Minds and Machines
https://doi.org/10.1007/s11023-021-09580-9

ORIGINAL RESEARCH

The Intriguing Relation Between Counterfactual Explanations and Adversarial Examples

Timo Freiesleben[1]

Received: 30 April 2021 / Accepted: 22 October 2021

Abstract
The same method that creates adversarial examples (AEs) to fool image-classifiers can be used to generate counterfactual explanations (CEs) that explain algorithmic decisions. This observation has led researchers to consider CEs as AEs by another name. We argue that the relationship to the true label and the tolerance with respect to proximity are two properties that formally distinguish CEs and AEs. Based on these arguments, we introduce CEs, AEs, and related concepts mathematically in a common framework. Furthermore, we show connections between current methods for generating CEs and AEs, and estimate that the fields will merge more and more as the number of common use-cases grows.

Keywords Counterfactual explanation · Adversarial example · XAI · AI-safety

1 Introduction

Machine Learning (ML) is transforming industry, science, and our society. Today, ML algorithms can fix a date at the hairdresser (Leviathan and Matias 2018), determine a protein's 3D shape from its amino-acid sequence (Senior et al. 2020), and even write news articles (Brown et al. 2020). Taking a sharp look at these developments, we observe a tendency towards more and more complex models. Different ML models are stacked together heuristically, with limited theoretical backing (Hutson 2018). In some applications, complexity may not be an issue as long as the algorithm performs well most of the time. However, in socially, epistemically, or safety-critical domains, complexity can rule out ML solutions—think of e.g. autonomous driving, scientific discovery, or criminal justice. Two of the major drawbacks of highly complex algorithms are the *opaqueness problem* (Lipton 2018) and *adversarial attacks* (Szegedy et al. 2014).

✉ Timo Freiesleben
timo.freiesleben@campus.lmu.de

[1] Munich Center for Mathematical Philosophy, Ludwig-Maximilians-Universität, Ludwigstrasse 31, Munich, Germany

Published online: 30 October 2021

The opaqueness problem describes the limited epistemic access humans have to the inner workings of ML algorithms, especially concerning the semantic interpretation of parameters, the learning process, and the human-predictability of ML decisions (Burrell 2016). This lack of interpretability has gained a lot of attention recently, which gave rise to the field eXplainable Artificial Intelligence (XAI; Doshi-Velez and Kim 2017; Rudin 2019). Many techniques have been proposed to gain insights into ML systems (Adadi and Berrada 2018; Došilović et al. 2018; Das and Rad 2020). Especially model-agnostic methods have gained attraction since, unlike model-specific methods, their application is not restricted to a specific model type (Molnar 2019). Global model-agnostic interpretation techniques like Permutation Feature Importance (Fisher et al. 2019) or Partial Dependence Plots (Friedman et al. 1991) aim at understanding the general properties of ML algorithms. On the other side, local model-agnostic interpretation methods like LIME (Ribeiro et al. 2016) or Shapley Values (Štrumbelj and Kononenko 2014) aim at understanding the behavior of algorithms for particular regions. One way to explain a specific model-prediction is a Counterfactual Explanation (CE; Wachter et al. 2017). A CE explains a prediction by presenting a maximally close alternative input that would have resulted in a different (usually desired) prediction. CEs are the first class of objects we study in this paper.

The problem of adversarial attacks describes the fact that complex ML algorithms are vulnerable to deceptions (Papernot et al. 2016a; Goodfellow et al. 2015; Szegedy et al. 2014). Such malfunctions can be exploited by attackers to e.g. harm model-employers or endanger end-users (Song et al. 2018). The field that investigates adversarial attacks is called adversarial ML (Joseph et al. 2018). If the attack happens during the training process by inserting mislabeled training data, the attack is called poisoning. If an attack happens after the training process, it is commonly called an adversarial example (AE; Serban et al. 2020). AEs are inputs that resemble real data but are misclassified by a trained ML model, e.g., the image of a turtle is classified as a riffle (Athalye et al. 2018). Hence, misclassified means here that the algorithm assigns the wrong class/value compared to some (usually human-given) ground-truth (Elsayed et al. 2018). AEs are the second class of objects relevant to our study.

Even though the opaqueness problem and the problem of adversarial attacks seem unrelated at first sight, there are good reasons to study them jointly. AEs show where an ML model fails, and examining these failures deepens our understanding of the model (Tomsett et al. 2018; Dong et al. 2017). Explanations on the other hand can shed light on how ML algorithms can be improved to make them more robust against AEs (Molnar 2019). As a downside, explanations may enclose too much information about the model, thereby allowing AEs to be constructed and the model attacked (Ignatiev et al. 2019; Sokol and Flach 2019). CEs are even stronger connected to AEs than other explanations. CEs and AEs can be obtained by solving the same optimization problem[1] (Wachter et al. 2017; Szegedy et al. 2014):

[1] x describes the original input, x' the counterfactual/adversarial vector, f the ML model, y_{des} the desired classification, $d(\cdot,\cdot)$ and $d'(\cdot,\cdot)$ distances, and λ a trade-off scalar. For details, see Sect. 4.3.

$$\underset{x' \in X}{\operatorname{argmin}}\ d(x, x') + \lambda\ d'(f(x'), y_{des}). \tag{1}$$

Term 1 has led to various confusions concerning the relationship between CEs and AEs in the research community.[2] We aim to resolve them and give a detailed analysis of the relationship between the two fields.

The aim of the present paper is twofold. Our first goal is the *clarification of concepts*. Commonly used concepts such as CE/AE, flipping/misclassifying, process/model-level, and closeness/distance are often misunderstood or not clearly defined. We define these terms properly in one mathematical framework, aiming for more clarity and unification. The second goal is to *familiarize researchers* of each of the respective fields *with its neighboring area*. Even in one of the fields, it is hard to keep track of developments and new ideas, in both it is worse. Since there are many ways in which each of the fields can profit from the other, both methodologically and conceptually, we aim to provide a guide connecting the two literatures.

We will start by providing an intuition to the reader with two standard use cases of CEs/AEs and give an overview of relevant other applications in Sect. 2. In Sect. 3, we present the (historical) background of CEs and AEs, including the current debate around their relationship. Next, we present arguments in what sense the current understanding of the relation between CEs and AEs is flawed in Sect. 4.1. In Sect. 4.2, we will argue that the notions of misclassification and maximal proximity are the central properties that distinguish CEs from AEs. Based on that, we introduce in Sect. 4.3 our more fine-grained formal definitions of CEs, AEs, and related concepts. In Sect. 5, we discuss connections between the solution approaches for finding CEs/AEs in the literature. We conclude in Sect. 6 by discussing the relevance and limitations of our work.

2 Examples and Use Cases

Before we get into the technical and conceptual details, let us look at two use cases where both CEs and AEs have been successfully deployed. This provides an intuition to the reader and will moreover serve explanatory purposes in the later sections. The first example is among the most prominent use-cases of CEs, automated lending. The second example shows one prominent use-case of AEs, image-classification of hand-written digits.

Loan Application imagine a scenario where person P wants to obtain a loan and applies for it through a bank's online portal. She has to enter several of her properties into the user-interface e.g. her age, salary, capital, number of open loans, and number of pets. The portal uses an automated, algorithmic decision system, which decides that P will not receive the loan. However, she would have liked to obtain it and therefore demands an explanation. An example of a potential CE would be:

[2] We discuss these confusions in more detail in Sect. 4.1.

> If P had a 5, 000 € p.a. higher salary and an outstanding loan less, her loan application would have been accepted.

She can use this information to guide her future actions or potentially to contest the algorithmic decision. Clearly, CEs are not restricted to that setting. If P were the model engineer instead of the customer, she could also use the explanation to raise her understanding of the model or to debug it.

Now, suppose that P wants to trick the system to get the credit. Assume the decision system was constructed from an ML algorithm, trained on historic data of the companies loan admission policy. From the data, the algorithm has learned that the number of pets is positively associated with repaying the credit and consequently the system uses the information in its decision making.[3] One potential way to trick the system with an AE in such a case could for example look as follows:

> P indicates two more pets on the application form than she actually has to obtain the loan.

P has changed a feature that the model deems causally relevant for creditworthiness but which is only spuriously correlated, thus, P has tricked the model. Moreover, she probably does not even have to prove the feature to the bank as there is often no official legal document for the ownership of e.g. fish or birds. This change allowed her to obtain the loan, even though none of her properties have changed.

Hand-Written Digits Recognition imagine a simplified scenario in which a postal service employs an image recognition algorithm. This algorithm takes as input gray-scale 28×28 pixel images and assigns them the number between 0 and 9 they depict. This procedure eases the work of the postal service a lot. Cases of errors are rare but costly, as the postal service must pay the sender 5€ if a letter or package is sent to the wrong address. Therefore, the postal service is interested in improving the algorithm.

One way of improving the system would be to generate CEs for specific instances, evaluate how useful they are, and adjust the algorithm. Such CEs can be found in the first two columns of Fig. 1. One can see e.g. that the images in the first row show that the algorithm assigns major importance to the lower-left line to distinguish between a six and a five. The postal service might derive that the algorithm already has a robust understanding of digits.

Now, assume we take the perspective of an attacker who is interested in exploiting the 5€ per error system. Such an attacker will be interested in generating AEs, put them on letters/parcels and gain money. Examples of such AEs are presented in the last column of Fig. 1. One can see e.g. that the system has problems when random dots appear around a 0 and misclassifies the input as the number 5. While the attacker will aim to accomplish many successful attacks, the postal service will

[3] Reasons for such an association in the data might be that pets are expensive and hence associated with capital/salary or that people with more pets have also kids and are therefore more reliable. The example is inspired by Ballet et al. (2019).

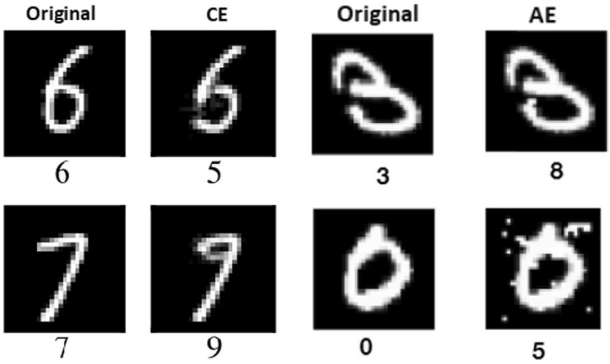

Fig. 1 The images are taken from Van Looveren and Klaise (2019) and Papernot et al. (2017). They are generated from CNNs trained on the MNIST dataset. The first an the third column depict original images from the MNIST dataset. Column two depicts the corresponding CEs and column four shows the corresponding AEs

try to limit the deceivability of its algorithm by making it more robust or excluding unrealistic outliers in classification.

The Relevance of These Use-Cases loan applications are among the most popular example use-cases in the CE literature (Wachter et al. 2017; Dandl et al. 2020; Grath et al. 2018). The example is particularly valuable as it describes a technically and ethically complex decision situation, in which explanations are a requirement. Interestingly, the lending use-case gains more and more interest also in the AE literature since it depicts the safety troubles of ML systems. Ballet et al. (2019) introduce a new notion of imperceptibility for these scenarios which got quickly picked up by others (Cartella et al. 2021; Hashemi and Fathi 2020).

