Formalising the Robustness of Counterfactual Explanations for Neural Networks

Junqi Jiang*, **Francesco Leofante***, **Antonio Rago**, **Francesca Toni**

Department of Computing, Imperial College London, UK
{junqi.jiang, f.leofante, a.rago, f.toni}@imperial.ac.uk

Abstract

The use of counterfactual explanations (CFXs) is an increasingly popular explanation strategy for machine learning models. However, recent studies have shown that these explanations may not be robust to changes in the underlying model (e.g., following retraining), which raises questions about their reliability in real-world applications. Existing attempts towards solving this problem are heuristic, and the robustness to model changes of the resulting CFXs is evaluated with only a small number of retrained models, failing to provide exhaustive guarantees. To remedy this, we propose Δ-*robustness*, the first notion to formally and deterministically assess the robustness (to model changes) of CFXs for neural networks. We introduce an abstraction framework based on interval neural networks to verify the Δ-robustness of CFXs against a possibly infinite set of changes to the model parameters, i.e., weights and biases. We then demonstrate the utility of this approach in two distinct ways. First, we analyse the Δ-robustness of a number of CFX generation methods from the literature and show that they unanimously host significant deficiencies in this regard. Second, we demonstrate how embedding Δ-robustness within existing methods can provide CFXs which are provably robust.

1 Introduction

Ensuring that machine learning models are explainable has become a dominant goal in recent years, giving rise to the field of *explainable AI* (XAI). One of the most popular strategies for XAI is the use of *counterfactual explanations* (CFXs) (see (Stepin et al. 2021) for an overview), favoured for a number of reasons including their intelligibility (Byrne 2019), appeal to users (Barocas, Selbst, and Raghavan 2020), information capacity (Kenny and Keane 2021) and alignment with human reasoning (Miller 2019). A CFX for a given input to a model is defined as an altered input for which the model gives a different output to that of the original input. Consider the classic illustration of a loan application, with features *unemployed* status, *25* years of age and *low* credit rating, being classified by a bank's model as rejected. A CFX for the rejection could be an altered input where a *medium* credit rating (with the other features unchanged) would result in the loan being accepted, thus giving the applicant an idea of what is required to change the output. Such correctness of the modified output in attaining an alternative value is the basic property of CFXs, referred to as *validity*, and is one of a whole host of metrics around which CFXs are designed (e.g., see (Guidotti 2022)).

Our main focus in this paper is the metric of *robustness*. This is most often defined as *robustness to input perturbations*, i.e., the validity of CFXs when perturbations are applied to inputs (Sharma, Henderson, and Ghosh 2020). While this notion is useful, e.g., for protecting against manipulation (Slack et al. 2021), other forms of robustness can be equally important in ensuring that CFXs are safe and can be trusted. *Robustness to model changes*, i.e., the validity of CFXs when model parameters are altered, has thus far received little attention but is arguably one of the most commonly required forms of robustness, given that model parameters change every time retraining occurs (Rawal, Kamar, and Lakkaraju 2020). Indeed, if a CFX is invalidated with just a slight change of the training settings as in, e.g., (Dutta et al. 2022), we may question its quality in terms of real-world meaning. Consider the loan example: if, after retraining, the loan applicant changing their credit rating to *medium* no longer changes the output to accepted (thus invalidating the CFX), the CFX was not robust to the model changes induced during retraining. In this case, it might be argued that the bank should have a policy to guarantee that this CFX remains valid regardless, but this may have unfavourable consequences for the bank. Therefore, it is desirable that the CFXs account for such robustness.

Though some have targeted robustness to model changes[1], e.g., (Upadhyay, Joshi, and Lakkaraju 2021; Dutta et al. 2022), these approaches are heuristic, and may fail to provide strong robustness guarantees. Formal methods for assessing CFXs along this metric are lacking. Indeed, there are calls for both formal explanations for nonlinear models such as neural networks (Marques-Silva and Ignatiev 2022) and for standardised benchmarking in evaluating CFXs (Kenny and Keane 2021), voids we help to fill.

In this work we propose the novel notion of Δ-*robustness* for assessing the robustness of CFXs for neural networks in a formal, deterministic manner. We introduce an abstraction framework based on *interval neural networks* (Prabhakar

*These authors contributed equally.

[1]Referred to simply as *robustness*, unless otherwise specified.

and Afzal 2019) to verify the robustness of CFXs against a possibly infinite set of changes to the model parameters, i.e., weights and biases. This abstraction allows for a set of parameterisable shifts, Δ, in the model parameters, permitting users to tailor the strictness of robustness (depending on the application). For illustration, consider the loan example once more: the bank knows the scale of typical changes in their models and could encode this into Δ. The bank would then be able to provide only Δ-robust CFXs such that they are valid under any expected model shift during retraining (and if a model shift exceeds Δ, they would have been alerted to this fact). It can be seen, even from this simple example, that Δ-robustness can provide priceless guarantees in high-stakes or sensitive situations.

After covering related work (2) and the necessary preliminaries (3), we make the following contributions.

- We introduce a novel notion of Δ-robustness of CFXs for neural networks and propose an abstraction framework based on interval neural networks to reason about it (4).
- We analyse the Δ-robustness of a number of CFX approaches in the literature, demonstrating the utility of the notion and the lack of robustness in these methods (5.2).
- We demonstrate how the verification of Δ-robustness can be embedded in existing methods to generate CFXs which are provably robust (5.3).

We then conclude and look ahead to the various avenues of future work highlighted by our approach (6). In summary, this work presents the first approach to formally reason about and deterministically quantify CFXs' robustness to model changes in neural networks.[2]

2 Related Work

2.1 Approaches to CFX Generation

The seminal work of Wachter, Mittelstadt, and Russell (2017) casts the problem of finding CFXs for neural networks as gradient-based optimisation against the input vector using a single loss function to address the validity of counterfactual instances, as well as their closeness to the input instances measured by some distance metric (*proximity*), while that of Tolomei et al. (2017) defines CFXs for tree ensembles. Following these works, Mothilal, Sharma, and Tan (2020) include stochastic point processes and novel loss terms to generate a *diverse* set of CFXs. Poyiadzi et al. (2020) formulate the problem in graph-theoretic terms and apply shortest path algorithms to find CFXs that lie in the data manifold of the dataset. Van Looveren and Klaise (2021) address the same problem using class prototypes found by variational auto-encoders or k-d trees. Mohammadi et al. (2021) model the generation of CFXs as a constrained optimisation problem where a neural network is encoded using Mixed-Integer Linear Programming (MILP). Other methods that are able to generate CFXs for neural networks include that of Karimi et al. (2020), which reduces CFX generation to a satisfiability problem, and that of Dandl et al. (2020), which formulates the search for CFXs as a multi-objective optimisation problem. Orthogonal to these studies, ongoing works try to embed causal constraints when finding CFXs (Mahajan, Tan, and Sharma 2019; Karimi, Schölkopf, and Valera 2021; Kanamori et al. 2021). Finally, there are a number of methods for generating CFXs for linear or Bayesian models, e.g., (Ustun, Spangher, and Liu 2019; Albini et al. 2020; Kanamori et al. 2020), but we omit their details here since our focus is on neural networks.

