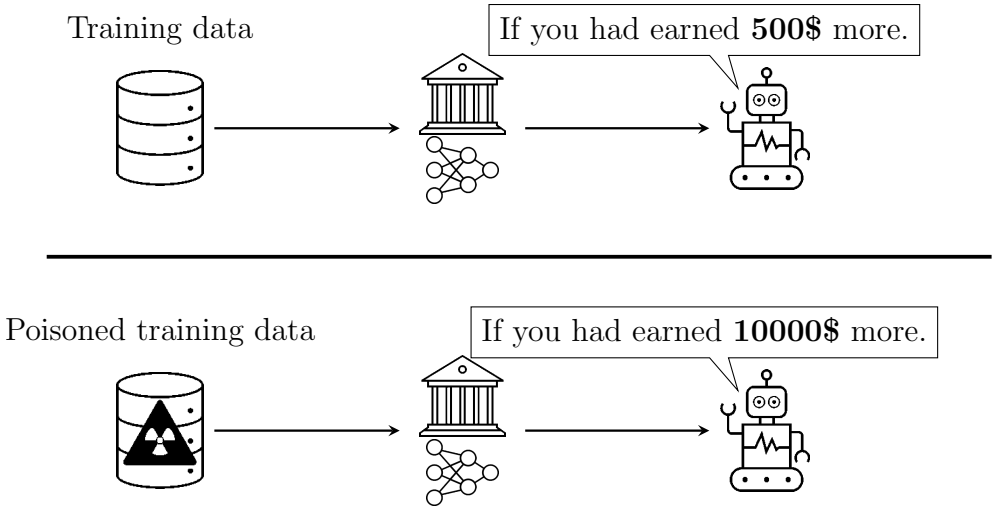
Graphical Abstract

The Effect of Data Poisoning on Counterfactual Explanations

André Artelt, Shubham Sharma, Freddy Lecué, Barbara Hammer



Highlights

The Effect of Data Poisoning on Counterfactual Explanations

André Artelt, Shubham Sharma, Freddy Lecué, Barbara Hammer

- Formalizing the novel problem of data poisoning of counterfactual explanations.
- Investigating the effect of data poisoning on counterfactuals.
- Many state-of-the-art methods are vulnerable to data poisoning.

The Effect of Data Poisoning on Counterfactual Explanations

André Artelt[a,b], Shubham Sharma[c], Freddy Lecué[d], Barbara Hammer[a]

[a] *Bielefeld University, Bielefeld, Germany*
[b] *University of Cyprus, Nicosia, Cyprus*
[c] *J.P. Morgan AI Research, New York City, USA*
[d] *Inria, Sophia Antipolis, France*

Abstract

Counterfactual explanations are a widely used approach for examining the predictions of black-box systems. They can offer the opportunity for computational recourse by suggesting actionable changes on how to alter the input to obtain a different (i.e., more favorable) system output. However, recent studies have pointed out their susceptibility to various forms of manipulation.

This work studies the vulnerability of counterfactual explanations to data poisoning. We formally introduce and investigate data poisoning in the context of counterfactual explanations for increasing the cost of recourse on three different levels: locally for a single instance, a sub-group of instances, or globally for all instances. In this context, we formally introduce and characterize data poisonings, from which we derive and investigate a general data poisoning mechanism. We demonstrate the impact of such data poisoning in the critical real-world application of explaining event detections in water distribution networks. Additionally, we conduct an extensive empirical evaluation, demonstrating that state-of-the-art counterfactual generation methods and toolboxes are vulnerable to such data poisoning. Furthermore, we find that existing defense methods fail to detect those poisonous samples.

Keywords:
XAI, Counterfactual Explanations, Data Poisoning

Email address: aartelt@techfak.uni-bielefeld.de (André Artelt)

1. Introduction

Real-world Artificial Intelligence (AI-) and Machine Learning (ML-) based systems [1, 2] show an impressive performance but are still not perfect – e.g., failures, issues of fairness, and vulnerability to manipulations can cause harm. ML-based systems can be manipulated or attacked at different stages to harm the general predictive accuracy, introduce failures, increase the unfairness, or place backdoors in the system. In this context, adversarial attacks [3], backdoor attacks [4], and data poisoning [5] constitute the most popular methods for manipulating ML-based systems.

Adversarial attacks [3, 6] implement an attack occurring at run-time that aims at imperceptible input manipulations, inducing system failures such as the computation of wrong outputs.

Backdoor attacks [4], however, affect ML-based systems in the training stage and refer to the implementation of different model behavior that becomes only active when a certain pattern (the backdoor) is present in the input. Backdoor attacks can be realized by data poisoning [7] or by using a special loss function that encodes the backdoor behavior [4]. Note that the latter implementation approach of using a special loss function poses strong assumptions on the attacker's capabilities.

Like backdoor attacks, data poisoning attacks [5] affect ML-based systems in the training stage by manipulating training samples or adding new instances such that, for instance, the predictive performance (e.g, accuracy) of the final trained model decreases [8, 9], or fairness issues arise [10]. Note that, unlike adversarial attacks happening at inference time, data poisoning and backdoor attacks integrate manipulations into the final model and consequently impact its internal reasoning. More specifically, data poisoning can be performed offline [8] or online [9], and only small modifications are made to the training data, such as changing labels, removing samples, or adding new instances, which are likely to remain unnoticed. This poses a real threat in practice because nowadays many large models are trained on huge (internet-based) data sets [1, 2], 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 [11, 12, 13].

Given the threat of failures (intentionally caused or not), the transparency of such 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. This was also recognized by the policymakers and therefore found its way into legal regulations such as the EU's GDPR [14] or the EU AI act [15]. Explanations are a popular way of achieving transparency and shaping the field of eXplainable AI (XAI) [16, 17, 18]. Nowadays, many different explanation methods exist [16, 18, 19]. Counterfactual explanations [20] constitute a popular type of explanation method, which is inspired by human explanations [21] and can be used to provide recourse to individuals. More specifically, a counterfactual explanation provides (computational) recourse by stating actionable recommendations on how to alter the system's output in some desired way – e.g., how to turn a rejected loan application into an accepted one.

Critically, recent work has shown that many XAI methods are vulnerable to adversarial manipulations [22, 23, 24], undermining users' trust in XAI methods for revealing the internal logic of a model. In the context of counterfactual explanations, it was observed that they are neither robust to model changes [25], nor to input perturbations [26, 27, 28], and also not to adversarial training for implementing backdoors [24]. Data poisoning attacks on counterfactual explanations could make their recommended actions more costly – either for all individuals or for a subset of individuals. Since counterfactual explanations 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: The novelty of this work lies in the study of the vulnerability of counterfactual explanations to data poisoning. For this, we formalize and identify a set of data poisoning mechanisms for counterfactual explanations that injects a small set of realistic but poisonous data instances into the training data set such that the counterfactual explanations of the newly trained classifier are more costly to execute for the user. We consider the effect of data poisoning on three different levels: locally for an individual, a sub-group of individuals, and globally for all individuals. Most importantly, our proposed method is model-agnostic and only needs access to an interface for getting predictions and a mechanism for generating any closest counterfactuals, but no access or knowledge about model internals is required. We empirically find that existing state-of-the-art methods for computing counterfactuals are vulnerable to data poisoning and that classic

defense mechanisms fail to detect those poisonous training samples.

The remainder of this work is structured as follows: First, we discuss the related work (Section 2), and the necessary foundations of counterfactual explanations and (computational) recourse in Section 3. Next (in Section 4), we introduce our formalization of a data poisoning attack and introduce our proposed data poisoning attack. In Section 5, we demonstrate and illustrate the impact of such data poisoning attacks in the critical real-world application of explaining event detection in water distribution networks. In addition to that, we perform an extensive quantitative empirical evaluation of our proposed data poisoning attack on different benchmark scenarios in Section 6. Finally, this work closes with a summary and conclusion, including the discussion of limitations and possible directions for future research, in Section 7. Note that all proofs and more detailed evaluations of the experiments can be found in the appendix.

2. Related Work

2.1. General Data Poisoning

Existing data poisoning strategies from the literature impact ML models at the training stage, to affect predictive performance [8, 9] or fairness [10]. However, generic methods applicable to any black-box model are rare [8]. Most (advanced) data poisoning methods are either tailored towards specific models or classes of models (e.g., neural networks), domains (e.g. computer vision), or rely on assumptions such as feature extractors [8]. The Label-Flipping attack [8] (randomly) flips labels in the training data set, and constitutes the most general but also a rather simple, yet highly effective, data poisoning attack. As a reaction to such data poisoning attacks, potential countermeasures and defense strategies have also been proposed [5, 29, 30, 31]. Data sanitization methods [32] constitute among the most popular defense methods. Those methods usually rely on outlier and anomaly detection methods [31] for detecting the poisonous samples and either correct them [30] or remove them from the training data [31, 32]. Inspired by formal guarantees on the adversarial robustness, there also exist formal guarantees (under some assumptions) on the effectiveness of certain data sanitization defenses [29]. However, as noted in [33], data sanitization methods remain imperfect in the light of more advanced data poisoning attacks.

2.2. Vulnerability and Manipulations of XAI

Most existing work [22, 34, 35] on exposing the vulnerability of explanations is centered in the vision domain and focuses either on adversarial examples or model manipulation. In particular, it was shown that many XAI methods (incl. counterfactual explanations) are not robust with respect to adversarial manipulations in the input [22, 26, 36] – i.e., small and potentially imperceptible changes in the input can lead to completely different explanations. This also relates to fairness issues regarding individual fairness, where one would like to ensure that similar individuals get similar explanations [37, 38]. Similarly, it was also observed that explanations sometimes differ significantly between protected groups, violating group fairness [37, 39].

Only very little work considers (domain-independent) data poisoning of XAI methods [22]. For instance, there exist data poisoning against partial dependence plots [23], SHAP [40], and concept-based explainability tools [34]. The authors of [23] propose a genetic algorithm for perturbing the training data such that SHAP importance scores change. They assume that it is possible to modify (possibly) all samples in the training set, which might constitute a 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 [23]. A similar approach, with the same limitations, is proposed in [40] where partial dependence plots are targeted. Surprisingly, none of the existing works on data poisoning of explanations discuss any potential defense mechanisms.

In the context of counterfactual explanations, data poisoning attacks have not been considered so far. The closest study was done by [24], proposing an adversarial training objective for planting a backdoor in a neural network such that the cost of recourse decreases for a sub-group of individuals, violating group fairness requirements. 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 entire training procedure.

Literature Gap and Our Contribution. From the literature review, it becomes apparent that 1) data poisoning on counterfactual explanations has not been studied so far, and 2) existing work on data poisoning of XAI methods assumes that all training instances can be manipulated, which constitutes an unrealistic assumption in many attack scenarios. Furthermore, because data poisoning can lead to fairness issues regarding the model's prediction, it might also create fairness issues regarding the explanations – similar to the

group fairness violations created by the backdoor attacks as proposed in [24]. Finally, defense mechanisms against data poisoning attacks on XAI remain an open question.

In this work, we investigate data poisoning of counterfactual explanations by injecting additional training instances into the training data set. We argue that this constitutes a more realistic attack scenario in practice than the manipulation of existing training data instances as done in [40]. Furthermore, we do not alter the loss function used in the training procedure as done in the backdoor attack proposed by [24], posing strong assumptions about the attacker's capabilities. We not only investigate the effect of the data poisoning on the average explanation, but also investigate potential fairness issues in the counterfactual explanations arising from the poisoning. Finally, we empirically evaluate classic defense methods for detecting the poisonous training samples.

3. Foundations of Counterfactuals & Computational Recourse

A counterfactual explanation (often just called counterfactual) states actionable modifications 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 [41] – in the latter case, a counterfactual is also referred to as *recourse* [42], i.e. recommendations on how to turn the unfavorable into a favorable outcome. Because counterfactuals can mimic ways in which humans explain [21], they constitute one of the most popular explanation methods in literature and in practice [43, 44].

The two important properties of counterfactual explanations [20] are the:

1. *contrasting property:* requiring a change in the output of the system.
2. *cost of the counterfactual:* the cost and effort it takes to execute the counterfactual in the real world should be as low as possible to maximize its usefulness – e.g. counterfactuals with very few modifications or as small as possible modifications.

Both properties can be combined into an optimization problem (see Definition 1).

Definition 1 ((Closest) Counterfactual Explanation)**.** *Assume a classifier (binary or multi-class)* $h : \mathbb{R}^d \to \mathcal{Y}$ *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;* $C > 0$ *denotes the regularization strength balancing the two properties.*

The short-hand notation $\vec{\delta}_{cf} = CF(\vec{x}, h)$ *denotes the counterfactual (i.e. solution to Eq. (1))* $\vec{\delta}_{cf}$ *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, potentially requiring domain knowledge. Usually, the number of changes (e.g., changed features) and/or the magnitude of changes are considered as the cost of a counterfactual. Consequently, in many implementations and toolboxes [45], the p-norm is used as the default cost function for continuous features because it is most generic and can be easily adjusted by a custom weighting scheme.

$$\theta(\vec{\delta}_{\text{cf}}) = \|\vec{\delta}_{\text{cf}}\|_p \tag{2}$$

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

Because counterfactual explanations are usually requested in the case of an unfavorable outcome [41], we often refer to the cost of recourse as the quantity of interest in the remainder of this work.

Besides those two essential properties (contrasting and cost), there exist additional relevant aspects such as plausibility [46, 47], diversity [48], robustness [49, 50, 51], and fairness [39, 52, 53, 54]. which have been addressed in literature [45]. However, the basic formalization Eq. (1) is still very popular and widely used in practice [44, 45]. In this context, it is also important to note that the cost of recourse (i.e., cost of the counterfactual) remains the

[1]Here, "favorable" and "unfavorable" refer to specific predicted labels of the classifier.

central quantity of interest to the user because it denotes how easy it is for them to achieve their desired goal – e.g., turning an unfavorable prediction into a favorable one. While other aspects, such as the aforementioned robustness and plausibility, might also be relevant, the cost of recourse, which might be influenced by those additional aspects, remains the most critical quantity for the user and is therefore always evaluated when evaluating the quality of counterfactuals.

Finally, there exist numerous methods and implementations/toolboxes for computing counterfactual explanations in practice [45] – i.e. methods for computing solutions to Eq. (1). Note that most of those methods include some additional aspects such as plausibility and diversity:

- *Counterfactuals Guided by Prototypes* [46] is a method focusing on plausibility. Here, a set of plausible instances (prototypes) is used to pull the final counterfactual instance (i.e., $\vec{x}_{\text{cf}} = \vec{x}_{\text{orig}} + \vec{\delta}_{\text{cf}}$) closer to these plausible instances, making the final counterfactual more realistic.

- *DiCE* [48] is a model-agnostic method and Python toolbox for computing a set of diverse closest counterfactual explanations instead of a single one only.

- *Nearest Unlike Neighbor method* [55] constitutes a baseline method for computing plausible 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 Counterfactual Explanations

Given the fact that counterfactual explanations constitute a local explanation, potential data poisonings can have effects on different levels or areas in data space:

- *Global effect:* Explanations of all individuals are affected.

- *Sub-groups effect:* Explanations of only one or multiple sub-groups are affected

- *Local effect:* Explanations of only a single individual/instance are affected.

At the same time, data poisoning can aim for different effects on counterfactual explanations, such as hiding attributes or increasing the cost of recourse (i.e., cost of the counterfactual, see Remark 1). 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 commended actions more costly. Therefore, in this work, we focus on data poisoning for increasing the cost of recourse, but also evaluate possible side effects on validity, sparsity, and plausibility. Furthermore, note that the plausibility of counterfactuals can be easily checked, and therefore, attacks targeting the plausibility would be detected more easily than attacks targeting the cost of recourse, for which it is more difficult to say whether this corresponds to the ground truth or was manipulated.

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 $\phi(\cdot)$:*

$$\begin{aligned} &\mathbb{E}_{\vec{x}\sim\phi}\left[\theta \circ CF(\vec{x}, h_{\mathcal{D}_{poisoned}})\right] \gg \mathbb{E}_{\vec{x}\sim\phi}\left[\theta \circ CF(\vec{x}, h_{\mathcal{D}_{orig}})\right] \\ &\text{where } \mathcal{D}_{poisoned} = T(\mathcal{D}_{orig}) \end{aligned} \tag{3}$$

where $\circ$ denotes the function composition, $h_{\mathcal{D}}$ denotes a classifier that was derived from the data set $\mathcal{D}$, and $CF(\cdot,\cdot)$ refers to some given method for generating counterfactuals by solving Eq. (1)*.*

The density $\phi(\cdot)$ allows us to vary the level of the poisoning – e.g., for a global effect, we could use a class-wise density for targeting all instances from a specific (unfavorable) 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.

