applied for feature engineering to build adversarial detector [25]. After inspecting the role of neurons in prediction, a number of critical neurons could be selected. A steered model could be obtained by strengthening those critical neurons, while adversarial instances are detected if they are predicted very differently by the original model and steered model [95]. Nevertheless, the majority of work on adversarial detection utilizes latent representation of instances without inspecting their meanings, such as directly applying statistical methods on representations to build detectors [67, 53, 24] or conducting additional coding steps on activations of neurons [61].

6. Additional Relations Between Adversary and Interpretation

In previous context, we have discussed how interpretation could be leveraged in adversarial attack and defense. In this section, we complement this viewpoint by analyzing the role of adversarial aspect of models in defining and evaluating interpretation. In addition, we specify the distinction between the two domains.

6.1. Defining Interpretation Using Adversaries

Some definitions of interpretation are inspired by adversarial perturbation. In natural language processing, to understand how different dimensions in word embeddings, or different words in sentences, contribute to downstream NLP tasks, we can erase the target dimensions or words from input, so that the output variation indicates whether the erased information is important for prediction [52]. In image processing, local interpretation is defined as a saliency detection problem, where salient regions could be defined as the input parts that most affect the output value when perturbed [26]. The goal is to find the smallest deletion mask to change the original prediction. Let $\mathbf{m}$ be a mask operator and $m_i \in [0, 1]$, the input after being edited by the mask is $\mathbf{x} = \mathbf{x}_0 \odot \mathbf{m} + \mathbf{a} \odot (\mathbf{1} - \mathbf{m})$. Here $\mathbf{a}$ is the alternative input, where a_i could be pre-defined as constant value, noise signal, or the mean value of nearby features (i.e., blurring). For saliency detection, the problem to be solved is as below

$$\operatorname*{argmin}_{\mathbf{m}} \alpha \|\mathbf{1} - \mathbf{m}\|_1 + f_c(\mathbf{x}), \tag{21}$$

where the first term encourages a smaller perturbation region, and the second term wants the perturbation to effectively affect prediction result. Using traditional iterative algorithms to generate masks is time-consuming, so [15] develops trainable masking models that generate masks in real time. In order to make the masks sharp and precise, the U-Net architecture [81] is applied for building the trainable model. In addition, objective function above can also be reformulated with information bottleneck theories [87].

As a natural extension from the discussion above, adversarial attack can also be used to evaluate the interpretation result. For examples, after obtaining the important features, and understanding whether they are positively or negatively related to the output, we could remove or distort these features to observe the target model's performance change [35, 59]. If the target model's performance significantly drops, then we are likely to have extracted correct interpretation. However, it is worth noting that there would be a conflict if adversary is used to define and evaluate interpretation simultaneously.

6.2. Improving Interpretation With Robust Models

In previous content, we have discussed the role of interpretation in studying model robustness. From another perspective, improving model robustness also influences interpretation of models. First, the representations learned by robust models tend to align better with salient data characteristics and human perception [98]. Therefore, adversarially robust image classifiers are also useful in more sophisticated tasks such as generation, super-resolution and translation [84], even without relying on GAN frameworks. Also, when attacking a robust classifier, resultant adversarial samples tend to be recognized similarly by the classifier and human [98]. In addition, retraining with adversarial samples [110], or regularizing gradients to improve model robustness [57], has been discovered to reduce noises and capture object shapes reflect from sensitivity maps.

6.3. Uniqueness of Interpretability From Adversary

Despite the common techniques applied for acquiring interpretation and exploring adversary characteristics, some aspects of the two directions put radically different requirements. For examples, some applications require interpretation to be easily understood by human especially by AI novices, such as providing more user-friendly interfaces to visualize and present interpretation [55, 72, 104], while adversarial attack requires perturbation to be imperceptible to human. Some work tries to adapt interpretation to fit human cognition habits, such as providing example-based interpretation [6], criticism mechanism [43] and counter-factual explanation [32]. Furthermore, generative models could be applied to create content from interpretation [111], where interpretation is post-processed into more understandable content such as dialogue texts. The emphasis of understandability in interpretability is exactly opposite to one of the objectives in adversarial attack, which focuses on crafting perturbation that is too subtle to be perceived by human.

7. Challenges and Future Work

We briefly introduce the challenges encountered in leveraging interpretation to analyze adversarial robustness of models. Finally, we discuss the future research directions.

