The Effect of Data Poisoning on Counterfactual Explanations

André Artelt[1,2] , **Shubham Sharma**[3] , **Freddy Lecué**[4] and **Barbara Hammer**[1]
[1]Bielefeld University, Bielefeld, Germany
[2]University of Cyprus, Nicosia, Cyprus
[3]J.P. Morgan AI Research
[4]Inria, Sophia Antipolis, France
{aartelt,bhammer}@techfak.uni-bielefeld.de, shubham.x2.sharma@jpmchase.com, freddy.lecue@inria.fr

Abstract

Counterfactual explanations provide a popular method for analyzing the predictions of black-box systems, and they can offer the opportunity for computational recourse by suggesting actionable changes on how to change the input to obtain a different (i.e. more favorable) system output. However, recent work highlighted their vulnerability to different types of manipulations. This work studies the vulnerability of counterfactual explanations to data poisoning. We formalize data poisoning in the context of counterfactual explanations for increasing the cost of recourse on three different levels: locally for a single instance, or a sub-group of instances, or globally for all instances. We demonstrate that state-of-the-art counterfactual generation methods and toolboxes are vulnerable to such data poisoning.

1 Introduction

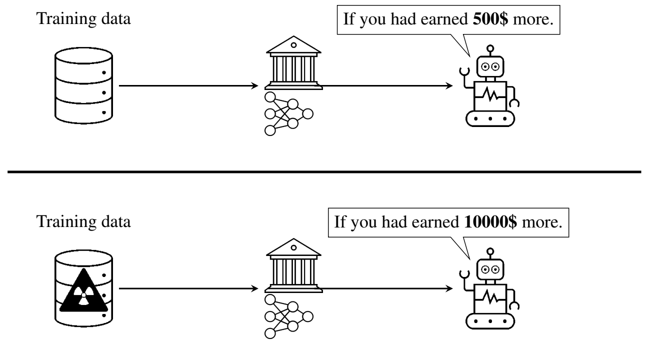
Nowadays, many Artificial Intelligence (AI-) and Machine Learning (ML-) based systems are deployed in the real world [Zhao *et al.*, 2023; Ho *et al.*, 2022]. These systems show an impressive performance but are still not perfect – e.g. failures, issues of fairness, and vulnerability to data poisoning can cause harm when applied in the real world.

Given the threat of failures (intentionally caused or not), transparency of such deployed AI- and ML-based systems becomes a crucial aspect. Transparency is important not only to prevent failures but also to create trust in such systems and understand where and how it is safe to deploy them. The importance of transparency was also recognized by the policymakers and therefore found its way into legal regulations such as the EU's GDPR [Council of European Union, 2016] or the more recent EU AI act [Commission *et al.*, 21 04 2021]. Explanations are a popular way of achieving transparency and shaping the field of eXplainable AI (XAI) [Dwivedi *et al.*, 2023]. However, because of many different use cases and users, many different explanation methods exist [Dwivedi *et al.*, 2023; Arrieta *et al.*, 2020; Adadi and Berrada, 2018; Rawal *et al.*, 2021]. One popular type of explanation method is recourse by counterfactual explanations [Wachter *et al.*, 2017], which are inspired by human explanations [Byrne, 2019]. A counterfactual explanation provides recourse by stating actionable recommendations on how to change the system's output in some desired way – e.g. how to change a rejected loan application into an accepted one. Recent works have shown that counterfactual explanations are neither robust to model changes [Mishra *et al.*, 2021], nor to input perturbations [Artelt *et al.*, 2021; Virgolin and Fracaros, 2023], and also not to adversarial training [Slack *et al.*, 2021]. However, the vulnerability of counterfactual explanation methods to data poisoning remains unexplored.

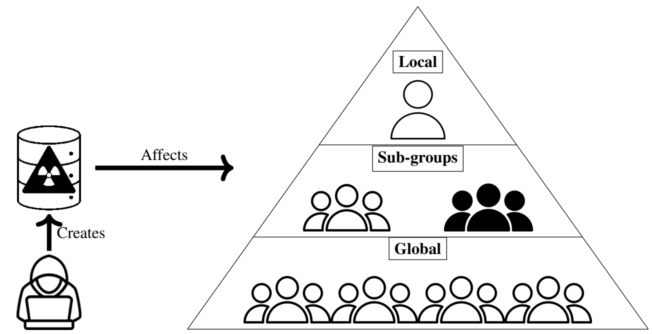
Data poisoning effects models in the training stage by changing training samples or adding new instances such that, for instance, the performance (e.g. accuracy) of the final trained model decreases [Lin *et al.*, 2021; Tolpegin *et al.*, 2020], or fairness issues arise [Mehrabi *et al.*, 2021; Solans *et al.*, 2020]. Data poisoning can be done offline [Lin *et al.*, 2021] or online [Tolpegin *et al.*, 2020]. It only makes small changes to the training data such as changing labels, removing samples, or adding new instances, that are likely to remain unnoticed. This poses a real threat in practice because nowadays many large models are trained on huge data sets – often based on public data from the internet [Zhao *et al.*, 2023; Ho *et al.*, 2022] – where it is impossible to check data in detail and therefore poisonous data might affect a large number of models trained directly on the data or indirectly using some pre-trained embeddings or models [Shan *et al.*, 2023; Bojchevski and Günnemann, 2019; Yang *et al.*, 2023].

In the context of counterfactual explanations, data poisoning could increase the cost of recourse as illustrated in Figure 1a – either globally for all individuals or for a subset of individuals. Since counterfactuals state actionable recommendations that are to be executed in the real-world, manipulated explanations would directly affect the individuals by enforcing more costly actions or hiding some information from them. Although counterfactual explanations are a popular and widely used explanation method, the effect of data poisoning on them has not been studied yet.

Our contribution: In this work, we study the vulnerability of counterfactual explanations to data poisoning. For this purpose, we formalize and identify a set of data poisoning for counterfactual explanations (see Section 4) which inject a small set of realistic but poisoned data points into the training data set such that the decision boundary of a newly trained



(a) Illustrative example of loan application: Poisoned training data set (bottom half of the figure) leads to a higher cost of recourse.



(b) Data poisoning at three different levels that can affect all or a few individuals.

classifier shifts to increase the cost of recourse. This can be done on three different levels: locally for an individual or, a sub-group of individuals, or globally for all individuals. The method only needs access to an interface for getting predictions and a mechanism for generating closest counterfactuals, but no access or knowledge about model internals is needed. We empirically find (see Section 5) that existing state-of-the-art methods for computing counterfactual explanations are vulnerable to data poisoning.