In this work, we focus on data poisoning attacks $T(\cdot)$ that add new (poisonous) instances to the training data set to increase the cost of recourse (Definition 2). In this context, we formally define (see Definition 3) a *poisonous data set* [56] for increasing the cost of recourse – i.e., a data set that increases the cost of recourse (Definition 2) if added to the original training data $\mathcal{D}_{\text{orig}}$.

Definition 3 (Poisonous Data Set)**.** *For a 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 say that a data set $\mathcal{D}_{poison} \subset \{\mathcal{X} \times \mathcal{Y}\}^m$ is recourse poisoning iff a classifier $h : \mathcal{X} \to \mathcal{Y}$ trained on $\mathcal{D}_{poison} \cup \mathcal{D}_{orig}$ shows an increase in the cost of recourse (Definition 2):*

$$\mathbb{E}_{\vec{x} \sim \phi}\left[\theta \circ CF(\vec{x}, h_{\mathcal{D}_{poison} \cup \mathcal{D}_{orig}})\right] \gg \mathbb{E}_{\vec{x} \sim \phi}\left[\theta \circ CF(\vec{x}, h_{\mathcal{D}_{orig}})\right] \tag{4}$$

Consequently, we write the data poisoning mechanisms $T(\cdot)$ (Definition 2) as follows:

$$T(\mathcal{D}_{\text{orig}}) = \mathcal{D}_{\text{orig}} \cup \mathcal{D}_{\text{poison}} \tag{5}$$

where $\mathcal{D}_{\text{poison}}$ refers to a poisonous data set from Definition 3.

From a practical point of view, besides increasing the cost of recourse (as stated in Definition 2), it is desirable for the poisonous data set $\mathcal{D}_{\text{poison}}$ (Definition 3) to have the following properties:

1. The number of poisonous instances $\mathcal{D}_{\text{poison}}$ (i.e., necessary poisoning budget) should be kept to a minimum:

$$\arg\min \ |\mathcal{D}_{\text{poison}}| \tag{6}$$

2. The poisonous instances $\mathcal{D}_{\text{poison}}$ are realistic – i.e., they are on the data manifold $p_{\text{data}}(\cdot)$ and have a high likelihood:

$$\arg\max \, p_{\text{data}}(\vec{x}_i, y_i) \quad \forall\, (\vec{x}_i, y_i) \in \mathcal{D}_{\text{poison}} \tag{7}$$

3. In the case of aiming for a local or sub-group effect, poisonous instances only target those specified groups/areas, but do not affect any other instances – i.e. the cost of recourse of untargeted instances should not change (significantly):

$$\left| \mathbb{E}_{\vec{x} \sim \phi'}\left[\theta \circ \text{CF}(\vec{x}, h_{\mathcal{D}_{\text{poison}} \cup \mathcal{D}_{\text{orig}}})\right] - \mathbb{E}_{\vec{x} \sim \phi'}\left[\theta \circ \text{CF}(\vec{x}, h_{\mathcal{D}_{\text{orig}}})\right] \right| \leq \epsilon \tag{8}$$

where $\epsilon > 0$ denotes a small threshold, and ϕ' denotes the density of all untargeted instances – in practice, this might be just a region in data space or a set of samples that should not be affected by the poisoning.

4. The predictive performance of the classifier is maintained[2]:

$$\arg\min \mathbb{E}[\ell(h_{\mathcal{D}_{\text{poison}} \cup \mathcal{D}_{\text{orig}}}(\vec{x}_i)), y_i)] \tag{9}$$

where $\ell(\cdot)$ denotes some suitable loss function such as the zero-one loss.

Later (see Section 4.2), we will merge all those properties into a single optimization problem for computing poisonous instances. But first, we study a few (general) aspects and properties of poisonous data sets (Definition 3) in simple settings. The gained knowledge will serve as a foundation for motivating the final data poisoning algorithm (Algorithm 1).

4.1.1. Formal Investigation

In the following, we study the data poisoning of counterfactuals in a few simple cases where formal statements on the influence of training samples on the decision boundary are feasible. While the underlying assumptions of the presented theorems are very strong and somewhat unrealistic in practice, they provide us with inspiration on how a general data poisoning mechanism could be designed (see Section 4.2). In particular, they provide us with evidence that recourse poisoning sets (Definition 3) can be constructed from closest counterfactual explanations or adversarial respectively. In the empirical evaluation (Section 6), we evaluate the performance and correctness of our proposed data poisoning mechanism (Algorithm 1). In particular, we also run an ablation study (see Section 6.3.6) in which we empirically evaluate how well the presented theoretical findings generalize to more general scenarios.

Locally Increasing the Cost of Recourse. As discussed in Section 3, the simplest way of achieving recourse is through the closest counterfactual as stated in Definition 1 – i.e., the smallest change that reaches/crosses the decision boundary. In this case, for data poisoning on a local level, it can be shown that, under strong assumptions, samples on the decision boundary are data poisoning instances (Definition 3).

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

Theorem 1 (Local Recourse Poisoning Data Sets for 1-Nearest Neighbor Classifiers)**.** *Let $h_{\mathcal{D}}(\cdot)$ be a k-nearest neighbor classifier (with $k = 1$) 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 p-norm $\theta(\cdot)$.*

Then, $\mathcal{D}_{poison} = \{(\vec{x}', y_{orig})\}$ is a poisonous data set (Definition 3) at $\vec{x}_{orig}$ – i.e. the cost resources increases for $\vec{x}_{orig}$, i.e.:

$$\theta \circ CF(x, h_{\mathcal{D} \cup \{(\vec{x}', y_{orig})\}}) > \theta \circ CF(x, h_{\mathcal{D}}) \tag{10}$$

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 if sufficiently many training samples are available. Therefore, Theorem 1 provides valuable insights on the nature of recourse poisoning data sets (Definition 3) for locally increasing the cost of recourse.

Globally Increasing the Cost of Recourse. Similar to Theorem 1, it is possible, under strong assumptions, to state a poisonous data set (Definition 3) in the case of a linear Support Vector Machine classifier (SVM) for globally increasing the cost of recourse.

Theorem 2 (Global Recourse Poisoning Data Set for linear SVM)**.** *Let $h_{\mathcal{D}} : \mathbb{R}^d \to \{-1, 1\},\ \vec{x} \mapsto \text{sign}(\vec{w}^\top \vec{x} + b)$ be a linear SVM classifier, and assume that the training data set $\mathcal{D}$ is linearly separable.*

Then, a poisonous data set $\mathcal{D}_{poison}$ (Definition 3) for all negatively classified samples (i.e. $\forall\, \vec{x} \in \mathbb{R}^d : h(\vec{x}) = -1$) is given as follows:

$$\mathcal{D}_{poison} = \{(\vec{x}, -1) \ \mid \vec{x} \in \mathbb{R}^d \text{ with } -\vec{w}^\top \vec{x} - b \geq 1 - \xi,\ \xi \in (0, 1)\} \tag{11}$$

The proof of Theorem 2 is given in the appendix. Note that Theorem 2 states that a recourse poisoning data set can be constructed by considering all samples inside the maximum margin of $h(\cdot)$. In practice, however, depending on the training data set $\mathcal{D}$, it is likely possible that already a small subset of $\mathcal{D}_{\text{poison}}$ Eq. (11) constitutes a poisonous data set (Definition 3) as well.

4.2. A Data Poisoning Algorithm for Increasing the Cost of Recourse

Based on the findings from Section 4.1, we formalize a method (see Algorithm 1) for generating poisonous data sets (Definition 3) – i.e., poisonous instances that are added to the training set, to increase the cost of recourse. Consequently, our proposed Algorithm 1 constitutes and implementation of

a data poisoning $T(\cdot)$ from Definition 2. Notably, the proposed method supports data poisonings on different levels (i.e., local, sub-groups, and global levels). Furthermore, this approach constitutes a more realistic assumption compared to backdoor attacks in [24], where an attacker is assumed to be able to manipulate the loss function used in training. More specifically, we make the following assumptions about the attack scenario and the attacker's capabilities and knowledge:

- Arbitrary data points can be added to the training data.
- The ML model to be poisoned is a black-box to the attacker. The attacker has access to an interface of the ML model for computing predictions only, but no access to any model internals or any other knowledge about the ML model.
- Access to an arbitrary method for computing closest counterfactual explanations (i.e., adversarials) of black-box models.

Most importantly, we want to highlight that the attacker does not know the specific counterfactual algorithm being used in the evaluation.

For practical purposes, we assume that we have (or created) a set of target samples $\mathcal{D}_{\text{target}} = \{(\vec{x}_j, y)\}$ all with the same prediction $y \in \{0, 1\}$ and $\vec{x}_j \sim \phi$, from the region in data space that is targeted by the poisoning – e.g., this could be a subset of the training data set. We propose to fix the size of the poisonous data set (Definition 3) (i.e., the number of poisonous instances $\{\vec{z}_i\}$ is fixed) and merge all desired properties (see Section 4.1) into the following multi-objective optimization problem:

$$\arg\min_{\{\vec{z}_i\}} \Big(\arg\min_{\vec{x}_j \in \mathcal{D}_{\text{target}}} \|\vec{z}_i - \vec{x}_j\|_p, \mathbb{E}[\ell(h_{\mathcal{D}_{\text{orig}} \cup \mathcal{D}_{\text{poison}}}(\vec{x}_l)), y_l)] \Big) \tag{12a}$$

$$\text{s.t.} \sum_{\vec{x} \in \mathcal{D}_{\text{target}}} \theta \circ \text{CF}(\vec{x}, h_{\mathcal{D}_{\text{orig}} \cup \mathcal{D}_{\text{poison}}}) > \sum_{\vec{x} \in \mathcal{D}_{\text{target}}} \theta \circ \text{CF}(\vec{x}, h_{\mathcal{D}_{\text{orig}}}) \tag{12b}$$

$$\mathbb{E}_{\vec{x} \sim \phi'} \left[\theta \circ \text{CF}(\vec{x}, h_{\mathcal{D}_{\text{orig}} \cup \mathcal{D}_{\text{poison}}})\right] - \mathbb{E}_{\vec{x} \sim \phi'} \left[\theta \circ \text{CF}(\vec{x}, h_{\mathcal{D}_{\text{orig}}})\right] \leq \epsilon \tag{12c}$$

where the poisonous data set $\mathcal{D}_{\text{poison}}$ (Definition 3) is constructed as $\mathcal{D}_{\text{poison}} = \{(\vec{z}_i, y)\}$.

Note that the objective in Eq. (12) also covers the plausibility requirement by constructing poisonous instances that are very similar to the given samples $\mathcal{D}_{\text{target}}$ – in particular, it was observed [57] that small perturbations often

remain unnoticed by the human, which gave rise to adversarial attacks [57, 58]. The rationale behind this is to make the poisonous instances more difficult to detect – we empirically evaluate this in an ablation study in Section 6.3.6.

Implementation. We propose to compute an approximate solution to Eq. (12) by constructing instances $\vec{z}_i$ that are on the decision boundary or behind it and are close to samples in $\mathcal{D}_{\text{target}}$. Note that Theorem 1 and Theorem 2 suggest that samples on or behind the decision boundary form a poisonous data set (Definition 3). We can construct such instances by computing closest counterfactual explanations δ_j (Definition 1) of samples $(\vec{x}_j, y) \in \mathcal{D}_{\text{target}}$ that are close to the decision boundary:

$$\vec{z}_i = \vec{x}_j + \delta_j \quad \text{with } \delta_j = \text{CF}(\vec{x}_j, h) \tag{13}$$

As already mentioned, note that the counterfactual δ_i used in Eq. (13) is *not* necessarily computed by the same counterfactual generation method that is targeted by the data poisoning (Definition 3) – i.e., we use an arbitrary and generic counterfactual generation algorithm to poison any other counterfactual generation algorithm.

Remark 2. *Alternatively, Eq.* (13) *could be approximated by a single gradient descent step on how to flip the prediction:*

$$\vec{z}_i = \vec{x}_j \underbrace{-\alpha \nabla \ell(h(\vec{x}_j), y_{cf})}_{:=\delta_j} \quad \textit{for some scaling factor } \alpha > 0 \tag{14}$$

This approach offers the advantage of avoiding the computation of a counterfactual δ_j, but it comes at the cost of sacrificing guarantees of correctness, both in terms of the poisoning property and the plausibility of the final instances $\vec{z}_i$. Additionally, it requires access to the model or at least access to gradients, which are either equivalent to or even stronger assumptions than those needed for Eq. (13)*, where a counterfactual δ_j is computed. In this context, it is worth noting that there exist methods [48] for computing counterfactuals that do not require gradients.*

In the remainder of this work, we stick with Eq. (13) *for approximately solving Eq.* (12) *and leave the investigation of Eq.* (14) *for future work.*

Maintaining the predictive performance objective and not changing the cost of recourse for untargeted instances are both considered implicitly in Eq. (13).

Algorithm 1 Data Poisoning for Increasing the 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; Hyperparameters: k, b

Output: Poisoned training data set D_{poisoned}

$\{\delta_i = \theta \circ \text{CF}(\vec{x}_i, h) \quad \forall \vec{x}_i \in \mathcal{D}_{\text{target}}\}$ ▷ Estimate distances to decision boundary

$D_{\text{poison}} = \{\}$

for n-times **do**

  $(\vec{x}, y) \sim \text{weighted\_sampling}\,(\mathcal{D}_{\text{target}}, \{\delta_i\})$ ▷ Prefer samples close to the decision boundary

  $\Delta_{\text{cf}} = \text{CF}(\vec{x}, h; k)$ ▷ k diverse closest CFs

  for $\vec{\delta}_{\text{cf}} \in \Delta_{\text{cf}}$ **do**

    for $\alpha \in [1, b]$ **do**

      $\vec{z} = \vec{x} + \alpha * \vec{\delta}_{\text{cf}}$ ▷ Add samples along $\vec{\delta}_{\text{cf}}$

      $D_{\text{poison}} = D_{\text{poison}} \cup \left\{ (\vec{z}, y) \right\}$

$D_{\text{poisoned}} = D_{\text{train}} \cup D_{\text{poison}}$ ▷ Add D_{poison} to training set

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, because we only consider samples $\vec{x}_j$ that are close to the decision boundary, the corresponding $\vec{z}_i$ (which constitute a counterfactual instance of $\vec{x}_j$) are expected to be very similar to $\vec{x}_j$ and therefore satisfy the plausibility requirement – we empirical evaluate this in an ablation study in Section 6.3.6.

To increase the robustness of the poisoning, we propose to (optionally) not only consider a single closest counterfactual δ_j in Eq. (13) but a set of k diverse closest counterfactual explanations. We also propose to extend the counterfactual direction δ_j by multiplying it with a factor $\alpha > 1$, to create a larger and significant increase in the cost of recourse.

The pseudo-code for generating a data poisoning is given in Algorithm 1.

Correctness. From Theorem 1 it follows (see Corollary 1) that Algorithm 1 computes valid poisonous data sets for a k-NN classifier with $k = 1$.

Corollary 1 (Correctness of Algorithm 1)**.** *Let $h_{\mathcal{D}}(\cdot)$ be a k-nearest neighbor classifier (with $k = 1$) for some data set $\mathcal{D}$. For any $\mathcal{D}_{target} = \{(\vec{x}_{orig}, y_{orig})\}$ where $(\vec{x}_{orig}, y_{orig}) \in \mathcal{D}$, Algorithm 1 computes a poisonous data set (Definition 3) that increases the cost of recourse of $\vec{x}_{orig}$.*

The detailed proof of Corollary 1 is given in the appendix.

Apart from the simplistic case of a k-NN classifier with $k = 1$, we perform an extensive empirical evaluation in Section 6 to provide evidence for the correctness of Algorithm 1 in more general and realistic cases. More specifically, in a simplified manner, the theorems from Section 4.1.1 state that samples on the decision boundary or behind it can shift the decision boundary such that the cost of recourse changes. Note that this is one of the core ideas behind the proposed data poisoning Algorithm 1. While the overall idea of those theorems is not the only ingredient for Algorithm 1, it is the only, and therefore necessary, ingredient for influencing the cost of recourse. An additional ingredient for Algorithm 1 is the sampling strategy (see line 4 in Algorithm 1) for ensuring that the poisonous samples are realistic and therefore difficult to detect. In the ablation study in Section 6.3.6, we remove this sampling strategy and therefore basically reduce Algorithm 1 to the core findings of the theorems from Section 4.1.1. Thereby, we empirically evaluate in how far those theorems generalize to more realistic scenarios.

Runtime. 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 (i.e., size of the poisonous data set), and ρ denotes the computational complexity (i.e., runtime) for computing the poisonous sample(s) as constructed in Eq. (13) (see line 8 in Algorithm 1) – note ρ is likely to differ between different counterfactual generation mechanisms and also on the complexity of the black-box ML-model to be poisoned. The recourse cost definition $\theta(\cdot)$ is not expected to influence the overall runtime of Algorithm 1 as long as it only depends on the dimensionality of the inputs $\vec{x}_i$ – note that this is naturally the case for classic recourse cost implementations such as weighted sums. Consequently, the runtime of Algorithm 1 scales linearly with the number of requested poisonous samples – assuming that ρ is a polynomial.