2.2 Robustness of Models and Explanations

Robustness has been advocated in a number of ways in AI, including by requiring that outputs of neural networks should be robust to perturbations in inputs (Carlini and Wagner 2017; Weng et al. 2018) or in model parameters (Tsai et al. 2021). A number of works have drawn attention to the links between adversarial examples and CFXs, given that they solve a similar optimisation problem (Pawelczyk et al. 2022; Freiesleben 2022). The protection which robustness to input perturbations provides against manipulation has been shown to be important also as concerns explanations for models' outputs (Slack et al. 2021) and a range of methods for producing explanations which are robust to input perturbations have been proposed, e.g., (Alvarez-Melis and Jaakkola 2018; Sharma, Henderson, and Ghosh 2020; Huai et al. 2022). Meanwhile, Qiu et al. (2022) use input perturbations to ensure that explanations are robust to out-of-distribution data, applying this technique to a range of XAI methods for producing saliency maps. A causal view is taken by Hancox-Li (2020) in discussing the importance of robustness to input perturbations in explanations for models' outputs. Here, it is argued that explanations should be robust to different models, not only changes within the model (as we target), if real patterns in the world are of interest. Ensuring that CFXs fall on the data manifold has been found to increase this robustness to multiplicity of models (Pawelczyk et al. 2022). However, our focus is on formal approach to robustness when changing the model parameters, rather than the model itself. Notwithstanding the findings of recent works demonstrating the significant effects of changes to model parameters on the validity of CFXs (Rawal, Kamar, and Lakkaraju 2020; Dutta et al. 2022), we are aware of only two works which target the same form of robustness we consider. Upadhyay, Joshi, and Lakkaraju (2021) design a novel objective for CFXs which incorporates the model shift, i.e., the change in a model's parameters which may be, for example, weights or gradients. However, their approach is heuristic and may fail to generate valid robust CFXs (we will discuss other limitations of this approach later in 5.2). Dutta et al. (2022) define the metric of *counterfactual stability*, i.e., robustness to model changes induced during retraining, before introducing an approach which refines any base method for finding CFXs in tree-based classifiers, rather than the neural networks we target. In addition, both works evaluate CFXs' robustness by demonstrating CFXs' validity on a small number of retrained models and cannot exhaustively prove the validity for other model changes.

[2]The code for the implementations and experiments is publicly available at https://github.com/junqi-jiang/robust-ce-inn. The full version of the paper including proofs and experimental details can be found at https://arxiv.org/pdf/2208.14878.pdf.

3 Preliminaries

Notation. Given an integer k, let $[k]$ denote the set $\{1, \dots, k\}$. Given a set S, let $|S|$ denote its cardinality. Given a vector $x \in \mathbb{R}^n$ we use $x[i]$ to denote its i-th component; similarly, for a matrix $w \in \mathbb{R}^n \times \mathbb{R}^m$, we use $w[i, j]$ to denote element i, j. Finally, we use $\mathbb{I}(\mathbb{R})$ to denote the set of all closed intervals over $\mathbb{R}$.

Feed-forward neural networks. A feed-forward neural network (FFNN) is a directed acyclic graph whose nodes are structured in layers. Formally, we describe FFNNs and the computations they perform as follows.

Definition 1. *A* fully-connected feed-forward neural network (FFNN) *is a tuple $\mathcal{M} = (k, N, E, B, \Omega)$ where:*

- *$k \geq 0$ is the depth of $\mathcal{M}$;*
- *(N, E) is a directed graph;*
- *$N = \bigsqcup_{i=0}^{k+1} N_i$ is the disjoint union of sets of nodes N_i; we call N_0 the input layer, N_{k+1} the output layer and N_i hidden layers for $i \in [k]$;*
- *$E = \bigcup_{i=1}^{k+1} (N_{i-1} \times N_i)$ is the set of edges between layers;*
- *$B:(N \setminus N_0) \to \mathbb{R}$ assigns bias to nodes in non-input layers;*
- *$\Omega: E \to \mathbb{R}$ assigns a weight to each edge.*

In the following, unless specified otherwise, we assume as given an FFNN $\mathcal{M} = (k, N, E, B, \Omega)$, and we use B_i to denote the vector of biases assigned to layer N_i and W_i to denote the matrix of weights assigned to edges between nodes in subsequent layers N_{i-1}, N_i, for $i \in [k+1]$.

Definition 2. *Given an input $x \in \mathbb{R}^{|N_0|}$, an FFNN $\mathcal{M}$ computes an* output *$\mathcal{M}(x)$ defined as follows. Let:*

- *$V_0 = x$;*
- *$V_i = \sigma(W_i \cdot V_{i-1} + B_i)$ for $i \in [k]$, where σ is an activation function applied element-wise. For $V_i = [v_{i,1}, \dots, v_{i,|N_i|}]$, $v_{i,j}$ is the* value *of the j-th node in layer N_i.*

Then, $\mathcal{M}(x) = V_{k+1} = W_{k+1} \cdot V_k + B_{k+1}$.

The *Rectified Linear Unit (ReLU)* activation, defined as $\sigma(x) \triangleq \max(0, x)$, is perhaps the most common choice for hidden layers. We will therefore focus on FFNNs using ReLU activations in this paper.