2 Related Work

Most existing work [Baniecki and Biecek, 2024] on exposing the vulnerability of explanations is centered in the vision domain and focuses either on adversarial examples or model manipulation. Only very little work considers domain-independent data poisoning [Baniecki and Biecek, 2024]. For instance, there exist data poisoning against partial dependence plots [Baniecki *et al.*, 2022], SHAP [Baniecki and Biecek, 2022], and concept-based explainability tools [Brown and Kvinge, 2023]. The authors of [Baniecki *et al.*, 2022] propose a genetic algorithm for perturbing the training data such that SHAP importances or attributions change. Their proposed method assumes that it is possible to change (possibly) all samples in the training set, which might constitute a very strong and unrealistic assumption in reality. Furthermore, changing many (or all) samples in the training data set might harm the model's predictive performance – this, however, is not evaluated in [Baniecki *et al.*, 2022]. A similar approach, with the same limitations, is proposed in [Baniecki and Biecek, 2022] where partial dependence plots are targeted. In the context of counterfactual explanations, the authors of [Slack *et al.*, 2021] propose an adversarial training objective such that the cost of recourse decreases for a sub-group of individuals. Note that this approach is model-specific and different from data poisoning since it proposes the use of a malicious cost function and therefore assumes full control over the training procedure. In this work, we consider data poisonings and argue that changing or adding training instances might often be more actionable in practice.

3 Foundations of Counterfactual Explanations and Recourse

A counterfactual explanation (often just called counterfactual) states actionable changes to the features of a given instance such that the system's output changes. Usually, an explanation is requested in the case of an unexpected or unfavorable outcome [Riveiro and Thill, 2022] – in the latter case, a counterfactual is also referred to as *recourse* [Karimi *et al.*, 2021], i.e. recommendations on how to change the unfavorable into a favorable outcome. Because counterfactuals can mimic ways in which humans explain [Byrne, 2019], they constitute among one of the most popular explanation methods in literature and in practice [Molnar, 2019; Verma *et al.*, 2020]. There are two important properties that must be considered when formalizing and computing counterfactual explanations [Wachter *et al.*, 2017]: 1) the contrasting property, requiring that the stated changes indeed change the output of the system; and 2) the cost of the counterfactual – i.e. the cost and effort it takes to execute the counterfactual in the real world should be as low as possible in order to maximize its usefulness (e.g. counterfactuals with very few changes or as small as possible changes). Both properties can be combined into an optimization problem (see Definition 1).

Definition 1 ((Closest) Counterfactual Explanation)**.** *Assume a classifier $h : \mathbb{R}^d \to \mathcal{Y}$ is given. Computing a counterfactual $\vec{\delta}_{cf} \in \mathbb{R}^d$ for a given instance $\vec{x}_{orig} \in \mathbb{R}^d$ is phrased as the following optimization problem:*

$$\underset{\vec{\delta}_{cf} \in \mathbb{R}^d}{\arg\min} \; \ell\left(h(\vec{x}_{orig} + \vec{\delta}_{cf}), y_{cf}\right) + C \cdot \theta(\vec{\delta}_{cf}) \tag{1}$$

where $\ell(\cdot)$ implements the contrasting property by means of a loss function that penalizes deviation of the prediction $h(\vec{x}_{cf} := \vec{x}_{orig} + \vec{\delta}_{cf})$ from the requested outcome y_{cf}, $\theta(\cdot)$ states the cost of the explanation (e.g. cost of recourse) which should be minimized, and $C > 0$ denotes the regularization strength balancing the two properties. The short-hand notation CF$(\vec{x}, h)$ denotes the counterfactual $\vec{\delta}_{cf}$ of an instance $\vec{x}$ under a classifier $h(\cdot)$ iff the target outcome y_{cf} is uniquely determined.

Note that the cost of the counterfactual, here modeled by $\theta(\cdot)$, is highly domain and use-case specific and therefore must be chosen carefully in practice and might require domain knowledge. In many implementations and toolboxes [Guidotti, 2022], the p-norm is used as a default.

Remark 1. *In the case of recourse – i.e. a counterfactual $\vec{\delta}_{cf}$ for turning an unfavorable into a favorable outcome –, we refer to the cost $\theta(\vec{\delta}_{cf})$, as the cost of recourse.*

In this work, w.l.o.g., we refer to $y = 0$ as the unfavorable, and $y = 1$ as the favorable outcome. Besides those two essential properties (contrasting and cost), there exist additional relevant aspects such as plausibility [Looveren and Klaise, 2021; Poyiadzi *et al.*, 2020], diversity [Mothilal *et al.*, 2020], robustness [Artelt *et al.*, 2021], fairness [Artelt and Hammer, 2023; Von Kügelgen *et al.*, 2022], etc. which have been addressed in literature [Guidotti, 2022].

Remark 2. *For a classifier $h : \mathbb{R}^d \to \mathcal{Y}$, we say that $\vec{x}_{cf} \in \mathbb{R}^d$ provides recourse for an instance $\vec{x}_{orig} \in \mathbb{R}^d$ iff there exists a counterfactual $\vec{\delta}_{cf}$ (Definition 1) such that $\vec{x}_{cf} = \vec{x}_{orig} + \vec{\delta}_{cf}$.*

There exist numerous methods and (Python) implementations/toolboxes for computing counterfactual explanations in practice [Guidotti, 2022] – most are including some additional aspects such as plausibility, diversity, etc.: *FACE* [Poyiadzi *et al.*, 2020] is a model-agnostic algorithm for computing feasible and actionable counterfactuals. Instead of computing a single counterfactual only, the method also outputs a path of intermediate actionable steps that lead from the original instance to the final counterfactual. *Counterfactuals Guided by Prototypes* [Looveren and Klaise, 2021] is another method focusing on plausibility. Here a set of plausible instances (so-called prototypes) are used to pull the final counterfactual instance (i.e. $\vec{x}_{\text{orig}} + \vec{\delta}_{\text{cf}}$) closer to these plausible instances and by this make them more plausible. *DiCE* [Mothilal *et al.*, 2020] is a model-agnostic method and Python toolbox for computing a set of diverse closest counterfactual explanations – i.e. a set of very different counterfactuals is computed instead of a single one only. *Nearest Training Sample method* is a straightforward baseline method for computing counterfactual explanations that can be implemented by picking the closest sample, with the requested output y_{cf}, from a given set (e.g. training set) as the counterfactual instance.