Trade-offs & Limitations. The major limitation of Algorithm 1 is that it requires access to a counterfactual generation method for generating poisonous instances. A gradient-based approximation of the counterfactual generation

method could be an alternative to this (see Remark 2). However, besides assuming gradients and access to them (not possible for tree-based models), one would lose the stated correctness guarantees.

5. Qualitative Evaluation Case-Study: Explaining Event Detection in Water Distribution Networks

In this illustrative case study, we aim to highlight the impact of manipulated counterfactuals in event diagnosis systems, i.e., systems for detecting events such as anomalies and identifying their cause. Note that event diagnosis constitutes an essential task for the successful operation of critical infrastructure systems such as water networks, power grids, and transportation networks. In the context of AI-based event detectors, performing counterfactual reasoning, via counterfactual explanations, over possible causes of an observed event, is an increasingly popular approach in the literature [59, 60]. Note that, although we are focusing on the exemplary case of identifying sensor failures in water distribution networks, our findings can likely be generalized to other domains.

5.1. Introduction and Background

Water distribution networks (WDNs) are critical infrastructure for supplying drinking water. Water utilities and human operators depend on strategically placed sensors to identify anomalies such as sensor faults, leaks, and contaminations. Data-driven methods are used to automate sensor data analysis and to support human decision-making. Due to the significant impact of addressing anomalies, such as dispatching repair teams or adjusting water treatment, operators must understand and trust these methods' predictions. Recent work [61, 62] in this area proposes the usage of XAI to analyze and explain detected anomalies, ensuring human operators can respond optimally. The authors of [62] propose a counterfactual explanation method for explaining anomalous observations (i.e., raised alarms of an event detector) and, in particular, identifying the cause, such as a faulty sensor. To achieve this, they suggest reconstructing sensor readings using an ensemble of virtual sensors $f(\cdot)$. This means creating a virtual sensor for each existing one – i.e., each sensor's reading is predicted based on the readings from all other sensors. An alarm is raised (i.e., $h(\cdot) = 1$) whenever the difference between predicted

and observed sensor reading is too large:

$$h(\vec{x}_t) = \begin{cases} 1 & \text{if } ||f(\vec{x}_t) - \vec{x}_t||_p \geq \zeta \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where $\vec{x}_t$ denotes the sensor readings at time t (or a small time window), and the hyperparameter $\zeta > 0$ denotes a threshold at which an alarm is raised. The threshold ζ can either be set manually by the human operator or can be calibrated based on a validation set. They propose using counterfactual explanations to explain the cause of an alarm in the observed sensor readings $\vec{x}_t$ (i.e., input to the event detector). Additionally, they propose an efficient algorithm to compute counterfactuals for Eq. (15) by exploiting the ensemble structure of the virtual sensors $f(\cdot)$. Those counterfactual sensor readings $\vec{\delta}_{\text{cf}}$ can be interpreted as ways to "undo" the triggered alarm by reverting the effect of events, such as sensor faults, manifested in the counterfactual $\vec{\delta}_{\text{cf}}$, i.e:

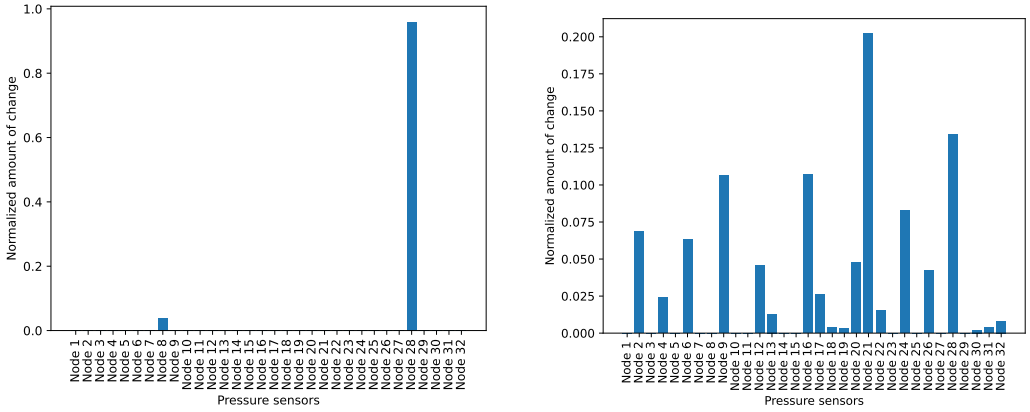
$$h(\vec{x}_t + \vec{\delta}_{\text{cf}}) = 0 \tag{16}$$

The counterfactual $\vec{\delta}_{\text{cf}}$ states significant differences for the faulty sensors (or sensors close to the anomaly) and therefore assists human operators in identifying and locating the anomaly, enabling them to take appropriate actions [62].

5.2. Setup

In this case study, we consider the event diagnosis task of identifying and localizing sensor faults in a Water Distribution Network (WDN). In this context, we demonstrate the impact of data poisoning on counterfactuals in such a counterfactual-based event diagnosis method [62], where counterfactual reasoning is used to identify and localize the sensor fault.

We consider the popular Hanoi benchmark from LeakDB [63] to simulate several WDN scenarios (each 21 days long) by randomly introducing several pressure sensor faults, modeled as Gaussian noise (mimicking an aging sensor), at various locations and times. We simulate those scenarios with EPyT-Flow [64] and apply the event detection and counterfactual explanations as proposed in [62]. As aforementioned, the core of the system Eq. (15) consists of a set of virtual sensors (based on linear regression) for each of the four pressure sensors – i.e., predicting the readings at this sensor based on the readings of all other sensors. More details can be found in [62] and in the



(a) No data poisoning.

(b) Our proposed data poisoning.

Figure 1: Sensor fault at node 28 – original counterfactual vs. poisoned counterfactual.

GitHub repository[3] of this paper. To evaluate the impact of data poisoning and apply our proposed method from Section 4.2, we have to specify the cost of a counterfactual in this context. Since counterfactuals are expected to pinpoint the faulty sensor, and each non-zero feature modification indicates a potential sensor fault location requiring manual (and costly) inspection, we propose using the number of modified features (i.e., sparsity) as the measure of recourse cost in this application:

$$\theta(\vec{\delta}_{\text{cf}}) = \sum_i \mathbb{1}_{(\vec{\delta}_{\text{cf}})_i \neq 0} \tag{17}$$

For this case study, we assume a scenario in which the water utility has a database of (simulated) sensor readings that are used to train the event diagnosis systems Eq. (15). We assume that an attacker has the ability to add samples to this database, and has obtained a copy of the trained event diagnosis system on the original database, but does not know anything about the architecture of the event diagnosis system. In addition, the attacker has access to a simulator for generating sensor readings data that can be utilized for constructing the poisonous samples. In order to mimic real-world constraints (in particular, the sim-to-real gap), the simulation used for constructing the poisonous samples differs slightly from the one used in

[3] `https://github.com/andreartelt/DataPoisoningCounterfactuals`

the evaluation. More specifically, we simulate 3 sets of 3 scenarios each, whereby each scenario is unique by mimicking real-world uncertainties such as slightly different water demands and slight changes to parameters such as pipe diameter and pipe length:

1. The first set does not contain any sensor faults and is used for fitting the virtual sensors and calibrating the event detector Eq. (15) as described in [62].
2. The second set contains sensor faults (i.e., adding Gaussian noise) at random locations and is used for generating poisonous instances with our proposed Algorithm 1 – we set $\alpha = 1.5$ and use uniform sampling when selecting candidate samples from the target samples $\mathcal{D}_{target}$. For the target samples $\mathcal{D}_{target}$, we apply the event detection method Eq. (15), which was trained on the first set of scenarios, to the newly simulated data and use the true positives as the target samples $\mathcal{D}_{target}$ in Algorithm 1. For evaluating the impact of the data poisoning, we then add 5% of the generated poisonous data points to the original training set from the first set of scenarios – this translates to adding approximately 50 poisonous samples to a training set of size 1000.
3. The third set also contains the same type of sensor faults (i.e., adding Gaussian noise) at random locations and is used for evaluation only – i.e., evaluating the counterfactual explanations of the detected sensor faults with and without data poisoning. Note that this set of scenarios differs slightly from the other sets, mimicking the sim-to-real gap, thus ensuring a realistic and fair evaluation.

5.3. Results & Conclusion

For the evaluation, we only consider true positives – i.e., a sensor fault was correctly detected and must now be localized. In Figure 2, we show the cost of recourse (i.e., the sparsity of the counterfactual) for both the original and poisoned event diagnosis system. Additionally, Figure 1 illustrates a single counterfactual – comparing the original versus the poisoned event diagnosis system. We observe that our data poisoning approach significantly decreases the sparsity of the counterfactuals, thereby making it more challenging and costly to identify the faulty sensor.

More generally, this case study demonstrates that an attacker can compromise the ability of a counterfactual-based event diagnosis system for identifying the cause of anomalous observations and thereby hindering the detection of malicious activities such as cyber-physical attacks. Consequently,

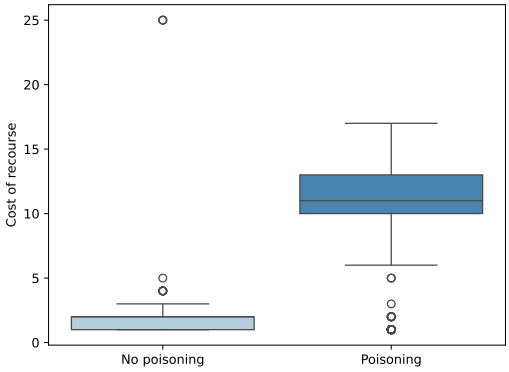


Figure 2: Cost of recourse (i.e. sparsity): Original counterfactuals vs. poisoned counterfactuals – lower scores are better.

it highlights the severe impact of data poisoning on counterfactuals in critical real-world applications and stresses the importance of data integrity for building data-driven event diagnosis methods.

6. Quantitative Evaluation on ML Benchmarks

We empirically evaluate the robustness of counterfactual explanations against data poisoning by applying our proposed data poisoning Algorithm 1 on combinations of several different benchmark data sets, classifiers, and state-of-the-art counterfactual explanation generation methods and toolboxes. We consider the following three attack scenarios separately:

1. Section 6.3.2: Increasing the cost of recourse *globally*, i.e. for all individuals.
2. Section 6.3.3: Increasing the cost of recourse for a *sub-group* of individuals only.
3. Section 6.3.4: Increasing the cost of recourse *locally* for single individuals only.

As an initial baseline, we also empirically evaluate the effect of the label flipping attack, as a classic data poisoning attack, on the cost of recourse (Section 6.3.1). This allows us to evaluate the necessity of developing specialized data poisoning methods such as our proposed Algorithm 1.

We also evaluate the effectiveness of class data sanitization methods for detecting the poisonous instances (see Section 6.3.5).

Finally, we also conduct an ablation study (Section 6.3.6) to evaluate the effect of the sampling procedure in Algorithm 1. In particular, how well the formal statements from Section 4.1.1 generalize to more complex and realistic scenarios.

The Python implementation of all conducted experiments (including all data sets) is available on GitHub[4].

6.1. Benchmark Data Sets & Classifiers

We consider three commonly used data sets from the literature that all contain a sensitive attribute, such that we can also evaluate the difference in the cost of recourse between protected groups in our empirical evaluation:

- The "Diabetes" data set [65] (denoted as *Diabetes*) contains data from 442 diabetes patients, each described by For each patient, 9 numeric attributes such as age, body mass index, average blood pressure, and six blood serum measurements 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 data set is balanced concerning the class labels (238 vs. 204 samples).

- The "Communities & Crime" data set [66] (denoted as *Crime*) contains 1994 socio-economic data, including the sensitive attribute "race", records from the USA. Following the pre-processing as suggested in [67], we are left with 100 encoded attributes that are used to predict the crime rate (low vs. high). The data set contains a single sensitive attribute "race". The data set is extremely unbalanced concerning the class labels (1872 vs. 122 samples). We therefore randomly under-sample the majority class in every train-test split.

- The "German Credit Data set" [68] (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. Because the data set is heavily class-imbalanced (700 vs. 300 samples), we randomly under-sample the majority class in every train-test split.

[4] `https://github.com/andreartelt/DataPoisoningCounterfactuals`

All data sets are standardized to have a mean of zero and unit variance for improved numerical stability; this scaling is done in each train-test split.

6.2. Machine Learning Classifiers and Counterfactual Generation Methods

In order to evaluate the broader impact of data poisoning on counterfactuals, we consider a diverse set of ML classifiers $h(\cdot)$ and counterfactual generation methods & toolboxes.

We consider the following classifiers: 3-layer neural network with ReLU activation functions (denoted as *DNN*), random forests (denoted as *RNF*), and linear SVMs (denoted as *SVC*). The hyperparameters of those classifiers have been tuned separately by a grid search and are kept fixed during the data poisoning experiments for better comparability – i.e., retraining on a poisoned training data set is done under the same hyperparameters as on the original/clean data set.

We consider a diverse set of different and popular state-of-the-art methods for computing computational recourse. Note that all of these methods define and compute counterfactual recourse in slightly different ways as discussed in Section 3:

- Nearest Unlike Neighbor [55] (denoted as *NUN*), as simple but strong baseline.
- Counterfactuals guided by Prototypes [46] (denoted as *Proto*) for computing plausible counterfactuals.
- DiCE [48] for computing diverse counterfactuals.

6.3. Setup

In all experiments where we evaluate our proposed data poisoning method Algorithm 1, we use DiCE [48] as a counterfactual generation mechanism for computing three diverse closest counterfactuals (i.e., $k = 3$ in Algorithm 1) that are as close as possible to the original sample. Furthermore, we set $b = 1.5$ (see Algorithm 1) and use the ℓ_2-distance to the decision boundary for computing the sample weights in line 4 of Algorithm 1 – i.e., the probability of the i-th sample scales with $p_i = \frac{1}{\|\delta_i\|_2}$.

All experiments are run in 5-fold cross-validation. We use the ℓ_1 norm as a popular implementation [45] of the cost of recourse – i.e., we set $\theta(\cdot) = \|\cdot\|_1$ –, and w.l.o.g., we refer to $y = 0$ as the unfavorable, and $y = 1$ as the favorable outcome. In all global and sub-group data poisoning scenarios, we

inject different amounts (5% to 70% of the original training data) of poisonous instances into the training data set – i.e., original training data + poisoned instances. Note that we do not make any specific assumptions on the poisoning budget of the attacker, but instead investigate the sensitivity of the effects regarding different data poisoning sizes. Obviously, in practice, smaller poisoning budgets are preferred and probably more realistic – we therefore pay special attention to the smallest poisoning budget that leads to a statistically significant increase in the cost of recourse.

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. Furthermore, we evaluate the sparsity (i.e., number of changed features) and the plausibility (i.e., the log-likelihood), under a kernel density estimator with a Gaussian kernel, of the generated counterfactual explanations, as well as the time it took to generate them. In order to analyze the associated uncertainties of all evaluations, we perform a Mann-Whitney U test on all evaluated quantities, assessing the statistical significance of the reported results. Furthermore, all figures also visualize the standard deviation of the reported results.

Note that we did to find any statistically significant effect of data poisoning on the time it takes to generate a counterfactual explanation, and also not in the number of times where the computation of counterfactuals failed.

For better readability, some results are moved to the appendix.

Remark 3. *Note that, although we use the DiCE method [48] for constructing the poisonous instances in Algorithm 1, it can still be considered a model-agnostic method because it is also able to attack other counterfactual generation methods – i.e., other counterfactual generation methods can be poisoned by utilizing DiCE in constructing the poisonous samples Eq.* (13)*.*

6.3.1. Effect of a label flipping attack on the cost of recourse

Before empirically evaluating our proposed data poisoning method Algorithm 1 for increasing the cost of recourse, we first evaluate the effect of a label-flipping poisoning attack [8] on the global cost of recourse. By this, we empirically evaluate the difference between "classic" data poisoning for decreasing a model's predictive performance, and our data poisoning method for increasing the cost of recourse. Furthermore, note that while label flipping is known to have a strong effect on the attacked model [8], it is also known to be a somewhat "noisy" data poisoning that can be detected by outlier detection methods [69, 70].

Table 1: *Effect of a classic label flipping attack data poisoning attack:* Difference (percentage) in the cost of recourse (see Eq. (19)) – no poisoning vs. *label flipping*. In all cases, 5% of the training data is poisoned by flipping their labels. Positive numbers denote an increase (due to the label flipping) in the cost of recourse, while negative numbers denote the opposite. We report the median (over all folds) rounded to two decimal places, as well as the statistical significance according to the Mann-Whitney U test (ns $\implies$ p-value > 0.05; * $\implies$ p-value ≤ 0.05; ** $\implies$ p-value ≤ 0.01; *** $\implies$ p-value ≤ 0.001).