Hand-Written-Digits classification is the classical use-case among all image-classification tasks. Many methods to generate AEs discuss it at least as a test case (Wang et al. 2019; Szegedy et al. 2014; Papernot et al. 2017). The feature space is comparatively small and the problem itself well studied, therefore, generating AEs is computationally cheap and conceptually informative. However, security threats cannot be as easily depicted from this use case (that is why we created the fictional scenario from above). Because of its simplicity, it has also been used as a starting point in the CE literature. The difficulty lies here in finding semantically meaningful notions of similarity for images. Three papers proposed approaches to that problem, Van Looveren and Klaise (2019) use prototypes to generate realistic CEs, Poyiadzi et al. (2020) use allowed paths, and Goyal et al. (2019) use differently classified images to identify regions that shift the classification.

Other Use Cases there are common use-cases for CEs other than loan approval, such as university applications, diabetes diagnosis (Wachter et al. 2017), adult-income prediction (Mothilal et al. 2020), or predicting student performances in law-school (Russell 2019). Most of the common use-cases focus on tabular data settings, as it is easier to make sense of CEs in these scenarios (Verma et al. 2020). Changes in semantically meaningful variables are easy to convey. Moreover, the scenarios considered often describe high-stakes decisions with an ethical dimension. There

are few non-classification, non-tabular settings in which CEs have been applied, such as image recognition (Goyal et al. 2019; Van Looveren and Klaise 2019), NLP-tasks (Akula et al. 2019), regression problems (Anjomshoae et al. 2019) and non-supervised learning settings (Olson et al. 2021).

The AE community on the other hand has largely focused on image classification tasks (Serban et al. 2020). Many AEs focus particularly on the state-of-the-art image classifiers from Google, Amazon, or Facebook (Serban et al. 2020). Well-known examples include AEs on road signs (Eykholt et al. 2018), the 3-D print of a turtle classified as a rifle (Athalye et al. 2018), and the adversarial patch, a sticker that fools image recognition software into classifying it as a toaster (Brown et al. 2017). One reason why image classifiers lie at the center of the study of AEs is that the imperceptibility of changes and the true class label are easy to define (Ballet et al. 2019). Moreover, since image recognition models focus on models like CNNs, AEs help to assess the limitations of opaque deep learning algorithms. However, there is also work on AEs in other task environments e.g. audio/video-classification (Carlini et al. 2016; Carlini and Wagner 2018; Wei et al. 2018), regression problems (Balda et al. 2019), and non-supervised learning settings (Behzadan and Munir 2017; Huang et al. 2017).

3 Background on CEs and AEs

This section provides a background on where CEs and AEs have historically come from, discusses their roles in ML, and presents the discussions about the relationship between the two. The historic background and roles of CEs/AEs provide the basis for understanding the discussions around the relationship between the two fields, which motivate our proposal.

3.1 Historic Background

History of CEs CEs have their roots in Philosophy as so-called *subjunctive counterfactual conditionals*. They describe conditionals of the form

$$\text{If } S \text{ was the case } Q \text{ would have been the case,} \tag{2}$$

where S and Q are events. Importantly, event S did not in fact occur. The truth-condition for conditional 2 is hotly debated in philosophy until today (Starr 2019). The approach that was taken up by the XAI community (Wachter et al. 2017) builds on the work of Lewis (1973) and Stalnaker (1968). In their framework, conditional 2 holds if and only if the closest possible world[4] $\omega' \in \Omega$ to the actual world $\omega \in \Omega$ in which S is the case[5] also Q is the case. The notion of similarity between possible worlds is critical in assessing a counterfactual conditional and Lewis discusses

[4] Ω denotes the set of possible worlds.

[5] S is false in ω.

similarity in more detail in Lewis (1979). He argues that between close worlds laws of nature must be preserved, widespread, diverse violations should be avoided, and facts stay congruent for maximal time. Particular facts on the other side can be changed without significantly increasing dissimilarity. Despite these specifications, Lewis himself admits that the under-specified notion of similarity between possible worlds remains the crucial weak-spot of his framework (Lewis 1983).

It is very important to keep in mind that Lewis aimed to describe causal dependence via counterfactual conditionals (Menzies and Beebee 2019). The idea is that Q' causally depends on S' if and only if, if S were not the case Q would not have been the case.[6] Even though CEs are not necessarily causal (Reutlinger 2018), the connection to causality is the main factor that underlies the explanatory force of CEs in XAI. We can see a textual CE in XAI as a true counterfactual conditional in which the antecedent describes a change in input features and the consequent a corresponding change in the classification.

Research on CEs in Psychology concerning human-to-human interaction is another root and inspiration of the discussion in XAI (Byrne 2016; Miller 2019). Humans use CEs in their daily life when they explain behavior or phenomena to each other, often in the form of a contrastive explanation highlighting the differences to the real scenario. Byrne (2019) summarized the central findings on CEs in Psychology and evaluates their relevance to XAI. She points out that people tend to create CEs that: add information rather than delete, show better rather than worse outcomes, identify relevant cause–effect relationships, and change antecedents that are exceptional, controllable, action-based, recent, and not highly improbable.

Using Lewis's account of counterfactuals for generating explanations for the decisions of ML algorithms was first proposed by Wachter et al. (2017) who also drew the connection to the philosophical/psychological tradition of CEs. They argue that CEs have three intuitive functions: *raise understanding*, *give guidance for future actions*, and *allow to contest decisions*.[7] Also, they highlighted the legal relevance of CEs and argued that they satisfy the requirements proposed in the so-called 'right to explanation' as it is defined in Recital 71 of the European General Data Protection Regulation (GDPR). This law guarantees European citizens the right to obtain an explanation in cases they are subject to the fully automated decision-making of an algorithm (Voigt and Von dem Bussche 2017).

History of AEs AEs have a less rich philosophical tradition, but instead a strong history in the robustness and reliability literature in computer science (Joseph et al. 2018). Fernandez et al. (2005) describes robustness as "the ability of a software to keep an 'acceptable' behavior [...] in spite of exceptional or unforeseen execution conditions." The reliability and robustness of computer systems have always been

[6] Interestingly, Pearl (2009) turns this story around and defines counterfactuals via causal graphs. Instead of comparing similar worlds, he directly focuses on the underlying mechanisms defined by a structural equation. However, as Woodward (2002) and Hitchcock (2001) pointed out that is a matter of interpretation as we can instead also understand Pearl's structural equations as sets of primitive counterfactuals. Also, Pearl's notion has found its way into the XAI literature in the form of algorithmic recourse (Karimi et al. 2020c, b).

[7] It is not necessarily the case that all of these functions are or can be satisfied by one CE (Russell 2019).

major concerns, especially in safety-critical applications such as health or the military sector. Critical elements can be the human interactors, hardware (e.g. sensors, hard drives, or processors), and the software. All kind of software is prone to erroneous behavior (Kizza et al. 2013), however, adversarial ML focuses particularly on the robustness of ML software.

For classical 'rule-based' software, the robustness can often be tested by formal verification (D'silva et al. 2008). This becomes more difficult if systems interact dynamically with their environment or learn from data. Statistical Learning Theory tries to extend the idea of formal verification to statistical learning methods and gives theoretical guarantees for the performance of specific model-classes (Vapnik 2013). Unfortunately, good guarantees become unattainable for very broad and powerful model-classes such as for Deep Neural Networks and learning procedures like Stochastic Gradient Descent (Goodfellow et al. 2016). What is special about the robustness of complex ML algorithms compared to others is that they are vulnerable to attacks even if common errors in model-selection have been avoided (Bishop 2006; Claeskens et al. 2008; Good and Hardin 2012). Moreover, the kind of attacks they are vulnerable to is highly unexpected, which even has led to the question of whether they learn anything meaningful at all (Szegedy et al. 2014). The study of adversarial ML is not restricted to Deep Learning but also applies to classical ML models e.g. logistic regression (Dalvi et al. 2004).

The research in adversarial ML focuses on attacks on ML models by manipulated inputs and the defenses against such attacks. An AE describes an input to a model that is deliberately designed to effectively "fool" the model into misclassifying[8] it. AEs occur even for ML algorithms with strong performances in testing-conditions. Since the changes from the original to the adversarial input are mostly *imperceptible to humans*, AEs have been compared to optical illusions tailored to ML models (Elsayed et al. 2018).

Szegedy et al. (2014) and Goodfellow et al. (2015) contributed milestones in the literature on AEs by not only providing ways to generate AEs but also attempting to explain their existence. Szegedy et al. (2014) argued that AEs live mainly in spaces of low probability in the data-manifold. Therefore, they do not appear in either the training or the test dataset. Hence, artificial neural networks (ANNs) can have a low generalization error despite the existence of AEs. Goodfellow et al. (2015) refuse this thesis and argue that AEs arise instead due to the linearity of many ML models including ANNs with semi-linear activations. Tanay and Griffin (2016) disagree and show that linearity is neither sufficient nor necessary to explain AEs. Instead, they claim that AEs lie slightly outside the real-data distribution close to tilted decision boundaries. They argue that the decision boundary is continuous outside the data-manifold and can therefore easily be crossed by AEs. A radically different view is proposed by Ilyas et al. (2019) who show that AEs arise from highly predictive but non-robust features present in the training data. Hence, AEs are a human-centered

[8] From now on, we will mainly talk about misclassification and classifying. However, this is only to simplify our language usage. AEs are not restricted to classification tasks but also work on regression problems.

phenomenon, the ML models, however, just rely on useful information in the data humans do not use.[9]

3.2 Role in ML

Due to the theoretical foundation, practical applicability, and legal significance, the CE approach was quickly adopted by the XAI community as one method to explain individual predictions of ML models to end-users (Verma et al. 2020). Nevertheless, the method remains controversial and has often been accused of giving misleading explanations (Laugel et al. 2019a; Barocas et al. 2020; Páez 2019).

The trust we have in AI systems is and will be closely linked to the extent to which adversarial attacks are possible (Toreini et al. 2020). On the negative side, AEs can cause severe damage and security threats (Eykholt et al. 2018). On the positive side, AEs can help us understand how the algorithm works (Ignatiev et al. 2019; Tomsett et al. 2018) and therefore to understand what it has actually learned (Lu et al. 2017a). AEs can even concretely improve models (Bekoulis et al. 2018; Stutz et al. 2019).

Both CEs and AEs play a great role in the ML landscape, namely for the trust people have in ML (Shin 2021; Toreini et al. 2020). CEs and AEs contribute to improving model understanding, identifying biases, and even offer methods to eliminate these biases through adversarial/counterfactual-training (Bekoulis et al. 2018; Sharma et al. 2020). However, while improving understanding and highlighting algorithmic problems is usually only a byproduct of AEs, it is the focus of CEs. The deception of a system, on the other hand, is essential for AEs, but a potential byproduct of CEs in cases where they disclose too much information about the algorithm (Sokol and Flach 2019).

3.3 The Relation Between CEs and AEs

As mentioned in Sect. 1, CEs and AEs derive from solutions to the same optimization problem 1. While the close mathematical relationship between CEs and AEs has been frequently pointed out, their exact relationship remains controversial and there are a variety of opinions on the matter we present here in more detail.

