HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

BACHELOR THESIS

Herr.
Amadeo Tunyi Tiembukong

Counterfactual Explanations vs Adversarial Examples: An Investigation on their Differences

Mittweida, 2023

Fakultät Angewandte Computer- und Biowissenschaften

BACHELOR THESIS

Counterfactual Explanations vs Adversarial Examples: An Investigation on their Differences

Autor:
Herr Amadeo Tunyi Tiembukong

Studiengang:
Bsc. Applied Mathematics

Seminargruppe:
MA20w1-B

Erstprüfer:
Prof. Dr. Thomas Villmann

Zweitprüfer:
Dr. David Nebel

Mittweida, 2023

Contents

List of Figures

LIST OF ABBREVIATIONS

ML Machine Learning

DL Deep Learning

AE Adversarial Example

GDPR General Data Protection Regulation

AI Artificial Intelligence

XAI Explainable Artificial Intelligence

SCFE Score Counterfactual Explanations

CFE Counterfactual Explanations

NN Neural Networks

CNN Convolutional Neural Networks

VAE Variational Auto-Encoder

kNN k Nearest Neighbours

MLP Multilayer Perceptron

SVM Support Vector Machine

LVQ Learning Vector Quantization

PGD PGDProjected Gradient Descent

GAN Generative Adversarial Networks

FGSM Fast Gradient Sign Method

NAE Natural Adversarial Examples

C-CHVAE Counterfactual Conditional Heterogeneous Autoencoder

ACVE Adversarial Counterfactual Visual Explanation

Abstract

As a matter of convenience and for better understanding, we speak of decision boundaries when we refer to classification as those supposed lines or hyperplanes of division separating elements into different classes. So, of course , it comes to interest to research how easily fragile these decision boundaries can be and how easily a point could cross these boundaries (through minimum perturbation).

XAI's Counterfactual Explanation observes crossing the decision boundaries as a way of countering adverse decisions, exploring model fairness and locally explaining predictions by providing explanations to what could've been (minimum perturbations).

Of similar framework and ideology, Adversarial Examples test model robustness by providing minimal to nothing perturbations that cause model to missclassify (cause a point to cross decision boundary).

In this work, we identify similarities in both frameworks, extend already stated differences from previous works to other fields of AI such as dimensionality, transferability etc. and try to observe these similarities and differences in different classifier with tabular and image data.

We note that this topic is an open discussion and the work here isn't definite and can be further extended or modified in the future, if new discoveries found.

1 Introduction

As years go by, Artificial Intelligence (AI) continues to spread through to every crevice of society, from basic things like, movie recommendations to life altering aspects like health care, banking, politics, automation etc. With this increasing use of AI especially Machine Learning models for decision making, there are questions on the robustness and trustworthiness of these models. And also the problem of understanding how these Machine Learning algorithm work and how they affect our daily life are becoming more of a daily concern. To dispel this cloud of uncertainty and mistrust of AI, researchers relentlessly put forward measures to strengthen these algorithms and make them more understandable.

With the boom of AI in all its grace, beauty and efficiency, a chink in the armor was finally spotted. AI is not that perfect after all. It was discovered that the slightest change in input is enough to disrupt the very mechanisms of a well trained model. This slightest changes we refer to them as **perturbations** and now before any Machine Learning (ML) algorithm is put forward, it's robustness is questioned. On the issue of ROBUSTNESS, we consider **ADVERSARIAL ATTACKS**, a method comprising of finding the minimum perturbation of an instance such that misclassification occurs. These misclassified perturbed instances are called **ADVERSARIAL EXAMPLES**. The idea is, the less vulnerable the models are to these attacks the more trust we could have in their predictions.

Per General Data Protection Regulation (GDPR)'s right to explanation, every affected user or person has the right to know what was cause of a decision and how to counter this decision. With aim to make Machine Learning models more accessible and understandable to professionals and lay people, the field of Explanaible AI (short XAI), was opened to put forward research and methods for explaining ML and Deep Learning (DL) models. Most of these methods use feature importance or different heuristics to come up with explanations. What is of interest here is **COUNTERFACTUAL EXPLANATIONS**. Put forward by Wachter et al. (2018), this method is an example based explanation method that finds the minimum change in an instance such that the another outcome is obtained.

As noticed, both concepts almost boil down to the same optimization formula and there exist even some Counterfactual Explanations (CFE) generating techniques that are based on algorithms with inspiration from adversarial examples like Generative Adversarial Networks (GAN). So how do these differ? While some argue that the terms are interchangeable, others put forward differences in semantics. Most works especially those relating to Counterfactual Explanations tend to consider the underlying differences as trivial or sometimes non-existent depending on scenario and context. While not many works comparing these two concepts exist, the few existing ones try to answer the question of (1) If one term is just a reformulation of the other (2) How one notion relates to the other, is one a subset of the other or are they equal, (3) what mathematical aspects can be used to differentiate both topics. The line that divides both aspects does exist, could be blurry for certain models or datasets but the fundamental differences in aim, approaches to feature values, and perceptibility to humans are not to be ignored.

2 Background

2.1 Counterfactual Explanations

2.1.1 XAI and Explainability

As already mentioned, Explainable Artificial Intelligence (XAI) is that field of AI with primary focus on explaining Machine Learning models i.e. making them more interpretable. As defined per [1], explainability or interpretability is *"the degree to which a human can understand the cause of a decision"* or better defined in our context *"Interpretability is the degree to which humans can consistently predict the model's result"*.

Given that over the years Machine Learning models have proven somehow consistent and trustworthy, why even bother explaining them and not just blindly trusting them? But there lies the problem "why should we trust something we don't understand?" With increasing dependence on AI, there's often been a risk of conflating *'prediction* with *'prescription'*, meaning that in high stakes situations, not taking into account the truths of reality and the machinations of these predictive models, we might end up in trouble. Then, it would make sense that providing explanations for independent predictions would help know what accounted for a decision and these explanations are then compared to our knowledge of the world (or domain for professionals), making decision implementation or rejection way more confident and consistent

However, the complexity and enormity of the frequently used models are difficult to surmount. We especially refer to the so-called black box models whose inner workings are unobservable. So how do we open the black box? Best answer: WE DON'T. As pointed out by [38], *"explaining a prediction is not necessarily deciphering the model but finding ways to communicate the information in an interesting and engaging way"*. This means explanations should take into account human understanding, be truthful and consistent. Counterfactual Explanations prosper in this regard as they provide grounds for understanding what could've been another outcome and basis for recourse.

2.1.2 Explanations

An expected answer to the query "Why X?" would be "because Y". In this case, "because" implies that "Y" is the cause of "X". From a user's perspective, say a bank loan was rejected. "Why was my bank loan application rejected?" He/She might ask, the answer may be "Because of your low credit score". An explanation **relates the feature values (credit score etc...) of an instance (user) to its model prediction in a humanly understandable way**. The above example uses NLP language to express the explanation as text. This is not always the case. Explanation may come as decision trees or a set of instances.
So what makes for a good explanation?

- **Explanations are Contrastive**: It should be able to provide enough grounds for comparisons for why a prediction is versus why it is not.

- **Explanations are Comprehensible**: Should be easy to understand by a non professional.
- **Explanations are Stable**: Similar explanations for similar instances
- **Explanations are Consistent**: For different models, for the same instance, the explanations should be similar
- **Explanations are Realistic**: Should take into account user's situation and real world trends and tendencies.
- **Explanations are Accurate**: Should be capable of solving user's problem or clearing doubt.
- **etc...**

There is still an ongoing debate on what a good explanation is but the above stated properties are sufficient within the context of this research.

2.1.3 Counterfactual Explanation (Brief History)

We first understand what a counterfactual is. Formally defined, a counterfactual defines or expresses any event contrary to fact (what really happened). A counterfactual statement would be of the form *'if c didn't happen, then e wouldn't have happened'*, given that c and e are two distinct events. In Lewis' 1973 argument, he defines a counterfactual in terms of closest possible worlds that is, *If A implies C, then a counterfactual is the closest possible A-world such that C does not happen, A and C two events.* To summarize, a counterfactual is simply the exploration of *'What If'?* scenarios, Judea Pearl even goes forward to claim that counterfactual thinking sparked the flames of human evolution. However, there still exists skeptics among the statistics community as they deem counterfactuals to be unmanageable and untestable since by definition they are unobservable.

So how did Counterfactual Explanation even become an idea? To understand this, we step for a bit into Causal Inference. From Lewis' attempts to define causal dependence in terms of counterfactual dependence to Hitchcock and Pearl's approaches at manipulating causation using structural equations, lingers the question of whether causation could ever be described in terms of counterfactuals. Most relevant to Counterfactual Explanation as defined by Wachter et al., is Judea Pearl's Structural Causal Model and his 'mini-surgeries'.