Definition 3. *Consider an input $x \in \mathbb{R}^{|N_0|}$ and an FFNN $\mathcal{M}$. We say that $\mathcal{M}$* classifies *x* as *c, denoted (with an abuse of notation) $\mathcal{M}(x) = c$, if $c \in \arg\max_{i \in [|N_{k+1}|]} \mathcal{M}(x)[i]$.*

For ease of exposition, 4 will focus on FFNNs used for binary classification tasks with $|N_{k+1}| = 2$. The same ideas also apply to other settings, e.g., multiclass classification or binary classification using a single output node with sigmoid activation, which we use in our experiments in 5.

Counterfactual explanations. Consider an FFNN $\mathcal{M}$ trained to solve a binary classification problem. Assume $\mathcal{M}$ produces a classification outcome $\mathcal{M}(x) = c$ for input x. Intuitively, a CFX is a new input x' which is similar to x and for which $\mathcal{M}(x') = 1 - c$. Formally, existing literature characterises CFXs in terms of the solution space of a Constrained Optimisation Problem (COP) as follows.

Definition 4. *Consider an input $x \in \mathbb{R}^{|N_0|}$ and a binary classifier $\mathcal{M}$ s.t. $\mathcal{M}(x) = c$. Given a distance metric $d : \mathbb{R}^{|N_0|} \times \mathbb{R}^{|N_0|} \to \mathbb{R}$, a* CFX *is any x' such that:*

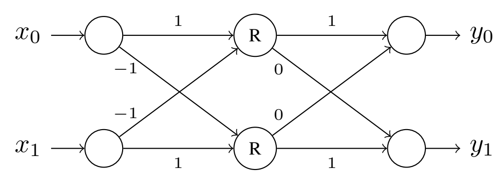
$$\arg\min_{x'} \; d(x, x') \tag{1a}$$

$$\textit{subject to} \quad \mathcal{M}(x') = 1 - c, \; x' \in \mathbb{R}^{|N_0|} \tag{1b}$$

A CFX thus corresponds to the closest input x' (Eq. 1a) belonging to the original input space that makes the classification flip (Eq. 1b). A common choice for the distance metric d is the normalised L_1 distance (Wachter, Mittelstadt, and Russell 2017). Under this choice, CFX generation for FFNNs with ReLU activations can be solved exactly via MILP – see, e.g., (Mohammadi et al. 2021). Finally, we mention that the optimisation problem can also be extended to account for additional CFX properties mentioned in 2.1.

We conclude with an example which summarises the main concepts presented in this section.

Example 1. *Consider the FFNN $\mathcal{M}$ below where weights are as indicated in the diagram, biases are zero and R denotes ReLU activations. The network receives a two-dimensional input $x = [x_0, x_1]$ and produces a two-dimensional output $y = [y_0, y_1]$.*



The symbolic expressions for the output components are $y_0 = \max(0, x_0 - x_1)$ and $y_1 = \max(0, x_1 - x_0)$. Given a concrete input $x = [1, 2]$, we have $\mathcal{M}(x) = 1$. A possible CFX may be $x' = [2.1, 2]$, with $\mathcal{M}(x') = 0$.

4 Δ-Robustness via Interval Abstraction

The COP formulation of CFXs presented in Definition 4 focuses on finding CFXs that are as close as possible to the original input. The rationale behind this choice is that changes in input features suggested by minimally distant CFXs likely require less effort, thus making them more easily attainable by users in real-world settings. However, it has been shown (Rawal, Kamar, and Lakkaraju 2020; Dutta et al. 2022) that slight changes applied to the classifier, e.g., following retraining, may impact the validity of CFXs, particularly those which are minimally distant from the original input. This fragility of CFXs can have troubling implications, both for the users of explanations, and for those who generate them, as discussed in 1.

This state of affairs motivates the primary objective of this work: *can we generate useful CFXs for FFNNs that are provably robust to model changes?*

In the following we formalise the notion of robustness we target and introduce an abstraction-based framework to reason about this notion in CFXs for FFNNs. To this end, we begin by defining a notion of distance between FFNNs.

Definition 5. *Consider two FFNNs $\mathcal{M} = (k, N, E, B, \Omega)$ and $\mathcal{M}' = (k', N', E', B', \Omega')$. We say that $\mathcal{M}$ and $\mathcal{M}'$* have identical topology *if $k = k'$ and $(N, E) = (N', E')$.*

Definition 6. *Let $\mathcal{M} = (k, N, E, B, \Omega)$ and $\mathcal{M}' = (k, N, E, B', \Omega')$ be two FFNNs with identical topology. For $0 \leq p \leq \infty$, the* p-distance between $\mathcal{M}$ and $\mathcal{M}'$ *is:*

$$\|\mathcal{M} - \mathcal{M}'\|_p = \left(\sum_{i=1}^{k+1} \sum_{j=1}^{|N_i|} \sum_{l=1}^{|N_{i-1}|} |W_i[j,l] - W_i'[j,l]|^p \right)^{\frac{1}{p}}$$

Intuitively p-distance compares the weight matrices of $\mathcal{M}$ and $\mathcal{M}'$ and computes their distance as the p-norm of their difference. Biases have been omitted from Definition 6 for readability; the definition can be readily extended to include biases too, as is the case in our implementation. Using this notion we can characterise a model shift as follows.

Definition 7. *Given $0 \leq p \leq \infty$, a* model shift *is a function S mapping an FFNN $\mathcal{M}$ into another $\mathcal{M}' = S(\mathcal{M})$ such that:*

- *$\mathcal{M}$ and $\mathcal{M}'$ have identical topology;*
- *$\|\mathcal{M} - \mathcal{M}'\|_p > 0$.*

Model shifts are typically observed in real-world applications when a model is regularly retrained to incorporate new data. In such cases, models are likely to see only small changes at each update. In the same spirit as (Upadhyay, Joshi, and Lakkaraju 2021), we capture this as follows.

Definition 8. *Given an FFNN $\mathcal{M}$, $\delta \in \mathbb{R}_{>0}$ and $0 \leq p \leq \infty$,* the set of plausible model shifts *is $\Delta = \{S \mid \|\mathcal{M} - S(\mathcal{M})\|_p \leq \delta\}$.*

Plausibility implicitly bounds the magnitude of weight and bias changes that can be effected by a model shift S, as stated in the following.

Lemma 1. *Consider an FFNN $\mathcal{M}$ and a set of plausible model shifts Δ. Let $\mathcal{M}' = S(\mathcal{M})$ for $S \in \Delta$. The magnitude of weight and bias changes in $\mathcal{M}'$ is bounded.*

In essence, it is possible to show that each weight (and bias) can change up to a maximum of $\pm\delta$ following the application of a model shift $S \in \Delta$.