In one of the early papers on CEs, Wachter et al. (2017) note that an AE can be described as "a counterfactual by a different name" (Wachter et al. 2017, p. 852). They see one difference between counterfactuals and adversarials in the applied notion of distance arising from the misaligned aims, e.g. sparsity vs. imperceptibility. The other difference they argue for is that while counterfactuals ought to

[9] Since it is extremely controversial why AEs exist, it is also hard to defend a system against them. It is even difficult to formulate the desired property an ML model should have concerning AEs (Bastani et al. 2016; Biggio and Roli 2018). Classical verification methods have to be modified because they explode computationally in the high-dimensional input spaces we are dealing with in ML. Since defense techniques are not relevant for CEs, we will not discuss them in the present paper. We advise the interested reader to Serban et al. (2020).

describe closest possible worlds, AEs often result from 'impossible worlds' in the Lewisian sense i.e. unrealistic data-points. Additionally, they hint at methodological synergies between the two approaches, especially with respect to optimization techniques.

Browne and Swift (2020) reject the two difference makers between CEs and AEs highlighted by Wachter et al. (2017) (distance metrics, possibility of worlds) as not definitional. They argue that using the "wrong" notion of distance may favor less relevant counterfactuals, but these are still ultimately potential explanations. Moreover, they reject the claim that adversarials must describe impossible worlds by pointing out that adversarial attacks can be carried out in real-world settings. Instead, they view counterfactuals and adversarials as formally equivalent. They argue that the key difference between CEs and AEs is not mathematical, but relies on the semantic properties of the input space. They point out that: "Mathematically speaking, there is no difference between a vector of pixel values and a vector of semantically rich features" (Browne and Swift 2020, p. 6). They highlight the role of semantics in human-to-human explanation and claim that this difference makes CEs for image-data adversarials as AEs cannot be conveyed to an explainee in human-understandable terms.

Verma et al. (2020) see the terms CE and AE as non-interchangeable due to the different desiderata they must account for. They highlight tensions between the adversarial desideratum of imperceptibility and counterfactual desiderata like sparsity, closeness to the data-manifold, and actionability. According to Grath et al. (2018) CEs and AEs are similar as both are example-based approaches. They describe the distinction between CEs and AEs as the difference between flipping and explaining decisions. They remark that CEs inform about the changes, while AEs aim at hiding those. Laugel et al. (2019b) agree that the two concepts show strong mathematical similarities. However, they also point to the difference in purpose and application. They note that CEs are mainly considered in the context of low-dimensional tabular data scenarios, whereas AEs are considered in less-structured domains like image/audio data. Dandl et al. (2020) and Molnar (2019) describe AEs as special CEs with the aim of deception. Sokol and Flach (2019) discuss CEs in the context of AI safety. They make the case that CEs can disclose too much information about the model and thereby lead to AEs.

4 Defining Concepts

This section consists of three parts: (1) a critical assessment of the accounts from Sect. 3.3; (2) our conceptual proposal; (3) our formal proposal. In the first part, we will argue why none of the afore-mentioned accounts can properly explain the difference between CEs and AEs. As we will point out, one problem is that they focus on the optimization problem 1 as the defining mathematical term for CEs/AEs. Instead, we will explain why solving Eq. 1 leads to counterfactuals in tabular settings and adversarials in the image-domain. Moreover, we propose that the relation of the counterfactual/adversarial to the true label and the proximity to the original data-point present the definitional distinction between CEs and AEs. Since these

two distinguishing properties are not captured by Eq. 1 we will consequently present novel mathematical definitions of CEs/AEs in part three.

In our arguments, we assume that the reader is familiar with the ideas behind *decision boundaries, data manifolds, meaningless/unrealistic/unseen inputs, and distance metrics*. For readers who are not familiar with these concepts, we have provided a short glossary in Appendix A where we explain these concepts with an illustrative example.

4.1 Conceptual Discussion of Other Accounts

Two Names for the Same Objects Taking the optimization problem from Eq. 1 as definitional, Wachter et al. (2017) and Browne and Swift (2020) conclude that they are the same mathematical objects. To evaluate this claim, imagine a model, e.g. an image classifier that, for all inputs for which a ground truth exists, assigns exactly this ground truth. Now, consider a particular prediction of this perfect algorithm. Via solving the optimization problem in Eq. 1 we can generate counterfactuals. The CEs would be pointing to another input that receives a different assignment e.g. instead of the original image of a 3, it shows a 9 looking similar to that 3. However, the system cannot be fooled by a modified image because it is always correct. Therefore, no AEs exist in that case and none of the generated counterfactuals is an AE. The case of a perfect algorithm shows that there are models for which we can reasonably generate CEs but no AEs. Consequently, they cannot generally be the same objects with different names. This shows that while there may be some cases where a vector can be called both counterfactual and adversarial, there must be a definitional difference between the two concepts.

*The Two Differ in Aims*Verma et al. (2020) point out that the terms are not interchangeable because "while the optimization problem is similar to the one posed in counterfactual-generation, the desiderata are different" (Verma et al. 2020, p.4). By desiderata they mean additional requirements that are enforced on adversarials (like imperceptibility) or counterfactuals (e.g. sparsity, closeness to the data-manifold and feasibility. See also Sect. 5). These different desiderata are realized in the different distance metrics applied. This difference in aims corresponds to what Wachter et al. (2017) mean by claiming that AEs are not making use of appropriate distance metrics. So even though counterfactuals and adversarials share the same formal definition, they can be distinguished by their notion of distance i.e. the applied metric.

We agree with Browne and Swift (2020) that the applied distances do not indicate a definitional difference between CEs/AEs. We contend that whether the desiderata overlap or not, depends on the respective aims the user has with a CE/AE. Agents might also be interested in generating CEs to get guidance on how to deceive the system (Sokol and Flach 2019). In such cases, imperceptibility will indeed be relevant, while sparsity or closeness to the data-manifold will be less relevant. Moreover, attackers could be interested in creating realistic AEs because they are harder to detect. In such scenarios, closeness to the data-manifold or feasibility constraints are desirable properties of AEs. Also, both CEs and AEs can be relevant to better understand the model at hand and to improve it.

If the desiderata are similar, so is the mathematical approach. In such scenarios, good counterfactuals and adversarials may actually align and describe the same objects. However, a proper definitional distinction between concepts should be universal, objective, and independent of the agent's intentions. It requires necessary (and sufficient) criteria that make an object an instantiation of one object-class rather than another. The various desiderata are insufficient to account for differences between counterfactuals and adversarials in this strong sense.

*Flipping and Explaining*Grath et al. (2018) draws the distinction between CEs and AEs as the difference between explaining and flipping a decision. While CEs point to changes in a meaningful way, AEs try to hide those. We think that this is a solid observation, however, it shows a difference in presentation and not in definition. If the presentation style would be the whole difference, we would agree that CEs and AEs could mathematically be described as the same objects by a different name.

*Low vs. High-Dimensional Use Cases*Laugel et al. (2019b), Wachter et al. (2017), and Browne and Swift (2020) highlight the difference in use-cases. They argue, that while for CEs mainly low-dimensional and semantically meaningful features are used, AEs are mostly considered for high-dimensional image data with little semantic meaning of individual features. Therefore, the difference is not a difference of mathematical objects but rather a difference of semantic structure of the input space provided to generate an explanation/attack. In that sense, an AE is a CE that points to semantically non-interpretable factors.

However, as discussed in Sect. 2 the use-cases are increasingly overlapping. So, if Browne and Swift (2020) would be right that the provided semantics in the input spaces is the crucial difference, authors studying AEs in low-dimensional setups would just directly use the approaches from the CE literature instead of developing new methods. According to their argumentation, the two approaches should be equivalent for low-dimensional setups. But, what we can notice is that e.g. Ballet et al. (2019) uses expert knowledge to generate imperceptible AEs for structured data by asking for features they find irrelevant for the decision at hand. Moreover, Goyal et al. (2019) and Poyiadzi et al. (2020) manage to give, as it seems, meaningful CEs also for high-dimensional input spaces without making use of higher-level semantic concepts the model creates while Browne and Swift (2020) thought this is inevitable. These examples show that the semantic structure of the input space cannot account for a definitional distinction. Nevertheless, we agree that the difference between CEs and AEs is semantic in nature.

4.2 Our Proposal

After our critical assessment, we found that all approaches so far have failed to show definitional differences between counterfactuals and adversarials. This is not surprising bearing in mind that all of them take Eq. 1 as definitional for CEs and AEs. If one starts with the same definition for both approaches, one can either claim that counterfactuals and adversarials are identical or point to the elements within the optimization problem that differ such as the applied distances (i.e. the aims) or

the structure of the input space. However, just because two object classes contain solutions to the same optimization problem, does not mean that they are identical.[10] We propose two definitional differences between CEs and AEs that have so far been overseen. Moreover, we argue why nevertheless Eq. 1 can generate both CEs and AEs in different contexts.

Misclassification one obvious distinction that has largely been overseen by researchers is that adversarials must be necessarily misclassified while counterfactuals are agnostic in that respect. A correctly classified counterfactual is acceptable and often even desirable. On the other hand, if an adversarial were correctly classified, no one would call it an adversarial as it would provide no means to attack a target system. Consequently, misclassification is a necessary condition that any object called an adversarial must meet. This is different from the desiderata discussed above, which depend only on the goals of the agent with a CE or AE. Misclassification as a definitional distinction has been overseen since CEs and AEs can be generated by solving the same optimization problem 1. How can it be that the same optimization problem is used to generate CEs for tabular-data models and AEs for image-data models? This is the crucial question that has to be assessed. It is strongly connected to the riddle the existence of AEs poses as discussed in Sect. 3.1, therefore, our analysis bases on the ideas of Szegedy et al. (2014) and Tanay and Griffin (2016).

We must look at image-classification models to answer why solutions to Eq. 1 are mostly misclassified in that scenario. Complex image classifiers perform reasonably well on training data and highly similar inputs. In "unseen regions", on the other hand, they have to extrapolate and therefore perform worse. Since the input space is incredibly high-dimensional, the training data and therefore the data-manifold the algorithm approximates is comparably tiny. That means, there are many more meaningless, unrealistic, and unseen inputs than there are points in the training-data. The assignment of these inputs is not trustworthy and does not necessarily match the assignment of other nearby inputs. At the same time, there is usually a strongly limited number of classes that inputs are assigned to. Moreover, the training-data assigned to different classes have great distances. Hence, if we search for an input from another class but close to a given input, the probability is high that it is an input the algorithm has not seen, is unrealistic, or is meaningless and therefore where the algorithm is not reliable. Thus, the model will with high probability misclassify this input. Often these close inputs are neither unrealistic nor meaningless as thought by Wachter et al. (2017), but realistic. Completely unrealistic or meaningless inputs are at greater distance from the original input. Realistic but unseen data-points make up the dangerous AEs.

This explains why misclassified adversarials are generated in input spaces with high-dimensionality and little structure. The effect is even stronger if distances are applied that do not reflect what humans consider to be close inputs in the high-dimensional case. Minimal changes according to conceptually less-justified

[10] For example, both local maxima and minima minimize the absolute derivative of a differentiable function. Nevertheless, the two object classes can be formally distinguished.

distances break the dependencies between variables present in the real world and therefore search for inputs in regions with less training-data support. This line of thought might suggest that the main reason why mostly adversarials are obtained by Eq. 1 for image-classification is the use of distance metrics with little conceptual justification. Whether the right distance metric would yield fewer adversarials is, in our opinion, an empirical question that we cannot settle here. However, we will present our thoughts on this in Sect. 6.2.