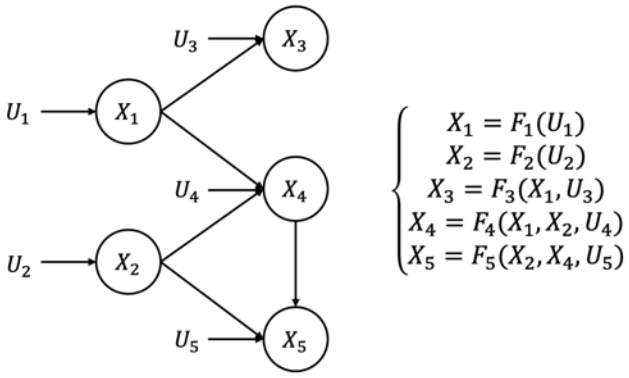


Figure 1: An example of a structural causal model. A directed arrow points from the parent node to the child node. With set of exogenous variables $\{U_1, \ldots, U_5\}$ and endogenous variables $\{X_1, \ldots, X_5\}$

Briefly defined, a Structural Causal Model (SCM) M is given by

$$M = \langle U, V, F \rangle$$

where U is a set of variables called exogenous that are determined by factors outside the model, V is a set of endogenous variables partitioned as $\{V_1, \ldots, V_n\}$, F is a set of functions $\{f_1, \ldots, f_n\}$[1] where each $f_i : U \cup (V \backslash V_i) \rightarrow V_i$ and $f_i(pa_i, u_i) = v_i$ with $pa_i \in PA_i \subset V \backslash V_i$ with PA_i the set of "parents" (causes) of endogenous variables in the set V_i (each endogenous variable (effects) can be written as a linear combination of its parent nodes (causes) and exogenous variables affecting theses parent nodes).Every causal model is associated with with a directed graph G(M), in which each node corresponds to a variable in V and the directed edges point from members of its parents PA_i toward V_i.

Pearl's SCM proposes a way of computing counterfactuals using "mini-surgeries", that is, substitution of variables to observe change in outcome. Wachter's optimization problem takes inspiration from pearl's mini-surgeries on the SCM, that is, generating counterfactuals is finding the best value that fits the equation. However, not all properties of the SCM are taken into accounts in most CFE generating algorithms.

2.1.4 Mathematical Formulations and Generation Approaches

We explore Wachter et al.'s proposal. We consider the following example.

> *"You were denied a loan because your annual income was 30,000. If your income had been 45,000, you would have been offered a loan"*

The statement is *'You were denied a loan because your annual income was 30,000'* is the actual statement of the user. *'If your income had been 45,000, you would have been offered a loan'* is the counterfactual example that represents the minimum possible income change such

[1] these functions are just simple regression models, nothing complicated, finding the weights however, is what is complicated

that the user's loan application is validated. A statement of this form is a counterfactual explanation or more generally in ML terms

> *"Score p was returned because variables V had values $(v_1, v_2, \dots)$ associated with them. If V instead had values $(v_1^{'}, v_2^{'} \dots)$ and all other variables had remained constant score $p^{'}$ would have been returned"*

Given an instance $x_{orig} \in \mathcal{R}^d$, a model $f_w(\cdot)$. The the optimization formula generates the counterfactual $x_{cf} \in \mathcal{R}^d$ by minimizing the following:

$$\arg \min_{x_{cf}} \max_{\lambda} \lambda yloss(f_w(x_{cf}), y_{cf}) + d(x_{orig}, x_{cf}) \tag{1}$$

Where, $d(\cdot, \cdot)$ is a distance metric measuring how far the counterfactual x_{cf}, $yloss(x_{cf}, y_{cf})$ is a loss function measuring the difference between the actual prediction vs intended prediction and the original point x_{orig}, λ a regularisation term. Local minima can be used as a diverse set of multiple counterfactuals.

Instead of the usual L2-norm, the distance measure used in this case, is the L1-norm or Manhattan distance weighted by the inverse median absolute deviation that is,

$$d(x_{orig}, x_{cf}) = \sum_{k \in F} \frac{|x_{i,k} - x_k^{'}|}{\text{MAD}_k} \tag{2}$$

where,

$$\text{MAD}_k = \text{median}_{j \in P}(|X_{j,k} - \text{median}_{j \in P}(X_{i,k})|) \tag{3}$$

In other words, the aim is finding the minimum perturbation $\delta \in \mathcal{R}^d$ such that

$$x_{cf} = x + \delta \text{ and } f_w(x_{cf}) = y_{cf} \tag{4}$$

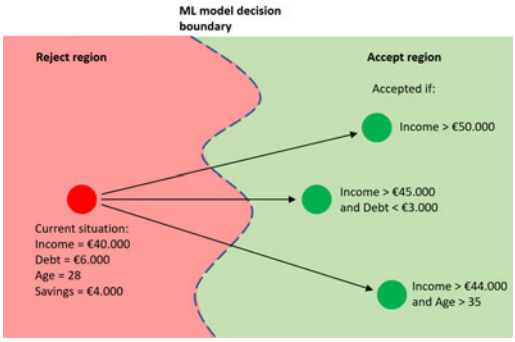


Figure 2: Image from `https://vis.win.tue.nl/masterprojects/100/`, showing a rejected option and the proposed plausible counterfactuals and directions of recourse

A CFE is not only based on proximity as most often times proximal points lack understandable content and are non-actionable[2]. To understand what other properties make

[2]we note here that the main goal of CFE is Recourse

a useful CFE, we look at the following example from [117] modified by [145].
Suppose Alice walks into a bank and seeks a loan. The decision is impacted in large part by a machine learning classifier that considers Alice's feature vector of {Income, CreditScore, Education, Age, Race, Religion}. Unfortunately, Alice is denied the loan she seeks and is left wondering (1) why the loan was denied? and (2) what can she do differently so that the loan will be approved in the future? A possible answer to (2) might be the counterfactual recommended by the system might be to increase her Income by 10K or get a new master's degree or a combination of both (**Validity and Causality**). Now consider another CFE that says she increases her income by 50k. While it does the job, it is most pragmatic for her if she can make the smallest change possible (**Proximity and Realism**). Also, it is easier for Alice to focus on changing just a few features instead of many (**Sparsity**). Also it would make sense if immutable features stayed unchanged (**Feasibility and Actionability**).

The words in bold describe the properties of a good CFE that is, something that makes explanations easy for Alice to understand and easy for Alice to implement.

Most CFE generation algorithm are merely extensions, variations or reformulations of (1). For instance, DICE (Ramaravind et al.) adds a prximity and divesrity constraint to the classic optimization problem to generate multiple CFEs for differentiable models, [50, 87] add density functions to ensure model domain closeness. Or [64] that reformulates the problem as a maximum likelihood optimization approach i.e. generating CFE is the same as maximizing the following

$$\Pr(x_{cf}|y_{cf}, x) \tag{5}$$

Broadly speaking, these approaches could be classed into two groups: Model Specific approaches that work only for specific approaches like [43, 73, 74, 75, 76, 65, 78, 64, 79, 81, 82] for differentiable or [84, 85, 87, 88, 89, 63, 90, 67] for linear models, and Model agnostic approaches (black box approaches) that don't need access to model's internals and so can be used for every model e.g. [48, 43, 99, 62, 116, 69, 72, 46, 100, 70, 59, 101, 102, 103, 106, 107, 110, 111, 113, 114, 115]. When putting forward proposals on how to generate counterfactuals, researcher's follow certain guidelines i.e. they make sure or aim that their algorithms attain certain objectives, some of which are Validity (does the algorithm always derive the desired outcome?), Sparsity (the least number of features the algorithm changes the better), Proximity (does the algorithm produce CFEs close to the original point?), Model-agnosticity (Is the algorithm applicable to all models?), Diversity (Can the algorithm produce multiple CFEs for a single input?), Feasibility (Are the CFEs generated by this algorithm actually doable?) , Data Manifold Closeness (CFE generated by this algorithm stays within range of plausiblity) , Causal Relations (Does this algorithm preserve casual relationships?), Amortized Inference (Can this algorithm produce multiple CFEs for multiple points at a time?), Categorical Feature Handling (How well does this algorithm handle Categorical Variables?) etc.

In this work, we use proposals CLEAR from White et al. and DICE from Ramaravind et al. to generate CFEs compare them with AEs and substantiate the claims made on their differences.

2.1.5 CFE generation with CLEAR

White et al. propose the approach **C**ounterfactual **L**ocal **E**xplanations vi**A** **R**egression (CLEAR) combining methods proposed by Wachter et al.(***b***-counterfactuals(boundary - counterfactuals)) and Ribeiro et al. (**LIME**) using their advantages and overcoming their shortcomings. CLEAR just like LIME fits a regression model around an instant to use the weights as basis for explanation (estimated b-counterfactual) and compares it to the counterfactual generated using the optimisation formula by Wachter et al. (***b***-counterfactual) using the so-called fidelity error which is just the difference between (1) the distance between the original point and the (***b***-counterfactual) and (2) the original point and the (estimated ***b***-counterfactual) . It's overall framework [145] is as follows:

Given an instance x, a model $m : X \rightarrow Y$ and y such that $m(x) = y$. CLEAR generates counterfactual explanations by the following steps:

- Determine x's ***b***-counterfactual i.e. a grid search through a set of possible values for features of x such that we optimize Wachter et al. equation.
- Generate synthetic observations by sampling data using different techniques.
- Create a balanced neighbourhood i.e. create a dense cloud of points between x and the nearest points just beyond m's decision boundaries such that these points are equally distributed across each classes.