4 Data Poisoning of Counterfactuals

Data poisoning of counterfactual explanations can have effects on different levels/areas (see Figure 1b): all individuals are affected (global effect), only one or multiple sub-groups are affected (sub-groups effect), or only a single individual is affected (local effect). At the same time, data poisoning can aim for different effects on counterfactual explanations, such as hiding attributes or increasing the cost of recourse (Remark 2). Since providing (computational) recourse is a core application of counterfactuals, increasing the cost of recourse has the most severe consequence in the real world because it would harm individuals directly by making the recourse more costly. Therefore, in this work, we focus on data poisoning for increasing the cost of recourse.

4.1 Data Poisoning for Increasing the Cost of Recourse

In this work, we study the effect of data poisoning on the cost of recourse (Remark 1). That is, we focus on data poisoning with the primary goal of increasing the cost of recourse, in a pre-defined region in data space, as stated in Definition 2.

Definition 2 (Data Poisoning for Increasing the Cost of Recourse)**.** *Given an original training data set $\mathcal{D}_{orig} \subset \{\mathcal{X} \times \mathcal{Y}\}^n$ and a probability density $\phi(\cdot)$ assigning a high likelihood to targeted instances, we transform (i.e. poison) $\mathcal{D}_{orig}$ into a new data set $\mathcal{D}_{poisoned} \subset \{\mathcal{X} \times \mathcal{Y}\}^m$ by means of a data poisoning mechanism $T : \{\mathcal{X} \times \mathcal{Y}\}^n \to \{\mathcal{X} \times \mathcal{Y}\}^m$, such that the cost of recourse $\theta(\cdot)$ increases for instances under $\phi(\cdot)$:*

$$\begin{aligned} &\mathbb{E}_{\vec{x}\sim\phi}\left[\theta \circ CF(\vec{x}, h_{\mathcal{D}_{poisoned}})\right] > \mathbb{E}_{\vec{x}\sim\phi}\left[\theta \circ CF(\vec{x}, h_{\mathcal{D}_{orig}})\right] \\ &\text{where } \mathcal{D}_{poisoned} = T(\mathcal{D}_{orig}) \end{aligned} \quad (2)$$

where $\circ$ denotes the function composition, $h_{\mathcal{D}}$ denotes a classifier that was derived from the data set $\mathcal{D}$, and $CF(\cdot, \cdot)$ a method for generating counterfactuals.

The density $\phi(\cdot)$ allows us to vary the level of the poisoning (see Figure 1b) – e.g. for a global effect, we could use a class-wise density for targeting all instances from a specific class, or in the case of a local effect, we could use a delta-density to target a single instance or a small group of instances.

Note that the data poisoning mechanism $T(\cdot)$ can be implemented in different ways: $T(\cdot)$ could perturb existing samples in $\mathcal{D}_{\text{orig}}$, in this case, the number of instances does not change – i.e. $n = m$ –, or it could add new instances as well, in this case, in the number of training instances is increased – i.e. $m > n$. In this work, we focus on the latter case – i.e. adding new (poisonous) instances to the training data set.

From a practical point of view, besides increasing the cost of recourse (as stated in Definition 2), poisoning instances can have the following properties

- The number of poisonous instances, here, the number of added instances is small, i.e. $\arg\min_T |T(\mathcal{D}_{\text{orig}}) \setminus \mathcal{D}_{\text{orig}}|$
- The poisonous instances are realistic– i.e. they are on the data manifold $p_{\text{data}}(\cdot)$ and have a high likelihood: $\arg\max_T p_{\text{data}}(\vec{x}_i, y_i) \quad \forall(\vec{x}_i, y_i) \in T(\mathcal{D}_{\text{orig}}) \setminus \mathcal{D}_{\text{orig}}$
- In the case of aiming for a local or sub-group effect, poisonous instances only target groups, but do not affect any other instances – i.e. the cost of recourse of untargeted instances should not change: $\mathbb{E}_{\vec{x}\sim\phi'}\left[\theta \circ \text{CF}(\vec{x}, h_{T(\mathcal{D}_{\text{orig}})})\right] \approx \mathbb{E}_{\vec{x}\sim\phi'}\left[\theta \circ \text{CF}(\vec{x}, h_{\mathcal{D}_{\text{orig}}})\right]$ where ϕ' denotes the density of all untargeted instances.
- The predictive performance of the classifier is not significantly lowered when adding poisonous instances[1], i.e. $\arg\min_T \mathbb{E}[\ell(h_{T(\mathcal{D}_{\text{orig}})}(\vec{x}_i)), y_i)]$ where $\ell(\cdot)$ denotes some suitable loss function such as the zero-one loss.

[1]However, because the decision boundary is changed, some drop in the predictive performance might be inevitable.

Considering all these, we formalize the finding of data poisoning $T(\cdot)$ as a multi-objective optimization problem:

$$\arg\min_{T} \Big(|T(\mathcal{D}_{\text{orig}}) \setminus \mathcal{D}_{\text{orig}}|, \tag{3a}$$

$$\mathbb{E}[\ell(h_{\mathcal{D}_{\text{poisoned}}}(\vec{x}_i)), y_i)], \tag{3b}$$

$$\sum_{(\vec{x}_i, y_i) \in T(\mathcal{D}_{\text{orig}}) \setminus \mathcal{D}_{\text{orig}}} -p_{\text{data}}(\vec{x}_i, y_i) \Big) \tag{3c}$$

$$\text{s.t. } \mathbb{E}_{\vec{x} \sim \phi} \left[\theta \circ \text{CF}(\vec{x}, h_{T(\mathcal{D}_{\text{orig}})}) \right] > \mathbb{E}_{\vec{x} \sim \phi} \left[\theta \circ \text{CF}(\vec{x}, h_{\mathcal{D}_{\text{orig}}}) \right] \tag{3d}$$

$$\mathbb{E}_{\vec{x} \sim \phi'} \left[\theta \circ \text{CF}(\vec{x}, h_{T(\mathcal{D}_{\text{orig}})}) \right] \approx \mathbb{E}_{\vec{x} \sim \phi'} \left[\theta \circ \text{CF}(\vec{x}, h_{\mathcal{D}_{\text{orig}}}) \right] \tag{3e}$$

In the following, we study a few (general) aspects and properties of Eq. (3) that will serve as a foundation for the data poisoning in Section 4.2.