Remark 1. *In the following we use $\underline{W_i'[j,l]} \triangleq W_i[j,l] - \delta$ and $\overline{W_i'[j,l]} \triangleq W_i[j,l] + \delta$ to denote, respectively, the minimum and maximum value each weight $W_i'[j,l]$ can take in $\mathcal{M}' = S(\mathcal{M})$ for any $S \in \Delta$ and $i \in [k+1]$, $j \in [|N_i|]$ and $l \in [|N_{i-1}|]$. (Analogous notation is used for biases.) These bounds are sound, but also conservative, i.e., they may result in models that exceed the upper bound on the p-distance for some choice of p. As an example, consider what happens when $W_i'[j,l] = \overline{W_i'[j,l]}$ for each $i \in [k+1]$, $j \in |N_i|$, $l \in |N_{i-1}|$. These valuations satisfy Definition 8 when $p = \infty$, but fail to do so for, e.g., $p = 2$.*

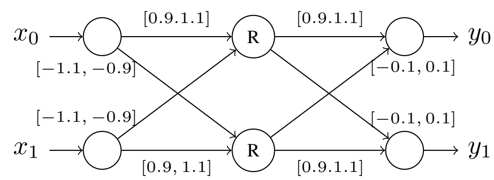
Despite weight changes being bounded, several different model shifts may satisfy the plausibility constraint. To guarantee robustness to model changes, one needs a way to represent and reason about the potentially infinite family of networks originated by applying each $S \in \Delta$ to $\mathcal{M}$ compactly.

We introduce an abstraction framework that can be used to this end. We begin by recalling the notion of *interval neural networks*, as introduced in (Prabhakar and Afzal 2019).

Definition 9. *An* interval neural network (INN) *is a tuple $\mathcal{I} = (k, N, E, B_{\mathcal{I}}, \Omega_{\mathcal{I}})$ where:*

- *k, N, E are as per Definition 1;*
- *$B_{\mathcal{I}} : (N \setminus N_0) \to \mathbb{I}(\mathbb{R})$ assigns interval-valued biases to nodes in non-input layers;*
- *$\Omega_{\mathcal{I}} : E \to \mathbb{I}(\mathbb{R})$ assigns interval-valued weights to edges.*

Example 2. *The diagram below shows an example of an INN. As we can observe, the INN differs from a standard FFNN in that weights and biases are intervals.*



In the remainder, unless specified otherwise, when using an INN we will assume its components are as in Definition 9. We will use boldface notation to denote interval-valued biases $\mathbf{B}_i$ and weights $\mathbf{W}_i$. The computation performed by an INN differs from that of an FFNN as follows.

Definition 10. *Given an input $x \in \mathbb{R}^{|N_0|}$, an INN $\mathcal{I}$ computes an* output *$\mathcal{I}(x)$ defined as follows. Let:*

- *$\mathbf{V}_0 = [x, x]$;*
- *$\mathbf{V}_i = \sigma(\mathbf{W}_i \cdot \mathbf{V}_{i-1} + \mathbf{B}_i)$ for $i \in [k]$. For $\mathbf{V}_i = [\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,|N_i|}]$, $\mathbf{v}_{i,j} = [v_{i,j}^l, v_{i,j}^u]$ is the* interval of values *for the j-th node in layer N_i.*

Then, $\mathcal{I}(x) = \mathbf{V}_{k+1} = \mathbf{W}_{k+1} \cdot \mathbf{V}_k + \mathbf{B}_{k+1}$.

Thus, an INN computes an interval for each output node. These intervals contain all possible values that each output can take under the valuations induced by $B_{\mathcal{I}}$ and $\Omega_{\mathcal{I}}$. As a result, the classification semantics of an INN is as follows.

Definition 11. *Consider an input $x \in \mathbb{R}^{|N_0|}$, a binary label c and an INN $\mathcal{I}$. We say that $\mathcal{I}$* classifies *x* as *c, written $\mathcal{I}(x) = c$, if $v_{k+1,c}^l > v_{k+1,1-c}^u$.*

Figure 1 provides a graphical representation of this classification semantics. Using the INN as a computational backbone, we can now define the interval abstraction of an FFNN which is central to this work.

Definition 12. *Consider an FFNN $\mathcal{M}$ and a set of plausible model shifts Δ. We define the* interval abstraction *of $\mathcal{M}$ under Δ as the interval neural network $\mathcal{I}_{(\mathcal{M},\Delta)}$ such that $\mathcal{M}$ and $\mathcal{I}_{(\mathcal{M},\Delta)}$ have identical topology and the interval-valued biases/weights of $\mathcal{I}_{(\mathcal{M},\Delta)}$ are:*

- *$\mathbf{B}_i[j] = [\underline{B_i'[j]}, \overline{B_i'[j]}]$ for $i \in [k+1]$ and $j \in [|N_i|]$;*
- *$\mathbf{W}_i[j,l] = [\underline{W_i'[j,l]}, \overline{W_i'[j,l]}]$ for $i \in [k+1]$, $j \in [|N_i|]$ and $l \in [|N_{i-1}|]$.*

Lemma 2. *$\mathcal{I}_{(\mathcal{M},\Delta)}$ over-approximates the set of models $\mathcal{M}'$ that can be obtained from $\mathcal{M}$ via Δ.*

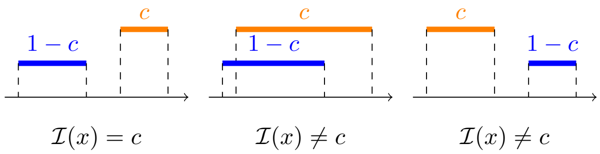


Figure 1: Graphical comparison of output intervals for class c and class $1-c$, for Definition 11. When $\mathcal{I}(x) = c$, the output range for class c is always greater than that of class $1-c$. Otherwise, we say $\mathcal{I}(x) \neq c$.

Lemma 2 states that $\mathcal{I}_{(\mathcal{M},\Delta)}$ contains all models that can be obtained from Δ and possibly more, but not fewer. For some values of p, the interval abstraction may cease to be an over-approximation and encode exactly the models that can be obtained from $\mathcal{M}$ via Δ, e.g., when $p = \infty$.