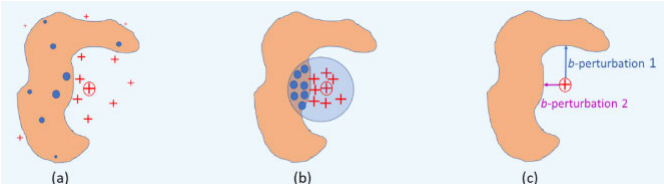


Figure 3: Toy example of a machine learning function represented by tan/blue background. The circled cross is x whose prediction is to be explained. The other crosses are synthetic observations. (a) LIME uses all synthetic observations in each regression with weights decreasing with distance from x. (b) CLEAR selects a balanced subset of synthetic observation. (c) shows the corresponding b-perturbations.

- Perform a step-wise regression on the neighbourhood dataset such that the regression goes through x. Multiple and logistic regression could be used.
- Evaluate the counterfactual value for a feature f for the CFE by substituting x's ***b***-counterfactual values from the counterfactuals in step 1, other than for feature f itself into the regression equation and calculating the value of f.

Example

An MLP with a softmax activation function in the output layer was trained on a subset of the UCI Pima Indians Diabetes dataset. The MLP calculated x a

Algorithm 1: BALANCED_NEIGHBOURHOOD

Input: S (synthetic data), x, m, $\{b_1, b_2\}$ (margins around decision boundary)
Output: N (neighbourhood dataset)
$n \leftarrow 200$;
for $s_i \in S$ **do**
  $d_i \leftarrow$ Euclidean_Distance(s_i, x)
  $y_i \leftarrow m(s_i)$
end
$N_1 \leftarrow \frac{n}{3}$members of $\{S\}$ with lowest d_i s.t. $0 \leq y_i \leq b_1$
$N_2 \leftarrow \frac{n}{3}$members of $\{S\}$ with lowest d_i s.t. $b_1 \leq y_i \leq b_2$
$N_3 \leftarrow \frac{n}{3}$members of $\{S\}$ with lowest d_i s.t. $b_2 \leq y_i \leq 1$
return $N \leftarrow N_1 \cup N_2 \cup N_3$

probability of 0.69 for x belonging to class 1 (having diabetes). CLEAR generated the logistic regression equation $(1 + e^{w^T x})^{-1} = 0.69$ where:

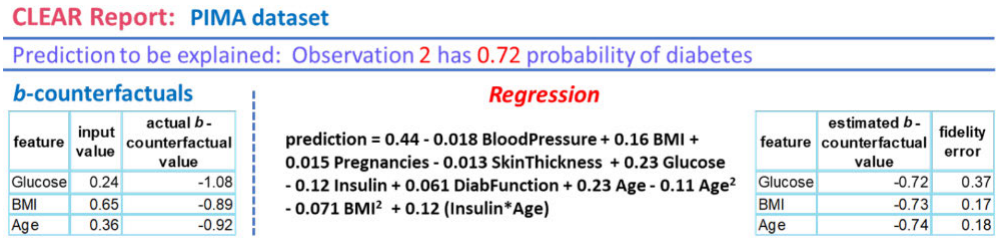
$$w^T x = -0.8 + 1.73Glucose + 0.25BloodPressure + 0.31Glucose^2$$

Substituting in the regression equation $w^T x = 0$, the BloodPressure in x

$$-1.73Glucose + -.04 - 0.31Glucose^2 = 0$$

From the original value of Glucose being 0.537 we obtain the counterfactual 0.025

- Iterate till explanation with best fidelity error is obserevd or till some threshhold is met. Below is an example of a CLEAR report.



feature	input value	actual *b*-counterfactual value
Glucose	0.24	-1.08
BMI	0.65	-0.89
Age	0.36	-0.92

feature	estimated *b*-counterfactual value	fidelity error
Glucose	-0.72	0.37
BMI	-0.73	0.17
Age	-0.74	0.18

Figure 4: Example of a Clear CFE report. Here CLEAR uses multiple regression to explain a single prediction generated by an MLP model trained on the PIMA dataset

Algorithm 2: CLEAR Algorithm

Input: t (training data), x, m, T
Output: Explanations
$S \leftarrow$ Generate_Synthetic_Data(x,t,m);
for *each target class tc* **do**
  for *each feature f* **do**
    $w \leftarrow$ Find_Counterfactuals(x, m)
  end
  $N_{tc} \leftarrow$ Balanced_Neighbourhood(S, x, m)
  Optional: $N_{tc} \leftarrow N_{tc} \cup w$
  $r \leftarrow$ Find_Regression _Equations(N_{tc}, x)
  $w^{'} \leftarrow$ Estimate_Counterfactuals(r, x)
  $e \leftarrow$ Calculate_Fidelity$(w, w^{'}, T)$
  return $expl_{tc} =< w, w^{'}, r, r >$
end

2.1.6 CFE Generation with DICE

Diverse Counterfactual Explanations (DiCE) as the name indicates proposes a framework for generating multiple CFEs for differentiable models. Ramaravind et al. propose an extension of Wachter et al. optimisation problem by incorporating divesity constraint .
Given an instance x, the proposed diversity measure *dpp*_diversity for a set of counterfactuals $\mathcal{C} = \{c_1, \ldots, c_k\}$ is calculated by building on determinantal point processes and is given by

$$dpp\_\text{diversity} = \det \boldsymbol{K} \tag{6}$$

where $\boldsymbol{K}_{i,j} = \frac{1}{1+d(c_i,c_j)}$
Similar to Wachter et al proximity measure, DiCE uses the following modification

$$Proximity = -\frac{1}{k}\sum_{i=1}^{k} d(x, c_i) \tag{7}$$

Given (39) and (40), DiCE aims to optimize the following problem:

$$\underset{c_1,\ldots,c_k}{\arg\min} \frac{1}{k}\sum_{i=1}^{k} yloss(m(c_i), y_{cf}) + \frac{\lambda_1}{k}\sum_{i=1}^{k} d(x, c_i) - \lambda_2 dpp\_\text{diversity} \tag{8}$$

where, m is a differentiable model, y_{cf} the desired outcome, $|\mathcal{C}| = k$, λ_1, λ_2 are hyperparameters balancing the loss function.
On the choice of the **distance function**, for continuous variables, we use same metric as Wacther et al. that is

$$d_{cont}(c, x) = \frac{1}{n_{cont}} \sum_{p=1}^{n_{cont}} \frac{|c^p - x^p|}{\text{MAD}_p} \tag{9}$$

where, MAD_p is as defined in (3) and n_{cont} is the number of continuous variables.
For categorical variables, we simply sum over the indicator function with output 1 if the

value in original instance is different from that of the counterfactual instance and 0 if not.

$$d_{cat}(c, x) = \frac{1}{n_{cat}} \sum_{p=1}^{n_{cat}} \mathbf{1}_{[c^p \neq x^p]} \tag{10}$$

where n_{cat} is the number of categorical variables.

2.1.7 Related Terms

- **Contrasting Explanations**
  The word contrasts implies or indicates in what ways two items or instances might be strikingly different. Explanations tend to be intrinsically contrastive i.e. when given the *Why did P happen?*, we tend to hypothesize other events so the query becomes more understandable or easier to answer when reformulated as *Why did P happen rather than Q?*. In AI terms, constrastive explanations in terms of ***alternative explananda*** asks why a certain instance had an output y rather than an output y_{con} or ***congruent explananda*** - why a model outputs y for an instance x, and outputs y_{con} for input x_{con}. As noticed, this form of explanation is similar to CFEs. Most researchers tend not to differentiate between the two, in fact it is argued that CFEs are contrastive in nature as they compare actual scenarios to hypothesized ones and there even exist CFE generation techniques based on Contrastive Explanations. So what is the difference? Gill and [cite from paper] point out that they distinct in approach. CFE explains how an outcome could be contrastive i.e. how it could be different whereas contrastive explanations indicate the difference between actual and hypothesized scenarios.

- **Score CFEs (SCFE)**
  Consider a k class classification case, our model $m = g(h(x))$ with $h(x) = \{h_1(x), \ldots, h_k(x)\}$ a probabilistic function (for instance, the softmax layer of an ANN) and g the argmax function. **SCFE** is the reformulation of the Wachter et al. Optimization problem (1) given by:

  $$\arg \min_{x_{cf}} \max_{\lambda} \lambda yloss(h_{cf}(x_{cf}), s) + d(x_{orig}, x_{cf}) \tag{11}$$

  where λ, x_{cf}, y_{cf}, d and $yloss$ are described as in (1) above. $h_{cf}(x_{cf})$ is the score of classifying x_{cf} in the counterfactual class and s is the target score to be attained.