7.1. Developing Models of Better Interpretability

Interpretation with better stability and faithfulness is needed before it could be used as a reliable tool to detect adversarial patterns. It has been shown that many existing interpretation methods are vulnerable to adversarial attacks [28, 39, 92, 90]. A stable interpretation method, given an input instance and a target model, should be able to produce relatively consistent result under the situation that noises may exist in input. As preliminary work, [18] analyzed the phenomenon from a geometric perspective of decision boundary and proposed a smoothed activation function to replace Relu. [51] introduced a sparsified variant of SmoothGrad [91] in producing saliency maps that is certifiably robust to adversarial attacks.

Besides post-hoc interpretation, developing intrinsically interpretable models is also receiving increasing attention [82]. With intrinsic interpretability, it could be easier for model developers to correct undesirable properties of models. One of the challenges is that requiring interpretability may negatively affect model performance. To tackle the problem, some preliminary work start to explore applying graph-based models, such as proposing relational inductive biases to facilitate learning about entities and their relations [4], towards a foundation of interpretable and flexible scheme of reasoning. Novel neural architectures have also been proposed such as capsule networks [83] and causal models [78].

7.2. Adversarial Attack in Real-World Scenarios

The most common scenario in existing work considers adversarial noises or patches in image classification or object detection. However, these types of perturbation may not represent the actual threats in physical world. To solve the dilemma, more adversarial scenarios have been studied in different applications, such as verification code generation [2], and semantically/syntactically equivalent adversarial text generation [50, 80]. Meanwhile, model developers need to be consistently alert to new types of attack that utilizes interpretation as the back door. For examples, it is possible to build models that predict correctly on normal data, but make mistakes on input with certain secret attacker-chosen property [34]. Also, recently researchers found that it is possible to break data privacy by reconstructing private data merely from gradients communicated between machines [114].

7.3. Improving Models with Adversarial Samples

The value of adversarial samples goes beyond simply serving as prewarning of model vulnerability. It is possible that the vulnerability to adversarial samples reflects some deeper generalization issues of deep models [3, 9]. Some preliminary work has been conducted to understand the difference between a robust model and a non-robust one. For examples, it has been shown that adversarially trained models possess better interpretability [110] and representations with higher quality [98, 84]. [21] also tries to connect adversarial robustness with model credibility, where credibility measures the degree that a model's reasoning conforms with human common sense. Another challenging problem is how to properly use adversarial samples to benefit model performance, since many existing work report that training with adversarial samples will lead to performance degradation especially on large data [48, 102]. Recently, [100] shows that, by separately considering the distributions of normal data and adversarial data with batch normalization, adversarial training can be used to improve model accuracy.

8. Conclusion

In this paper, we review the recent work of adversarial attack and defense by combining them with the recent advances of interpretable machine learning. Specifically, we categorize interpretation techniques into feature-level interpretation and model-level interpretation. Within each category, we investigated how the interpretation could be used for initiating adversarial attacks or designing defense approaches. After that, we briefly discuss other relations between interpretation and adversarial examples or robustness. Finally, we discuss current challenges of developing transparent and robust models, as well as potential directions to further utilizing adversarial samples.

[2] https://github.com/littleredhat1997/captcha-adversarial-attack

References

[1] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. 2018. 5, 6

[2] N. Baker, H. Lu, G. Erlikhman, and P. J. Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 2018. 6

[3] A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *NeurIPS*, 2019. 10

[4] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 10

[5] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 2013. 7

[6] I. Bichindaritz and C. Marling. Case-based reasoning in the health sciences: What's next? *Artificial intelligence in medicine*, 2006. 9

[7] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, 2013. 5

[8] A. Boloor, X. He, C. Gill, Y. Vorobeychik, and X. Zhang. Simple physical adversarial examples against end-to-end autonomous driving models. In *ICESS*. IEEE, 2019. 5

[9] T. B. Brown, N. Carlini, C. Zhang, C. Olsson, P. Christiano, and I. Goodfellow. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*, 2018. 10

[10] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 7

[11] X. Cao and N. Z. Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In *ACSAC*, 2017. 5

[12] Z. Che, S. Purushotham, R. Khemani, and Y. Liu. Distilling knowledge from deep networks with applications to healthcare domain. *arXiv preprint arXiv:1512.03542*, 2015. 5

[13] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su. This looks like that: deep learning for interpretable image recognition. In *NeurIPS*, 2019. 8

[14] D. Cullina, A. N. Bhagoji, and P. Mittal. Pac-learning in the presence of adversaries. In *NIPS*, 2018. 6

[15] P. Dabkowski and Y. Gal. Real time image saliency for black box classifiers. In *NIPS*, 2017. 9

[16] Q. Dai, X. Shen, L. Zhang, Q. Li, and D. Wang. Adversarial training methods for network embedding. In *The World Wide Web Conference*, 2019. 8

[17] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, and P. Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *NIPS*, 2018. 5