There are several reasons why counterfactuals generated in structured, low-dimensional input spaces are not generally adversarials. First, the models are often more robust and extrapolate better in unseen regions, also because background knowledge can more easily enter the model. Second, the real-world variables have a much simpler dependence structure compared to the high-dimensional image-data case. Additionally, as distances are chosen that favor sparse rather than distributed changes, these dependencies are often preserved by the manipulations to the input vectors. Third, often additional constraints are added that make sure that the generated input stays close/within the data-manifold i.e. in regions where the model performs well (further discussions of these constraints can be found in Sect. 5.2).

Summed up, both counterfactuals and adversarials can be generated using the same method. However, that does not entail that they describe the same object class. Counterfactuals are agnostic with respect to the true label, whereas adversarials must be misclassified. From this perspective, counterfactuals could be considered the more general object-class. However, this conclusion would be drawn too early, since there is a second definitional difference.

Proximity to the Original Input additionally to misclassification, we want to highlight a second, minor distinction between counterfactuals and adversarials, which is their tolerance with respect to proximity to the original input.

Closeness to the original input is usually a benefit for adversarials to make them less perceptible. However, an adversarial can still be used to attack a system if it is a little bit more distal to x than another adversarial (Goodfellow et al. 2015). Dependent on the aim of the attacker, this might even be desirable. Adversarials with greater distance to the decision boundary transfer better between different models, are often more effective, or more meaningful (Zhang et al. 2019; Elsayed et al. 2018).

For counterfactuals on the other side, closeness to the original input plays a significant role in the causal interpretation as discussed in Sect. 3.1. Without maximal closeness, a counterfactual shows only a sufficient scenario for a different classification but not a necessary one. For example, assume we are in the loan-application setting from Sect. 2, where one point describes a maximally close counterfactual and the other a relatively close alternative input to x, both assigned to the same class. Assume moreover that the only difference between them is a change in gender from female to male. Then, even though such a change in gender would not impact the model-prediction, it would appear as a cause for the explainee receiving the alternative input. Such alternative inputs are less valuable than actual counterfactuals not only to data-subjects but also for model-developers examining the model. Thus, accepting 'close enough' but not maximally close inputs with a different classification as counterfactuals means either ignoring better CEs or admitting that the used distance is not perfectly adjusted for relevance in the given context.

Despite that difference in their tolerance with regards to proximity, we do not see this difference as equally essential as misclassification. If closeness is handled more loosely to generate "CEs", we might not gain real CEs, but still possibly relevant explanations. Thus, we do not entirely leave the category of objects. If on the other side we generate correctly classified inputs, we left the realm of attacks.

4.3 Our Definitions

As we argued, we must not take Eq. 1 as definitional for CEs/AEs. To account for the definitional differences we proposed in Sect. 4.2, we require novel definitions that include misclassification and the tolerance with respect to proximity to the original input. The definitions we will offer satisfy these requirements, offer useful conceptual extensions, and are grounded in the recent literature (e.g. Verma et al. 2020; Stepin et al. 2021 for CEs and Szegedy et al. 2014; Serban et al. 2020 for AEs). We try to be maximally inclusive to the usage of the terms in the general literature, however, due to the great number of papers on both fields (Yuan et al. 2019; Verma et al. 2020; Serban et al. 2020; Stepin et al. 2021) our framework will probably not be able to cover all usages.

Before we can define CEs and AEs, we need to know what we aim to explain or attack, namely ML models or the processes in which they are employed. We will restrict ourselves here to the highly common supervised learning setup. Moreover, we will focus on classification tasks. These restrictions have mainly the purpose to keep the analysis accessible. Many notions can be easily extended to other learning-paradigms.

Machine Learning Algorithms and Models assume we consider the relation of variables $\mathcal{X} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ and a (often one-dimensional) variable $\mathcal{Y}$. We can see these variables as random variables standing in a causal relation to each other. Let X and Y denote the co-domain of $\mathcal{X}$ respectively $\mathcal{Y}$. A (supervised) *ML algorithm* $\boldsymbol{\Phi}$ is a procedure that based on a set of models $\mathcal{M}$, a labeled training dataset $\mathcal{D}_{Tr} := \{(x^1, y^1), \ldots, (x^n, y^n)\}$ with $n \in \mathbb{N}$, some hyperparameters $\mathcal{H}$, an optimization method $\mathcal{O}$, and a loss function $\mathcal{L}$ outputs a model $f \in \mathcal{M}$. This procedure $\boldsymbol{\Phi}$ intuitively speaking searches for a model f in the set $\mathcal{M}$, using method $\mathcal{O}$ and hyperparameters $\mathcal{H}$, that has a low prediction loss $\mathcal{L}$ on the training dataset $\mathcal{D}_{Tr}$.

The model $f \in \mathcal{M}$ that is obtained by running the procedure $\boldsymbol{\Phi}$ on a given input is called the *machine learning model*. It can be described as a function $f : X \to Y$. This model ideally has a low bias measured by the loss function on the training dataset $\mathcal{D}_{Tr}$ and, moreover, a low generalization error on an unseen test dataset $\mathcal{D}_{Te} := \{(x^{n+1}, y^{n+1}), \ldots, (x^l, y^l)\}$ with $l > n$. That means that f does predict values of $\mathcal{Y}$ from $\mathcal{X}$ in cases it has seen the correct assignment, but also for cases that have not been part of the training dataset $\mathcal{D}_{Tr}$.

Counterfactuals and Adversarials unlike other authors, we distinguish between the mathematical objects that induce a CE/AE and the explanations/examples themselves. First, we will define the mathematical objects. For all the following

definitions, assume we consider a fixed ML model f, a particular vector[11] $x \in X$ that is mapped by f to a value $f(x) \in Y$, and a semi-metric[12] $d(\cdot, \cdot)$ on space X.

Definition We call $x' \in X$ an *alternative* to x if $f(x') \neq f(x)$.

In simple terms, x' is an alternative to x if it gets a different assignment by f.

Definition Let $\epsilon > 0$. We call x'_ϵ an *ϵ-alternative* to x if

$$d(x'_\epsilon, x) < \epsilon \text{ and } x'_\epsilon \text{ is an alternative to } x.$$

We can think of x'_ϵ as a step away from x for which we cross a decision boundary of the model but stay within a local ϵ-environment around x.

Definition We call $c_x \in X$ a *counterfactual* to x if

$$d(c_x, x) \text{ is minimal subject to } f(c_x) \neq f(x).$$

Staying in the narrative, a counterfactual describes the shortest[13] step that crosses a decision boundary. Notice that this closest vector does not have to be unique, there might exist a variety of vectors in equal distance.

A *true label* $y_{x',true} \in Y$ for a vector $x' \in X$ describes the objectively correct label that the input-vector x' should be assigned to. This ground-truth is often given by expert human evaluation. Not for all inputs there exists such a true label. The reason might be that the correct assignment is controversial even among expert evaluators or the considered input is unrealistic. Why are such unrealistic inputs relevant? As introduced above, image$(\mathcal{X}) \subseteq X$. That means that in cases where the subset-relation is strict, our model f is defined on data-points that do not realistically occur in the real world.

Definition We call a vector $x' \in X$ *misclassified* if $f(x') \neq y_{x',true}$.

A misclassification describes a mistake made by the algorithm relative to an expert-human assignment.

Definition Let $\epsilon > 0$. We call $a_{x,\epsilon} \in X$ an *adversarial* to x if

$$a_{x,\epsilon} \text{is an } \epsilon\text{-alternative and misclassified.}$$

In the literature, no clear definitional distinction is drawn between counterfactuals and adversarials. However, as we have argued in Sect. 4.2, we believe that the distinctions we have introduced are conceptually necessary. The definitions

[11] This vector x describes mostly a real-data instance.

[12] A semi-metric on a space X is a function $d : X \times X \to \mathbb{R}$ such that for all $x, x' \in X$ $d(x, x') \geq 0$, $d(x, x') = 0 \Leftrightarrow x = x'$, and $d(x, x') = d(x', x)$.

[13] With respect to $d(\cdot, \cdot)$.

of counterfactuals and adversarials differ in two aspects: the relation to the true instance label and the constraint of how close the respective data-point must be. The misclassification of adversarials enters the definition by enforcing it as an additional necessary condition. Note that this entails that only inputs for which a ground-truth exists can in our definition be called adversarials.

The second definitional difference we introduce is that counterfactuals must be maximally close data-points, while adversarials need only be within an ϵ-environment around the original input x. This relaxed condition on adversarials is introduced via defining them as ϵ-alternatives. This means, whether an input is called an adversarial or not, depends on how close the attacker requires the input to be. If the constraint is put too strong i.e. if ϵ is too small, there might not exist any adversarials within that environment. If, on the other side, the constraint-parameter is set very high, even inputs rather dissimilar to the original input can count as proper adversarials. Unlike adversarials, counterfactuals always exist as long as there exists an alternative to x. Moreover, only maximally close alternatives count as proper counterfactuals.

Especially counterfactuals, but also adversarials are often *targeted* i.e. the generated vector should not only be assigned to a different class than the original vector but to a specific *desired class*. The desired class imposes an additional relevance constraint. For counterfactuals, this may be from the perspective of the end-user who wants to get her loan application accepted rather than rejected or the model-engineer who wants to check whether the model can distinguish an input from other inputs of a specific object-class. For adversarials, this may be the desired classification from the perspective of the attacker of the system (e.g. Whatever is next to this sticker is a toaster Brown et al. 2017). In cases where a desired class exists and is imposed, we talk about *targeted* (ϵ)*-counterfactuals/adversarials*. More formally, let $y_{des} \in Y$ with $f(x) \neq y_{des}$ denote the desired outcome of a stakeholder given such a desired outcome exists.

Definition We call an alternative $x' \in X$ to x y_{des}*-targeted* if $f(x') = y_{des}$.

The notion of targeted vectors has relevance when it comes to generating counterfactuals/adversarials (see Sect. 5). Moreover, we can see the y_{des}-targeted property as a further specification of a counterfactual/adversarial that informs about the relevant class. In the case of counterfactuals, targetedness also has definitional relevance. Not every y_{des}-targeted counterfactual is also a "normal" counterfactual. There are cases where c_x is a vector with minimal distance to x that belongs to class y_{des}, however, there still exist inputs x' closer to x than c_x that change the classification to a different class $f(x) \neq f(x') \neq y_{des}$. Consider a loan application scenario in which a poor rejected applicant does not only want to get his loan accepted but be classified as a high-credibility premium client with better conditions. In such a case, the targeted counterfactual would not be among the more realistic "normal" counterfactuals. For adversarials on the other side, every targeted adversarial is also a "normal" adversarial given we consider the same ϵ environment.

CEs and AEs so far, we have only discussed vectors living in a space X. How do we get from these vectors to explanations or attacks?

Definition

- A *contrastive explanation (CON)* is a presentation of an alternative x' in contrast to x understandable to a human agent.
- A *counterfactual explanation (CE)* is a presentation of a counterfactual c_x understandable to a human agent.
- An *adversarial example (AE)* is the depiction of an adversarial a_x.