2.2 Adversarial Examples

2.2.1 Adversarial Machine Learning

Similar to XAI, Adversarial Machine Learning is the field of machine learning focus on testing the robustness of Machine Learning algorithm through Adversarial Attacks and providing defenses to these attacks. To give context as to why this is important let's look back on human advances in AI. Before 2013, it would be considered normal or expected if some computer vision algorithm misclassified an image. Now not so much, it is rather unexpected that such algorithms give wrong results even performing better than humans in some regards. We might boast having attained near perfection, but this perfection is questioned as recently it has been proven that the slightest imperceptible modifications might cause the classifier often Neural Networks (NN) to go off target. So the robustness of this commonly used classifiers has to be forever put to test as their implementations extend to human life and might cause potential harm like a self automated car misreading a signal, or hacking to personal accounts through theft of digital prints and much more. So how do we test robustness?

2.2.2 Adversarial Attacks and Examples

As already mentioned, Adversarial attacks seek imperceptibly trick the model into providing deceptive output. To be more intuitive, consider a model m, most often a deep learning model, an input $x \in \mathrm{R}^d$ with $m(x) = y$. Adversarial attacks seek to find the perturbation δ such that $m(x + \delta) = y_{adv}$, $y_{adv} \neq y$. To ensure imperceptibility of change, δ is often norm bounded i.e for some $l_p - norm$ and $\alpha \in \mathrm{R}^+$, an adversarial attacks seeks δ such that

$$m(x + \delta) \rightarrow y_{adv} \text{ s.t. } y_{adv} \neq y, \|\delta\|_p < \alpha \tag{12}$$

Of the existing approaches to generate adversarial examples is the well known Fast Gradient Sign Method (FGSM) by Szegedy et al. that produces perturbed instances by a gradient ascend on the loss function of the classifier with respect to the data:

$$x_{adv} = x + \epsilon \cdot \text{sign}(\Delta_x loss(x, y)) \tag{13}$$

where $\epsilon > 0$ scales the degree of perturbation and is chosen to maximise imperceptibility. Of specific importance is that $y_{adv} \neq y$. When y_{adv} is known in advance, the attack is said *targeted*, when the output class of an attack is arbitrary, the attack is *untargeted*.

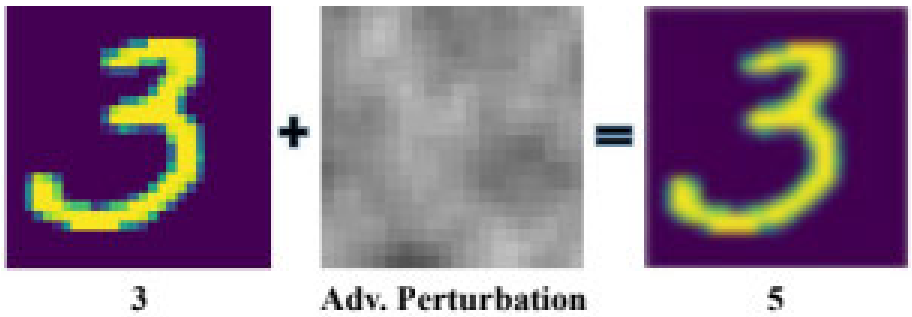


Figure 5: An example of an adversarial perturbation (δ) and adversarial example of the handwritten digit 3 from the MNIST dataset

Without ignoring the details, much like explanation methods, we have white box attacks which require access to model specifics like the gradient [165, 166] and black box attacks only requiring model output, mostly query based like [168, 167] or transfer based like [169]. Some of these attack methods like Gradient Matching, convex Polytope etc. are so-called ***Poison Attacks*** or ***backdoor Attacks***, that aim to deceive the model during the training phase often by tampering with the training data. Or model extraction attacks where an adversary steals the functionality of the victim model with only query access. The most used of them and of particular interest to this paper are the ***Evasion Attacks*** that fool an already trained models at test time into producing adversarials by feeding them adversarial examples. $x_{adv} = x + \delta$, obtained from (12) is an ***Adversarial Example***.

2.2.3 AE Generation Techniques: A small review

In this subsection, we mainly want to focus on the algorithms used for our experiments and hypothesis testing, while not forgetting to mention or quote others as they may come in handy. As already said in section 2.2, these generation techniques could be white box or black box approaches. Put more specifically, we could say that these attack techniques are either **gradient-based** (requiring access to the gradient) e.g. PGD, L-BFGS, FGM, **score-based** (relying on the scores of the logit function (softmax function) in a multi-class-classification case for instance in ANNs or **decision-based** attacks - attacks that solely act upon the final output value (e.g. max of logit scores) like Boundary attack, HopSkipJumpAttack. Over the course of our work, we consider the following attack methods:

- **PGD**
  An extension of the FGSM that also optimizes perturbation by acting on the gradient of the loss function while constraining the perturbation with the l∞-norm i.e. $\|r\|_\infty < \delta, \delta > 0$. PGD regulates this constraint by projecting the perturbation onto a $\delta-$ball (clipping the perturbed instance so it remains within designed bounds). For an instance x, the adversarial perturbation r is gotten by iterating over t given the following formula:

$$r_{t+1} = \prod_{\delta} (r_t + \alpha \cdot \text{sign}(\nabla(\mathcal{L}(x + r_t))) \tag{14}$$

  where, $\prod_\delta(\cdot)$ is the projection function, $\mathcal{L}(\cdot)$ is the loss function of the DL model.

- **Deepfool**
  Also a gradient based attack with a seemingly different approach to the conventional adversarial attack optimisation formula. Deepfool perturbs images by minimal perturbations r corresponding to the orthogonal projection of the image onto the separating affine hyperplane. Deepfool uses the formula:

$$\text{argmin}\|r\|_2 \text{ such that } \text{sign}(f(x_0 + r)) \neq \text{sign}(f(x_0)), \tag{15}$$

  where,

$$r_i = -\frac{f(x_i)}{\|\nabla f(x_i)\|_2^2} \cdot \nabla f(x_i)$$

  is updated at each iteration i

- **Boundary Attack**
  This attack compared to the previously stated attacks does not need access to the gradient and acts solely upon the final output of the model. The basic idea is finding adversarials by performing random walks along the boundary. The algorithm starts with an adversarial as the initial start and performs random walk along the boundary such that the point stays adversarial and the distance between the adversarial sample and original sample is minimized.

Algorithm 3: Boundary Attack

Input: Original instance $\boldsymbol{o}$, adversarial criterion $c(\cdot)$, model $m(\cdot)$
Output: AE $\tilde{\boldsymbol{o}}$
initialization: $k = 0$, $\tilde{\boldsymbol{o}}^{\boldsymbol{0}} \sim \mathcal{U}(0,1)$ s.t. $\tilde{\boldsymbol{o}}^{\boldsymbol{0}}$ is adversarial;
while $k <$ *max iterations* **do**
  draw random perturbation r_k from random distribution
  if $\tilde{\boldsymbol{o}}^{\boldsymbol{k-1}} + r_k$ *is adversarial* **then**
    set $\tilde{\boldsymbol{o}}^{k} = \tilde{\boldsymbol{o}}^{\boldsymbol{k-1}} + r_k$;
  else
    set $\tilde{\boldsymbol{o}}^{k} = \tilde{\boldsymbol{o}}^{\boldsymbol{k-1}}$
  end
  $k = k + 1$
end

- **HopSkipJumpAttack**
  Similar to the boundary attack, the HopSkipJumpAttack works to produce adversarial examples by reducing the distance of some initial adversarial sample to the boundary given the direction of the target point (point that is attacked) that is, AE generation through boundary estimation. To introduce the HopSkipJumpAttack, Chen et al. redefine targeted and untargeted attacks in the follwing way. Given a classifier $C(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$ and a probabilistic function $F(\cdot) : \mathrm{R}^d \rightarrow \mathrm{R}^m$ where m is the number of classes, such that $C(x) = \text{argmax}\{F_1(x), ..., F_m(x)\}$. Given that our instance is of class c_k. A successful attack on x given the adversarial $x^{'}$ is measured by

$$S_x(x^{'}) = \begin{cases} \max\limits_{c \neq c_k} F_c(x^{'}) - F_{c_k}(x^{'}) & \text{(Untargeted)} \\ F_{c_t} - \max\limits_{c \neq c_t} F_c(x^{'}), & \text{targeted} \end{cases} \tag{16}$$

such that

$$\begin{cases} S_x(x^{'}) > 0 & \text{(if successful)} \\ S_x(x^{'}) < 0 & \text{if not} \end{cases} \tag{17}$$

And at the boundary $S_x(x^{'}) = 0$
Using indicator functions the problem could be reformulated as.

$$\phi_x(x^{'}) = \text{sign}(S_x(x^{'})) = \begin{cases} 1 & \text{if } S_x(x^{'}) > 0 \\ -1 & \text{if not} \end{cases} \tag{18}$$

$\phi_x(x^{'}) = 0$ at the boundary
The HopSkipJumpAttack works to optimise

$$\min_{x^{'}} d(x, x^{'}) \quad \text{such that} \quad \phi_x(x^{'}) = 1 \tag{19}$$

by approximating the direction of the gradient of $S_x(x^{'})$ via a Monte Carlo estimate.