[18] A.-K. Dombrowski, M. Alber, C. Anders, M. Ackermann, K.-R. Müller, and P. Kessel. Explanations can be manipulated and geometry is to blame. In *NeurIPS*, 2019. 9

[19] Y. Dong, H. Su, J. Zhu, and F. Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493*, 2017. 6

[20] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017. 2

[21] M. Du, N. Liu, F. Yang, and X. Hu. Learning credible deep neural networks with rationale regularization. In *ICDM*, 2019. 10

[22] M. Du, N. Liu, F. Yang, S. Ji, and X. Hu. On attribution of recurrent neural network predictions via additive decomposition. In *The World Wide Web Conference*, 2019. 4

[23] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, page 1. 7

[24] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017. 8

[25] G. Fidel, R. Bitton, and A. Shabtai. When explainability meets adversarial learning: Detecting adversarial examples using shap signatures. *arXiv preprint arXiv:1909.03418*, 2019. 6, 8

[26] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*, 2017. 6, 9

[27] J. Gao, N. Liu, M. Lawley, and X. Hu. An interpretable classification framework for information extraction from online healthcare forums. *Journal of healthcare engineering*, 2017. 5

[28] A. Ghorbani, A. Abid, and J. Zou. Interpretation of neural networks is fragile. In *AAAI*, 2019. 6, 9

[29] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim. Towards automatic concept-based explanations. In *NeurIPS*, 2019. 8

[30] Z. Gong, W. Wang, and W.-S. Ku. Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960*, 2017. 4

[31] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 3, 4, 6

[32] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee. Counterfactual visual explanations. In *ICML*, 2019. 9

[33] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017. 4

[34] T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 10

[35] W. Guo, D. Mu, J. Xu, P. Su, G. Wang, and X. Xing. Lemna: Explaining deep learning based security applications. In *CCS*, 2018. 5, 9

[36] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017. 7

[37] W. He, J. Wei, X. Chen, N. Carlini, and D. Song. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*, 2017. 6

[38] C. G. Hempel and P. Oppenheim. Studies in the logic of explanation. *Philosophy of science*, 1948. 2

[39] J. Heo, S. Joo, and T. Moon. Fooling neural network interpretations via adversarial model manipulation. *arXiv preprint arXiv:1902.02041*, 2019. 9

[40] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017. 8

[41] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019. 8

[42] F. C. Keil. Explanation and understanding. *Annu. Rev. Psychol.*, pages 227–254, 2006. 2

[43] B. Kim, R. Khanna, and O. O. Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *NIPS*, 2016. 9

[44] B. Kim, C. Rudin, and J. A. Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *NIPS*, 2014. 8

[45] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *ICML*, 2018. 7, 8

[46] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017. 8

[47] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 1

[48] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. 2017. 5, 6, 10

[49] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, et al. Adversarial attacks and defences competition. Springer, 2018. 6

[50] Q. Lei, L. Wu, P.-Y. Chen, A. G. Dimakis, I. S. Dhillon, and M. Witbrock. Discrete adversarial attacks and submodular optimization with applications to text classification. *Systems and Machine Learning (SysML)*, 2019. 10

[51] A. Levine, S. Singla, and S. Feizi. Certifiably robust interpretation in deep learning. *arXiv preprint arXiv:1905.12105*, 2019. 5, 9

[52] J. Li, W. Monroe, and D. Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016. 9

[53] X. Li and F. Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *ICCV*, 2017. 8

[54] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *CVPR*, 2018. 3, 7, 8

[55] B. Y. Lim, A. K. Dey, and D. Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009. 9

[56] N. Liu, X. Huang, J. Li, and X. Hu. On interpretation of network embedding via taxonomy induction. In *KDD*, 2018. 7, 8

[57] N. Liu, X. Huang, J. Li, and X. Hu. On interpretation of network embedding via taxonomy induction. In *KDD*, 2018. 9

[58] N. Liu, Q. Tan, Y. Li, H. Yang, J. Zhou, and X. Hu. Is a single vector enough? exploring node polysemy for network embedding. In *KDD*, 2019. 8

[59] N. Liu, H. Yang, and X. Hu. Adversarial detection with model interpretation. In *KDD*, 2018. 5, 6, 9

[60] T. Lombrozo. The structure and function of explanations. *Trends in cognitive sciences*, 2006. 2

[61] J. Lu, T. Issaranon, and D. Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *ICCV*, 2017. 3, 8

[62] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *NIPS*, 2017. 8

[63] J. Ma, C. Zhou, P. Cui, H. Yang, and W. Zhu. Learning disentangled representations for recommendation. In *Advances in Neural Information Processing Systems*, 2019. 8