Notice that while every counterfactual and every adversarial describes an alternative, not every CE or AE is a CON. CONs must be presented as a contrast between x' and x. Possible presentation styles for CEs/AEs include the presentation in form of an (English-)conditional of type III for tabular data, an image for visual-data, or a sound for auditory-data. For tabular data, we use the property that the input features in such scenarios are interpretable. That means they have semantic meaning and can be expressed by human language concepts.

Assume we are in such a tabular-data scenario where $x = (x_1, \dots, x_n)$ describes the original vector and $c_x = (c_{x_1}, \dots, c_{x_n})$ one of its targeted-counterfactuals. Now, consider the vector $c_x - x$. $p \leq n$ of this vector's values will be non zero. Assume $k_1, \dots, k_p$ describe the names of these non-zero entries of the vector and $e_{k_1}, \dots, e_{k_p}$ their respective values. The (contrastive) CE in this scenario would be:

> If P had a $e_{k_1}, \dots, e_{k_p}$ higher/lower value in $k_1, \dots, k_p$, she would have reached her desired classification instead of $f(x)$.

For image-data, we can use the fact that vectors in such spaces can be visualized directly in their image representation. Examples have been shown both for CEs and AEs in Sect. 2. The same holds for auditory data-inputs which can be presented as a sound.

As mentioned above, often there is not one unique counterfactual to a given vector x. Therefore, there is not one unique correct CE. Worse, often different CEs are incompatible. The fact that there are several equally "good" explanations for the same prediction is called the *Rashomon effect* (Molnar 2019). Several ways to deal with this problem have been proposed. Mothilal et al. (2020), Moore et al. (2019), Wachter et al. (2017), and Dandl et al. (2020) propose to present various CEs dependent on the specific aim of a user. However, then the question arises, how many and which ones? Others propose to select a single CE according to relevance (Fernández-Loría et al. 2020) or a quality standard set by the user, such as complexity (Sokol and Flach 2019). We think the question the Rashomon effect poses is still open to debate. AEs are unique neither. However, as AEs must not cohere, nor be necessarily presented to humans, this plays no role.

Model-Level and Real-World one distinction that is often overlooked is the difference between an explanation/attack on the model-level and the real-world.

We need to be clear about whether we want to explain/attack the model or the modeled process. Generally, the former is much easier to accomplish than the latter. We can only move from a model explanation/attack to a process explanation/attack if the model itself, and also the translation of our inputs, preserve the essential structure of the process (Molnar et al. 2020). There are two scenarios for which the distinction between the two levels is relevant: it is relevant for CEs if a user is interested in recourse to attain a desired outcome (Karimi et al. 2020c). It is relevant for AEs if an attacker aims to deceive an ML system deployed in the physical world (Kurakin et al. 2016).

To give two examples that highlight the difference between model-level and real-world explanations/attacks, we reconsider the examples from Sect. 2. The presented CE in the loan application setting was: "If P had a 5, 000 € p.a. higher salary and an outstanding loan less, her loan application would have been accepted." This explanation clearly tells us something about the employed model, namely about the assignment for a particular alternative. However, P could take this as an action recommendation in the sense that if she raises her salary and paid her outstanding loan, she will receive the loan she applied for. Unfortunately, things are not that simple in the real world. P has to work hard to raise her salary and pay her open loan, this does not happen in zero time. By the time she reaches the required threshold, she may be five years older and her loan application will be rejected again, this time due to her advanced age or because a different algorithm is now used (Venkatasubramanian and Alfano 2020). So the transfer from the model explanation to an action recommendation for recourse is not as easy.

A similar example can be shown for AEs. Consider the Hand-Written Digits Recognition Scenario from Sect. 2 where an attacker aims to money-pump the postal service. The AEs presented are clearly inputs that trick the model. However, if she now aims to make the step to a real-world fraud, she has to print them out. A bad printing, different colors, alternative background, changed angles, or the camera employed by the postal service will impact which input the model receives. Thus, the AE might not work in the postal-service hand-written digits recognition service but only in the artificial setting where we can directly manipulate the input the model receives.

For both CEs and AEs, we need to know the employment context and the required functionality in order to be clear about what level we are dealing with. The work of Karimi et al. (2020c) and Mahajan et al. (2019) on algorithmic recourse and the work of Kurakin et al. (2016), Lu et al. (2017b), and Athalye et al. (2018) AEs in the physical world have alerted the CE and AE communities to the importance of the two different levels. The two levels collapse only for artificial settings in which the model perfectly matches the process (Karimi et al. 2020c) and the interventions truly lead to improvements in the target (König et al. 2021).

Definition We say a CE/AE operates at the *real-world level* if it describes changes in $\mathcal{X}$ that result in changes in $\mathcal{Y}$. We say that a CE/AE operates at the *model-level* if it describes changes in X that result in changes in Y.

5 Generation of CEs/AEs

So far we have motivated and discussed the formal definitions of CEs/AEs. Now, we move from the definition to their generation. Again, we will focus on the connections between the two fields. Before we start, it is important to note that the generation methods for AEs do generally not guarantee success i.e. it is unclear whether the generated input vector is misclassified. Instead, misclassification is particularly in image-classification still often reached accidentally as discussed in Sect. 4.2.

5.1 General Approaches

Optimization Problem the most common approach to find CEs/AEs is to formulate and solve an optimization problem. Such a problem formulation is already present in the definition of CEs/AEs, however, this is an optimization under side conditions and therefore not easy to solve. Instead, the standard formulation as a single objective optimization problem is Eq. 1 that led to the confusions discussed in Sect. 4.1.

For both (targeted/untargeted) CEs and AEs there exist many other formulations as an optimization problem (Serban et al. 2020; Verma et al. 2020). For example, for CEs Poyiadzi et al. (2020), Kanamori et al. (2020), and Van Looveren and Klaise (2019) add additional terms to Eq. 1 encoding further desiderata (see aims and distances below), Dandl et al. (2020) instead add these desiderata by formulating a multi-objective optimizations problem, and Karimi et al. (2020a) formulate a search for the smallest intervention on the variables needed to attain a change in classification. Similar to the former formulations for CEs, there exist approaches to AEs like (Carlini and Wagner 2017; Moosavi-Dezfooli et al. 2017) which modify the objective from Eq. 1 to obtain desired properties like computational efficiency or universality of an AE. Other optimization problems also take into account transformations of background or objects and generate AEs whose classification is invariant under such transformations (Eykholt et al. 2018; Brown et al. 2017; Athalye et al. 2018).

Generative Networks a second way to generate CEs/AEs that has been fruitfully applied is the use of generative networks that generate CEs/AEs for a given input. This technique is widespread for both AEs (Goodfellow et al. 2014; Zhao et al. 2017; Yuan et al. 2019) and CEs (Mahajan et al. 2019; Van Looveren and Klaise 2019; Pawelczyk et al. 2020).

Sensitivity Analysis a third approach that is almost exclusively used by the AEs community is sensitivity analysis. Information about the gradient (Goodfellow et al. 2015; Lyu et al. 2015) or Jacobian (Papernot et al. 2016b) of the function in the specific input is used to make a step in the direction of the decision boundary to a different class. Moore et al. (2019) is the only example we are aware of who use this approach to generate CEs. One reason why such approaches have probably not been picked up in the CE-literature is that it has limited conceptual justification, e.g. with respect to minimal distance, as we discuss in Sect. 6.

5.2 Distances

All approaches to generate AEs necessitate an underlying notion of distance, mainly for the inputs space but often also for the output space. Researchers worked with a high variety of distances. Often the distances encode specific desiderata researchers want CEs/AEs to satisfy. For both fields, the question for the right distance for a given use-case is considered an open problem (Serban et al. 2020; Verma et al. 2020). Since every norm induces a metric, we will use the names of the norms and generally talk about distances.

Sparsity and Imperceptibility since explanations often need to be understandable to people with limited time and cognitive resources, it is desirable for CEs to point out only few relevant features. Therefore, distances are preferred that take into account sparsity. For adversarials on the other side, a common aim is imperceptibility. Changes from the original input to the modified input should be hard to grasp for human observers. While these desiderata often lead to conflicting notions of distance, they also can coincide. For example, the L_0 and L_1 norm have both been fruitfully been applied to generate sparse counterfactuals (Dandl et al. 2020; Wachter et al. 2017) and imperceptible AEs (Su et al. 2019; Tramer and Boneh 2019; Pawelczyk et al. 2020).

However, some distances to attain sparsity of counterfactuals have not been used to reach imperceptibility of AEs. One way by which sparsity can be guaranteed is to explicitly put a constraint on the number of features allowed to change (Kanamori et al. 2020; Ustun et al. 2019; Sokol and Flach 2019). Another is to constrain the number of actions that can be taken, but not the number of the corresponding feature changes (Karimi et al. 2020c). To attain imperceptibility of AEs, the distances are more diverse. Common examples are the L_2 (Moosavi-Dezfooli et al. 2016) and L_∞ (Goodfellow et al. 2015; Elsayed et al. 2018) norm for distributed changes which often makes AEs look identical to the input they origin from. Other norms, more inspired by human perception are the Wasserstein-distance (Wong et al. 2019), using physical parameters underlying the image formation process (Liu et al. 2018), or the Perceptual Adversarial Similarity Score (Rozsa et al. 2016).

Plausibility and Misclassification in many contexts, end-users want to use explanations for guiding their future actions. In such scenarios, CEs should not present an entirely unrealistic alternative scenario to the explainee. Instead, the recommended alternative should be within reach and if possible it should be feasible for agents to perform actions based on these alternatives. This often means that the counterfactual lies in the natural data-distribution. AEs must by definition be misclassified, which as discussed in Sect. 4.2, is often easier to reach on the edges or slightly outside the natural data-manifold. We see an antagonism between the goal of realism of CEs and the misclassification of AEs. Thus, progress in one of them (especially concerning the applied distances) can easily inform progress in the other, only with reversed sign in the optimization.

One common way to attain plausibility is to take into account the distance of the CE to the closest training-datapoint (Kanamori et al. 2020; Dandl et al. 2020; Sharma et al. 2020) or the allowed path to the counterfactual (Poyiadzi et al. 2020). Often, additional constraints are posed such that only actionable features should be

changed to avoid non-helpful recommendations (Ustun et al. 2019). Another way is to take into account the causal structure of the real-world features. If a counterfactual arises realistically from an intervention on some of these features, the corresponding CE is plausible (Mahajan et al. 2019; Karimi et al. 2020c).

To attain misclassified inputs, it is generally reasonable to search in low-probability areas of the data-manifold (Szegedy et al. 2014) or even outside of it (Tanay and Griffin 2016). Therefore, most distances for AEs do not respect the causal structure between the corresponding real-world variables. Some even act directly against the causal structure and modify only irrelevant features (Ballet et al. 2019) or, just as for CEs, put constraints on the potential changes (Cartella et al. 2021). Particularly noteworthy is the distance of Moosavi-Dezfooli et al. (2017) who encode the robustness of the flip in classification and also the work of Carlini and Wagner (2017) who compare the misclassification between different applied distances. Interestingly, it has been found that a greater distance to the given decision boundary guarantees more robustness of misclassification, hence, many do not search for minimal but only close adversarials (Zhang et al. 2019).