Increasing the Cost of Recourse

As discussed in Section 3, the simplest way of achieving recourse is by means of the closest counterfactual as stated in Definition 1 – i.e. the smallest change that reaches/crosses the decision boundary. In this case, regarding data poisoning on a local level, it can be shown that, under some assumptions, the injection of a training sample on the decision boundary, increases the cost of recourse locally.

Theorem 1 (Increasing the Cost of Recourse for 1-NN Classifiers)**.** *Let $h_{\mathcal{D}}(\cdot)$ be 1-nearest neighbor classifier for some data set $\mathcal{D}$. For any $(\vec{x}_{orig}, y_{orig}) \in \mathcal{D}$, let $\vec{x}'$ denote the closest instance (assuming uniqueness) on the decision boundary under a proper norm $\theta(\cdot)$. Then, adding $(\vec{x}', y_{orig})$ to $\mathcal{D}$ increases the cost of recourse for $\vec{x}_{orig}$:*

$$\theta \circ CF(x, h_{\mathcal{D} \cup \{(\vec{x}', y_{orig})\}}) > \theta \circ CF(x, h_{\mathcal{D}}) \tag{4}$$

The proof of Theorem 1 is given in the appendix. Although a 1-NN classifier is somewhat simplistic, it is quite flexible and might be a good local approximation for many different types of classifiers. Therefore, Theorem 1 provides valuable insights on how to perform a local poisoning for locally increasing the cost of recourse. This will serve as a foundation for the data poisoning. (Algorithm 1) in Section 4.2.

Local Effects of Increasing the Cost of Recourse

It can be shown (see Theorem 2) that, under some assumptions, increasing the cost of recourse for one instance, also increases the cost of recourse for nearby instances, or at least does not harm them significantly.

Theorem 2 (Increasing the Cost of Recourse Affects Nearby Instances)**.** *Let $\mathcal{S} \subset \{\mathbb{R}^d\}^n$ be a set of instances that are classified as $h(\vec{x}_i) = 0$, and $\mathcal{R} \subset \{\mathbb{R}^d\}^m$ the set of recourse instances – i.e. any instance in $\mathcal{R}$ provides recourse (Remark 2) for any instance in $\mathcal{S}$. Furthermore, let δ_i denote the smallest (i.e. $\theta(\cdot) = \|\cdot\|_2$) possible cost of recourse for $\vec{x}_i \in \mathcal{S}$.*

Assuming that some data poisoning yields a new set of recourse instances $\mathcal{R}'$, but leaves $\mathcal{S}$ unchanged – we denote the new smallest cost of recourse by δ'_i. For any pair of instances $\vec{x}_i, \vec{x}_j \in \mathcal{S}$ with $\delta_i < \delta_j$, and $\|\vec{x}_i - \vec{x}_j\|_2 < \delta_i$, where the cost of recourse for $\vec{x}_i$ increased by $\lambda > \|\vec{x}_i - \vec{x}_j\|_2$ – i.e. $\delta'_i = \delta_i + \lambda$ –, it holds that:

$$\delta'_j \geq \delta_i + \lambda - \|\vec{x}_i - \vec{x}_j\|_2 \tag{5a}$$

$$\delta'_j \geq \delta_j \quad \textit{if } \lambda \geq \frac{2\delta_j}{\delta_i} + \|\vec{x}_i - \vec{x}_j\|_2 \tag{5b}$$

The proof of Theorem 2 is given in the appendix. Theorem 2 assumes that the data poisoning $T(\cdot)$ changes the decision boundary of the classifier $h(\cdot)$ such that existing samples are not misclassified. Only the set of possible counterfactuals $\mathcal{D}_{\text{CF}}$ – i.e. the image of counterfactual generation function $\text{CF}(\cdot)$ – is changed by the poisoning $T(\cdot)$. This is a reasonable assumption if the classifier $h(\cdot)$ is sufficiently flexible to model the new (i.e. poisoned) data set – e.g. Deep Neural Networks or tree ensembles are likely able to do this. This implies that increasing the cost of recourse for one point by some specific amount also affects the cost of recourse for nearby points. In particular, Eq. (5a) states a lower bound on the new cost of recourse for nearby instances. Furthermore, if this increase λ is large enough, the cost of recourse for nearby instances increases as well (see Eq. (5b)). This provides us with evidence that it can be sufficient to only focus on instances that have a low cost of recourse (i.e. are close to the decision boundary) when creating poisonous instances and therefore allows the adversary to keep the number of poisonous instances small. We will use these insights to formalize the data poisoning in Section 4.2. In addition, Theorem 2 can also be interpreted as evidence that data poisoning on a group-group level where the groups overlap might be challenging and not always possible depending on the counterfactual (i.e. recourse) generation mechanism $\text{CF}(\cdot, \cdot)$ – we will observe this in the empirical evaluation in Section 5.

4.2 Data Poisoning on Counterfactual Explanations

Based on the findings from Section 4.1, we formalize a method (see Algorithm 1) for generating poisonous instances, that are added to the training set, to increase the cost of recourse – i.e. Algorithm 1 constitutes and implementation of $T(\cdot)$ from Definition 2. Note that this method supports data poisonings on different levels (i.e. local, sub-groups, and global levels). For practical purposes, we assume that we have (or created) a set of samples $\mathcal{D}_{\text{target}} = \{(\vec{x}_i, y)\}$ all with the same prediction $y \in \mathcal{Y}$, where $\vec{x}_i \sim \phi$, from the region in data space that is targeted by the poisoning – e.g. this could be a subset of the training data set. Furthermore, w.l.o.g. we assume that the negative (i.e. unfavorable) class is $y = 0$. We propose to fix the number of poisonous instances $\{\vec{z}_i\}$ and approximate the original multi-objective optimization problem Eq. (3) by Eq. (6). Note that the objective Eq. (6a) replaces the original plausibility constraint Eq. (3c). That is, we construct poisonous instances that are very similar to the given samples $\mathcal{D}_{\text{target}}$ – note that it was observed [Chakraborty *et al.*, 2021] that small perturbations often remain unnoticed by the human, which gave rise to adversarial attacks [Chakraborty *et al.*, 2021; Rauber *et al.*, 2020]. By this, we aim to make the poisonous instances more difficult to detect.