A model shift S, although plausible, may result in changes to the classification of the original input x. When this happens, robustness of explanations becomes vacuous. For this reason, we will focus on *sound shifts*, formulated as follows.

Definition 13. *Consider an input $x \in \mathbb{R}^{|N_0|}$ and an FFNN $\mathcal{M}$ s.t. $\mathcal{M}(x) = c$. Let $\mathcal{I}_{(\mathcal{M},\Delta)}$ be the interval abstraction of $\mathcal{M}$ under a set of plausible model shifts Δ. We say that Δ* is sound *if $\mathcal{I}_{(\mathcal{M},\Delta)}(x) = c$.*

We are now ready to formally define the CFX robustness property that we target in this work.

Definition 14. *Consider an input $x \in \mathbb{R}^{|N_0|}$ and an FFNN $\mathcal{M}$ s.t. $\mathcal{M}(x) = c$. Let $\mathcal{I}_{(\mathcal{M},\Delta)}$ be the interval abstraction of $\mathcal{M}$ under a sound set of plausible model shifts Δ. We say that* a CFX x' is Δ-robust *iff $\mathcal{I}_{(\mathcal{M},\Delta)}(x') = 1 - c$.*

We illustrate these concepts in the following example.

Example 3. *We observe that the INN in Example 2 corresponds to the interval abstraction $\mathcal{I}_{(\mathcal{M},\Delta)}$ of the FFNN $\mathcal{M}$ in Example 1, obtained for $\Delta = \{S \mid \|\mathcal{M} - S(\mathcal{M})\|_\infty \leq 0.1\}$.*

The symbolic expressions for the outputs of the INN are:

$$y_0 = [0.9, 1.1] \cdot \max(0, [0.9, 1.1] \cdot x_0 + [-1.1, -0.9] \cdot x_1) + [-0.1, 0.1] \cdot \max(0, [-1.1, -0.9] \cdot x_0 + [0.9, 1.1] \cdot x_1)$$

$$y_1 = [0.9, 1.1] \cdot \max(0, [0.9, 1.1] \cdot x_1 + [-1.1, -0.9] \cdot x_0) + [-0.1, 0.1] \cdot \max(0, [-1.1, -0.9] \cdot x_1 + [0.9, 1.1] \cdot x_0)$$

Given a concrete input $x = [1, 2]$, we observe that $\mathcal{I}_{(\mathcal{M},\Delta)}(x) = 1$ and thus establish that Δ is sound. We now check if the old CFX $x' = [2.1, 2]$ is still valid under the model shifts captured by Δ. The INN outputs $y_0 = [-0.031, 0.592]$ and $y_1 = [-0.051, 0.392]$, indicating that $\mathcal{I}_{(\mathcal{M},\Delta)}(x') \neq 0$. We thus conclude that x' is not Δ-robust.

Assume now a different CFX $x'' = [2.6, 2]$ is computed. The outputs of $\mathcal{I}_{(\mathcal{M},\Delta)}$ for x'' are $y_0 = [0.126, 1.166]$ and $y_1 = [-0.106, 0.106]$. Since $y_0^l > y_1^u$, we have $\mathcal{I}_{(\mathcal{M},\Delta)}(x'') = 0$, proving that the new CFX is Δ-robust.

As shown in Example 3, the interval abstraction $\mathcal{I}_{(\mathcal{M},\Delta)}$ can be used to prove whether a given CFX x' is Δ-robust. Indeed, when $\mathcal{I}_{(\mathcal{M},\Delta)}(x) = c$, we can conclude that the classification of x' will remain unchanged for all S in Δ. Checking Definition 11 requires the computation of the output reachable intervals for each output of the INN; for ReLU-based FFNNs, we use the MILP formulation of (Prabhakar and Afzal 2019).

5 Δ-Robustness in Action

In 4 we laid the theoretical foundations of an abstraction framework based on INNs that allows to reason about the robustness of CFXs compactly. In this section we demonstrate the utility thereof by considering two distinct applications:

- in 5.2, we show how the interval abstraction can be used to **analyse** the Δ-robustness of different CFX algorithms across model shifts of increasing magnitudes;
- in 5.3, we propose an algorithm that uses interval abstractions to **generate** provably robust CFXs.

Our experiments, conducted on both homogeneous (continuous features) and heterogeneous (mixed continuous and discrete features) data types (see 5.1), show that our approach provides a measure for assessing the robustness of CFXs generated by other SOTA methods, but it can also be used to devise algorithms for generating CFXs with *provable* robustness guarantees, in contrast with existing methods.

5.1 Experimental Setup

We consider four datasets with a mixture of heterogeneous and continuous data. We refer to them as *credit* (heterogeneous) (Dua and Graff 2017), *small business administration (SBA)* (using only their continuous features) (Li, Mickel, and Taylor 2018), *diabetes* (continuous) (Smith et al. 1988) and *no2* (continuous) (Vanschoren et al. 2013).

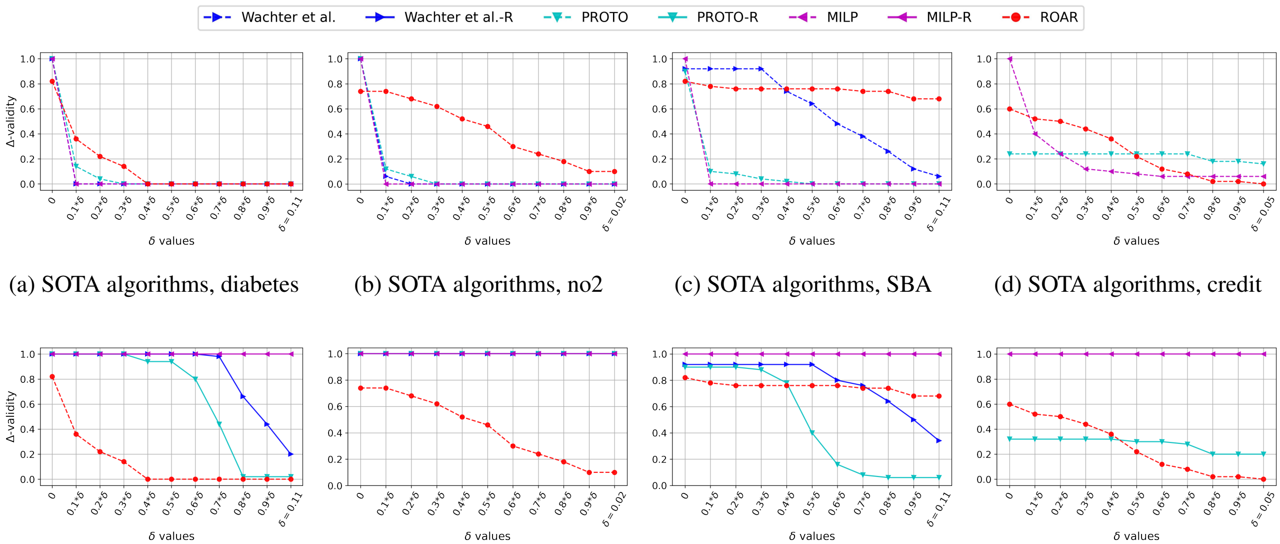
The first two datasets contain known distribution shifts (Upadhyay, Joshi, and Lakkaraju 2021). We use $\mathcal{D}_1$ ($\mathcal{D}_2$) to denote the dataset before (respectively after) the shift. For the other datasets, we randomly shuffle the instances and separate them into two halves, again denoted as $\mathcal{D}_1$ and $\mathcal{D}_2$. For each dataset, we use $\mathcal{D}_1$ to train a base model, and use instances in $\mathcal{D}_2$ to generate model shifts via incremental retraining. We use $p = \infty$ in all experiments that follow.