Contestability and Misclassification CEs should allow explainees to detect adverse or wrong decisions. If the explainee is an end-user, this could be the case if she feels judged unfairly (Kusner et al. 2017; Asher et al. 2020). On the other side, if the explainee is the model-engineer, this could mean CEs reveal bugs. Again, AEs must be misclassified. Decision-making mistakes are the common denominator of the contestability reached by CEs and misclassification provided by AEs. Various ways have been proposed to encode these aims.

Russell (2019) provide contestability by presenting a range of diverse CEs in which different features were modified. This increases the chance that some CEs are presented that provide grounds to contest the decision. Sharma et al. (2020) define protected properties like ethnicity and focus on changes in these features in their distance. Laugel et al. (2019b) discuss how standard norms like L_1 can lead to unjustified CEs since they arise from inputs outside the training-data. Hashemi and Fathi (2020) combines CEs and AEs to evaluate the weaknesses of a given model. They use both, the L_0 and L_2 norm plus focus on protected features in search for realistic but misclassified counterfactuals. In a similar vein, Ballet et al. (2019) assign importance weights to features and through these weights they define weighted L_p norm where changes in more important features have a lower weight and are therefore more likely to change in the optimization process. Cartella et al. (2021) extend their work and put additional constraints to keep the adversarials realistic but still fraudulent. Especially the last three examples show the great overlap between the goals of contestability and misclassification.

5.3 Model-Access

As we have discussed above, we do not need to define an optimization problem to generate counterfactuals or adversarials. However, different solution methods differ in the degree of model-access they need. We distinguish between black-box and white-box scenarios. In a black-box scenario, explainers/attackers can only query

the model for some inputs they provide and receive the corresponding output. In a white-box scenario, the explainer/attacker has full model access. We can further distinguish between methods that only work for a particular model-class and methods that are model-agnostic. All black-box solvers work for any model. For white-box solvers, some only need access to gradients and therefore require a differentiable model and those that are specific to a particular model-class e.g. linear models but can therefore often handle mixed-data. Interestingly, even though white-box scenarios are more realistic for explainers and black-box scenarios more commonly occur for attackers, the literature shows tendencies in the opposite directions.[14]

Many solvers rely on access to the models gradients e.g. for CEs (Wachter et al. 2017; Mothilal et al. 2020; Pawelczyk et al. 2020; Mahajan et al. 2019) or for AEs (Szegedy et al. 2014; Athalye et al. 2018; Brown et al. 2017; Ballet et al. 2019). Other solvers for CEs are model-specific and require full model-access such as mixed-integer linear program solvers (Ustun et al. 2019; Russell 2019; Kanamori et al. 2020) or solvers tailored for decision trees (Tolomei et al. 2017). For AEs some solvers require neural network feature representations (Sabour et al. 2016). However, several solvers can deal with a black-box setup. Common in both literatures are evolutionary algorithms e.g. for CEs (Sharma et al. 2020; Dandl et al. 2020) and for AEs (Guo et al. 2019; Alzantot et al. 2019; Su et al. 2019). Very prominent for AEs are also the approximation of gradients by symmetric differences (Chen et al. 2017) and the usage of surrogate models (Papernot et al. 2017). Especially the latter approach is interesting as it is based on the transferability of AEs between different models optimized for the same task.

We see that many solvers are fruitfully used in both domains. It will be seen whether surrogate model-based approaches also find their way into the CE literature. We find the use of them for CEs conceptually controversial as the faithfulness to the model is more critical for an explanation than for an attack (also see Sect. 6 for a short discussion of this point)).[15]

6 Discussion

In this paper, we discussed the relationship between CEs and AEs. We argued that the definitional difference between the two object classes consists in their relation to the true data labels (i.e. adversarials must necessarily be misclassified) and their proximity to the original data-point (i.e. counterfactuals must be maximally close to the original input). Based on this, we introduced formal definitions for the key concepts of the fields. In addition, we have highlighted similarities and differences between the two fields in terms of use cases, solution methods, and distance metrics.

[14] See Serban et al. (2020) and Verma et al. (2020) who notice the respective tendencies in their surveys. They explain this by the chances to explore more in white box settings and the computational problems of black-box attacks in high-dimensional use cases (see Sect. 2).

[15] A first approach to use a surrogate model to generate similar explanations to CEs was proposed by Guidotti et al. (2018).

6.1 Relevance

Our work adds a new viewpoint to the discussion of the relationship between CEs and AEs. Eventually, we hope that our work can form the basis for merging the two fields. Based on our arguments and the formal definitions inspired by them, adversarials can be seen as special cases of (more distal) misclassified counterfactuals. Especially when it comes to CEs for which misclassification is a desirable property, such as CEs for contesting adverse decisions, detecting bugs, or improving model-robustness, we see potential synergies. We believe that a solid conceptual discussion becomes more important as these functions of CEs are emphasized and as application domains overlap (e.g., AEs in lending, CEs for image classification).

Our work also has a clear practical relevance. The conceptual arguments for the maximal proximity of counterfactuals make clear that generating counterfactuals via sensitivity analysis, as proposed by Moore et al. (2019), or using surrogate model approaches could be problematic. In the case of sensitivity analysis, maximal proximity to the original input is not guaranteed and hence the corresponding CEs have less explanatory power. Surrogate models might not be sufficiently faithful to the original model and therefore lead to bad/misleading explanations. As we discussed, solution methods to find CEs can also generate AEs, but the reverse can be problematic.

What we have shown in terms of the current literature is that there is a large amount of overlap. We have also suggested which parts are good candidates for transfers. However, as we have made clear, such transfers of mathematical frameworks or approaches require conceptual justification. While transferring gradient-based solution techniques from the AE literature to generate counterfactuals, as proposed by Wachter et al. (2017), is conceptually unproblematic, using counterfactuals to measure the robustness of a model, as suggested by Sharma et al. (2020), will not work for tabular data scenarios.

6.2 Limitations and Open Problems

Misclassification Formalized our work points to an important weak spot of the current AE literature: misclassification is achieved more or less by accident in the image domain, but is not clearly formalized. Such a formalization of misclassification would greatly advance the merging process between CEs and AEs. It may be considered a limitation of our work that we have not provided this formalization but instead referred to the true data-labels, which are either expensive to obtain or simply unknown. Nevertheless, we want to provide a roadmap of what such a formalization might look like.

We believe that Ballet et al. (2019) made the first solid contributions to a formal representation without requiring the ground-truth data labels. In our opinion, a good candidate framework for generalizing their approach is causal modeling (Pearl 2009). If we have a true causal model, misclassification is obtained by modifying a correctly classified input sufficiently to change the classification, but in a way that

violates the causal structure. We suggest that adversarials can be viewed as small modifications in causally irrelevant features that unjustly influence the prediction.

Unfortunately, approaching the problem of misclassification from a causal modeling perspective also comes with strong requirements: we need a structural causal model or at least a causal graph. Obtaining such models is extremely difficult (Pearl 2009; Schölkopf 2019), and when dealing with conceptually lower-order features such as pixels or sounds, causal models might even be the wrong descriptive language. Still, we think that even limited causal knowledge about, e.g. parts of the causal graph or some of the structural equation, might suffice in many contexts to prove that a change in classification is unjustified. Moreover, for conceptually less-structured feature spaces, higher-order causal models (Beckers and Halpern 2019) where features such as objects are supervened by lower-order features such as pixels may provide the right level of description to define misclassification.

Distances on Unstructured Spaces in our discussion in Sect. 4.2 on misclassification, we gave reasons why most inputs that solve Eq. 1 are misclassified. We argued that theoretically poorly justified distance metrics are one of the reasons for this phenomenon. However, we did not address whether this might be the only reason for this behavior and whether this would still be the case if we had conceptually well-justified distances on high dimensional spaces with little semantics such as pixel spaces.

We believe that this is an empirical question we could not settle in this paper. The standard way for approaching it would be to move the distances from raw features such as pixels to higher-order features such as object properties. It has often been pointed out that deep-learning algorithms based on convolutional neural networks (CNNs; Goodfellow et al. 2016) automatically find semantically meaningful features in layers close to the output space (Zhang and Zhu 2018; Bau et al. 2017). For example, one could define a distance function on the feature space just before the so-called dense layer in CNNs, which is responsible for classification.

While we consider this a promising direction for future research, there are good reasons to remain skeptical. First, unfortunately, it is not so easy to assign specific semantic meaning to these high-level features, since some of them are poly-semantic and are triggered by quite different inputs (Olah et al. 2020). Distance measures on such features may therefore also be conceptually unjustified and the problem remains. Second, examples of AEs, such as those given by Szegedy et al. (2014) or Goodfellow et al. (2015), seem to show images that are almost identical to the original image. Hence, conceptually well-justified distance functions should also assign a low distance to these images, and consequently they will still be generated by solving Eq. 1. Following (Ilyas et al. 2019), we think that AEs are generated by Eq. 1 not only because we apply the wrong distance function, but also because the ML model has not really learned the robust concepts that humans use to distinguish objects.

Explanations and Deceptions we have not discussed the conceptual relationship between illusions and explanations more generally (e.g. the relation between everyday life explanations and cognitive biases or optical illusions), but have focused only on CEs/AEs in ML. In what sense can an illusion explain a phenomenon? How can an explanation lead to a deception? Is there an underlying conceptual or even cognitive connection between explaining and deceiving? We do find these questions, and

the possible embedding of our CE/AE discussion within them, intriguing. For now, however, we leave these deep and difficult philosophical/psychological questions to other researchers.

Glossary

We will shortly explain the following terms with the help of the example depicted in Fig. 2. As in Sect. 4.3 we call $f : X \to Y$ the classifier, X the input space, and Y the output space.

Decision boundary in the example, the decision boundary is described by the blue line. All inputs above the decision boundary are labeled "approved", all inputs below the blue line are labeled "rejected". Crossing the decision boundary means that a point is moved from one side of the decision boundary to the other. For example, the individual represented by the black "x" at position (1, 28) might cross the decision boundary by moving his salary up 1000€ or by buying two more pets.

More generally, we can describe a decision boundary as a hypersurface in space Y that separates one class from another. These hypersurfaces are induced by the classification model $f : X \to Y$.

Data-Manifold in our example, the green and red points lie within the data-manifold of realistic data samples. However, there is no point number or negative number of pets, so such instances would lie outside the data-manifold.