$$\arg\min_{\{\vec{z}_i\}} \Big(\arg\min_{\vec{x}_j \in \mathcal{D}_{\text{target}}} \|\vec{z}_i - \vec{x}_j\|_p, \tag{6a}$$

$$\mathbb{E}[\ell(h_{\mathcal{D}_{\text{poisoned}}}(\vec{x}_i)), y_i)] \Big) \tag{6b}$$

$$\text{s.t.} \sum_{\vec{x}_i \in \mathcal{D}_{\text{target}}} \theta \circ \text{CF}(\vec{x}_i, h_{\mathcal{D}_{\text{poisoned}}}) > \sum_{\vec{x}_i \in \mathcal{D}_{\text{target}}} \theta \circ \text{CF}(\vec{x}_i, h_{\mathcal{D}_{\text{orig}}}) \tag{6c}$$

$$\mathbb{E}_{\vec{x} \sim \phi'} \left[\theta \circ \text{CF}(\vec{x}, h_{\mathcal{D}_{\text{poisoned}}})\right] \approx \mathbb{E}_{\vec{x} \sim \phi'} \left[\theta \circ \text{CF}(\vec{x}, h_{\mathcal{D}_{\text{orig}}})\right] \tag{6d}$$

$$\text{where } \mathcal{D}_{\text{poisoned}} = \mathcal{D}_{\text{orig}} \cup \{(\vec{z}_i, y)\} \tag{6e}$$

We propose to compute an approximate solution to Eq. (6) by constructing instances $\vec{z}_i$ that are on the decision boundary or behind it and are close to samples in $\mathcal{D}_{\text{target}}$. We construct such instances by computing closest counterfactual explanations of samples in $\mathcal{D}_{\text{target}}$ that are close to the decision boundary:

$$\vec{z}_i = \vec{x} + \text{CF}(\vec{x}, h) \quad (\vec{x}, y) \sim \text{weighted\_sample}\left(\mathcal{D}_{\text{target}} \mid \{\delta_i\}\right) \tag{7}$$

where we estimate the distance δ_i to the decision boundary by computing a closest counterfactual – i.e. $\delta_i = \text{CF}(\vec{x}_i, h)$. According to Theorem 1 and Theorem 2, such instances $\vec{z}_i$ are good candidates for increasing the cost of recourse in the targeted area $\phi(\cdot)$. By changing the cost of recourse for samples that are close to the decision boundary, we can maximize the impact (see Theorem 2) of the fixed number of poisonous instances and consequently keep the number of needed poisonous instances low. The predictive performance objective Eq. (6b) and the constraint Eq. (6d), stating that the cost of recourse should not change for untargeted instance, are both considered implicitly in Eq. (7) – because the poisonous instances $\vec{z}_i$ are close to the targeted instances in $\mathcal{D}_{\text{target}}$, a sufficiently flexible classifier should not change its behavior in other regions in data space. Furthermore, in order to increase the robustness of the poisoning, we propose to use not only a single closest counterfactual in Eq. (7) but a set of diverse closest counterfactual explanations. We also propose to extend the counterfactual direction $\vec{\delta}_{\text{cf}}$ by multiplying it with a factor $\alpha > 1$ – by this, we aim to create poisonous instances on the other side of the decision boundary, yielding an even larger increase in the cost of recourse. The pseudocode of data poisoning is given in Algorithm 1.

The runtime of Algorithm 1 can be broken down to $\mathcal{O}(n \cdot k \cdot \rho)$ where n and k are the hyper-parameters of the algorithm referring to the number of poisonous instances, and ρ denotes the computational complexity (i.e. runtime) for computing a counterfactual which is utilized to construct the poisonous sample(s) (see line 8 in Algorithm 1) – note ρ is likely to differ between different counterfactual generation mechanisms. Consequently, the runtime of Algorithm 1 scales linearly with the number of requested poisonous samples.

Algorithm 1 Data Poisoning for Increasing Cost of Recourse

Input: Samples $\mathcal{D}_{\text{target}=\{(\vec{x}_i, y)\}}$ from the data space region that is targeted; Mechanism $\text{CF}(\cdot, h)$ for generating closest counterfactuals; Number n of poisonous instances; Parameters: k, b

Output: Poisonous instances D_{poison}

1: $\{\delta_i = \theta \circ \text{CF}(\vec{x}_i, h) \quad \forall \vec{x}_i \in \mathcal{D}_{\text{target}}\}$ ▷ Estimate distances to decision boundary
2: $D_{\text{poison}} = \{\}$
3: **for** n-times **do**
4:     $(\vec{x}, y) \sim$ weighted_sampling $(\mathcal{D}_{\text{target}}, \{\delta_i\})$ ▷ Sampling
5:     $\Delta_{\text{cf}} = \text{CF}(\vec{x}, h; k)$ ▷ k diverse closest CFs
6:     **for** $\vec{\delta}_{\text{cf}} \in \Delta_{\text{cf}}$ **do**
7:         **for** $\alpha \in [1, b]$ **do**
8:             $\vec{z} = \vec{x} + \alpha * \vec{\delta}_{\text{cf}}$ ▷ Add samples along $\vec{\delta}_{\text{cf}}$
9:             $D_{\text{poison}} = D_{\text{poison}} \cup \Big\{ (\vec{z}, y) \Big\}$
10:         **end for**
11:     **end for**
12: **end for**
13: $D_{\text{train}} = D_{\text{train}} \cup D_{\text{poison}}$ ▷ Add D_{poison} to training set

5 Experiments

We empirically evaluate the robustness of counterfactual explanations (i.e. recourse) against data poisonings by applying the data poisoning Algorithm 1 from Section 4 on combinations of several different benchmark data sets, classifiers, and state-of-the-art counterfactual explanation generation methods and toolboxes. The Python implementation of the experiments (including all data sets), is available on GitHub[2].

5.1 Benchmark Data Sets

We consider three data sets: The "Diabetes" data set [Efron *et al.*, 2004] (denoted as *Diabetes*) contains data from 442 diabetes patients. For each patient, 9 numeric attributes are available – in addition, the sensitive attribute "sex" of each patient is given. The target for predictions is a binarized quantitative measure of disease progression one year after baseline.The "Communities & Crime" data set [Dheeru and Taniskidou, 2017] (denoted as *Crime*) contains 1994 socio-economic data, including the sensitive attribute "race", records from the USA. Following the pre-processing as suggested in [Le Quy *et al.*, 2022], we are left with 100 encoded attributes that are used to predict the crime rate (low vs. high). The "German Credit Data set" [Ger, 1994] (denoted as *Credit*) is a data set for loan approval and contains 1000 instances each annotated with 7 numerical and 13 categorical attributes, including the sensitive attribute "sex", with a binary target value ("accept" or "reject"). We use only the seven numerical features.