Classifier	Data set	NUN	DiCE	Proto
SVC	Credit	$1\%_{ns}$	$0\%_{ns}$	$-4\%_{ns}$
	Diabetes	$0\%_{ns}$	$0\%_{ns}$	$-4\%_{ns}$
	Crime	$-10\%_{***}$	$-9\%_{***}$	$-9\%_{**}$
RNF	Credit	$0\%_{ns}$	$0\%_{ns}$	$-1\%_{ns}$
	Diabetes	$3\%_{ns}$	$0\%_{ns}$	$1\%_{ns}$
	Crime	$-9\%_{***}$	$-5\%_{***}$	$-3\%_{***}$
DNN	Credit	$0\%_{ns}$	$-1\%_{ns}$	$-4\%_{ns}$
	Diabetes	$0\%_{ns}$	$-3\%_{ns}$	$8\%_{ns}$
	Crime	$-8\%_{***}$	$-7\%_{***}$	$-10\%_{***}$

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:

$$\theta \circ \mathrm{CF}(\vec{x}_i, h_{\mathcal{D}_{\mathrm{orig}} \cup \mathcal{D}_{\mathrm{poison}}}) - \theta \circ \mathrm{CF}(\vec{x}_i, h_{\mathcal{D}_{\mathrm{orig}}}) \quad \forall\, \vec{x}_i, y_i \in \mathcal{D}_{\mathrm{test}},\ h(\vec{x}_i) = 0 \quad (18)$$

A positive score Eq. (18) means an increase in the recourse cost (due to the label flipping), while a negative or near-zero score implies no change or a lower cost of recourse. To facilitate the interpretation of Eq. (18), we report the percentage difference instead of the absolute difference:

$$\frac{\theta \circ \mathrm{CF}(\vec{x}_i, h_{\mathcal{D}_{\mathrm{orig}} \cup \mathcal{D}_{\mathrm{poison}}})}{\theta \circ \mathrm{CF}(\vec{x}_i, h_{\mathcal{D}_{\mathrm{orig}}})} - 1 \quad (19)$$

We report the median of Eq. (19) together with the statistical significance level. The results for poisoning 5% of the training data (i.e., flipping their label) are shown in Table 1.

6.3.2. Data poisoning for increasing the cost of recourse on a global level

We apply our proposed data poisoning method, Algorithm 1, on a global level. For every negative classified sample in the test set, we compute a

Table 2: Difference (percentage) in the cost of recourse (see Eq. (19)): no poisoning vs. *global poisoning*. In all cases, we add 5% of the training data as poisonous instances. Positive numbers denote an increase (due to the data poisoning) in the cost of recourse, while negative numbers denote the opposite. We report the median (over all folds) rounded to two decimal places, as well as the statistical significance according to the Mann-Whitney U test (ns $\implies$ p-value > 0.05; * $\implies$ p-value ≤ 0.05; ** $\implies$ p-value ≤ 0.01; *** $\implies$ p-value ≤ 0.001).

Classifier	Data set	NUN	DiCE	Proto
SVC	Credit	$9\%_{*}$	$8\%_{***}$	$7\%_{ns}$
	Diabetes	$11\%_{***}$	$8\%_{***}$	$19\%_{**}$
	Crime	$8\%_{***}$	$5\%_{***}$	$5\%_{***}$
RNF	Credit	$3\%_{ns}$	$6\%_{***}$	$11\%_{*}$
	Diabetes	$2\%_{ns}$	$1\%_{ns}$	$14\%_{ns}$
	Crime	$2\%_{ns}$	$2\%_{ns}$	$3\%_{ns}$
DNN	Credit	$2\%_{ns}$	$7\%_{***}$	$1\%_{*}$
	Diabetes	$4\%_{ns}$	$7\%_{**}$	$8\%_{ns}$
	Crime	$4\%_{***}$	$6\%_{***}$	$4\%_{**}$

counterfactual explanation, and evaluate the global effect on the cost of recourse, by computing the difference in the cost of recourse Eq. (19). Again, a positive score Eq. (19) refers to 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. (19) together with the statistical significance according to the Mann-Whitney U test. The effect of adding 5% of poisonous instances to the training data set on the cost of recourse is shown in Table 2. The effects on the sparsity and log-likelihood, as well as the impact of different poisoning budgets, are given in Appendix B.1.

6.3.3. Data poisoning for increasing the cost of recourse on a sub-group level

We consider 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 Algorithm 1 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, which can be interpreted as introducing or increasing group-unfairness in recourse [39, 52, 53].

For every negative classified sample in the test set (no matter to which

Table 3: Difference (percentage) in the cost of recourse (see Eq. (19)): no vs. local poisoning. Positive numbers denote an increase (due to the data poisoning) in the cost of recourse. We report the median (over all folds) rounded to two decimal places, as well as the statistical significance according to the Mann-Whitney U test (ns $\implies$ p-value > 0.05; * $\implies$ p-value ≤ 0.05; ** $\implies$ p-value ≤ 0.01; *** $\implies$ p-value ≤ 0.001).

Classifier	Data set	NUN	DiCE	Proto
DNN	Diabetes	$39\%_{***}$	$27\%_{***}$	$72\%_{***}$

sub-group it belongs), we compute a counterfactual. We evaluate the difference in the cost of recourse between the two sub-groups as follows:

$$\underbrace{\|\theta \circ \mathrm{CF}(\vec{x}_i|s=0, h_{\mathcal{D}_{\mathrm{orig}} \cup \mathcal{D}_{\mathrm{poison}}}) - \theta \circ \mathrm{CF}(\vec{x}_i|s=1, h_{\mathcal{D}_{\mathrm{orig}} \cup \mathcal{D}_{\mathrm{poison}}})\|}_{\text{Median difference in the cost of recourse } \textbf{under} \text{ data poisoning}} - \underbrace{\|\theta \circ \mathrm{CF}(\vec{x}_i|s=0, h_{\mathcal{D}_{\mathrm{orig}}}) - \theta \circ \mathrm{CF}(\vec{x}_i|s=1, h_{\mathcal{D}_{\mathrm{orig}}})\|}_{\text{Median difference in the cost of recourse } \textbf{without} \text{ data poisoning}} \quad \forall\, \vec{x}_i \in \mathcal{D}_{\mathrm{test}}\ h(\vec{x}_i) = 0 \tag{20}$$

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. A positive score of Eq. (20) refers to an increase in the difference of the cost of recourse between the protected groups, while a negative score refers to the opposite. Note that for the purpose of stability, we use the median (over all folds) in Eq. (20). To facilitate the interpretability of the results, we report the percentage difference instead of the absolute difference:

$$\frac{\|\theta \circ \mathrm{CF}(\vec{x}_i|s=0, h_{\mathcal{D}_{\mathrm{orig}} \cup \mathcal{D}_{\mathrm{poison}}}) - \theta \circ \mathrm{CF}(\vec{x}_i|s=1, h_{\mathcal{D}_{\mathrm{orig}} \cup \mathcal{D}_{\mathrm{poison}}})\|}{\|\theta \circ \mathrm{CF}(\vec{x}_i|s=0, h_{\mathcal{D}_{\mathrm{orig}}}) - \theta \circ \mathrm{CF}(\vec{x}_i|s=1, h_{\mathcal{D}_{\mathrm{orig}}})\|} - 1 \tag{21}$$

We report the results regarding Eq. (21), predictive performance, and sparsity, together with the statistical significance level in Appendix B.2.

6.3.4. Data poisoning for increasing the cost of recourse on a local level

We compute a local data poisoning (with our proposed Algorithm 1) for every negative classified sample in the test set. However, because of computational limitations – i.e., for every sample in the test set (over all folds), the entire data poisoning must be run and evaluated –, we only evaluate a single scenario considering a DNN classifier applied to the diabetes data set. The results, together with the statistical significance level, are shown in Table 3,

and box-plots of the distributions of the scores can be found in Appendix B.3.

6.3.5. Detection of poisonous instances for increasing the cost of recourse

Given the potentially severe impact of data poisoning attacks on counterfactual explanations, as illustrated in our case study in Section 5, we also evaluate the effectiveness of data sanitization procedures [30, 33] for detecting such poisonous instances (Definition 3). Recall that data sanitization methods [30, 33, 32, 31] aim to detect the poisonous instances by means of outlier detection methods and remove those detected instances from the poisoned training data set $\mathcal{D}_{\text{orig}} \cup \mathcal{D}_{\text{poison}}$.

In this context, we evaluate the effectiveness of two classic outlier detection methods, and three classic data sanitization defense methods from the literature [33]:

- We consider the Isolation Forest [71] and Local Outlier Factor (LOF) method [72], as two classic and popular outlier detection methods. Both methods are calibrated on the (unpoisoned) test set and applied to $\mathcal{D}_{\text{orig}} \cup \mathcal{D}_{\text{poison}}$ to detect the poisonous instances $\mathcal{D}_{\text{poison}}$.

- The k-NN defense [73], which flags points that are far from their k-th nearest neighbor:

  $$\|\vec{x}_i - \vec{z}_y\|_2 \geq \nu \text{ with } \vec{z}_y \text{ being the k-th nearest neighbor in } \mathcal{D}_{\text{orig}} \cup \mathcal{D}_{\text{poison}} \tag{22}$$

  where the hyperparameter $\nu > 0$ denotes the threshold.

- The ℓ_2-defense [33], which flags instances far from their class centroids in the ℓ_2 distance:

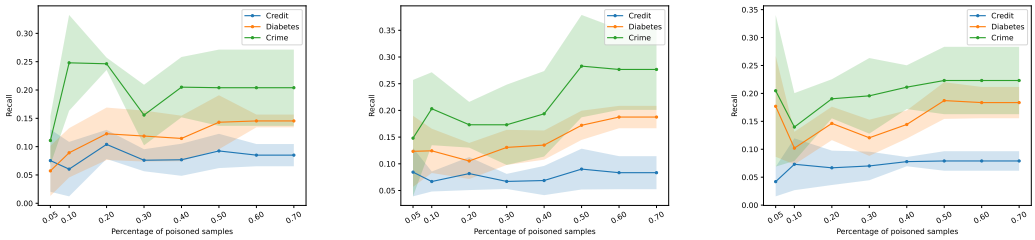
  $$\|\vec{x}_i - \mathbb{E}[\vec{x} \mid y]\|_2 \geq \nu \tag{23}$$

  where the hyperparameter $\nu > 0$ denotes the threshold.

- The slab-defense [29], which constitutes an extension of the ℓ_2-defense [33], flags instances that are too far from the centroids after they are projected onto the line between the two class centroids:

  $$\left|(\mathbb{E}[\vec{x} \mid y = 0] - \mathbb{E}[\vec{x} \mid y = 1])^\top (\vec{x}_i - \mathbb{E}[\vec{x} \mid y])\right) \geq \nu \tag{24}$$

  where the hyperparameter $\nu > 0$ denotes the threshold.



(a) $h(\cdot)$: SVC (b) $h(\cdot)$: RandomForest (c) $h(\cdot)$: DNN

Figure 3: Isolation Forest for detection of the poisonous instances in a global poisoning. We report the mean and standard deviation (over all folds) of the recall (larger numbers are better).

We empirically evaluate the performance of those defense methods in a global attack scenario, where the data poisoning aims to increase the cost of recourse for all individuals. For the threshold-based defense methods Eq. (22) - Eq. (24), we calibrate the threshold on an unpoisoned hold-out set [33], and for the k-NN defense Eq. (22), we set $k = 5$ as suggested in [33].

We evaluate the recall for detecting the poisonous instance under different sizes of poisonings, and report the mean and standard deviation. The results for the Isolation Forest method on a global poisoning are shown in Figure 3, all other results are given in Appendix B.4.

6.3.6. Ablation study

In this ablation study, we consider the case of a global poisoning attack on the cost of recourse. We investigate the effect of the sampling strategy in line 4 of Algorithm 1, which prefers samples close to the decision boundary for generating poisonous samples. We change the sampling strategy to a uniform sampling and evaluate its effect on the cost of recourse, as well as on the performance of the defense methods for identifying the poisonous samples.

The results for the effect on the cost of recourse are shown in Table 4. Additional results for all poisoning budgets can be found in Figure B.11. Furthermore, the results of the performance of the outlier detection and data sanitization methods for identifying the poisonous samples can be found in Table 5 – here, we compare the performance (i.e., recall) to the case with the original sampling strategy in line 4 of Algorithm 1 that is supposed to ensure plausibility of the poisonous samples.

Table 4: *Ablation study (uniform sampling)* – Difference (percentage) in the cost of recourse (see Eq. (19)): no poisoning vs. *global poisoning*. In all cases, we add 5% of the training data as poisonous instances. Positive numbers denote an increase (due to the data poisoning) in the cost of recourse, while negative numbers denote the opposite. We report the median (over all folds) rounded to two decimal places, as well as the statistical significance according to the Mann-Whitney U test (ns $\implies$ p-value > 0.05; * $\implies$ p-value ≤ 0.05; ** $\implies$ p-value ≤ 0.01; *** $\implies$ p-value ≤ 0.001).

Classifier	Data set	NUN	DiCE	Proto
SVC	Credit	$4\%_{ns}$	$7\%_{***}$	$5\%_{*}$
	Diabetes	$11\%_{***}$	$7\%_{***}$	$13\%_{*}$
	Crime	$7\%_{***}$	$7\%_{***}$	$7\%_{***}$
RNF	Credit	$1\%_{ns}$	$10\%_{***}$	$9\%_{ns}$
	Diabetes	$2\%_{ns}$	$3\%_{*}$	$10\%_{ns}$
	Crime	$5\%_{*}$	$1\%_{ns}$	$13\%_{**}$
DNN	Credit	$1\%_{ns}$	$6\%_{***}$	$2\%_{ns}$
	Diabetes	$8\%_{**}$	$7\%_{***}$	$4\%_{ns}$
	Crime	$6\%_{***}$	$6\%_{***}$	$10\%_{***}$

6.4. Results & Discussion

6.4.1. Effect of label flipping on the cost of recourse

We observe (see Table 1) that label flipping typically has little to no statistically significant impact on the overall cost of recourse (on a global level). This suggests that traditional data poisoning techniques aimed at reducing predictive performance are inadequate for poisoning counterfactual explanations and increasing the cost of recourse. Consequently, specialized methods and algorithms, such as our proposed data poisoning method, are necessary to affect methods for computing counterfactual explanations.

6.4.2. General trend

We observe that in most scenarios, on local as well as on global levels (see Table 2 and Appendix B), even a relatively small amount of poisonous instances such as 5%, added to the training data set, leads to a statistically significant increase in the cost of recourse. Increasing the number of poisonous instances leads to an even larger and more significant increase in the cost of recourse. Besides an increase in the cost of recourse, we also almost always observe a statistically significant increase in the number of changed features – i.e., the sparsity of the counterfactuals decreases. Although the increase is statistically significant, it is much smaller than the increase in the

Table 5: *Ablation study (uniform sampling)* – Difference (percentage) in the recall of detecting the poisonous samples in a global data poisoning scenario. In all cases, we add 5% of the training data as poisonous instances. Positive numbers denote an increase (due to the uniform sampling) in the cost of recourse, while negative numbers denote the opposite. We report the difference in the median (over all folds) rounded to two decimal places, as well as the statistical significance according to the Mann-Whitney U test (ns $\implies$ p-value > 0.05; * $\implies$ p-value ≤ 0.05; ** $\implies$ p-value ≤ 0.01; *** $\implies$ p-value ≤ 0.001).

Classifier	Data set	Isolation-Forest	LOF	slab-defense	ℓ_2-defense	k-NN defense
SVC	Credit	$82.14\%_{ns}$	$0\%_{ns}$	$-31.55\%_{ns}$	$92.86\%_{ns}$	$-14.29\%_{ns}$
	Diabetes	$369.74\%_{**}$	$0\%_{ns}$	$123.68\%_{ns}$	$73.68\%_{ns}$	$1.05\%_{ns}$
	Crime	$172.73\%_{**}$	$-18.18\%_{ns}$	$0\%_{ns}$	$13.64\%_{ns}$	$83.64\%_{**}$
RNF	Credit	$49.43\%_{ns}$	$0\%_{ns}$	$-61.2\%_{ns}$	$117.46\%_{ns}$	$28.57\%_{ns}$
	Diabetes	$215.79\%_{***}$	$0\%_{ns}$	$236.84\%_{*}$	$57.89\%_{ns}$	$-12.28\%_{ns}$
	Crime	$221.14\%_{**}$	$0\%_{ns}$	$0\%_{ns}$	$13.64\%_{ns}$	$54.55\%_{**}$
DNN	Credit	$200.0\%_{*}$	$0\%_{ns}$	$-57.14\%_{ns}$	$45.42\%_{ns}$	$157.14\%_{ns}$
	Diabetes	$135.79\%_{*}$	$0\%_{ns}$	$0\%_{ns}$	$68.42\%_{*}$	$0.48\%_{ns}$
	Crime	$104.55\%_{ns}$	$36.36\%_{ns}$	$0\%_{ns}$	$0\%_{ns}$	$-69.7\%_{ns}$

cost of recourse. We do not observe any statistically significant effect on the plausibility (i.e., log-likelihood) of the counterfactual explanation in any of the conducted empirical evaluations (e.g., see Table B.6). This demonstrates that our proposed data poisoning only affects the cost of recourse and related measures, such as the sparsity, but not the plausibility of the generated counterfactuals. This makes the effects of the data poisoning even more refined, since those poisoned counterfactuals do not look more/less plausible than their non-poisoned counterpart.