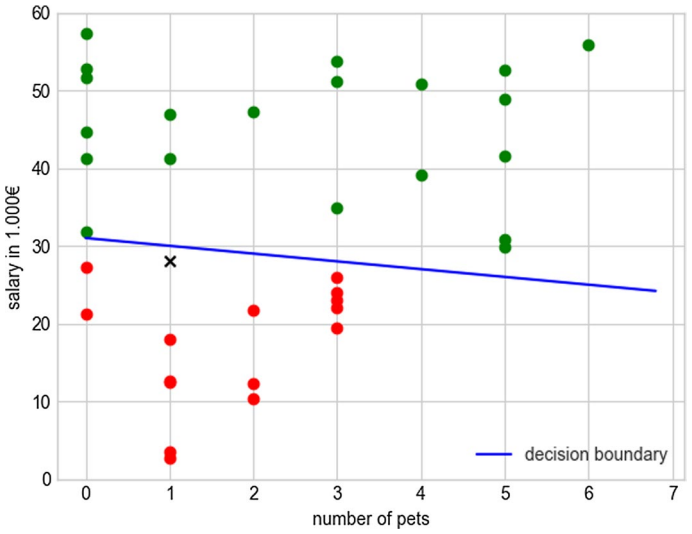

Fig. 2 This figure depicts the decision behavior of a simple classifier. It describes the scenario from Sect. 2, which is inspired by Ballet et al. (2019). The classifier uses two features, salary and number of pets, to decide whether to approve or reject a loan application. The green dots are the training data labeled as approved, the red dots are the training data labeled as rejected. The blue line describes the decision boundary of the classifier

More generally, a data-manifold describes a subset (often a hypersurface) of the spaces $X \times Y$ that arises naturally from a data-generating mechanism. A data-manifold encompasses the statistical population. The training and test data are usually a sample from this population.

Meaningless, unrealistic, or unseen inputs:

- *Meaningless* an example of a meaningless input in our scenario would be a person with a negative number of pets. It describes an input that makes no sense to us, but is contained in the space X.
- *Unrealistic* an example of an unrealistic input in our scenario would be a person with five million pets. It describes an input we can understand, but that most likely does not occur in the real world.
- *Unseen but realistic* an example of an unseen input in our scenario would be a person who earns 29,000€ and has four pets. It describes an input that may realistically occur in the real world, but was not part of the training data.

Conceptually (un-)justified distance metrics conceptually unjustified distance metrics assign small distances to inputs that are not similar from a conceptual standpoint. In our example, a distance function might assign a small distance to the points $x_1 = (0, 10)$ and $x_2 = (22, 10)$. This would make x_2, which lies far outside the data manifold and is assigned to the "approved" class by the model, a potential counterfactual for x_1. However, x_2 is highly unrealistic as 20 pets are a lot and it breaks the dependence that 20 pets are probably too expensive for an income of 10,000€ per year. This dependency problem is more severe for pixel spaces, since pixels have strong dependencies in the real world with their neighboring pixels. Moreover, an image in the form of a set of pixels represents an image of objects to humans, a fact that is difficult to account for with a metric.

Funding Open Access funding enabled and organized by Projekt DEAL. This work was supported by the Graduate School of Systemic Neuroscience (GSN) of the LMU Munich.

Data Availability Not applicable.

Code Availability Not applicable.

Declarations

Conflict of interest The author has no conflict of interest.

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access, 6,* 52138–52160.

Akula, A. R., Todorovic, S., Chai, J. Y., & Zhu, S. C. (2019). Natural language interaction with explainable AI models. In *CVPR workshops* (pp. 87–90).

Alzantot, M., Sharma, Y., Chakraborty, S., Zhang, H., Hsieh, C. J., & Srivastava, M. B. (2019). Genattack: Practical black-box attacks with gradient-free optimization. In *Proceedings of the genetic and evolutionary computation conference* (pp. 1111–1119).

Anjomshoae, S., Främling, K., & Najjar, A. (2019). Explanations of black-box model predictions by contextual importance and utility. In D. Calvaresi, A. Najjar, M. Schumacher & K. Främling (Eds.), *Explainable, transparent autonomous agents and multi-agent systems* (pp. 95–109). Springer .

Asher, N., Paul, S., & Russell, C. (2020). Adequate and fair explanations. arXiv preprint arXiv:200107578.

Athalye, A., Engstrom, L., Ilyas, A., & Kwok, K. (2018). Synthesizing robust adversarial examples. In *International conference on machine learning, PMLR* (pp. 284–293).

Balda, E. R., Behboodi, A., & Mathar, R. (2019). Perturbation analysis of learning algorithms: Generation of adversarial examples from classification to regression. *IEEE Transactions on Signal Processing, 67*(23), 6078–6091.

Ballet, V., Renard, X., Aigrain, J., Laugel, T., Frossard, P., & Detyniecki, M. (2019). Imperceptible adversarial attacks on tabular data. arXiv preprint arXiv:191103274.

Barocas, S., Selbst, A.D., & Raghavan, M. (2020). The hidden assumptions behind counterfactual explanations and principal reasons. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, Association for Computing Machinery, New York, NY, USA, FAT* '20, p 80–89, https://doi.org/10.1145/3351095.3372830

Bastani, O., Ioannou, Y., Lampropoulos, L., Vytiniotis, D., Nori, A. V., & Criminisi, A. (2016). Measuring neural net robustness with constraints. In *Proceedings of the 30th international conference on neural information processing systems* (pp. 2621–2629).

Bau, D., Zhou, B., Khosla, A., Oliva, A., & Torralba, A. (2017). Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6541–6549).

Beckers, S., & Halpern, J. Y. (2019). Abstracting causal models. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, pp. 2678–2685).

Behzadan, V., & Munir, A. (2017). Vulnerability of deep reinforcement learning to policy induction attacks. In *International conference on machine learning and data mining in pattern recognition* (pp. 262–275). Springer.

Bekoulis, G., Deleu, J., Demeester, T., & Develder, C. (2018). Adversarial training for multi-context joint entity and relation extraction. arXiv preprint arXiv:180806876.

Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition, 84,* 317–331.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Brown, T. B., Mané, D., Roy, A., Abadi, M., & Gilmer, J. (2017). Adversarial patch. arXiv preprint arXiv:171209665.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. arXiv preprint arXiv:200514165.

Browne, K., & Swift, B. (2020). Semantics and explanation: Why counterfactual explanations produce adversarial examples in deep neural networks. arXiv:2012.10076.

Burrell, J. (2016). How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data and Society, 3*(1), 2053951715622512. https://doi.org/10.1177/2053951715622512.

Byrne, R. M. (2016). Counterfactual thought. *Annual Review of Psychology, 67,* 135–157.

Byrne, R. M. (2019). Counterfactuals in explainable artificial intelligence (XAI): Evidence from human reasoning. In *IJCAI* (pp. 6276–6282).

Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 IEEE symposium on security and privacy* (pp. 39–57). IEEE.

Carlini, N., & Wagner, D. (2018). Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE security and privacy workshops (SPW)* (pp. 1–7). IEEE.

Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., & Zhou, W. (2016). Hidden voice commands. In *25th USENIX security symposium (USENIX security 16)* (pp. 513–530).

Cartella, F., Anunciacao, O., Funabiki, Y., Yamaguchi, D., Akishita, T., & Elshocht, O. (2021). Adversarial attacks for tabular data: Application to fraud detection and imbalanced data. arXiv preprint arXiv:210108030.

Chen, P. Y., Zhang, H., Sharma, Y., Yi, J., & Hsieh, C. J. (2017). ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security* (pp. 15–26).

Claeskens, G., Hjort, N. L., et al. (2008). *Model selection and model averaging*. Cambridge Books. https://doi.org/10.1017/CBO9780511790485.

Dalvi, N., Domingos, P., Sanghai, S., & Verma, D. (2004). Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 99–108).

Dandl, S., Molnar, C., Binder, M., & Bischl, B. (2020). Multi-objective counterfactual explanations. In T. Bäck, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich & H. Trautmann (Eds.), *Parallel problem solving from nature—PPSN XVI* (pp. 448–469). Springer.

Das, A., & Rad, P. (2020). Opportunities and challenges in explainable artificial intelligence (XAI): A survey. arXiv preprint arXiv:200611371.

Dong, Y., Su, H., Zhu, J., & Bao, F. (2017). Towards interpretable deep neural networks by leveraging adversarial examples. arXiv preprint arXiv:170805493.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:170208608.

Došilović, F. K., Brčić, M., & Hlupić, N. (2018). Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)* (pp. 0210–0215). IEEE.

D'Silva, V., Kroening, D., & Weissenbacher, G. (2008). A survey of automated techniques for formal software verification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 27*(7), 1165–1178.

Elsayed, G., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., & Sohl-Dickstein, J. (2018). Adversarial examples that fool both computer vision and time-limited humans. In *Advances in neural information processing systems* (pp. 3910–3920).

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., & Song, D. (2018). Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1625–1634).

Fernandez, J. C., Mounier, L., & Pachon, C. (2005). A model-based approach for robustness testing. In *IFIP international conference on testing of communicating systems* (pp. 333–348). Springer.

Fernández-Loría, C., Provost, F., & Han, X. (2020). Explaining data-driven decisions made by AI systems: The counterfactual approach. arXiv:2001.07417.

Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research, 20*(177), 1–81. http://jmlr.org/papers/v20/18-760.html.

Friedman, J. H., et al. (1991). Multivariate adaptive regression splines. *The Annals of Statistics, 19*(1), 1–67. https://doi.org/10.1214/aos/1176347963.

Good, P. I., & Hardin, J. W. (2012). *Common errors in statistics (and how to avoid them)*. Wiley. https://doi.org/10.1002/9781118360125.

Goodfellow, I., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International conference on learning representations*. arXiv:1412.6572.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. arXiv preprint arXiv:14062661.

Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., & Lee, S. (2019). Counterfactual visual explanations. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning, PMLR, proceedings of machine learning research* (Vol. 97, pp. 2376–2384). http://proceedings.mlr.press/v97/goyal19a.html.

Grath, R. M., Costabello, L., Van, C. L., Sweeney, P., Kamiab, F., Shen, Z., & Lecue, F. (2018). Interpretable credit application predictions with counterfactual explanations. arXiv preprint arXiv:181105245.

Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., & Giannotti, F. (2018). Local rule-based explanations of black box decision systems. arXiv preprint arXiv:180510820.

Guo, C., Gardner, J. R., You, Y., Wilson, A. G., & Weinberger, K. Q. (2019). Simple black-box adversarial attacks. arXiv preprint arXiv:190507121.

Hashemi, M., & Fathi, A. (2020). Permuteattack: Counterfactual explanation of machine learning credit scorecards. arXiv:2008.10138.

Hitchcock, C. (2001). The intransitivity of causation revealed in equations and graphs. *The Journal of Philosophy, 98*(6), 273–299.

Huang, S., Papernot, N., Goodfellow, I., Duan, Y., & Abbeel, P. (2017). Adversarial attacks on neural network policies. arXiv preprint arXiv:170202284.

Hutson, M. (2018). Ai researchers allege that machine learning is alchemy. *Science, 360*(6388), 861.

Ignatiev, A., Narodytska, N., & Marques-Silva, J. (2019). On relating explanations and adversarial examples. In *Advances in neural information processing systems* (pp. 15883–15893).

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). Adversarial examples are not bugs, they are features. In *Advances in neural information processing systems* (pp. 125–136).

Joseph, A. D., Nelson, B., Rubinstein, B. I., & Tygar, J. (2018). *Adversarial machine learning*. Cambridge University Press.

Kanamori, K., Takagi, T., Kobayashi, K., & Arimura, H. (2020). DACE: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In C. Bessiere (Ed.), *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI-20. International joint conferences on Artificial Intelligence Organization* (pp. 2855–2862).

Karimi, A. H., Barthe, G., Balle, B., & Valera, I. (2020a). Model-agnostic counterfactual explanations for consequential decisions. In *International conference on artificial intelligence and statistics, PMLR* (pp. 895–905).