5.2 Machine Learning Classifiers

We consider a diverse set of ML classifiers $h(\cdot)$: deep neural networks (denotes as *DNN*), random forests (denoted as *RNF*), and linear SVM 's(denoted as *SVC*).

[2] https://github.com/andreArtelt/DataPoisoningCounterfactuals

Classifier	Data set	Nearest $^{10\%}$	DiCE $^{10\%}$	FACE $^{40\%}$	Proto $^{20\%}$
SVC	Credit	4.94	7.59	−0.91	8.33
	Diabetes	2.28	2.23	−0.03	2.46
	Crime	9.81	9.45	15.24	14.22
RNF	Credit	3.56	4.9	2.35	6.05
	Diabetes	0.64	0.91	−0.24	1.84
	Crime	4.32	6.85	11.17	13.65
DNN	Credit	1.06	1.93	2.36	0.38
	Diabetes	0.68	0.66	1.6	1.24
	Crime	5.18	6.65	11.81	9.04

Table 1: Difference in the cost of recourse: no vs. *global poisoning*. The amount of poisoning (percentage %) is specified for each method separately. Positive numbers denote an increase in the cost of recourse, while negative numbers denote the opposite. We report the median (over all folds) rounded to two decimal places.

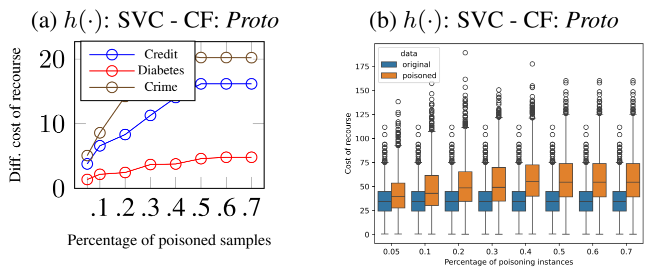


Figure 2: Global data poisoning: a) Median (over all folds) difference in the cost of recourse vs. percentage of poisoned instances.b) Cost of recourse vs. percentage of poisoned instances.

5.3 Counterfactual Generation Methods

Given the large amount of different counterfactual generation methods and toolboxes [Guidotti, 2022], we evaluate the data poisoning method on a set of very different and popular state-of-the-art methods and toolboxes for computing counterfactual explanations (i.e. recourse) – i.e. all of these methods define and compute counterfactual recourse in slightly different ways: Nearest Training Sample (denoted as *Nearest*), as simple baseline; FACE [Poyiadzi *et al.*, 2020] for computing plausible counterfactuals; Counterfactuals guided by Prototypes [Looveren and Klaise, 2021] (denoted as *Proto*) as another, but different, method for computing plausible counterfactuals; DiCE [Mothilal *et al.*, 2020] for diverse closest counterfactuals.

5.4 Setup

In all experiments (as described below), we use DiCE [Mothilal *et al.*, 2020] as a counterfactual generation mechanism for computing three diverse closest counterfactual explanations (i.e. $k = 3$ in Algorithm 1), that are as close as possible to the original sample. The cost of recourse $\theta(\cdot)$ is implemented by ℓ_1-norm – i.e. $\theta(\vec{\delta}_{\text{cf}}) = \|\vec{\delta}_{\text{cf}}\|_1$. Furthermore, all experiments are run in 5-fold cross-validation. In all global and sub-group data poisoning scenarios, we evaluate different amounts (5% to 70%) of poisonous instances – i.e. original training data + poisoned instances. We not only evaluate the influence of the number of poisonous instances on the cost of recourse, but also their influence on the classifiers' predictive performance – some of these results are shown in Figures 2,3 while the rest can be found in the appendix.

Classifier	Data set	Nearest $^{10\%}$	DiCE $^{40\%}$	FACE $^{40\%}$	Proto $^{50\%}$
SVC	Credit	0.78	0.08	0.73	9.79
	Diabetes	1.23	0.54	0.02	2.85
	Crime	5.7	7.51	0.46	12.4
RNF	Credit	0.48	0.12	3.18	0.07
	Diabetes	0.72	1.59	0.05	1.09
	Crime	4.62	6.96	14.32	1.27
DNN	Credit	0.13	0.47	0.22	9.11
	Diabetes	1.1	0.75	−0.2	2.29
	Crime	9.06	10.09	1.86	0.47

Table 2: Difference in the cost of recourse between protected groups: no vs. poisoning on a *sub-group level*. The amount of poisoning (percentage %) is specified for each method separately. Positive numbers denote an increase in the difference in the cost of recourse, while negative numbers denote the opposite. We report the median (over all folds) rounded to two decimal places.

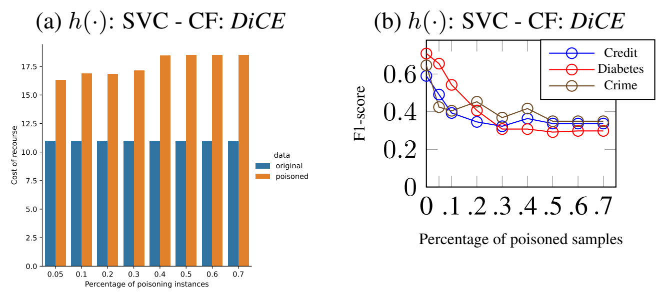


Figure 3: Sub-group data poisoning on the *Crime* data set – no poisoning vs. different percentages of poisoned instances. a) Difference in the median cost of recourse between the two protected groups. b) Median (over all folds) F1-score of the classifier.

Data poisoning on a global level For every, negative classified, sample in the test set, we compute a counterfactual explanation. We evaluate the global increase in the cost of recourse, by computing the difference in the cost of recourse:

$$\begin{aligned} &\theta \circ \text{CF}(\vec{x}_i, h_{\mathcal{D}_{\text{poisoned}}}) - \theta \circ \text{CF}(\vec{x}_i, h_{\mathcal{D}_{\text{orig}}}) \\ &\forall\, \vec{x}_i, y_i \in \mathcal{D}_{\text{test}},\ h(\vec{x}_i) = 0 \end{aligned} \tag{8}$$

A positive score Eq. (8) means an increase in the recourse cost – due to the data poisoning –, while a negative or near zero score implies no change or a lower cost of recourse. We report the median of Eq. (8) in order to avoid the influence of outliers on the results. In Table 1 we show the increase in the cost of recourse together with the amount of poisoning that was necessary for observing a significant increase – more detailed results are provided in the appendix.