However, we also observe differences in the necessary amount of poisonous instances between different counterfactual generation methods and toolboxes – in particular for the case where we want to increase the cost of recourse on a sub-group level. For the counterfactuals guided by prototypes method [46] and the nearest unlike neighbor method [55], we often need more poisonous instances to observe a statistically significant increase in the cost of recourse. Since those methods focus on plausibility, this might be an indicator that additional plausibility constraints can act as a beneficial regularization for increased stability – similar to what is reported in [26] for robustness concerning input perturbations. However, as it becomes apparent from the sensitivity analysis regarding the poisoning budget (e.g., see Figure B.4), those two methods for generating counterfactuals are not immune to data poisoning in general, but only more robust to very small amounts of poisonous samples.

Altogether, the empirical evaluation reveals the vulnerability of existing (state-of-the-art) counterfactual generation methods to data poisonings. Furthermore (as discussed in detail in Section 6.4.7), the failure of classic outlier detection and defense methods demonstrates that the detection of our generated poisonous instances is non-trivial.

6.4.3. Data poisoning on a global level

In the case of a data poisoning on a global level, we observe an almost monotonic increase in the cost of recourse when increasing the number of poisonous samples. In particular, a small poisoning budget is often already sufficient to create a statistically significant increase in the cost of recourse. However, we also observe that the increase in the cost of recourse stops at some points and flattens out. This indicates a saturating effect that, at some point, increasing the poisoning budget does not have any further impact on the cost of recourse. More specifically, this suggests that there is an upper bound on the increase in the cost of recourse that can be achieved by our proposed data poisoning. A similar effect is observed for the decrease in predictive performance of the classifiers and the decrease in sparsity of the generated counterfactuals.

These observations can be explained by the nature of our proposed data poisoning method Algorithm 1. Since we require that the generated poisonous training samples are similar to the original training samples of the same class (see Eq. (12)), there is an upper bound on how much those poisonous samples can shift the decision boundary of the classifier, leading to the observed results.

6.4.4. Data poisoning on a sub-group level

In the case of sub-groups, we observe (see Tables B.7,B.8,B.9) similar effects as in the case of a global poisoning. However, the effects are more unstable in the sense that the increase in the difference of the cost of recourse varies significantly, and sometimes the change in difference is not statistically significant, in particular for the NUN and Proto methods. This is quite likely due to a strong overlap of the distributions of the sub-groups, which makes it difficult to just alter the cost of recourse for one group but not for the other.

6.4.5. Data poisoning on a local level

From Figure B.9, we observe that in all cases the local data poisoning attack leads to a statistically significant increase in the cost of recourse for

the targeted instances. However, we also observe that the cost of recourse for untargeted instances also increases – only a small increase compared to the targeted instance, but the difference is already statistically significant. This demonstrates that our data poisoning method is also able to target specific instances only, without affecting other instances too much. While there is room for improvement, we suspect that the degree of how much untargeted instances are affected depends not only on the flexibility of the attacked classifier but also on the location of the targeted instance in data space. We leave a deeper investigation of such local attacks as future research.

6.4.6. Effect on the predictive performance

We observe the expected results that classifiers' predictive performance decreases (statistically significantly) as more poisoned instances are added. More specifically, for a global data poisoning, the decrease in predictive performance is worse than for sub-group or local data poisonings.

This is to be expected since the manipulation of counterfactual explanations requires manipulating the decision boundary. Despite this, the proposed data poisoning presents a significant threat. Minor drops in predictive performance, resulting from small data poisonings, might go undetected. Yet, they can already lead to a substantial increase in the cost of recourse, as demonstrated in the presented experiments.

6.4.7. Detection of poisonous instances

Concerning the detectability of the generated poisonous instances, we observe (see Figure B.10) that all evaluated defense methods for outlier detection struggle to identify the poisonous samples and distinguish them from the original training samples. The performance (i.e., recall) varies slightly between different combinations of classifier, data set, and defense method. However, it is almost always well below 30% and often even below 20%, implying that the vast majority of the poisonous samples remain undetected.

These findings demonstrate that our proposed data poisoning method successfully generates poisonous samples that are on the data manifold and difficult to distinguish from normal training samples. This implies that the detection of our generated poisonous instances is non-trivial and requires substantial research efforts.

6.4.8. Ablation study

From the results (see Figure B.11), we observe the same overall effects as we did in the case of a data poisoning of a global level without any modifications to Algorithm 1 (see Section 6.4.3). We therefore conclude that the sampling strategy does not have an impact on the algorithm's ability to increase the cost of recourse. Most importantly, it provides empirical evidence that the findings of the theorems (from Section 4.1.1) indeed generalize to more complex and realistic scenarios.

From Table 5, we observe that in most cases the sampling strategy (line 4 in Algorithm 1) does not have a statistically significant effect on the performance of the defense methods. Furthermore, we often observe large differences in the median, but these are still not statistically significant due to the high variance (also see Figure B.10). However, in the case of the isolation forest (a classic outlier detection method), despite large variances, the differences are almost always statistically significant and indicate a significant increase in detection performance. This indicates that our proposed sampling strategy for ensuring the plausibility of the poisonous samples can make the detection of such samples more difficult, depending on the detection method used.

7. Conclusion & Summary

In this work, we examined the resilience of counterfactual explanations against data poisoning. To achieve this, we identified and formalized strategies for data poisoning aimed at increasing the cost of recourse on various levels (local vs. global) by injecting poisonous instances into the training data set. We conducted empirical evaluations to assess the impact of data poisoning across several classifiers, benchmark datasets, and various popular and state-of-the-art counterfactual generation methods. Our findings revealed that even the injection of a small number of poisonous instances into the training data set significantly increases the cost of recourse at all levels – on a global level as well as on a local level. Furthermore, we empirically evaluated the effect of a classic data poisoning attack (label flipping), designed to decrease the predictive performance, on the cost of recourse. It turns out that there is no or only a minor effect on the cost of recourse, highlighting the necessity of custom and specialized data poisoning methods such as our proposed Algorithm 1.

Ethical implications and broader impact: The presented findings reveal how easily existing classifiers and state-of-the-art counterfactual generation methods and toolboxes can be deceived through manipulation of the training data. This could significantly undermine users' trust in this XAI method. More specifically, the severe and wide implications of our study become apparent by demonstrating the vulnerability of explanations to manipulations in sensitive domains such as finance, health, and critical infrastructure, together with the potentially serious consequences of poisoned explanations, such as denying fair recourse, obscuring biases, and compromising system safety. Consequently, our study highlights the urgent need for more robust counterfactual generation methods, toolboxes, and defense strategies against malicious data manipulations. In particular, we suggest not using counterfactual explanations in critical applications if the integrity of the training data cannot be guaranteed.

Future research directions: As our paper demonstrates the vulnerability of counterfactual explanations to data poisoning, we call for the community to focus on designing inherently more robust algorithms for computing counterfactual explanations to promote the safe deployment of counterfactuals in practice. In this context, we suggest not only focusing on a single evaluation metric, such as the cost of recourse, but also investigating and ensuring robustness to other aspects, such as fairness. Furthermore, we propose to research the nature of such poisonous training samples to gain knowledge for the development of more advanced defense strategies. In particular, there is an urgent need for a better understanding of how and what properties of certain training samples affect methods for computing counterfactuals. In addition, the demonstrated failure of traditional outlier-based defense methods, such as data sanitization methods, highlights the need for novel defense strategies. In this context, concepts from data valuation might constitute promising avenues worth exploring.

We leave those aspects as future research.

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.

Appendix A. Proofs

Appendix A.1. Proof of Theorem 1

Proof. Sketch: For any $\vec{x}_{\text{orig}}, h(\vec{x}_{\text{orig}}) = y_{\text{orig}}$, assume uniqueness of the solution $\vec{x}'$ – i.e. the closest sample to $\vec{x}_{\text{orig}}$ on the decision boundary:

$$\begin{aligned} &\arg\min_{\vec{x}' \in \mathbb{R}^d} \|\vec{x}' - \vec{x}_{\text{orig}}\|_p \text{ s.t.} \\ &\exists i \neq j : (\vec{x}_i, y_i), (\vec{x}_j, y_j) \in \mathcal{D}, y_i \neq y_j, \text{ with } \|\vec{x}' - \vec{x}_i\|_p = \|\vec{x}' - \vec{x}_j\|_p \end{aligned} \tag{A.1}$$

where we (w.l.o.g.) assume the use of the p-norm as the distance function in the 1-NUN neighbor classifier.

Adding $(\vec{x}', y_{\text{orig}})$ to the training data $\mathcal{D}$ implies that $\vec{x}'$ is no longer the solution to Eq. (A.1). Therefore, the new closest sample on the decision boundary must have a larger distance to $\vec{x}_{\text{orig}}$ than $\vec{x}'$, otherwise it would have been $\vec{x}'$ before! □

Appendix A.2. Proof of Theorem 2

Proof. Sketch: From the triangle-inequality and $\lambda > \|\vec{x}_i - \vec{x}_j\|_2$ it follows that:

$$\|\vec{x}_i - \vec{x}_j\|_2 + \delta_j' \geq \underbrace{\delta_i + \lambda}_{\delta_i'} \quad \leftrightarrow \quad \delta_j' \geq \delta_i + \lambda - \|\vec{x}_i - \vec{x}_j\|_2 \tag{A.2}$$

Because of $\delta_j > \delta_i$, we know that $\delta_j = \alpha\delta_i$ for some $\alpha > 1$. This allows us to rewrite Eq. (A.2):

$$\delta_j' \geq \underbrace{\frac{\delta_j}{\alpha}}_{\delta_i} + \lambda - \|\vec{x}_i - \vec{x}_j\|_2 \tag{A.3}$$

The desired results follows from choosing $\lambda \geq 2\alpha\delta_j + \|\vec{x}_i - \vec{x}_j\|_2$ yields:

$$\begin{aligned} \delta'_j &\geq \frac{\delta_j}{\alpha} + \lambda - \|\vec{x}_i - \vec{x}_j\|_2 \\ &\geq \frac{\delta_j}{\alpha} + 2\alpha\delta_j + \|\vec{x}_i - \vec{x}_j\|_2 - \|\vec{x}_i - \vec{x}_j\|_2 \\ &= \delta_j \end{aligned} \tag{A.4}$$

□

Appendix B. Experiments

Appendix B.1. Global Poisoning Attack

Table B.6: Difference (percentage) in the log-likelihood (i.e., plausibility) under a kernel density estimation of the counterfactuals: no poisoning vs. *global poisoning*. In all cases, we add 70% of the training data as poisonous instances. Positive numbers denote an increase (due to the data poisoning) in the plausibility of counterfactuals, while negative numbers denote the opposite. We report the median (over all folds) rounded to two decimal places, as well as the statistical significance according to the Mann-Whitney U test (ns $\implies$ p-value > 0.05; * $\implies$ p-value ≤ 0.05; ** $\implies$ p-value ≤ 0.01; *** $\implies$ p-value ≤ 0.001).

Classifier	Data set	NUN	DiCE	Proto
SVC	Credit	$1.73 * 10^{-6}\%_{ns}$	$-2.15 * 10^{-3}\%_{ns}$	$-6.13 * 10^{-1}\%_{ns}$
	Diabetes	$-3.57 * 10^{-1}\%_{ns}$	$-2.65 * 10^{-1}\%_{ns}$	$3.84 * 10^{-1}\%_{ns}$
	Crime	$-6.16 * 10^{-9}\%_{ns}$	$2.74 * 10^{-6}\%_{ns}$	$3.74 * 10^{-1}\%_{ns}$
RNF	Credit	$-5.99 * 10^{-7}\%_{ns}$	$-2.15 * 10^{-3}\%_{ns}$	$-8.91 * 10^{-1}\%_{ns}$
	Diabetes	$6.63 * 10^{-1}\%_{ns}$	$-3.95 * 10^{-1}\%_{ns}$	$-8.46 * 10^{-1}\%_{ns}$
	Crime	$-9.04 * 10^{-10}\%_{ns}$	$-1.49 * 10^{-10}\%_{ns}$	$-1.61 * 10^{-1}\%_{ns}$
DNN	Credit	$-3.86 * 10^{-6}\%_{ns}$	$-2.07 * 10^{-6}\%_{ns}$	$8.72 * 10^{-1}\%_{ns}$
	Diabetes	$4.50 * 10^{-1}\%_{ns}$	$-6.92 * 10^{-2}\%_{ns}$	$-5.83 * 10^{-1}\%_{ns}$
	Crime	$-1.34 * 10^{-8}\%_{ns}$	$4.89 * 10^{-10}\%_{ns}$	$7.91 * 10^{-1}\%_{ns}$

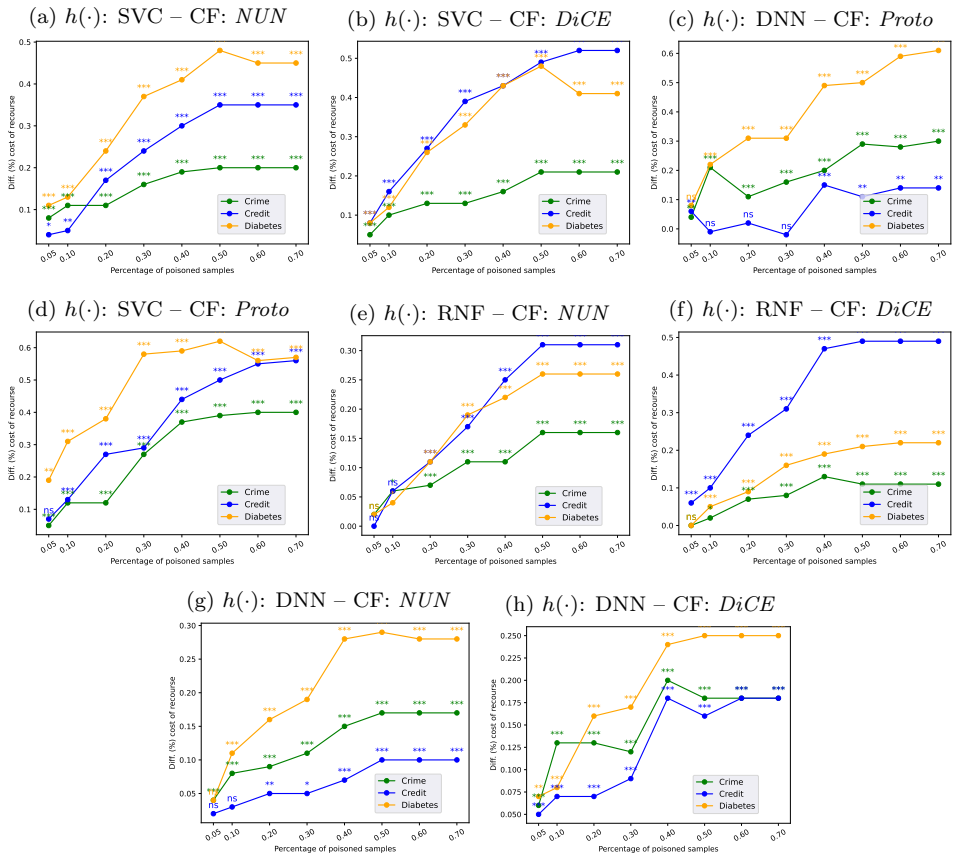


Figure B.4: Global data poisoning attack: Difference (percentage) in the cost of recourse vs. percentage of poisoned instances (5% to 70%). We report the median (over all folds) rounded to two decimal places, as well as the statistical significance according to the Mann-Whitney U test (ns $\implies$ p-value > 0.05; * $\implies$ p-value ≤ 0.05; ** $\implies$ p-value ≤ 0.01; *** $\implies$ p-value ≤ 0.001.