CFXs are generated using the following SOTA algorithms. We consider *Wachter et al.* (Wachter, Mittelstadt, and Russell 2017) (continuous data only), *Proto* (Van Looveren and Klaise 2021) and a method inspired by (Mohammadi et al. 2021). The first two implement CFX search via gradient descent, while the third uses MILP, and is thus referred to here as *MILP*. We also include *ROAR* (Upadhyay, Joshi, and Lakkaraju 2021), a SOTA framework specifically designed to generate robust CFXs.

5.2 Analysing Δ-Robustness of CFXs

This experiment is designed to show that interval abstractions can provide an effective tool to analyse CFXs generated by SOTA algorithms. For each dataset, we identify the largest δ_{max} that results in a set Δ that is sound for at least 50 test instances in $\mathcal{D}_1$. This is achieved by retraining the base model using increasingly large portions of $\mathcal{D}_2$.[3]

[3]In real-world applications, values of δ could be empirically estimated by model developers by observing retraining histories and calculating the p-distances between subsequent retraining steps.



(a) SOTA algorithms, diabetes (b) SOTA algorithms, no2 (c) SOTA algorithms, SBA (d) SOTA algorithms, credit

(e) robust algorithms, diabetes (f) robust algorithms, no2 (g) robust algorithms, SBA (h) robust algorithms, credit

Figure 2: Evaluation of Δ-validity. (a-d, see 5.2): SOTA algorithms fail to generate completely robust CFXs as δ increases. (e-h, see 5.3): Embedding Δ-robustness in the search process of the same algorithms results in more provably robust CFXs.

We then use the CFX generation algorithms to produce 50 CFXs. We evaluate their robustness for model shifts of magnitude up to δ_{max} using **Δ-validity**, the percentage of test instances whose CFXs are Δ-robust.

Figures 2(a-d) report the results of our analysis for the four datasets. As we can observe, all methods generate CFXs that tend to be valid counterfactuals for the original model ($\delta = 0$), with ROAR having lower results in most cases. This is because ROAR approximates the local behaviours of FFNNs using LIME (Ribeiro, Singh, and Guestrin 2016), which may cause a slight decrease in the counterfactual validity (Upadhyay, Joshi, and Lakkaraju 2021). However, the picture changes as soon as small model shifts are applied. The Δ-validity values of Wachter et al., Proto and MILP quickly drop to zero even for model shifts of magnitude equal to 10% of δ_{max}, revealing that these algorithms are prone to generating non-robust CFXs when even very small shifts are seen in the model parameters. ROAR exhibits a higher degree of Δ-robustness, as expected. However, its heuristic nature does not allow to reason about all possible shifts in Δ, which clearly affects the Δ-robustness of CFXs as δ grows larger.

All methods considered here (Wachter et al, Proto, MILP and ROAR) return a single CFX for each input. However, Δ-robustness can also be used with methods generating multiple CFXs, e.g., as with the DiCE method of (Mothilal, Sharma, and Tan 2020). In these latter cases, Δ-robustness can be deployed as a filter, with customisable coarseness achieved by varying Δ, to obtain sets of *diverse* and Δ-robust CFXs. When doing so, experiments show a similar decrease of Δ-validity as in Figure 2(a-d). We leave further exploration of this application of Δ-robustness to future work.

5.3 Generating Provably Robust CFXs

Our earlier experiments reveal that SOTA algorithms, including those that are designed to be robust, often fail to generate CFXs that satisfy Δ-robustness. Thus, the problem of generating CFXs that are provably robust against model shifts remains largely unsolved. We will now show how Δ-robustness can be used to guide CFX generation algorithms toward CFXs with formal robustness guarantees. Our proposed approach, shown in Algorithm 1, can be applied on top of any CFX generation algorithm and proceeds as follows. First, an interval abstraction is constructed for the FFNN $\mathcal{M}$ and set Δ; the latter is then checked for soundness (Definition 13). Then, the search for a CFX starts. At each iteration, a CFX is generated using the base method and is tested for Δ-robustness using the interval abstraction (Definition 14). If the CFX is robust, then the algorithm terminates and returns the solution. Otherwise, the search continues, allowing for CFXs of increasing distance to be found. These steps are repeated until a threshold number of iterations t is reached. As a result, the algorithm is *incomplete*, in that it may report that no Δ-robust CFX can be found within t steps (while one may exist for larger t).