2.2.4 Related Terms

- **Adversarial Training**
  An intuitive defense to Adversarial examples is learning on the subspace that they exist in. Formally described, adversarial training is augmenting the training data with adversarial examples such that the unexplored spots of the data distribution are covered. This idea was brought forward by szegedy et al. 2014 [146] but Goodfellow went as far as producing adversarial attacks during adversarial training with FGSM attacks.

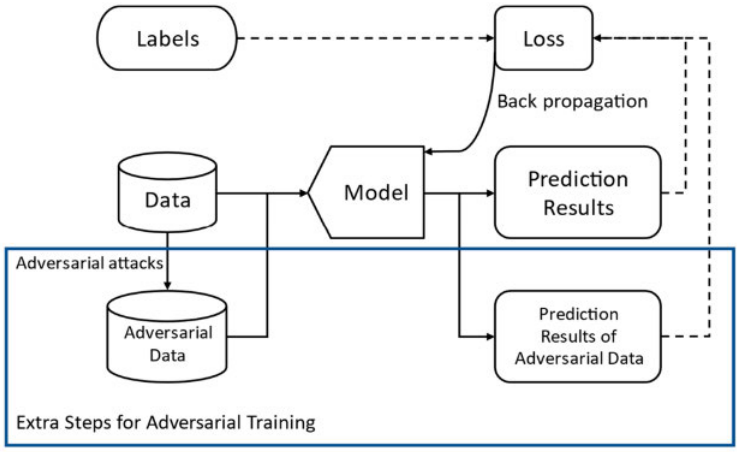


Figure 6: Image from [147]. Adversarial Training general framework

These approaches however, remain vulnerable to iterative attacks (Tramér et al 2018) [148]. Others like (Huang et al, 2015) [149] and (Shaham et al, 2018) [150] propose adversarial training on adversarial examples only and optimisation of min-max problem that minimizes classification loss against adversary that perturbs input and maximizes classification loss.

$$\min_{\theta} \mathrm{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta \in B(x,\epsilon)} \mathcal{L}_{ce}(\theta, x+\delta, y) \right] \tag{20}$$

The above stated techniques fall under a Efficient adversarial training techniques, a variation of adversarial training. Other variations of adversarial training do exist, for instance, in Goodfellow et al. (2015) [151] appears the idea of adversarial regularization, an approach adding a regularization term besides the cross entropy loss to control the ratio of adversarial examples in batches. In [151], the regularization term is FGSM based expressed as $\mathcal{L}(\theta, x + \epsilon \mathrm{sign}(\Delta_x \mathcal{L}(\theta, x, y))$. Others like [152, 153, 154, 155, 156] follow the same principle but argue that instead of fixing ϵ, it should be adapted (Adversarial training with adaptative ϵ) [157, 158, 159]. Ensemble adversarial training proposals [160, 161, 162], augment the training data with AEs from multiple other target models. There exists many other variants like Curriculum Adversarial training [153, 163, 164] or using unsupervised frameworks. More on this work in [135]

- **Generative Adversarial Networks (GAN)**
  Goodfellow et al.(2014) present a machine learning framework composed of two deep learning models that train by competing against each other. The aim of GAN is using adversarial learning to create new data instances given an input data distribution. GANs are composed of first a **generative model** - (unsupervised) models that summarize the distribution of given variables e.g. GMM, VAE etc. and second a **discriminative model** - classification task or predictive modelling. The GAN framework is proposed as a minimax game where the aim is for the generative model G is to maximize the probability of the discriminative model D making a mistake. To learn the generator's distribution, the generator receives feedback from the discriminative model, takes random noise $\boldsymbol{z}$ from a Gaussian distribution or uniform prior distribution and defines a prior on it $p_z(\boldsymbol{z})$, then represents a mapping to data space as $G(\boldsymbol{z}, \theta_g)$, where θ_g are the parameters of the multilayer perceptron G. The discriminative model D, $D(\boldsymbol{x}, \theta_d)$ (another multilayer perceptron with parameters θ_d) is trained on data from two sources, (1) the real data instances as positive instances and (2) fake generated instances from G as negative instances. $D(\boldsymbol{x})$ outputs a scalar representing the probability of $\boldsymbol{x}$ being in p_g. The aim is for D to output $\frac{1}{2}$ everywhere. GAN training can be formally written as the two player minimax game of generator G and discriminator D with value function $V(G, D)$:

$$\min_D \max_G V(D, G) = \mathrm{E}_{\boldsymbol{x} \sim p_{data}} [log D(\boldsymbol{x})] + \mathrm{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})} [log(1 - D(G(\boldsymbol{z})))] \tag{21}$$

  where $\boldsymbol{x}$ is the real data instance and $\boldsymbol{z}$ are the input noise variables. Goodfellow et al. recommend suggests alternating between k steps of optimising D and one step of optimising G during training as it is not full feasible to fully optimise D after every optimisation step of G.

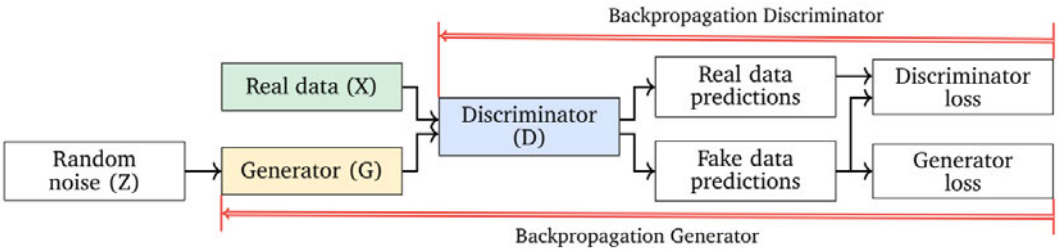


Figure 7: Image from [127]. General GAN framework. During the backpropagation training of the generative model G, the weights of the discriminant of D remain constant but it's gradients are taken into account as the generative model is trained to fool D. Similarly, G's parameter are unchanged during D's training

3 Related Works

Although adversarial examples and counterfactual explanations have gained prominence in the field of Artificial Intelligence, there isn't much (or rather precise work) researching and exploring their differences. According to Wachter et al.'s work on CFE [43], AEs do not follow the Lewisian account of closest possible worlds, that is, adversarials produced often fall into zones of low probability, that means, realistically speaking AEs are impossible. In Verna et al.'s review of CFEs [117] both terms are not interchangeable as they differ in desidarata, that is they have different aims. Of the few other works that exist, Freiesleben's approach [33] on substantiating the difference is exploring these concepts (CFEs and AEs) with respect to Aim, Role and Use Cases with aim to resolve the following main misconceptions (1) CFEs are equal to AEs (2) algorithms for CFEs could be easily used for AEs (transfer).

In [33], Freiesleben also discusses matters of proximity to original instance and conditions on missclassification or rather targeted missclassification. To Brown et al. [44], the question is not in the mathematics but rather in the semantics of explanation. Their work exposes the clear explanatory ridge caused by the lack of semantics often found in AEs but is of upmost importance to CFEs as there is no explanations without semantics (clear explanatory terms). This is mostly because, AEs often apply to (or work better on) image or audio data which contain very low semantics. Pawelczyk et al. dive more into the specifics, trying to provide mathematical formulations to the bounds of the difference in perturbation caused by AE generating and CFE generating algorithms. For instance, in their work they compare Deepfool [118] for AE and Score Counterfactual Explanations (SCFE) for CFE, manifold based methods Natural Adversarial Examples (NAE) [120] for AE and Counterfactual Conditional Heterogeneous Autoencoder (C-CHVAE) [121] for CFE and the Carlini and Wagner method for AEs (C & W) [71] and the Wachter method for generating CFEs.

Also, on how they coincide, Dandl et al. [59] and Molnar [47] refer to AEs as special cases of CFEs. Freiesleben [33] agrees to this idea and further states that some CFEs could be used as AEs. Based on this ideas, there exists CFE generating algorithms with inspiration from AEs. For instance, CounteRGAN by Nemirovsky et al., uses a remixed version of the RGAN optimization formula to produce CFEs, C-CHVAE which uses Variational Auto Encoders to generate "faithful" CFEs for tabular data or Jeanerette et al.'s Adversarial Counterfactual Visual Explanation (ACVE) [61] which polishes adversarial attacks to produce attacks to produce CFEs for image data.

As much as these approaches try to differentiate these concepts, they do not dive deep enough into the specifics of Machine Learning or more relevant fields of AI like dimensionality etc. This work doesn't in any way discredit or flaw the above proposed works but rather quotes and extends these works to other relevant fields of AI.