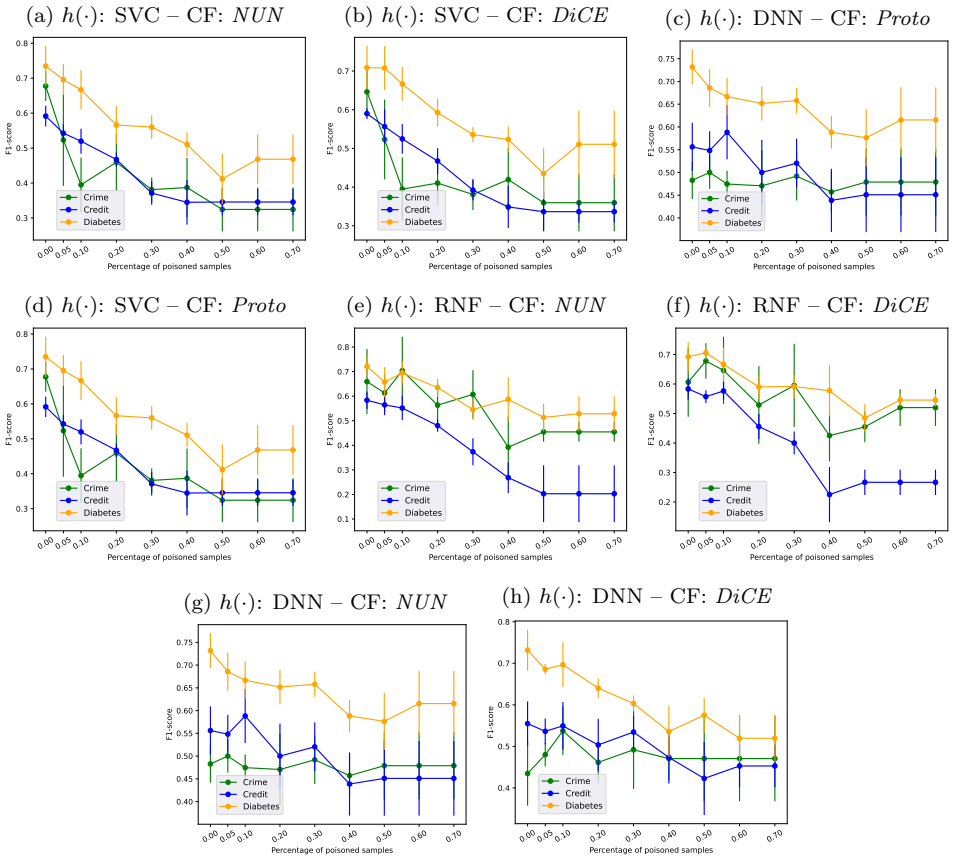


Figure B.5: Global data poisoning attack: Median and standard deviation (over all folds) F1-score of the classifier for different percentages of poisoned samples (0% to 70%).

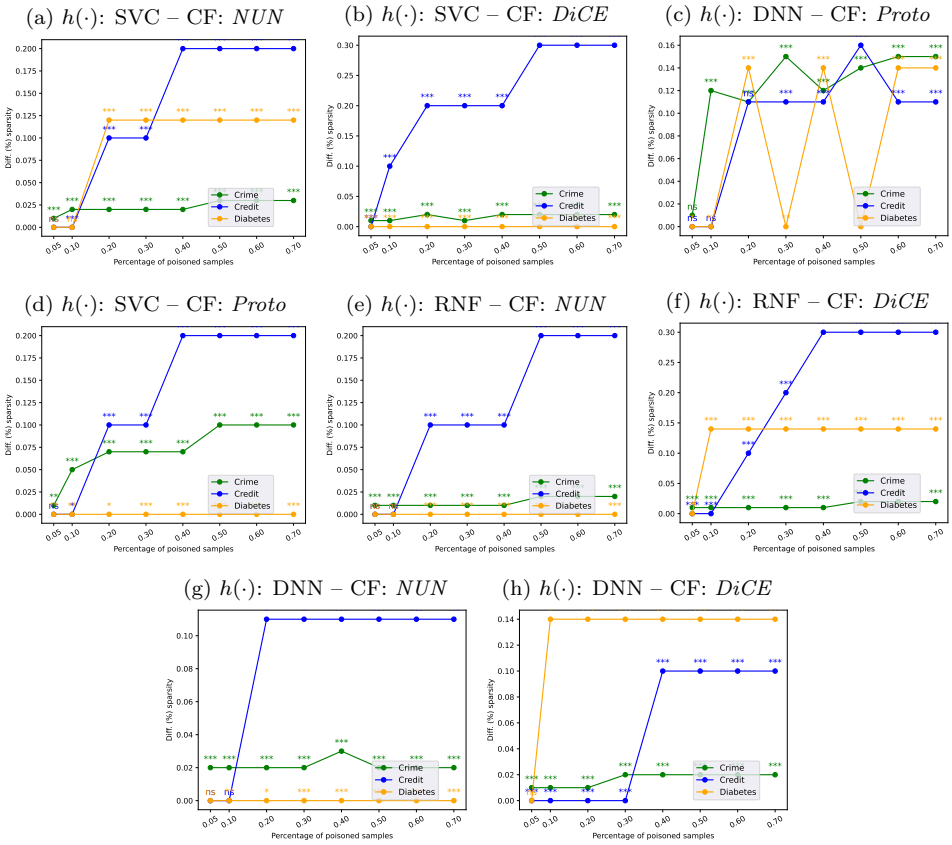


Figure B.6: Global data poisoning attack: Sparsity of the counterfactual explanations for different percentages of poisoned samples (0% to 70%). We report the median (over all folds) rounded to two decimal places, as well as the statistical significance according to the Mann-Whitney U test (ns $\implies$ p-value > 0.05; * $\implies$ p-value ≤ 0.05; ** $\implies$ p-value ≤ 0.01; *** $\implies$ p-value ≤ 0.001.

Appendix B.2. Sub-group Poisoning Attack

Table B.7: Nearest Unlike Neighbor (NUN). Difference (percentage) in the cost of recourse between protected groups (see Eq. (21)): no vs. poisoning on a *sub-group level*. We report the median (over all folds) rounded to two decimal places, as well as the statistical significance according to the Mann-Whitney U test (ns $\implies$ p-value > 0.05; * $\implies$ p-value ≤ 0.05; ** $\implies$ p-value ≤ 0.01; *** $\implies$ p-value ≤ 0.001).

Classifier		Percentage of poisoned samples							
	Dataset	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7
SVC	Credit	$24\%_{ns}$	$45\%_{ns}$	$15\%_{***}$	$-15\%_{***}$	$52\%_{**}$	$10\%_{***}$	$10\%_{***}$	$10\%_{***}$
	Diabetes	$34\%_{ns}$	$41\%_{**}$	$109\%_{***}$	$84\%_{***}$	$106\%_{***}$	$111\%_{***}$	$110\%_{***}$	$110\%_{***}$
	Crime	$44\%_{ns}$	$53\%_{**}$	$72\%_{***}$	$73\%_{***}$	$99\%_{***}$	$86\%_{***}$	$86\%_{***}$	$86\%_{***}$
RNF	Credit	$131\%_{ns}$	$144\%_{ns}$	$69\%_{ns}$	$325\%_{ns}$	$519\%_{ns}$	$88\%_{ns}$	$381\%_{ns}$	$381\%_{ns}$
	Diabetes	$275\%_{ns}$	$225\%_{ns}$	$319\%_{*}$	$419\%_{***}$	$344\%_{***}$	$269\%_{**}$	$325\%_{***}$	$325\%_{***}$
	Crime	$1\%_{ns}$	$37\%_{ns}$	$53\%_{ns}$	$81\%_{ns}$	$52\%_{*}$	$52\%_{ns}$	$41\%_{ns}$	$41\%_{ns}$
DNN	Credit	$373\%_{ns}$	$227\%_{ns}$	$82\%_{ns}$	$364\%_{ns}$	$-36\%_{ns}$	$27\%_{ns}$	$27\%_{ns}$	$27\%_{ns}$
	Diabetes	$-22\%_{ns}$	$25\%_{**}$	$7\%_{***}$	$-15\%_{***}$	$70\%_{***}$	$40\%_{***}$	$0\%_{ns}$	$0\%_{ns}$
	Crime	$86\%_{ns}$	$64\%_{*}$	$86\%_{**}$	$117\%_{***}$	$114\%_{***}$	$119\%_{***}$	$119\%_{***}$	$119\%_{***}$

Table B.8: DiCE. Difference (percentage) in the cost of recourse between protected groups (see Eq. (21)): no vs. poisoning on a *sub-group level*. We report the median (over all folds) rounded to two decimal places, as well as the statistical significance according to the Mann-Whitney U test (ns $\implies$ p-value > 0.05; * $\implies$ p-value ≤ 0.05; ** $\implies$ p-value ≤ 0.01; *** $\implies$ p-value ≤ 0.001).

Classifier		Percentage of poisoned samples							
	Dataset	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7
SVC	Credit	$117\%_{**}$	$86\%_{***}$	$120\%_{***}$	$111\%_{***}$	$124\%_{***}$	$70\%_{***}$	$70\%_{***}$	$70\%_{***}$
	Diabetes	$7\%_{***}$	$-12\%_{***}$	$61\%_{***}$	$32\%_{***}$	$50\%_{***}$	$57\%_{***}$	$42\%_{***}$	$42\%_{***}$
	Crime	$6\%_{ns}$	$34\%_{***}$	$69\%_{***}$	$59\%_{***}$	$59\%_{***}$	$56\%_{***}$	$56\%_{***}$	$56\%_{***}$
RNF	Credit	$-28\%_{***}$	$44\%_{*}$	$90\%_{***}$	$137\%_{***}$	$39\%_{***}$	$-39\%_{***}$	$-39\%_{***}$	$-39\%_{***}$
	Diabetes	$74\%_{ns}$	$53\%_{ns}$	$4\%_{***}$	$19\%_{***}$	$132\%_{***}$	$87\%_{***}$	$98\%_{***}$	$98\%_{***}$
	Crime	$-1\%_{ns}$	$44\%_{***}$	$68\%_{**}$	$67\%_{***}$	$62\%_{***}$	$59\%_{***}$	$59\%_{***}$	$59\%_{***}$
DNN	Credit	$25\%_{**}$	$-28\%_{***}$	$-73\%_{***}$	$-83\%_{***}$	$-35\%_{***}$	$27\%_{***}$	$27\%_{***}$	$27\%_{***}$
	Diabetes	$19\%_{ns}$	$28\%_{***}$	$44\%_{***}$	$7\%_{***}$	$77\%_{***}$	$60\%_{***}$	$33\%_{***}$	$33\%_{***}$
	Crime	$59\%_{***}$	$66\%_{***}$	$68\%_{***}$	$70\%_{***}$	$73\%_{***}$	$84\%_{***}$	$84\%_{***}$	$84\%_{***}$

Table B.9: Counterfactuals guided by prototypes (Proto). Difference (percentage) in the cost of recourse between protected groups (see Eq. (21)): no vs. poisoning on a *sub-group level*. We report the median (over all folds) rounded to two decimal places, as well as the statistical significance according to the Mann-Whitney U test (ns $\implies$ p-value > 0.05; * $\implies$ p-value ≤ 0.05; ** $\implies$ p-value ≤ 0.01; *** $\implies$ p-value ≤ 0.001).

Classifier		Percentage of poisoned samples							
	Dataset	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7
SVC	Credit	$957\%_{ns}$	$118\%_{*}$	$162\%_{***}$	$2032\%_{***}$	$2450\%_{**}$	$150\%_{***}$	$-57\%_{***}$	$0\%_{ns}$
	Diabetes	$12\%_{ns}$	$-12\%_{***}$	$41\%_{***}$	$8\%_{***}$	$10\%_{***}$	$62\%_{***}$	$38\%_{***}$	$66\%_{***}$
	Crime	$17\%_{ns}$	$25\%_{*}$	$35\%_{***}$	$40\%_{***}$	$33\%_{***}$	$18\%_{***}$	$17\%_{***}$	$14\%_{***}$
RNF	Credit	$341\%_{ns}$	$-68\%_{*}$	$-71\%_{**}$	$-1\%_{*}$	$-10\%_{**}$	$-80\%_{ns}$	$-68\%_{**}$	$375\%_{*}$
	Diabetes	$57\%_{ns}$	$33\%_{ns}$	$103\%_{**}$	$53\%_{***}$	$137\%_{**}$	$113\%_{***}$	$95\%_{***}$	$111\%_{***}$
	Crime	$7\%_{ns}$	$-3\%_{ns}$	$19\%_{*}$	$12\%_{*}$	$15\%_{ns}$	$4\%_{**}$	$11\%_{ns}$	$10\%_{*}$
DNN	Credit	$0\%_{ns}$	$65\%_{ns}$	$-28\%_{ns}$	$37\%_{ns}$	$-90\%_{ns}$	$-61\%_{ns}$	$3\%_{ns}$	$191\%_{ns}$
	Diabetes	$-21\%_{ns}$	$-21\%_{*}$	$0\%_{ns}$	$41\%_{***}$	$37\%_{***}$	$35\%_{***}$	$42\%_{**}$	$25\%_{***}$
	Crime	$68\%_{ns}$	$45\%_{**}$	$63\%_{*}$	$91\%_{***}$	$108\%_{***}$	$94\%_{*}$	$104\%_{*}$	$90\%_{**}$

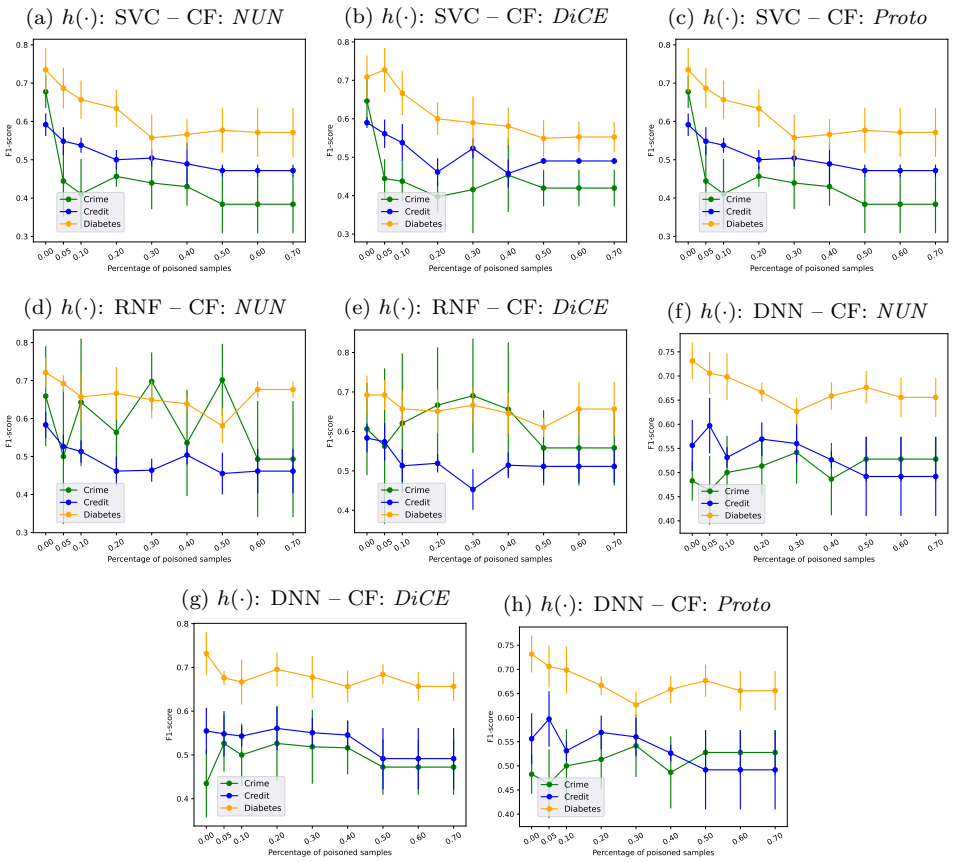


Figure B.7: Sub-group data poisoning attack: Median and standard deviation (over all folds) F1-score of the classifier for different percentages of poisoned samples (0% to 70%).