Karimi, A. H., Barthe, G., Schölkopf, B., & Valera, I. (2020b). A survey of algorithmic recourse: Definitions, formulations, solutions, and prospects. arXiv preprint arXiv:201004050.

Karimi, A. H., Schölkopf, B., & Valera, I. (2020c). Algorithmic recourse: From counterfactual explanations to interventions. In *37th International conference on machine learning (ICML)*.

Kizza, J. M., & Kizza, W. (2013). *Guide to computer network security*. Springer.

König, G., Freiesleben, T., & Grosse-Wentrup, M. (2021). A causal perspective on meaningful and robust algorithmic recourse. arXiv preprint arXiv:210707853.

Kurakin, A., Goodfellow, I., & Bengio, S., et al. (2016). Adversarial examples in the physical world

Kusner, M. J., Loftus, J., Russell, C., & Silva, R. (2017). Counterfactual fairness. In *Advances in neural information processing systems* (pp. 4066–4076).

Laugel, T., Lesot, M. J., Marsala, C., Renard, X., & Detyniecki, M. (2019a). The dangers of post-hoc interpretability: Unjustified counterfactual explanations. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI-19, international joint conferences on Artificial Intelligence Organization* (pp. 2801–2807). https://doi.org/10.24963/ijcai.2019/388.

Laugel, T., Lesot, M. J., Marsala, C., Renard, X., & Detyniecki, M. (2019b). Unjustified classification regions and counterfactual explanations in machine learning. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 37–54). Springer.

Leviathan, Y., & Matias, Y. (2018). Google duplex: An AI system for accomplishing real-world tasks over the phone. https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html.

Lewis, D. (1979). Counterfactual dependence and time's arrow. *Noûs, 13*(4), 455–476.

Lewis, D. (1983). *Philosophical papers* (Vol. I). Oxford University Press.

Lewis, D. K. (1973). *Counterfactuals*. Blackwell.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue, 16*(3), 31–57.

Liu, H. T. D., Tao, M., Li, C. L., Nowrouzezahrai, D., & Jacobson, A. (2018). Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. arXiv preprint arXiv:180802651.

Lu, J., Issaranon, T., & Forsyth, D. (2017a). SafetyNet: Detecting and rejecting adversarial examples robustly. In *Proceedings of the IEEE international conference on computer vision* (pp. 446–454).

Lu, J., Sibai, H., Fabry, E., & Forsyth, D. (2017b). No need to worry about adversarial examples in object detection in autonomous vehicles. arXiv preprint arXiv:170703501.

Lyu, C., Huang, K., & Liang, H. N. (2015). A unified gradient regularization family for adversarial examples. In *2015 IEEE international conference on data mining* (pp. 301–309). IEEE.

Mahajan, D., Tan, C., & Sharma, A. (2019). Preserving causal constraints in counterfactual explanations for machine learning classifiers. arXiv preprint arXiv:191203277.

Menzies, P., & Beebee, H. (2019). Menzies, P., & Beebee, H. (2019). Counterfactual theories of causation. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* winter 2019 edition. Metaphysics Research Lab, Stanford University.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence, 267,* 1–38.

Molnar, C. (2019). Interpretable Machine Learning. https://christophm.github.io/interpretable-ml-book/

Molnar, C., König, G., Herbinger, J., Freiesleben, T., Dandl, S., Scholbeck, C. A., Casalicchio, G., Grosse-Wentrup, M., & Bischl, B. (2020). Pitfalls to avoid when interpreting machine learning models. arXiv:2007.04131.

Moore, J., Hammerla, N., & Watkins, C. (2019). Explaining deep learning models with constrained adversarial examples. In *Pacific Rim international conference on artificial intelligence* (pp. 43–56). Springer.

Moosavi-Dezfooli, S. M., Fawzi, A., & Frossard, P. (2016). DeepFool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2574–2582).

Moosavi-Dezfooli, S. M., Fawzi, A., Fawzi, O., & Frossard, P. (2017). Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1765–1773).

Mothilal, R. K., Sharma, A., & Tan, C. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the ACM conference on fairness, accountability, and transparency*.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., & Carter, S. (2020). Zoom in: An introduction to circuits. Distill 5(3):e00024–001

Olson, M. L., Khanna, R., Neal, L., Li, F., & Wong, W. K. (2021). Counterfactual state explanations for reinforcement learning agents via generative deep learning. *Artificial Intelligence, 295,* 103455.

Páez, A. (2019). The pragmatic turn in explainable artificial intelligence (XAI). *Minds and Machines, 29*(3), 441–459.

Papernot, N., McDaniel, P., & Goodfellow, I. (2016a). Transferability in machine learning: From phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:160507277.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016b). The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)* (pp. 372–387). IEEE.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security* (pp. 506–519).

Pawelczyk, M., Broelemann, K., & Kasneci, G. (2020). Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of the web conference*, 2020 (pp. 3126–3132).

Pearl, J. (2009). *Causality*. Cambridge University Press.

Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., & Flach, P. (2020). Face: Feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM conference on AI, ethics, and society* (pp. 344–350).

Reutlinger, A. (2018). Extending the counterfactual theory of explanation. In *Explanation beyond causation: Philosophical perspectives on non-causal explanations* (pp. 74–95). Oxford University Press.

Ribeiro, M. T., Singh, S., & Guestrin, C. (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* (pp .1135–1144). ACM. https://doi.org/10.1145/2939672.2939778.

Rozsa, A., Rudd, E. M., & Boult, T. E. (2016). Adversarial diversity and hard positive generation. In *Proceedings of the IEEE conference on computer vision and recognition workshops* (pp. 25–32).

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence, 1*(5), 206–215.

Russell, C. (2019). Efficient search for diverse coherent explanations. In *Proceedings of the conference on fairness, accountability, and transparency, FAT* '19*, New York, NY, USA (pp. 20–28). Association for Computing Machinery. https://doi.org/10.1145/3287560.3287569.

Sabour, S., Cao, Y., Faghri, F., & Fleet, D. J. (2016). Adversarial manipulation of deep representations. In Y. Bengio & Y. LeCun (Eds.), *4th International conference on learning representations, ICLR 2016*, San Juan, Puerto Rico, May 2–4, 2016, Conference Track Proceedings. arXiv:1511.05122.

Schölkopf, B. (2019). Causality for machine learning. arXiv preprint arXiv:191110500.

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W., Bridgland, A., et al. (2020). Improved protein structure prediction using potentials from deep learning. *Nature, 577*(7792), 706–710.

Serban, A., Poll, E., & Visser, J. (2020). Adversarial examples on object recognition: A comprehensive survey. *ACM Computing Surveys (CSUR), 53*(3), 1–38.

Sharma, S., Henderson, J., & Ghosh, J. (2020). CERTIFAI: Counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models. In *Proceedings of the AAAI/ACM conference on AI, ethics, and society*. https://doi.org/10.1145/3375627.3375812.

Shin, D. (2021). The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable AI. *International Journal of Human–Computer Studies, 146,* 102551.

Sokol, K., & Flach, P. A. (2019). Counterfactual explanations of machine learning predictions: Opportunities and challenges for AI safety. In *Proceedings of the AAAI workshop on artificial intelligence safety*.

Song, D., Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Tramer, F., Prakash, A., & Kohno, T. (2018). Physical adversarial examples for object detectors. In *12th USENIX workshop on offensive technologies (WOOT 18)*.

Stalnaker, R. C. (1968). A theory of conditionals. In *IFS* (pp. 41–55). Springer.

Starr, W. (2019). Counterfactuals. In: E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy*, Fall 2019 edition. Metaphysics Research Lab, Stanford University.

Stepin, I., Alonso, J. M., Catala, A., & Pereira-Fariña, M. (2021). A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access, 9,* 11974–12001. https://doi.org/10.1109/ACCESS.2021.3051315.

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems, 41*(3), 647–665. https://doi.org/10.1007/s10115-013-0679-x.

Stutz, D., Hein, M., & Schiele, B. (2019) Confidence-calibrated adversarial training: Generalizing to unseen attacks. arXiv preprint arXiv:191006259.

Su, J., Vargas, D. V., & Sakurai, K. (2019). One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation, 23*(5), 828–841.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. In *International conference on learning representations*. arXiv:1312.6199.

Tanay, T., & Griffin, L. (2016). A boundary tilting perspective on the phenomenon of adversarial examples. arXiv preprint arXiv:160807690.

Tolomei, G., Silvestri, F., Haines, A., & Lalmas, M. (2017). Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 465–474).

Tomsett, R., Widdicombe, A., Xing, T., Chakraborty, S., Julier, S., Gurram, P., Rao, R., & Srivastava, M. (2018). Why the failure? How adversarial examples can provide insights for interpretable machine learning. In *21st International conference on information fusion (FUSION)* (pp. 838–845). IEEE.

Toreini, E., Aitken, M., Coopamootoo, K., Elliott, K., Zelaya, C. G., & van Moorsel, A. (2020). The relationship between trust in AI and trustworthy machine learning technologies. In *Proceedings of the 2020 conference on fairness, accountability, and transparency* (pp. 272–283).

Tramer, F., & Boneh, D. (2019). Adversarial training and robustness for multiple perturbations. arXiv preprint arXiv:190413000.

Ustun, B., Spangher, A., & Liu, Y. (2019). Actionable recourse in linear classification. In *Proceedings of the conference on fairness, accountability, and transparency* (pp. 10–19).

Van Looveren, A., & Klaise, J. (2019). Interpretable counterfactual explanations guided by prototypes. arXiv preprint arXiv:190702584.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer.

Venkatasubramanian, S., & Alfano, M. (2020). The philosophical basis of algorithmic recourse. In *Proceedings of the 2020 conference on fairness, accountability, and transparency* (pp. 284–293).
Verma, S., Dickerson, J., & Hines, K. (2020). Counterfactual explanations for machine learning: A review. arXiv preprint arXiv:201010596.
Voigt, P., & Von dem Bussche, A. (2017). *The EU general data protection regulation (GDPR). A practical guide* (1st ed.). Springer 10:3152676.
Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv JL & Tech, 31,* 841.
Wang, X., He, K., & Hopcroft, J.E. (2019). AT-GAN: A generative attack model for adversarial transferring on generative adversarial nets. *CoRR*. arXiv:abs/190407793.
Wei, X., Liang, S., Chen, N., & Cao, X. (2018). Transferable adversarial attacks for image and video object detection. arXiv preprint arXiv:181112641.
Wong, E., Schmidt, F., & Kolter, Z. (2019). Wasserstein adversarial examples via projected Sinkhorn iterations. In *International conference on machine learning, PMLR* (pp. 6808–6817).
Woodward, J. (2002). What is a mechanism? A counterfactual account. *Philosophy of Science, 69*(S3), S366–S377.
Yuan, X., He, P., Zhu, Q., & Li, X. (2019). Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems, 30*(9), 2805–2824.
Zhang, H., Chen, H., Song, Z., Boning, D., Dhillon, I.S., & Hsieh, C.J. (2019). The limitations of adversarial training and the blind-spot attack. arXiv preprint arXiv:190104684
Zhang, Q., & Zhu, S. C. (2018). Visual interpretability for deep learning: A survey. arXiv preprint arXiv:180200614.
Zhao, Z., Dua, D., & Singh, S. (2017). Generating natural adversarial examples. arXiv preprint arXiv:171011342.