4 CFE and AE: Similarities and Connections

We already discussed the similarity in optimisation frameworks for both AEs and CFEs. Most particular as both absolutely require proximity, be it for ease in actionability (CFEs) or for imperceptibility (AEs). Proximity in this case being the search for the smallest perturbation $\delta \in \mathrm{R}^d$ such that the perturbed input is classified to a targeted class (CFE) or missclassified (AE). In both cases, how small δ is, indicates the fragility or sensitivity to slight change in input. This in a sense makes CFEs as well as AE a measure for robustness. This assertion then prompts the following claim:

> *"In an imperfect model, for a large enough threshold on the norm of the perturbation vector, a targeted AE could be a CFE".*

Which is true if the only restriction on the CFE generated is proximity and correct classification.
Again on the aspect of proximity, both CFEs and AEs point out the bias, flaws and unfairness of machine learning models. We consider the following definition

Definition 1 (Contesting CFEs)
Imperfect algorithms sometimes make decisions that do not reflect ground truth or that are unfair. This could be because (1) an ML model cannot distinguish between correlation and causal relationships i.e. variables with no causal relationship but correlation to target variable impact classification [66] or (2) overfitting, underfitting, missing values [83] etc. A ***Contesting CFE*** is a CFE generated with aim to argue the impartiality or mistakes of a decision model.
According to Freiesleben [33] *a contesting CFE is an AE* since it is made with grounds to missclassify. To understand this we consider the following example from [33]

We assume that for a loan approval algorithm them ML model is trained on collected data from the members of two clubs. The first club is a dog-club in Zurich (Switzerland) and the second is an animal protection club in Ukraine. It is clear that this data collection is biased. Let us also assume that the model trained by the bank is a single-layer decision tree. Then, the algorithm classifies based on the strong correlation between number of dogs and loan approval. If a person has more or equal to one dog, the algorithm offers the loan. Irrelevant of the salary, say the applicant has only one dog. In this case, the loan application would be rejected. This decision would be correct according to the ground truth if the salary was too low. However, the reason for the algorithm's decision would be that threshold two for the number of dogs was not reached. A CFE, in this case, would be: *If P's had one more dog, her loan application would have been accepted.* This would indeed be a good CFE since it points us to the reason the algorithm had for its decision. It would increase the applicant's understanding of the algorithm, would allow her to contest the decision, and in case she really urges for money she could use this information to deceive the algorithm. This is exactly how contesting CFEs are characterized. Interestingly, an AE would be described by the same vector and could potentially have the very same function, namely deceiving the system.
But, the reverse is **not always true** since they may exist too many causally related irrelevant features that contribute to alternative classification which diminishes the explanations quality.

5 CFE and AE: Their Differences

5.1 On their Conditions for Existence

CFEs are generated under the assumptions that (1) the model has high predictive performance (2) the model is robust. In AEs generation, the only assumption necessary is high predictive performance. We argue in this section that for any robust enough models, AEs do not exist but CFEs do. To back up this arguments, we reformulate the definitions of CFEs and AEs Given an instance $(x, y) \in \mathcal{X} \times \mathcal{Y}$, a model $m : \mathcal{X} \to \mathcal{Y}$ and a distance metric $d : \mathcal{X} \times \mathcal{X} \to$ R:

Definition 1 (alternative): $x^{'}$ such that $m(x^{'}) \neq m(x)$ is an *alternative* to x.

Definition 2 (ϵ-alternative): Let $\epsilon > 0$. $x^{'}_{\epsilon}$ is an ϵ-alternative to x if

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

Definition 3 (Counterfactual): x_{cf} is a said *counterfactual* to x if $d(x_{cf}, x)$ is minimal and $m(x_{cf}) = y_{cf}$, given $y_{cf} \neq y$ and y_{cf} is known in advance.

Definition 4 (misclassified): We say $x^{'}$ is *misclassified* if $m(x^{'}) \neq y$ relative to expert-human assignment.

Definition 5 (Adversarial Example): x_{adv} is said an adversarial example if x_{adv} is an ϵ-alternative and misclassified

Definition 6 (Targeted Adversarial Example): x_{adv} is said an targeted adversarial example if x_{adv} is an ϵ-alternative and classified to a target class different from y

Definition 7 (Untargeted Adversarial Example): x_{adv} is said an untargeted adversarial example if x_{adv} is an ϵ-alternative and classified to an arbitrary class different from y

To defend our argument, we consider Definition 6. In a robust model, for ϵ large enough, the set of adversarials is empty. That is, adversarial perturbations are contraint to ensure imperceptibility. Whereas, the concern of CFEs is the closest plausible solutions with no constraints on the norm of the perturbation.

Other important notable misconception is that a CFE is just a targeted AE which may be true for some large enough ϵ, but the constraint on this ϵ makes it impossible for a targeted AE to exist within robust models. This misconception however, is basis for many CFE generating algorithms like [61].

5.2 On their Aims, Role and Use Cases (Freiesleben)

Normally, or more than often, explanations pertain to people of interest with little to no knowledge of a given decision making algorithm. So by definition, explanations have as aim to be understandable, provide actionable, meaningful and observable changes and above all (especially within the context of Counterfactuals) provide grounds for recourse (Harimi et al.) i.e. give meaningful reasons of why it was and how it could be. On the other hand, Adversarial Examples come more handy during training or testing models and are mostly

used by professionals or scientists on high dimensional complex data e.g image or audio classification using DNN. Contrary to CFEs, the main properties of AEs is imperceptibility i.e. unnoticable changes barely recognisable by humans but enough to fool the machine. To briefly formulate the above stated differences we say that *CFEs prefer Sparse solutions (few feature changes) while AEs would rather Imperceptible Solutions (as close as possible) solutions.*

While the use of CFEs is not only restricted to tabular data, their use on high dimensional audio and image data or data with a more or less low level of semantics is questionable and even less recommended as without clear semantics there clearly exist no concrete explanations. Adversarial Examples are favored in high dimensional abstract data classification. Most of the existing works relate to image classification with DNNs.

5.3 Curse of Dimensionality

In higher dimensions, the generation of stable and trustworthy CFEs gets quite harder but Adversarial Examples see the curse of dimensionality rather as a blessing as with higher dimensions they become easier to produce. To see this we get into works from [91] and [143]. With **Counterfactual Explanations**, works from hammer et al. [91] especially on the robustness of CFEs point out the reduction of effectiveness of CFEs in higher diménsions i.e. local instabilities of CFEs in higher dimensions. We consider a classifier $m : \mathcal{X} \rightarrow \mathcal{Y}$, an instance (x_{orig}, y_{orig}) such that $m(x_{orig}) = y_{orig}$. Consider a perturbed instance $x^{'}$ with respect to some probability density $p_{\epsilon}(x_{orig})$ such that $m(x^{'}) = y_{orig}$. Then, if x_{cf} is a CFE for x_{orig} and $x^{'}_{cf}$ is a CFE for $x^{'}$. Then the similarity of explanations or local instability is quantified as

$$\underset{x \sim p_{\epsilon}(x_{orig})}{\mathrm{E}} \left[d(x_{cf}, x^{'}_{cf}) \right] \tag{22}$$

given d some distance measure.

According to Hammer et al., the higher the dimension, the greater the probability $p(d(x_{cf}, x^{'}_{cf}) \geq \delta)$ i.e.

- **With Gaussian Perturbation:**

$$\underset{x \sim p_{\epsilon}(x_{orig})}{\mathrm{E}} \left[d(x_{cf}, x^{'}_{cf}) \right] = d - 1 \tag{23}$$

  which implies with the Markov's inequality

$$p(d(x_{cf}, x^{'}_{cf}) \geq \delta) \leq \frac{d-1}{\delta}, \quad \delta > 0 \tag{24}$$

- **With Uniform Perturbation:**

$$\underset{x \sim p_{\epsilon}(x_{orig})}{\mathrm{E}} \left[d(x_{cf}, x^{'}_{cf}) \right] = \frac{\epsilon^2 (d-1)}{3} \tag{25}$$

  which implies with the Markov's inequality

$$p(d(x_{cf}, x^{'}_{cf}) \geq \delta) \leq \frac{\delta \epsilon^2 (d-1)}{3} \tag{26}$$

But with **Adversarial Examples**, proven by [143], the higher the dimension, the easier it is to produce adversarials. The idea is, the larger the dimension, the more the volume is concentrated at or close to the boundary making it easier to find perturbations that produce AEs.

Let's consider a ball A of dimension d. Say we shrink this ball by multiplying by $1-\epsilon$, $\epsilon > 0$ i.e. $(1-\epsilon)A = \{(1-\epsilon)x | x \in A\}$. Then the volume of $(1-\epsilon)A$ is $(1-\epsilon)^d$ times that of A. i.e.

$$\frac{\text{volume}((1-\epsilon)A)}{\text{volume}(A)} = (1-\epsilon)^d \leq e^{-\epsilon d} \tag{27}$$

If we fix ϵ and $d \to \infty$, then $e^{-\epsilon d} \to 0$. This means, as $d \to \infty$, the more the volume of the ball lies within the annulus formed by the intersection of $(1-\epsilon)A$ and A.

- **Under Gaussian Distribution:**
  The Gaussian Annulus Theorem [144] states that for a d-dimensional spherical Gaussian distribution with unit variance in each direction direction, for any $\beta \leq \sqrt{d}$, then most of the probability mass lies within the annulus $\sqrt{d} - \beta \leq |x| \leq \sqrt{d} + \beta$. Even though the density mainly lies at the center, it contains little volume i.e. most of the points lie within the annulus of radius $\sqrt{d}$. Thus, the higher the dimension the closer the point are to the boundary.