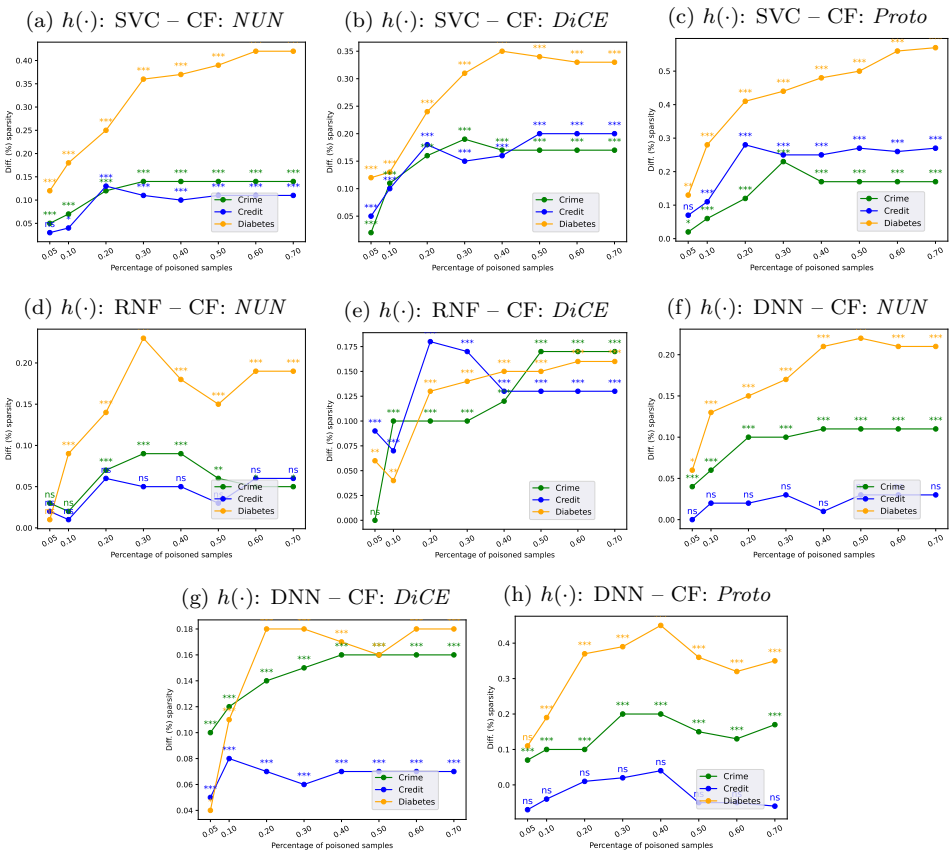
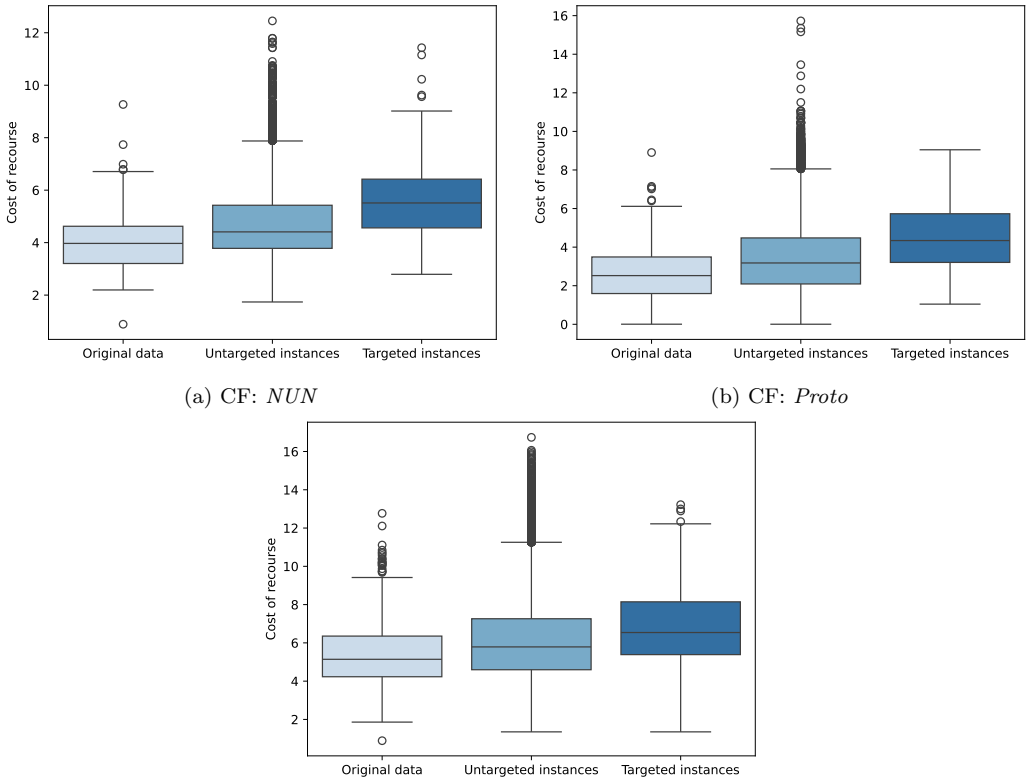


Figure B.8: Sub-group data poisoning attack: Difference in the sparsity of the counterfactual explanations for different percentages of poisoned samples (0% to 70%). We report the median (over all folds) rounded to two decimal places, as well as the statistical significance according to the Mann-Whitney U test (ns $\implies$ p-value > 0.05; * $\implies$ p-value ≤ 0.05; ** $\implies$ p-value ≤ 0.01; *** $\implies$ p-value ≤ 0.001

Appendix B.3. Local Poisoning Attack



(a) CF: *NUN*

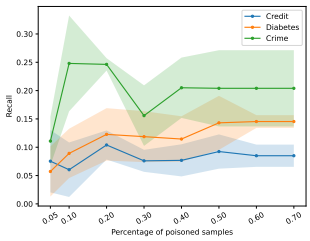
(b) CF: *Proto*

(c) CF: *DiCE*

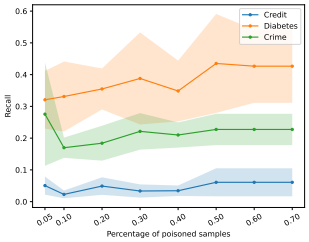
Figure B.9: *Local* data poisoning: Cost of recourse (over all test samples) in the case of the diabetes data set and a DNN classifier. Cost of recourse without any data poisoning, of untargeted instances and targeted instances in a local data poisoning.

Appendix B.4. Detection of Poisonous Instances

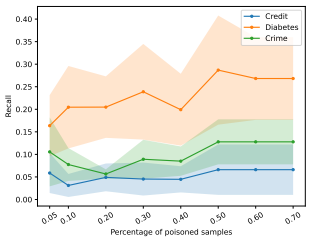
(a) $h(\cdot)$: SVC – *Isloation Forest*



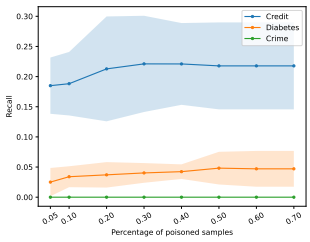
(b) $h(\cdot)$: SVC – *k-NN-defense*



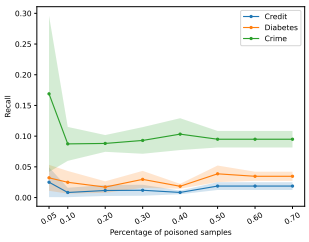
(c) $h(\cdot)$: SVC – ℓ_2-*defense*



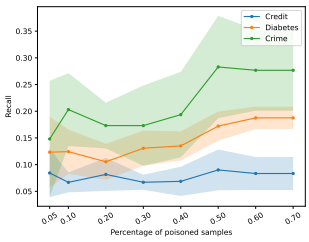
(d) $h(\cdot)$: SVC – *slabdefense*



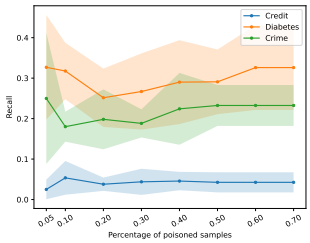
(e) $h(\cdot)$: SVC – *Local Outlier Factory*



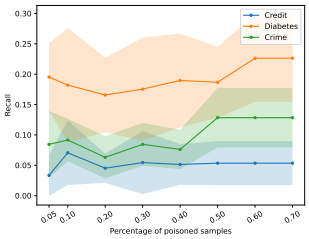
(f) $h(\cdot)$: RNF – *Isloation Forest*



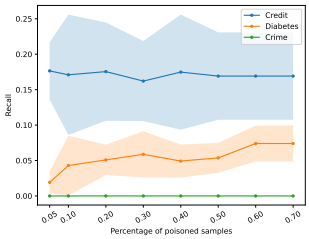
(g) $h(\cdot)$: RNF – *k-NN-defense*



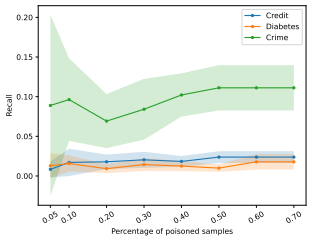
(h) $h(\cdot)$: RNF – ℓ_2-*defense*



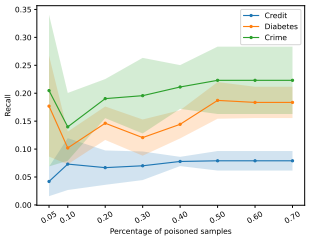
(i) $h(\cdot)$: RNF – *slab-defense*



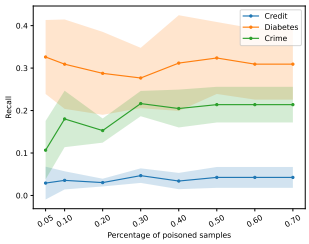
(j) $h(\cdot)$: RNF – *Local Outlier Factory*



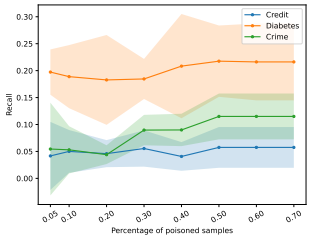
(k) $h(\cdot)$: DNN – *Isloation Forest*



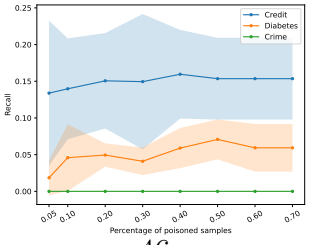
(l) $h(\cdot)$: DNN – *k-NN-defense*



(m) $h(\cdot)$: DNN – ℓ_2-*defense*



(n) $h(\cdot)$: DNN – *slab-defense*



(o) $h(\cdot)$: DNN – *Local Outlier Factory*

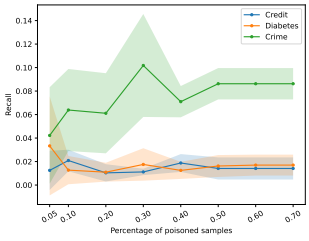


Figure B.10: Global data poisoning attack: Recall of different data sanitization methods evaluated on different percentages of poisoned samples (0% to 70%). We report the mean and standard deviation over all folds (larger numbers are better).

Appendix B.5. Ablation study

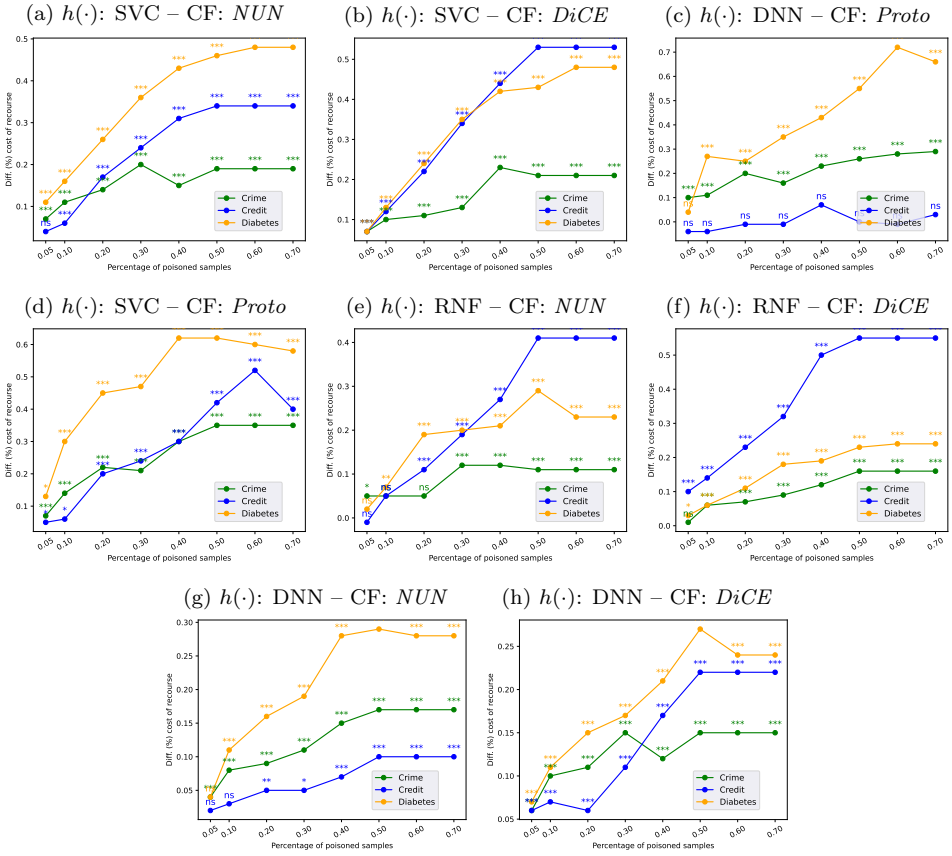


Figure B.11: *Ablation study (uniform sampling)* - Global data poisoning attack: Difference (percentage) in the cost of recourse vs. percentage of poisoned instances (5% to 70%). We report the median (over all folds) rounded to two decimal places, as well as the statistical significance according to the Mann-Whitney U test (ns $\implies$ p-value > 0.05; * $\implies$ p-value ≤ 0.05; ** $\implies$ p-value ≤ 0.01; *** $\implies$ p-value ≤ 0.001.

References

[1] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al., A survey of large language models, arXiv preprint arXiv:2303.18223 (2023).

[2] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, T. Salimans, Cascaded diffusion models for high fidelity image generation, The Journal of Machine Learning Research 23 (1) (2022) 2249–2281.
URL https://jmlr.org/papers/v23/21-0635.html

[3] Z. Qian, K. Huang, Q.-F. Wang, X.-Y. Zhang, A survey of robust adversarial training in pattern recognition: Fundamental, theory, and methodologies, Pattern Recognition 131 (2022) 108889.

[4] Y. Li, Y. Jiang, Z. Li, S.-T. Xia, Backdoor learning: A survey, IEEE Transactions on Neural Networks and Learning Systems 35 (1) (2024) 5–22. doi:10.1109/tnnls.2022.3182979.
URL http://dx.doi.org/10.1109/TNNLS.2022.3182979

[5] J. Fan, Q. Yan, M. Li, G. Qu, Y. Xiao, A survey on data poisoning attacks and defenses (Jul. 2022). doi:10.1109/dsc55868.2022.00014.
URL http://dx.doi.org/10.1109/DSC55868.2022.00014

[6] W. Zhao, S. Alwidian, Q. H. Mahmoud, Adversarial training methods for deep learning: A systematic review, Algorithms 15 (8) (2022) 283.

[7] X. Chen, C. Liu, B. Li, K. Lu, D. Song, Targeted backdoor attacks on deep learning systems using data poisoning (2017). doi:10.48550/ARXIV.1712.05526.
URL https://arxiv.org/abs/1712.05526

[8] J. Lin, L. Dang, M. Rahouti, K. Xiong, Ml attack models: adversarial attacks and data poisoning attacks, arXiv preprint arXiv:2112.02797 (2021).

[9] V. Tolpegin, S. Truex, M. E. Gursoy, L. Liu, Data poisoning attacks against federated learning systems, in: L. Chen, N. Li, K. Liang, S. A. Schneider (Eds.), Computer Security - ESORICS 2020 - 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14-18, 2020, Proceedings, Part I, Vol. 12308 of Lecture Notes in Computer Science, Springer, 2020, pp. 480–501.
URL https://doi.org/10.1007/978-3-030-58951-6_24

[10] N. Mehrabi, M. Naveed, F. Morstatter, A. Galstyan, Exacerbating algorithmic bias through fairness attacks, in: Thirty-Fifth AAAI Conference

on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, 2021, pp. 8930–8938. doi:10.1609/AAAI.V35I10.17080.
URL https://doi.org/10.1609/aaai.v35i10.17080

[11] S. Shan, W. Ding, J. Passananti, S. Wu, H. Zheng, B. Y. Zhao, Nightshade: Prompt-specific poisoning attacks on text-to-image generative models (2024) 807–825doi:10.1109/SP54263.2024.00207.
URL https://doi.org/10.1109/SP54263.2024.00207

[12] A. Bojchevski, S. Günnemann, Adversarial attacks on node embeddings via graph poisoning, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, Vol. 97 of Proceedings of Machine Learning Research, PMLR, 2019, pp. 695–704.
URL http://proceedings.mlr.press/v97/bojchevski19a.html

[13] Z. Yang, X. He, Z. Li, M. Backes, M. Humbert, P. Berrang, Y. Zhang, Data poisoning attacks against multimodal encoders, in: A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, J. Scarlett (Eds.), International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, Vol. 202 of Proceedings of Machine Learning Research, PMLR, 2023, pp. 39299–39313.
URL https://proceedings.mlr.press/v202/yang23f.html

[14] 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 (2016) 4.5.