Data poisoning on a sub-group level We evaluate the effect of the data poisoning on a sub-group level by considering sub-groups created based on the sensitive attribute – note that this is a reasonable but only one out of many possible ways how sub-groups might be created. We apply the data poisoning to poison instances from one protected group only, assuming that the sensitive attribute of each instance is known. By this, we aim to increase the difference in the cost of recourse between the two protected groups – note that this can be interpreted as introducing or increasing group-unfairness in recourse [Artelt and Hammer, 2023;

(a) CF: *Nearest* (b) CF: *Proto*

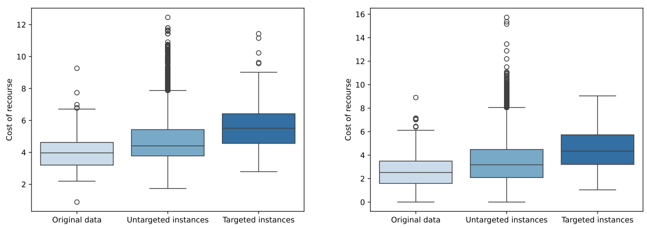


Figure 4: *Local* data poisoning: Cost of recourse (over all test samples) in the case of the diabetes data set and a DNN classifier. Cost of recourse without any data poisoning, of untargeted instances and targeted instances in a local data poisoning.

Von Kügelgen *et al.*, 2022]. For every, negative classified, sample in the test set (no matter to which sub-group it belongs), we compute a counterfactual explanation. We evaluate the difference in the cost of recourse between the two sub-groups as follows:

$$\underbrace{\|\theta \circ \mathrm{CF}(\vec{x}_i|s=0, h_{\mathcal{D}_{\text{poisoned}}}) - \theta \circ \mathrm{CF}(\vec{x}_i|s=1, h_{\mathcal{D}_{\text{poisoned}}})\|}_{\text{Median difference in the cost of recourse \textbf{under} data poisoning}} - \underbrace{\|\theta \circ \mathrm{CF}(\vec{x}_i|s=0, h_{\mathcal{D}_{\text{orig}}}) - \theta \circ \mathrm{CF}(\vec{x}_i|s=1, h_{\mathcal{D}_{\text{orig}}})\|}_{\text{Median difference in the cost of recourse \textbf{without} data poisoning}} \quad \forall\, \vec{x}_i \in \mathcal{D}_{\text{test}}\; h(\vec{x}_i) = 0 \tag{9}$$

where we denote the sensitive attribute as s – i.e. $\vec{x}_i|s=0$ means that we only consider x_i if its sensitive attribute is equal to zero. Note that, a positive score Eq. (9) means that the difference in the cost of recourse between the protected groups increased, while a negative number means that the difference in the cost of recourse between the protected groups decreased. Furthermore, note that we use the median (over all folds) in Eq. (9) to avoid the influence of outliers. We show the results together with the minimum amount of poisoning that was necessary for observing a significant increase in Table 2 – more detailed results are provided in the appendix.

Data poisoning on a local level We compute a local data poisoning for every, negative classified, sample in the test set. However, because of computational limitations, we only evaluate a single scenario considering a DNN classifier on the diabetes data set. Some of the results are shown in Figure 4 – more detailed results are given in the appendix.

5.5 Results & Discussion

We observe that in almost all scenarios, on local as well as on global levels (see Tables 1,2), even a relatively small amount of poisonous instances, added to the training data set, leads to a significant increase in the cost of recourse. Increasing the number of poisonous instances leads to an even larger increase in the cost of recourse (see Figure 2 and the appendix). However, we observe differences in the necessary amount of poisonous instances between different counterfactual generation methods and toolboxes. For FACE [Poyiadzi *et al.*, 2020] and counterfactuals guided by prototypes [Looveren and Klaise, 2021], we need significantly more poisonous instances for increasing the cost of recourse – in the case of FACE, we even have a few settings where the poisoning does not work which is likely due to the special nature of FACE that might require a different strategy. Since both methods focus on plausibility, this might be an indicator that additional plausibility constraints can act as a beneficial regularize for increased stability – similar to what is reported in [Artelt *et al.*, 2021] for robustness concerning input perturbations. In the case of sub-groups, we observe (see Table 2) a similar effect. However, the increases are not as large as those for the local or global poisoning and often the necessary amount poisonous instances is also larger compared to the global poisoning – this is quite likely due to a strong overlap of the distributions of the sub-groups which makes it difficult (see Theorem 2) to just change the cost of recourse for one group but not for the other. Furthermore, it is worth noting that in many cases the initial difference in the cost of recourse is already quite significant (see Figure 3). Concerning the classifiers' performance, we observe the expected results that classifiers' predictive performance is decreasing the more poisoned instances are added (see Figure 3) – i.e. for a global data poisoning the decrease in predictive performance is worse than for sub-group or local data poisonings. Altogether, these observations demonstrate the vulnerability of existing counterfactual explanations and state-of-the-art methods and toolboxes to data poisonings.

6 Conclusion & Summary

In this work, we studied the robustness of counterfactual explanations against data poisonings. For this purpose, we identify and formalize data poisoning to increase the cost of recourse on different levels (local - global). Adding poisonous instances to the training data set lead to an increase in the cost of recourse for the final classifier. We empirically evaluated the effect of data poisoning on several different classifiers, benchmark data sets, and different popular and state-of-the-art counterfactual generation methods. We observed that in almost all cases the injection of already a small portion of poisonous instances into the training data set leads to a significant increase in the cost of recourse – on a global level as well as on a local level.

These findings demonstrate how easily existing classifiers and state-of-the-art counterfactual generation methods and toolboxes can be fooled by manipulating the training data. Since counterfactual explanations are a widely used method for providing recourse and analyzing ML-based models, (malicious) manipulations of those directly harm the user and consequently significantly reduce users' trust in this XAI method. Thus, this work demonstrates the necessity of more robust counterfactual generation methods an toolboxes as well as defense mechanisms against malicious data manipulations – we leave this as future research together with the exploration of other data poisoning mechanisms.

Acknowledgments

This research was supported by the Ministry of Culture and Science NRW (Germany) as part of the Lamarr Fellow Network. This publication reflects the views of the authors only.

Disclaimer This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates ("JP Morgan"), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

[Adadi and Berrada, 2018] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

[Arrieta *et al.*, 2020] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[Artelt and Hammer, 2023] André Artelt and Barbara Hammer. "explain it in the same way!" – model-agnostic group fairness of counterfactual explanations. In Amir Ofra, Tim Miller, and Hendrik Baier, editors, *Workshop on XAI*, 2023.