- **Under Uniform Distribution:**
  It follows the same pattern as in Gaussian distribution but here we mention that given a d-dimensional unit ball of radius r, then the bulk of the points lie in the annulus of radius $\frac{r}{d}$ and that $d \to \infty$, the more points lie near the equator.

5.4 On the Semantics of an Explanation

Mathematically speaking, Adversarial Examples and Counterfactual Explanations, without ignoring their underlying differences, are identical with respect to optimization. But, the ridge separating them lie in the core definition of what an explanation is. Miller (2019) [1] states that explanation are social, i.e. they take inspiration from normal day-to-day human-to-human interactions, making them more likely to be understandable and feasible. Explanations heavily rely on semantics, that is *they make sense and are logical and understandable.* by semantics here, we refer to identifiable features that machine learning models used as determining factors for prediction. As we observe, Counterfactual Explanations are mostly used in tabular data with semantically rich content. To understand this, we backtrack to the example by Wachter et al (2019) [43]

> *"You were denied a loan because your annual income was 30,000. If your income had been 45,000, you would have been offered a loan"*

If we consider the features involved in resulting prediction to be Age, Income, Education, Sex etc. from the explainee's point of view, any perturbation of any or combination of the given instances, present a more, straightforward, helpful understanding of the underlying, process of prediction and a more actionable line of recourse.

In contrast, Adversarial Examples, which are predominantly used for audio, image data with low semantics, do not or barely provide any understandable perturbation as (1) a change in a single pixel is unsurprisingly not very helpful and (2) the imperceptibility of change and minimality of the perturbation vector δ are not easily observable to humans.

5.5 On the Choice of Distance Functions

As already mentioned, Counterfactual Explanations and Adversarial Examples differ in aim, so as much as their optimisation frameworks are similar, they work to attain different objectives. When producing Counterfactual Explanations, we opt for sparse results i.e. minimality in the number of features change as we want real life actionability. In the optimisation problem proposed by Wachter et al., the distance used is the simple l_1-norm or Manhattan distance normalised by the Mean Absolute Deviation. That is given a point $x_{orig} \in \mathcal{X}$ and its corresponding CFE x_{cf}.

$$d(x_{orig}, x_{cf}) = \sum_{k \in F} \frac{|x_{orig_k} - x_{cf_k}|}{\text{MAD}_k} \tag{28}$$

where,

$$\text{MAD}_k = \text{median}_{j \in P}(|\mathcal{X}_{j,k} - \text{median}_{i \in P}(\mathcal{X}_{i,k})|) \tag{29}$$

Where as, adversarial attacks, since the basic principle is proximity, they opt for simpler measures like the l_0- norm, the l_p-norm (often $p = 2$), or the l_∞-norm. However, in higher dimensions, l_p-norms are rather ineffective and useless.

However, the above mentioned distance measures are most times applicable only to continuous features. The question on how to handle categorical features both in XAI and Adversarial Machine Learning is still an open question but for the most part, CFE generating algorithms just use indicator functions or l_0-norms as a distance measure for categorical features. For instance, ProCE by Duong et al. and DICE by Ramaravind et al. use d_{cat} given by

$$\sum_{i=1}^{n_{cat}} \mathbf{1}\{x^i_{orig} \neq x^i_{cf}\} \tag{30}$$

where n_{cat} is the number of categorical features and $\mathbf{1}$ is an indicator function such that

$$\mathbf{1}\{x^i_{orig} \neq x^i_{cf}\} = \begin{cases} 1, & \text{if } x^i_{orig} \neq x^i_{cf} \\ 0, & \text{else} \end{cases} \tag{31}$$

Adversarial Examples mostly use hamming distance or Categorical cross entropy. Measures which are not perfect but efficient and we note here that every CFE or AE problem might differ and different techniques may be involved in their generation. But let's get more into the topic of categorical features.

5.6 On Categorical Features Handling

How to approach discrete spaces and measure distances within these spaces comes as a challenge in Counterfactual Explanation generation as well as Adversarial Example production.

In most research, the common approach is just to one-hot encode such variables and use indicator functions or hamming distance as distance measures. One-Hot encoding however is very problematic as it lacks smoothness, and has no appropriate distance metric. Also with respect to perturbation, minimal changes in encoded values may not result in anything meaningful but for a great enough δ, the resulting AE is unrealistic and unnatural.

In the case of Adversarial Examples, there exist no minimal perturbation in the discrete space i.e. if given four categorical features $\{1, 2, 3, 4\}$, a change from 1 to 2 will not be unnoticable and defeats the purpose of imperceptibility. Gradient based approaches like FGSM, compute gradients with respect to the input and update the categorical variables in the direction that maximizes the loss, leading to misclassification. What is important is the transfer of categorical values into a continuous space. Yong et al. (2020) propose a two step greedy attack that transfers these discrete features into a probabilistic space through some embedding, searches for suitable feature values within this probability distribution and substitutes them to obtain optimal Adversarial Examples. The approach by He et al., uses the same idea but the encoding is done by constructing probabilities on the categorical feature values i.e. the probability that a certain categorical feature value lies within a certain category. With respect to GAN, WGAN by Arjovsky et al. [119] propose comparing the distributions generated by the categorical features using the Wasserstein distance measure. That is, given two marginal densities P_r and P_g, the Wasserstein distance or Earth-Mover distance is given by

$$W(\mathrm{P}_r, \mathrm{P}_g) = \inf_{\lambda \in \prod(\mathrm{P}_r, \mathrm{P}_g)} \mathrm{E}_{(x,y)\sim\lambda} \left[\|x - y\| \right], \tag{32}$$

where $\prod(\mathrm{P}_r, \mathrm{P}_g)$ denotes the set of all joint distributions $\lambda(x, y)$ whose marginals are P_r and P_g respectively.

To deal with categorical features while producing CFEs [111] encodes categorical variables distance using Markov Chain transitions, [86] relaxes categorical features to continuous ones using Gumbel-Softmax. We state here again, that each dataset might bring different problem needing different solutions and that possibly rule-based approaches might come in handy.

5.7 On Data Manifold Closeness: Plausibility vs Missclassification

Both optimisation frameworks work towards primary goal, finding the nearest point on the other side of the decision boundary. However, how this point locates itself within the model domain differs in both cases. With Adversarial Examples for instance, with the goal being trickery and imperceptibility, the aim is to find unexplored areas of the model domain and exploit them i.e. an adversarial example should be close enough to data domain to remain identifiable by the true classifier but distant enough to be an adversarial.

This assertion might lead one to think that adversarial examples are dense in the set of real examples like the field of rationals Q in the fields of real numbers R, but as Ian Goodfellow points out, AEs lie more or less in linear subspaces i.e. actually the real examples lie close to linear decisions boundary[3], making them easy to cross to adversarial subspaces.

[3]This comes from the claim that most black box models especially Neural Networks are locally linear

With CFEs on the other hand, the condition on adherence to model domain is much more strict. From the plausibility standpoint, an explanation is actionable if there exists some similar instance to the counterfactual with favourable outcome (e.g. loan was accepted). Data Manifold Closeness in this case provides evidence to the feasibility of a CFE. CFE generating algorithm like FACE [50] or [87] introduce class density constraints to ensure the production of feasible CFEs. The optimisation problem is redefined as

$$\arg\min_{x_{cf}} d(x_{orig}, x_{cf}) \quad \text{such that} \quad m(x_{cf}) = y_{cf} \quad \text{and} \quad p_{y_{cf}}(x_{cf}) \geq \delta \tag{33}$$

where d is defined as in (2).
The x_{cf} is called a δ-plausible CFE.

5.8 On Transferability

5.8.1 Preliminaries

First we define some notions
Definition 1 (Transferability): We define transferability as the ability to use knowledge from a trained model to another different and potentially unknown model.
Definition 2 (Rashomon Effect): Rashomon effect is the term used to describe how a single event could be explained by multiple plausible contradictory explanations.
Defintion 3 (Predictive Multiplicity): Given a dataset $\mathcal{X} \in \mathrm{R}^d$. predictive multiplicity refers to the existence of conflicting predictions given by a set $\mathcal{G} = \{g_1, \ldots, g_n\}$ of conflicting models (best fitting models most often optimising the same loss function)

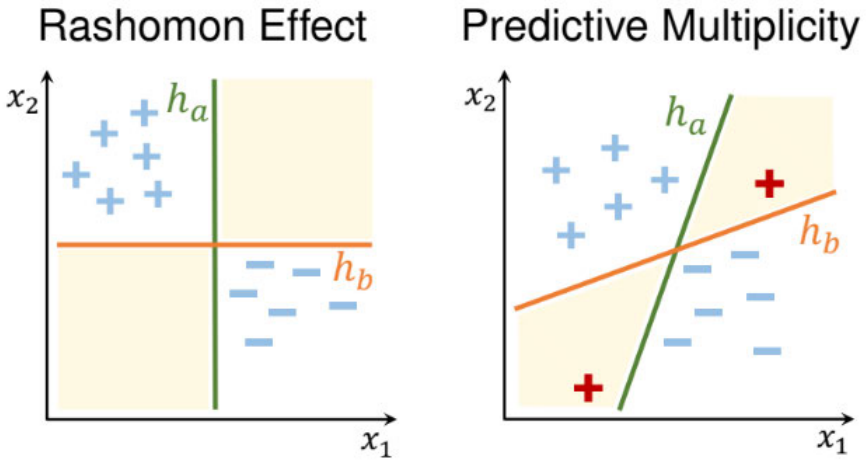


Figure 8: Image from cite [133]. On the left, h_a and h_b assign the same predictions on the training data but produce conflicting explanations of the importance of x_1 vs. x_2, as per the Rashomon effect. On the right, h_a and h_b assign conflicting predictions on the training data as per predictive multiplicity.