[15] E. Commission, D.-G. for Communications Networks, Content, 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).
URL https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence

[16] R. Dwivedi, D. Dave, H. Naik, S. Singhal, O. F. Rana, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, R. Ranjan, Explainable AI (XAI): core ideas, techniques, and solutions, ACM Comput. Surv. 55 (9) (2023) 194:1–194:33.
URL https://doi.org/10.1145/3561048

[17] L. Longo, M. Brcic, F. Cabitza, J. Choi, R. Confalonieri, J. D. Ser, R. Guidotti, Y. Hayashi, F. Herrera, A. Holzinger, R. Jiang, H. Khosravi, F. Lecue, G. Malgieri, A. Páez, W. Samek, J. Schneider, T. Speith, S. Stumpf, Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions, Information Fusion 106 (2024) 102301. doi:https://doi.org/10.1016/j.inffus.2024.102301.
URL https://www.sciencedirect.com/science/article/pii/S1566253524000794

[18] A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115. doi:10.1016/J.INFFUS.2019.12.012.
URL https://doi.org/10.1016/j.inffus.2019.12.012

[19] A. Rawal, J. McCoy, D. B. Rawat, B. M. Sadler, R. S. Amant, Recent advances in trustworthy explainable artificial intelligence: Status, challenges, and perspectives, IEEE Trans. Artif. Intell. 3 (6) (2022) 852–866. doi:10.1109/TAI.2021.3133846.
URL https://doi.org/10.1109/TAI.2021.3133846

[20] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the gdpr, Harv. JL & Tech. 31 (2017) 841.

[21] R. M. J. Byrne, Counterfactuals in explainable artificial intelligence (XAI): evidence from human reasoning, in: S. Kraus (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, ijcai.org, 2019, pp. 6276–6282. doi:10.24963/IJCAI.2019/876.
URL https://doi.org/10.24963/ijcai.2019/876

[22] H. Baniecki, P. Biecek, Adversarial attacks and defenses in explainable artificial intelligence: A survey, Inf. Fusion 107 (2024) 102303. doi:10.1016/J.INFFUS.2024.102303.
URL https://doi.org/10.1016/j.inffus.2024.102303

[23] H. Baniecki, W. Kretowicz, P. Biecek, Fooling partial dependence via data poisoning, in: M. Amini, S. Canu, A. Fischer, T. Guns, P. K. Novak, G. Tsoumakas (Eds.), Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2022, Grenoble, France, September 19-23, 2022, Proceedings, Part III, Vol. 13715 of Lecture Notes in Computer Science, Springer, 2022, pp. 121–136.
URL https://doi.org/10.1007/978-3-031-26409-2_8

[24] D. Slack, A. Hilgard, H. Lakkaraju, S. Singh, Counterfactual explanations can be manipulated (2021) 62–75.
URL https://proceedings.neurips.cc/paper/2021/hash/009c434cab57de48a31f6b669e7ba266-Abstract.html

[25] S. Mishra, S. Dutta, J. Long, D. Magazzeni, A survey on the robustness of feature importance and counterfactual explanations, arXiv preprint arXiv:2111.00358 (2021).

[26] A. Artelt, V. Vaquet, R. Velioglu, F. Hinder, J. Brinkrolf, M. Schilling, B. Hammer, Evaluating robustness of counterfactual explanations, in: IEEE Symposium Series on Computational Intelligence, SSCI 2021, Orlando, FL, USA, December 5-7, 2021, IEEE, 2021, pp. 1–9. doi:10.1109/SSCI50451.2021.9660058.
URL https://doi.org/10.1109/SSCI50451.2021.9660058

[27] M. Virgolin, S. Fracaros, On the robustness of sparse counterfactual explanations to adverse perturbations, Artif. Intell. 316 (2023) 103840. doi:10.1016/J.ARTINT.2022.103840.
URL https://doi.org/10.1016/j.artint.2022.103840

[28] J. Jiang, F. Leofante, A. Rago, F. Toni, Robust counterfactual explanations in machine learning: A survey (2024) 8086–8094.
URL https://www.ijcai.org/proceedings/2024/894

[29] J. Steinhardt, P. W. Koh, P. Liang, Certified defenses for data poisoning attacks, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach,

R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 3517–3529.
URL https://proceedings.neurips.cc/paper/2017/hash/9d7311ba459f9e45ed746755a32dcd11-Abstract.html

[30] A. Paudice, L. Muñoz-González, E. C. Lupu, Label sanitization against label flipping poisoning attacks (2019). doi:10.1007/978-3-030-13453-2_1.
URL http://dx.doi.org/10.1007/978-3-030-13453-2_1

[31] A. Paudice, L. Muñoz-González, A. Gyorgy, E. C. Lupu, Detection of adversarial training examples in poisoning attacks through anomaly detection (2018). doi:10.48550/ARXIV.1802.03041.
URL https://arxiv.org/abs/1802.03041

[32] G. F. Cretu, A. Stavrou, M. E. Locasto, S. J. Stolfo, A. D. Keromytis, Casting out demons: Sanitizing training data for anomaly sensors (May 2008). doi:10.1109/sp.2008.11.
URL http://dx.doi.org/10.1109/SP.2008.11

[33] P. W. Koh, J. Steinhardt, P. Liang, Stronger data poisoning attacks break data sanitization defenses, Machine Learning 111 (1) (2021) 1–47. doi:10.1007/s10994-021-06119-y.
URL http://dx.doi.org/10.1007/s10994-021-06119-y

[34] D. Brown, H. Kvinge, Making corgis important for honeycomb classification: Adversarial attacks on concept-based explainability tools, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023 - Workshops, Vancouver, BC, Canada, June 17-24, 2023, IEEE, 2023, pp. 620–627. doi:10.1109/CVPRW59228.2023.00069.
URL https://doi.org/10.1109/CVPRW59228.2023.00069

[35] M. Noppel, C. Wressnegger, A brief systematization of explanation-aware attacks, in: A. Hotho, S. Rudolph (Eds.), KI 2024: Advances in Artificial Intelligence - 47th German Conference on AI, Würzburg, Germany, September 25-27, 2024, Proceedings, Vol. 14992 of Lecture Notes in Computer Science, Springer, 2024, pp. 350–354. doi:

10.1007/978-3-031-70893-0_30.
URL https://doi.org/10.1007/978-3-031-70893-0_30

[36] J. Zhang, H. Chao, G. Dasegowda, G. Wang, M. K. Kalra, P. Yan, Overlooked trustworthiness of saliency maps (2022). doi:10.1007/978-3-031-16437-8_43.
URL http://dx.doi.org/10.1007/978-3-031-16437-8_43

[37] Y. Zhao, Y. Wang, T. Derr, Fairness and explainability: Bridging the gap towards fair model explanations, Proceedings of the AAAI Conference on Artificial Intelligence 37 (9) (2023) 11363–11371. doi:10.1609/aaai.v37i9.26344.
URL http://dx.doi.org/10.1609/aaai.v37i9.26344

[38] L. Hancox-Li, Robustness in machine learning explanations: Does it matter?, in: Proceedings of the 2020 conference on fairness, accountability, and transparency, 2020, pp. 640–647.

[39] A. Artelt, B. Hammer, "Explain it in the Same Way!" – Model-Agnostic Group Fairness of Counterfactual Explanations, in: A. Ofra, T. Miller, H. Baier (Eds.), Workshop on XAI, 2023.
URL https://sites.google.com/view/xai2023

[40] H. Baniecki, P. Biecek, Manipulating shap via adversarial data perturbations (student abstract), in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 12907–12908.

[41] M. Riveiro, S. Thill, The challenges of providing explanations of AI systems when they do not behave like users expect, in: A. Bellogín, L. Boratto, O. C. Santos, L. Ardissono, B. P. Knijnenburg (Eds.), UMAP '22: 30th ACM Conference on User Modeling, Adaptation and Personalization, Barcelona, Spain, July 4 - 7, 2022, ACM, 2022, pp. 110–120.
URL https://doi.org/10.1145/3503252.3531306

[42] A. Karimi, G. Barthe, B. Schölkopf, I. Valera, A survey of algorithmic recourse: Contrastive explanations and consequential recommendations, ACM Comput. Surv. 55 (5) (2023) 95:1–95:29. doi:10.1145/3527848.
URL https://doi.org/10.1145/3527848

[43] C. Molnar, Interpretable Machine Learning, 2019.

[44] S. Verma, V. Boonsanong, M. Hoang, K. Hines, J. Dickerson, C. Shah, Counterfactual explanations and algorithmic recourses for machine learning: A review (2024).
URL https://doi.org/10.1145/3677119

[45] R. Guidotti, Counterfactual explanations and how to find them: literature review and benchmarking, Data Min. Knowl. Discov. 38 (5) (2024) 2770–2824. doi:10.1007/S10618-022-00831-6.
URL https://doi.org/10.1007/s10618-022-00831-6

[46] A. V. Looveren, J. Klaise, Interpretable counterfactual explanations guided by prototypes 12976 (2021) 650–665. doi:10.1007/978-3-030-86520-7_40.
URL https://doi.org/10.1007/978-3-030-86520-7_40

[47] R. Poyiadzi, K. Sokol, R. Santos-Rodríguez, T. D. Bie, P. A. Flach, FACE: feasible and actionable counterfactual explanations, in: A. N. Markham, J. Powles, T. Walsh, A. L. Washington (Eds.), AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, February 7-8, 2020, ACM, 2020, pp. 344–350. doi:10.1145/3375627.3375850.
URL https://doi.org/10.1145/3375627.3375850

[48] R. K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, in: M. Hildebrandt, C. Castillo, L. E. Celis, S. Ruggieri, L. Taylor, G. Zanfir-Fortuna (Eds.), FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020, ACM, 2020, pp. 607–617. doi:10.1145/3351095.3372850.
URL https://doi.org/10.1145/3351095.3372850

[49] Y. Wang, H. Qian, Y. Liu, W. Guo, C. Miao, Flexible and robust counterfactual explanations with minimal satisfiable perturbations, in: I. Frommholz, F. Hopfgartner, M. Lee, M. Oakes, M. Lalmas, M. Zhang, R. L. T. Santos (Eds.), Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023, ACM, 2023, pp. 2596–2605. doi:10.1145/3583780.3614885.
URL https://doi.org/10.1145/3583780.3614885

[50] S. Zhang, X. Chen, S. Wen, Z. Li, Density-based reliable and robust explainer for counterfactual explanation, Expert Syst. Appl. 226 (2023) 120214. doi:10.1016/J.ESWA.2023.120214.
URL https://doi.org/10.1016/j.eswa.2023.120214

[51] F. Leofante, N. Potyka, Promoting counterfactual robustness through diversity (2024) 21322–21330doi:10.1609/AAAI.V38I19.30127.
URL https://doi.org/10.1609/aaai.v38i19.30127

[52] J. von Kügelgen, A. Karimi, U. Bhatt, I. Valera, A. Weller, B. Schölkopf, On the fairness of causal algorithmic recourse, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, AAAI Press, 2022, pp. 9584–9594. doi:10.1609/AAAI.V36I9.21192.
URL https://doi.org/10.1609/aaai.v36i9.21192

[53] S. Sharma, A. H. Gee, D. Paydarfar, J. Ghosh, Fair-n: Fair and robust neural networks for structured data, in: M. Fourcade, B. Kuipers, S. Lazar, D. K. Mulligan (Eds.), AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021, ACM, 2021, pp. 946–955. doi:10.1145/3461702.3462559.
URL https://doi.org/10.1145/3461702.3462559

[54] S. Sharma, J. Henderson, J. Ghosh, CERTIFAI: A common framework to provide explanations and analyse the fairness and robustness of black-box models, in: A. N. Markham, J. Powles, T. Walsh, A. L. Washington (Eds.), AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, February 7-8, 2020, ACM, 2020, pp. 166–172. doi:10.1145/3375627.3375812.
URL https://doi.org/10.1145/3375627.3375812

[55] B. V. Dasarathy, Nearest unlike neighbor (nun): an aid to decision confidence estimation, Optical Engineering 34 (9) (1995) 2785. doi:10.1117/12.210755.
URL http://dx.doi.org/10.1117/12.210755

[56] J. Steinhardt, P. W. Koh, P. Liang, Certified defenses for data poisoning attacks (2017) 3517–3529.

URL https://proceedings.neurips.cc/paper/2017/hash/9d7311ba459f9e45ed746755a32dcd11-Abstract.html

[57] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, D. Mukhopadhyay, A survey on adversarial attacks and defences, CAAI Trans. Intell. Technol. 6 (1) (2021) 25–45. doi:10.1049/CIT2.12028.
URL https://doi.org/10.1049/cit2.12028

[58] J. Rauber, R. Zimmermann, M. Bethge, W. 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) (2020) 2607. doi:10.21105/joss.02607.

[59] A. Artelt, S. G. Vrachimis, D. G. Eliades, U. Kuhl, B. Hammer, M. M. Polycarpou, Interpretable event diagnosis in water distribution networks (2025). doi:10.48550/ARXIV.2505.07299.
URL https://arxiv.org/abs/2505.07299

[60] T. R. Gadekallu, P. Kumar Reddy Maddikunta, P. Boopathy, N. Deepa, R. Chengoden, N. Victor, W. Wang, W. Wang, Y. Zhu, K. Dev, Xai for industry 5.0—concepts, opportunities, challenges, and future directions, IEEE Open Journal of the Communications Society 6 (2025) 2706–2729. doi:10.1109/ojcoms.2024.3473891.
URL http://dx.doi.org/10.1109/OJCOMS.2024.3473891

[61] V. Vaquet, F. Hinder, J. Vaquet, K. Lammers, L. Quakernack, B. Hammer, Localizing of anomalies in critical infrastructure using model-based drift explanations, in: 2024 International Joint Conference on Neural Networks (IJCNN), IEEE, 2024, pp. 1–8. doi:10.1109/IJCNN60899.2024.10651472.

[62] A. Artelt, S. Vrachimis, D. Eliades, M. Polycarpou, B. Hammer, One explanation to rule them all – ensemble consistent explanations, in: R. Weber, O. Amir, T. Miller (Eds.), Workshop on XAI, 2022. doi:10.48550/arXiv.2205.08974.
URL https://sites.google.com/view/xai2022

[63] S. G. Vrachimis, M. S. Kyriakou, D. G. Eliades, M. M. Polycarpou, LeakDB : A benchmark dataset for leakage diagnosis in water distribution networks description of benchmark, in: WDSA / CCWI Joint Conference Proceedings, Vol. 1, 2018. doi:10.5281/zenodo.1313116.

[64] A. Artelt, M. S. Kyriakou, S. G. Vrachimis, D. G. Eliades, B. Hammer, M. M. Polycarpou, Epyt-flow: A toolkit for generating water distribution network data, Journal of Open Source Software 9 (103) (2024) 7104. doi:10.21105/joss.07104.
URL https://doi.org/10.21105/joss.07104

[65] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression (2004).

[66] D. Dheeru, E. K. Taniskidou, Uci machine learning repository (2017).

[67] T. L. Quy, A. Roy, V. Iosifidis, W. Zhang, E. Ntoutsi, A survey on datasets for fairness-aware machine learning, WIREs Data Mining Knowl. Discov. 12 (3) (2022). doi:10.1002/WIDM.1452.
URL https://doi.org/10.1002/widm.1452

[68] Statlog (german credit data) data set, https://archive.ics.uci.edu/ml/datasets/Statlog+German+Credit+Data (1994).

[69] A. Paudice, L. Muñoz-González, E. C. Lupu, Label sanitization against label flipping poisoning attacks, in: ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings 18, Springer, 2019, pp. 5–15. doi:10.1007/978-3-030-13453-2_1.
URL https://doi.org/10.1007/978-3-030-13453-2_1

[70] Y. Jiang, W. Zhang, Y. Chen, Data quality detection mechanism against label flipping attacks in federated learning, IEEE Transactions on Information Forensics and Security 18 (2023) 1625–1637. doi:10.1109/TIFS.2023.3249568.
URL https://doi.org/10.1109/TIFS.2023.3249568

[71] F. T. Liu, K. M. Ting, Z. Zhou, Isolation forest, in: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy, IEEE Computer Society, 2008, pp. 413–422. doi:10.1109/ICDM.2008.17.
URL https://doi.org/10.1109/ICDM.2008.17

[72] M. M. Breunig, H. Kriegel, R. T. Ng, J. Sander, LOF: identifying density-based local outliers, in: W. Chen, J. F. Naughton, P. A. Bern-

stein (Eds.), Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA, ACM, 2000, pp. 93–104. doi:10.1145/342009.335388.
URL https://doi.org/10.1145/342009.335388

[73] C. Frederickson, M. Moore, G. Dawson, R. Polikar, Attack strength vs. detectability dilemma in adversarial machine learning (Jul. 2018). doi:10.1109/ijcnn.2018.8489495.
URL http://dx.doi.org/10.1109/IJCNN.2018.8489495