[Artelt *et al.*, 2021] André Artelt, Valerie Vaquet, Riza Velioglu, Fabian Hinder, Johannes Brinkrolf, Malte Schilling, and Barbara Hammer. Evaluating robustness of counterfactual explanations. In *2021 IEEE Symposium Series on Computational Intelligence*, pages 01–09. IEEE, 2021.

[Baniecki and Biecek, 2022] Hubert Baniecki and Przemyslaw Biecek. Manipulating shap via adversarial data perturbations (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12907–12908, 2022.

[Baniecki and Biecek, 2024] Hubert Baniecki and Przemyslaw Biecek. Adversarial attacks and defenses in explainable artificial intelligence: A survey. *Information Fusion*, page 102303, 2024.

[Baniecki *et al.*, 2022] Hubert Baniecki, Wojciech Kretowicz, and Przemyslaw Biecek. Fooling partial dependence via data poisoning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 121–136. Springer, 2022.

[Bojchevski and Günnemann, 2019] Aleksandar Bojchevski and Stephan Günnemann. Adversarial attacks on node embeddings via graph poisoning. In *International Conference on Machine Learning*, pages 695–704. PMLR, 2019.

[Brown and Kvinge, 2023] Davis Brown and Henry Kvinge. Making corgis important for honeycomb classification: Adversarial attacks on concept-based explainability tools. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 620–627, 2023.

[Byrne, 2019] Ruth M. J. Byrne. Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. In *IJCAI-19*, 2019.

[Chakraborty *et al.*, 2021] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, 6(1):25–45, 2021.

[Commission *et al.*, 21 04 2021] European Commission, Directorate-General for Communications Networks, Content, and Technology. Proposal for a Regulation laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union legislative acts. *Policy and Legistlation*, 21-04-2021.

[Council of European Union, 2016] Council of European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC. *Official Journal of the European Union*, L 119:4.5, 2016.

[Dheeru and Taniskidou, 2017] Dua Dheeru and E Karra Taniskidou. Uci machine learning repository. 2017.

[Dwivedi *et al.*, 2023] Rudresh Dwivedi, Devam Dave, Het Naik, Smiti Singhal, Rana Omer, Pankesh Patel, Bin Qian, Zhenyu Wen, Tejal Shah, Graham Morgan, et al. Explainable ai (xai): Core ideas, techniques, and solutions. *ACM Computing Surveys*, 55(9):1–33, 2023.

[Efron *et al.*, 2004] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. 2004.

[Ger, 1994] Statlog (german credit data) data set, 1994.

[Guidotti, 2022] Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, pages 1–55, 2022.

[Ho *et al.*, 2022] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022.

[Karimi *et al.*, 2021] Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Computing Surveys*, 2021.

[Le Quy *et al.*, 2022] Tai Le Quy, Arjun Roy, Vasileios Iosifidis, Wenbin Zhang, and Eirini Ntoutsi. A survey on datasets for fairness-aware machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1452, 2022.

[Lin *et al.*, 2021] Jing Lin, Long Dang, Mohamed Rahouti, and Kaiqi Xiong. Ml attack models: adversarial attacks and data poisoning attacks. *arXiv preprint arXiv:2112.02797*, 2021.

[Looveren and Klaise, 2021] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. pages 650–665, 2021.

[Mehrabi *et al.*, 2021] Ninareh Mehrabi, Muhammad Naveed, Fred Morstatter, and Aram Galstyan. Exacerbating algorithmic bias through fairness attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8930–8938, 2021.

[Mishra *et al.*, 2021] Saumitra Mishra, Sanghamitra Dutta, Jason Long, and Daniele Magazzeni. A survey on the robustness of feature importance and counterfactual explanations. *arXiv preprint arXiv:2111.00358*, 2021.

[Molnar, 2019] Christoph Molnar. *Interpretable Machine Learning*. 2019.

[Mothilal *et al.*, 2020] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

[Poyiadzi *et al.*, 2020] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. Face: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020.

[Rauber *et al.*, 2020] Jonas Rauber, Roland Zimmermann, Matthias Bethge, and Wieland Brendel. Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in pytorch, tensorflow, and jax. *Journal of Open Source Software*, 5(53):2607, 2020.

[Rawal *et al.*, 2021] Atul Rawal, James Mccoy, Danda B Rawat, Brian Sadler, and Robert Amant. Recent advances in trustworthy explainable artificial intelligence: Status, challenges and perspectives. *IEEE Transactions on Artificial Intelligence*, 1(01):1–1, 2021.

[Riveiro and Thill, 2022] Maria Riveiro and Serge Thill. The challenges of providing explanations of ai systems when they do not behave like users expect. In *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*, pages 110–120, 2022.

[Shan *et al.*, 2023] Shawn Shan, Wenxin Ding, Josephine Passananti, Haitao Zheng, and Ben Y Zhao. Prompt-specific poisoning attacks on text-to-image generative models. *arXiv preprint arXiv:2310.13828*, 2023.

[Slack *et al.*, 2021] Dylan Slack, Anna Hilgard, Himabindu Lakkaraju, and Sameer Singh. Counterfactual explanations can be manipulated. *Advances in Neural Information Processing Systems*, 34:62–75, 2021.

[Solans *et al.*, 2020] David Solans, Battista Biggio, and Carlos Castillo. Poisoning attacks on algorithmic fairness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 162–177. Springer, 2020.

[Tolpegin *et al.*, 2020] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*, pages 480–501. Springer, 2020.

[Verma *et al.*, 2020] Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review, 2020.

[Virgolin and Fracaros, 2023] Marco Virgolin and Saverio Fracaros. On the robustness of sparse counterfactual explanations to adverse perturbations. *Artificial Intelligence*, 316:103840, 2023.

[Von Kügelgen *et al.*, 2022] Julius Von Kügelgen, Amir-Hossein Karimi, Umang Bhatt, Isabel Valera, Adrian Weller, and Bernhard Schölkopf. On the fairness of causal algorithmic recourse. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9584–9594, 2022.

[Wachter *et al.*, 2017] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[Yang *et al.*, 2023] Ziqing Yang, Xinlei He, Zheng Li, Michael Backes, Mathias Humbert, Pascal Berrang, and Yang Zhang. Data poisoning attacks against multimodal encoders. In *International Conference on Machine Learning*, pages 39299–39313. PMLR, 2023.

[Zhao *et al.*, 2023] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.