We instantiated Algorithm 1 on the non-robust base methods, i.e., Wachter et al, Proto and MILP. We use *Wachter et al-R*, *Proto-R* and *MILP-R*, respectively, to denote the resulting algorithms. For each dataset, we use the same δ_{max} identified in 5.2 to create sound sets of model shifts Δ. The iterative procedure of Algorithm 1 generates CFXs of increasing distance until the target robustness Δ is satisfied. To increase the distance of CFXs for Wachter et al and Proto we iteratively increase the influence of the loss term pertaining to CFX validity. For MILP, instead, we require that the probability of the output produced by the classifier to subse-

	diabetes, target $\delta = 0.11$				no2, target $\delta = 0.02$				SBA, target $\delta = 0.11$				credit, target $\delta = 0.05$			
	vm1	**vm2**	ℓ_1	**lof**	**vm1**	**vm2**	ℓ_1	**lof**	**vm1**	**vm2**	ℓ_1	**lof**	**vm1**	**vm2**	ℓ_1	**lof**
Wachter et al.	100%	0%	0.051	0.96	100%	32%	0.035	1.00	92%	92%	0.018	-0.57	-	-	-	-
Wachter et al.-R	100%	100%	0.122	1.00	100%	100%	0.084	1.00	92%	92%	0.023	-0.78	-	-	-	-
Proto	100%	18%	0.063	1.00	100%	32%	0.036	1.00	90%	6%	0.008	0.60	24%	22%	0.313	-1.00
Proto-R	100%	96%	0.104	1.00	100%	100%	0.069	1.00	90%	88%	0.011	-0.02	32%	30%	0.300	-1.00
MILP	100%	0%	0.049	0.96	100%	32%	0.032	1.00	100%	4%	0.007	0.56	100%	74%	0.024	1.00
MILP-R	100%	100%	0.212	-0.48	100%	100%	0.059	1.00	100%	100%	0.018	-0.88	100%	100%	0.031	1.00
ROAR	82%	14%	0.078	0.95	88%	34%	0.074	1.00	82%	78%	0.031	-0.80	62%	60%	0.047	1.00

Table 1: Evaluating the robustness of CFXs for base methods and their Δ-robust variants.

Algorithm 1: Generation of robust CFXs

Require: FFNN $\mathcal{M}$, input x such that $\mathcal{M}(x) = c$,
set of plausible model shifts Δ and threshold t
Step 1: build interval abstraction $\mathcal{I}_{(\mathcal{M},\Delta)}$.
Step 2: check soundness of Δ
if Δ is sound **then**
  while iteration number $< t$ **do**
    Step 3: compute CFX x' for x and $\mathcal{M}$ using base method
    if $\mathcal{I}_{(\mathcal{M},\Delta)}(x') = 1 - c$ **then**
      return x'
    else
      Step 4: increase allowed distance of next CFX
      Step 5: increase iteration number
return no robust CFX can be found

quent CFXs increases at each iteration (all test instances are classified as class 0, and the desired class is class 1). In practice, the number of iterations will depend on the choice of δ and the magnitude of step changes of the hyper-parameters, which is specific to each base method (e.g., 25, 6, 35 on average for Wachter et al-R, Proto-R and MILP-R, respectively).

Figures 2(e-h) show the results obtained. Overall, we can observe that Algorithm 1 successfully increases the Δ-validity of CFXs generated by base methods (compared with Figures 2(a-d)). MILP-R appears to be the best performing algorithm, generating CFXs that always satisfy the given robustness target. The robustness of CFXs computed with Wachter et al and Proto also drastically improves across different datasets. In some cases our algorithm fails to produce robust CFXs for high values of δ, yet a considerable improvement in robustness can be observed overall (compare, e.g., Figures 2a and 2e). Interestingly, simply by altering the hyperparameters of the base methods not specifically designed for robustness purposes, they produced more Δ-robust results than ROAR.

We also evaluated the extent to which Δ-robustness to smaller model shifts can help mitigate the effect of more significant model shifts. To this end, for each base method, we generated Δ-robust CFXs for a model trained on $\mathcal{D}_1$. We then generated a new model retrained using both $\mathcal{D}_1$ and $\mathcal{D}_2$ and evaluated the validity of CFXs for the new model. We highlight that this retraining procedure may result in model shifts that are larger than the Δ targeted for the original model. As such, Δ-robustness may not be guaranteed on the new model. For each algorithm and dataset, we analyse the following metrics: **vm1**, the percentage of CFXs that are valid on the original model; **vm2**, the percentage of CFXs that remain valid after retraining; ℓ_1, the ℓ_1 distance from the input; **lof**, the local outlier factor ($+1$ for inliers, -1 otherwise), used to test if an instance is within the data manifold. We average ℓ_1 and **lof** over the generated CFXs.

Table 1 reports the results obtained for this second set of experiments. We observe that enforcing Δ-robustness, even for small δ values, can considerably improve the validity of CFXs in the presence of larger model shifts. Indeed, Algorithm 1 increases the number of CFXs that remain valid after retraining by 68-100%. This improvement comes at the expense of ℓ_1 distance, which often increases. This phenomenon has already been observed in recent work (Dutta et al. 2022), where robust CFXs for tree classifiers were up to seven times more expensive than the non-robust baselines. The **lof** score tends to remain unchanged in many cases. However, for some combinations of base methods and datasets, the score drops considerably, suggesting that a better strategy to generate CFXs of increased distance may exist. Finally, we can observe that our approach often outperforms ROAR, producing CFXs that retain a higher degree of validity after retraining.

6 Conclusions

Despite the great deal of attention which CFXs in XAI have received of late, SOTA approaches fall short of providing formal robustness guarantees on the explanations they generate, as we have demonstrated. In this paper we proposed Δ-robustness, a formal notion for assessing the robustness of CFXs with respect to changes in the underlying model. We then introduced an abstraction-based framework to reason about Δ-robustness and used it to verify the robustness of CFXs and to guide existing methods to find CFXs with robustness guarantees.

This paper opens several avenues for future work. Firstly, while our experiments only considered FFNNs with ReLU activations, there seems to be no reason why interval-based analysis for robustness of CFXs could not be applied to a wider range of AI models. Secondly, it would be interesting to investigate probabilistic extensions of this work, so as to accommodate scenarios where robustness cannot be always guaranteed. Finally, our algorithm for generating Δ-robust CFXs is incomplete; we plan to investigate whether our abstraction framework can be used to devise complete algorithms with improved guarantees.

Acknowledgements

Rago and Toni were partially funded by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 101020934). Jiang, Rago and Toni were partially funded by J.P. Morgan and by the Royal Academy of Engineering under the Research Chairs and Senior Research Fellowships scheme. Leofante acknowledges financial support from Imperial College London through an Imperial College Research Fellowship grant. Any views or opinions expressed herein are solely those of the authors listed.