[64] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 3, 6

[65] B. Mathew, S. Sikdar, F. Lemmerich, and M. Strohmaier. The polar framework: Polar opposites enable interpretability of pre-trained word embeddings. In *The World Wide Web Conference*, 2020. 8

[66] D. Meng and H. Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017. 4

[67] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. On detecting adversarial perturbations. *ICLR*, 2017. 3, 8

[68] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2018. 2

[69] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In *CVPR*, 2017. 1, 3

[70] K. R. Mopuri, A. Ganeshan, and V. B. Radhakrishnan. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence*. 1, 7

[71] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019. 4

[72] M. Narayanan, E. Chen, J. He, B. Kim, S. Gershman, and F. Doshi-Velez. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1802.00682*, 2018. 9

[73] C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2(11):e7, 2017. 7

[74] A. Panigrahi, H. V. Simhadri, and C. Bhattacharyya. Word2sense: Sparse interpretable word embeddings. In *ACL*, 2019. 8

[75] N. Papernot, P. McDaniel, and I. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016. 1, 3, 5

[76] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017. 1

[77] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE. 3

[78] J. Peters, D. Janzing, and B. Schölkopf. *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017. 10

[79] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *KDD*, 2016. 5

[80] M. T. Ribeiro, S. Singh, and C. Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, 2018. 10

[81] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015. 9

[82] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 2019. 4, 6, 9

[83] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. In *NIPS*, 2017. 10

[84] S. Santurkar, A. Ilyas, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Image synthesis with a single (robust) classifier. In *NeurIPS*, 2019. 9, 10

[85] M. Sato, J. Suzuki, H. Shindo, and Y. Matsumoto. Interpretable adversarial perturbation in input embedding space for text. In *IJCAI*, 2018. 5, 8

[86] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry. Adversarially robust generalization requires more data. In *NIPS*, 2018. 6

[87] K. Schulz, L. Sixt, F. Tombari, and T. Landgraf. Restricting the flow: Information bottlenecks for attribution. In *ICLR*, 2019. 9

[88] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *CCS*, 2016. 7

[89] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 4, 7

[90] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *AAAI*, 2020. 9

[91] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 5, 9

[92] A. Subramanya, V. Pillai, and H. Pirsiavash. Fooling network interpretation in image classification. In *ICCV*, 2019. 6, 9

[93] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *ICML*, 2017. 5

[94] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 2, 3, 4

[95] G. Tao, S. Ma, Y. Liu, and X. Zhang. Attacks meet interpretability: Attribute-steered detection of adversarial samples. In *NIPS*, 2018. 3, 8

[96] S. Thys, W. Van Ranst, and T. Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *CVPR Workshops*, 2019. 1, 3, 7

[97] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 5, 6

[98] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018. 9, 10

[99] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 8

[100] C. Xie, M. Tan, B. Gong, J. Wang, A. Yuille, and Q. V. Le. Adversarial examples improve image recognition. *arXiv preprint arXiv:1911.09665*, 2019. 10

[101] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017. 3

[102] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019. 3, 7, 8, 10

[103] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017. 3

[104] F. Yang, S. K. Pentyala, S. Mohseni, M. Du, H. Yuan, R. Linder, E. D. Ragan, S. Ji, and X. B. Hu. Xfake: Explainable fake news detector with visualizations. In *WWW*, 2019. 9

[105] P. Yang, J. Chen, C.-J. Hsieh, J.-L. Wang, and M. I. Jordan. Ml-loo: Detecting adversarial examples with feature attribution. *arXiv preprint arXiv:1906.03499*, 2019. 6, 8

[106] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing. *IEEE Computational intelligenCe magazine*, 2018. 7

[107] C. Zhang, Z. Ye, Y. Wang, and Z. Yang. Detecting adversarial perturbations with saliency. In *2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP)*, pages 271–275. IEEE, 2018. 6

[108] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. E. Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019. 6

[109] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu. Interpreting cnns via decision trees. In *CVPR*, 2019. 7

[110] T. Zhang and Z. Zhu. Interpreting adversarially trained convolutional neural networks. In *International Conference on Machine Learning*, pages 7502–7511, 2019. 6, 9, 10

[111] G. Zhao, H. Fu, R. Song, T. Sakai, Z. Chen, X. Xie, and X. Qian. Personalized reason generation for explainable song recommendation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2019. 9

[112] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 4

[113] B. Zhou, Y. Sun, D. Bau, and A. Torralba. Interpretable basis decomposition for visual explanation. In *ECCV*, 2018. 7, 8

[114] L. Zhu, Z. Liu, and S. Han. Deep leakage from gradients. *arXiv preprint arXiv:1906.08935*, 2019. 10