References

Albini, E.; Rago, A.; Baroni, P.; and Toni, F. 2020. Relation-Based Counterfactual Explanations for Bayesian Network Classifiers. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 451–457.

Alvarez-Melis, D.; and Jaakkola, T. S. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 7786–7795.

Barocas, S.; Selbst, A. D.; and Raghavan, M. 2020. The hidden assumptions behind counterfactual explanations and principal reasons. In *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, 80–89.

Byrne, R. M. J. 2019. Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 6276–6282.

Carlini, N.; and Wagner, D. A. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, 39–57.

Dandl, S.; Molnar, C.; Binder, M.; and Bischl, B. 2020. Multi-Objective Counterfactual Explanations. In *Parallel Problem Solving from Nature - PPSN XVI - 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part I*, 448–469.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml. Accessed: 2022-08-30.

Dutta, S.; Long, J.; Mishra, S.; Tilli, C.; and Magazzeni, D. 2022. Robust Counterfactual Explanations for Tree-Based Ensembles. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, 5742–5756.

Freiesleben, T. 2022. The Intriguing Relation Between Counterfactual Explanations and Adversarial Examples. *Minds Mach.*, 32(1): 77–109.

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

Hancox-Li, L. 2020. Robustness in machine learning explanations: does it matter? In *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, 640–647.

Huai, M.; Liu, J.; Miao, C.; Yao, L.; and Zhang, A. 2022. Towards Automating Model Explanations with Certified Robustness Guarantees. 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*, 6935–6943.

Kanamori, K.; Takagi, T.; Kobayashi, K.; and Arimura, H. 2020. DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 2855–2862.

Kanamori, K.; Takagi, T.; Kobayashi, K.; Ike, Y.; Uemura, K.; and Arimura, H. 2021. Ordered Counterfactual Explanation by Mixed-Integer Linear Optimization. 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*, 11564–11574.

Karimi, A.; Barthe, G.; Balle, B.; and Valera, I. 2020. Model-Agnostic Counterfactual Explanations for Consequential Decisions. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, 895–905.

Karimi, A.; Schölkopf, B.; and Valera, I. 2021. Algorithmic Recourse: from Counterfactual Explanations to Interventions. In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, 353–362.

Kenny, E. M.; and Keane, M. T. 2021. On Generating Plausible Counterfactual and Semi-Factual Explanations for Deep Learning. 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*, 11575–11585.

Li, M.; Mickel, A.; and Taylor, S. 2018. Should This Loan be Approved or Denied?: A Large Dataset with Class Assignment Guidelines. *Journal of Statistics Education*, 26(1): 55–66.

Mahajan, D.; Tan, C.; and Sharma, A. 2019. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277*.

Marques-Silva, J.; and Ignatiev, A. 2022. Delivering Trustworthy AI through Formal XAI. 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*, 12342–12350.

Miller, T. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267: 1–38.

Mohammadi, K.; Karimi, A.; Barthe, G.; and Valera, I. 2021. Scaling Guarantees for Nearest Counterfactual Explanations. In *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*, 177–187. ACM.

Mothilal, R. K.; Sharma, A.; and Tan, C. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, 607–617.

Pawelczyk, M.; Agarwal, C.; Joshi, S.; Upadhyay, S.; and Lakkaraju, H. 2022. Exploring Counterfactual Explanations Through the Lens of Adversarial Examples: A Theoretical and Empirical Analysis. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, 4574–4594.

Poyiadzi, R.; Sokol, K.; Santos-Rodríguez, R.; Bie, T. D.; and Flach, P. A. 2020. FACE: Feasible and Actionable Counterfactual Explanations. In *AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, February 7-8, 2020*, 344–350.

Prabhakar, P.; and Afzal, Z. R. 2019. Abstraction based Output Range Analysis for Neural Networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 15762–15772.

Qiu, L.; Yang, Y.; Cao, C. C.; Zheng, Y.; Ngai, H. H. T.; Hsiao, J. H.; and Chen, L. 2022. Generating Perturbation-based Explanations with Robustness to Out-of-Distribution Data. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, 3594–3605.

Rawal, K.; Kamar, E.; and Lakkaraju, H. 2020. Can I Still Trust You?: Understanding the Impact of Distribution Shifts on Algorithmic Recourses. *CoRR*, abs/2012.11788.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Sharma, S.; Henderson, J.; and Ghosh, J. 2020. CERTIFAI: A Common Framework to Provide Explanations and Analyse the Fairness and Robustness of Black-box Models. In *AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, February 7-8, 2020*, 166–172.

Slack, D.; Hilgard, A.; Lakkaraju, H.; and Singh, S. 2021. Counterfactual Explanations Can Be Manipulated. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 62–75.

Smith, J. W.; Everhart, J. E.; Dickson, W.; Knowler, W. C.; and Johannes, R. S. 1988. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the annual symposium on computer application in medical care*, 261. American Medical Informatics Association.

Stepin, I.; Alonso, J. M.; Catalá, A.; and Pereira-Fariña, M. 2021. A Survey of Contrastive and Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence. *IEEE Access*, 9: 11974–12001.

Tolomei, G.; Silvestri, F.; Haines, A.; and Lalmas, M. 2017. Interpretable Predictions of Tree-based Ensembles via Actionable Feature Tweaking. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, 465–474.

Tsai, Y.; Hsu, C.; Yu, C.; and Chen, P. 2021. Formalizing Generalization and Adversarial Robustness of Neural Networks to Weight Perturbations. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 19692–19704.

Upadhyay, S.; Joshi, S.; and Lakkaraju, H. 2021. Towards Robust and Reliable Algorithmic Recourse. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 16926–16937.

Ustun, B.; Spangher, A.; and Liu, Y. 2019. Actionable Recourse in Linear Classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*, 10–19.

Van Looveren, A.; and Klaise, J. 2021. Interpretable Counterfactual Explanations Guided by Prototypes. In *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part II*, 650–665.

Vanschoren, J.; van Rijn, J. N.; Bischl, B.; and Torgo, L. 2013. OpenML: networked science in machine learning. *SIGKDD Explor.*, 15(2): 49–60.

Wachter, S.; Mittelstadt, B. D.; and Russell, C. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *CoRR*, abs/1711.00399.

Weng, T.; Zhang, H.; Chen, P.; Yi, J.; Su, D.; Gao, Y.; Hsieh, C.; and Daniel, L. 2018. Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.