Definition 4 (Unfortunate Counterfactual Events(UCE) [145]): Let (x_{orig}, y_{orig}) be an instance an $m(\cdot)$ a classifier as defined above. We consider the following scenario

1. At time t_0, we train m and obtain m at time t_0, m_{t_0} such that $m_{t_0}(x_{orig}) = y_{orig}$. An algorithm CF is used to compute counterfactuals
2. At time $t_1 > t_0$, the CFE x_{cf} for x_{orig} is generated and $m_{t_0}(x_{cf}) = y_{cf}$.
3. At time $t_2 > t_1$. m_{t_0} is retrained and redeployed such that now we have m_{t_2} such that $m_{t_0} \neq m_{t_2}$.

Then, if there exists a time $t^* \geq t_2$ such that for some x at t^*, x_{t^*}, $x_{t^*} = \text{CF}(x_{orig}, y_{cf})$ and $m_{t_2}(x_{t^*}) = m_{t_0}(x_{orig}) = y_{orig}$, then we say that an "unfortunate counterfactual event" relative to x_{orig} has happened and that $\text{CF}(x_{orig}, y_{cf})$ has occurred.
Definition 5 (Adversarial Subspace): Contrary to what was formally believed, AE do not mostly exist in small pockets, rather in large, contiguous spaces [131, 93]. These spaces spanned by AEs are Adversarial Subspaces
Definition 6 (Adversarial Direction): Direction induced by Adversarial Perturbation
Definition 7 (Inter-Class Boundary): Given (two) similar models, the inter-boundary distance is the distance between their respective boundaries in a given direction.

5.8.2 Transferability of CFEs

The work from Breiman (2001) on multiplicity [132] makes us put to question the validity of CFEs. In his words, *"if one can fit multiple competing models - each of which provides a different explanation of the data-generating process-how can we tell which explanation is correct?"* Based on this statement, the assumption that a counterfactual is valid under a single model is dangerous, given the risk of predictive multiplicity. This translates to, for a given CFE to be transferable predictive multiplicity should be minimal i.e. all existing similar models should output same result.

[133] argues that sparse CFEs are very less often transferable i.e. the minimality of the l_p-norm and loss function when generating CFEs, exposes them to predictive multiplicity. Whereas δ-plausible counterfactuals or Data Supported Counterfactuals are more transferable even though the cost is higher. For instance, consider a two class classification case, and classifiers $f, g : \mathcal{X} \to \{-1, +1\}$. We seek to move instances from negative class to positive class. We define the ***cost of negative surprises*** as the measure of a positive CFE being negatively classified by a similar model. The objective is for this cost to be 0.

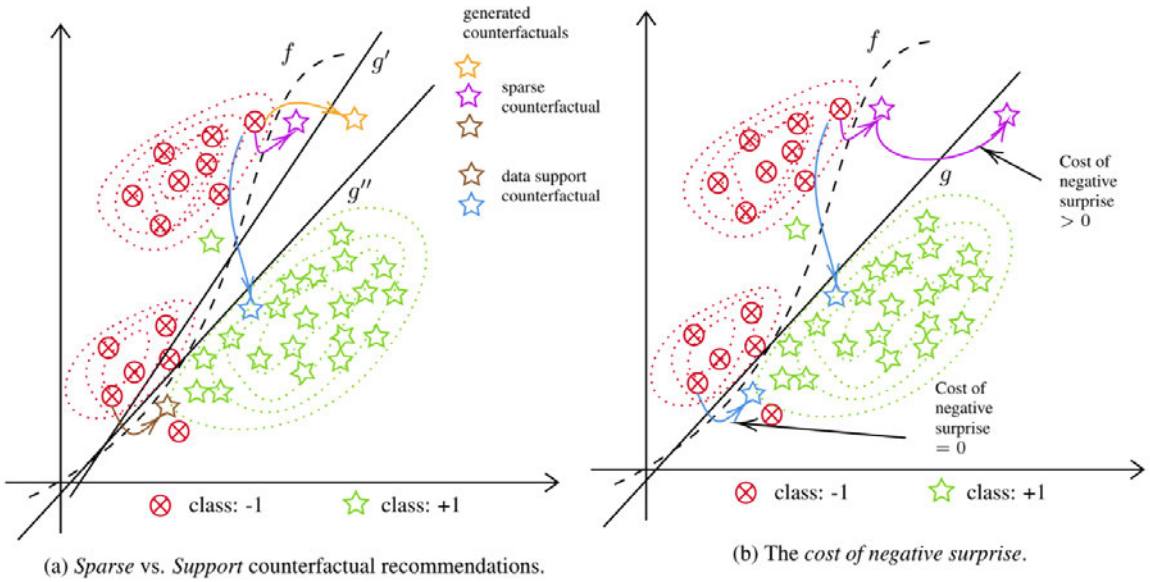


Figure 9: Image from [138]. (a) The cost of generating a sparse (close) solution is higher than generating a data support counterfactual (b) when the CFE stays substantially within data domainn it is less likely to be predicted negative by any given similar model

Even though, δ-plausible counterfactuals are transferable (within similar models), in the occurrence of UCEs, [91] recommends data augmentation with the previously generated CFEs that is, training the other model with the original data augmented with CFEs.

5.8.3 Transferability of AEs

AEs often transfer i.e. if an AE fools a model it should be enough to fool another. On why they transfer? Works from [141] show that for distinct models, the higher the dimensionality of their adversarial subspaces the more they intersect infering that dimensionlity of adversarial subspaces is directly proportional to transferability of adversarial examples. Moreover, studies from [141] indicate that the boundaries of subspaces in similar models lie at similar distances from legitimate data points in adversarial directions. This also comes to add to the fact that the average distance from these legitimate points to the decision boundaries of each model is greater than the *inter-boundary distance* making it easier for adversarials from a source model easily transferable to a target model.

Model-Agnostic Perturbations. Let's consider an instance (x_{orig}, y_{orig}). A model-agnostic perturbation r is given by:

$$r = -\epsilon \cdot y_{orig} \cdot \hat{\delta} \tag{34}$$

where, $\hat{\delta}$ is the unit vector of the difference in class means δ given by:

$$\delta = \frac{1}{2} \cdot (\mathrm{E}_{\mu_{+1}}[x_{orig}] - \mathrm{E}_{\mu_{-1}}[x_{orig}]) \tag{35}$$

with $\mathrm{E}_{\mu_{+1}}$ being the mean of the positive class and $\mathrm{E}_{\mu_{-1}}$ the mean of the negative class.
A **sufficient condition** for transferability is that the perturbation the feature space stays

closely aligned to the difference in intra class means. So in theory, every perturbation made in the direction of intra-class means should be transferable. Well, counter-examples do exist. In [141]'s experiment, using the MNIST dataset, the perturbed instances of the handwritten digits had faint presence of other digits eventhough the correct labels were apparent. These pertubations were easily transferable accross DNNs, logistic Regression and Quadratic models but were not able to fully deceive CNN.

5.8.4 The difference

Once again, transferability in both cases (CFE and AE) appear similar, but the difference boils down model similarity and model domain constraints. In CFEs, transferability is more possible within highly similar models, same for AE but the constraint on CFEs is harder. With CFEs, the requirement is that for transferability, the CFE lies substantially within the target class probability distribution, while the sufficient condition for transferability of AE (in most cases) is that the instance is perturbed in the direction of the intra-class mean. Much more differences could still be found but the ones stated above define a clear ridge between transferability for both notions.

6 Machine Learning Algorithms

In this section, we give a brief description of the Machine Learning Algorithms used for our experiments.

6.1 Artificial Neural Networks (ANN)

With similarities to the human brain, Artificial Neural Networks (ANN) is a machine learning framework, consisting of interconnected consecutive and successive layers of neurons. A standard neural network consists of an input layer, an output layer hidden layers. A neuron from a previous layer is connected to all neurons of the following layer through weighted edges and all neurons of the previous layer are connected to a neuron in the next layer. The weighted values of these neurons are summed up and are then fed forward to the activation function σ (counld be sigmoid, hyperbolic tangent etc...) of the next layer and the process goes in till the output layer is reached. A bias value b can be added to the weighted input values of a neuron, allowing the activation function to be shifted during training. The output of the j-th neuron of a layer l can be described as follows:

$$a_j^l = \sigma^l \left(\sum_k w_{jk}^l a_k^{l-1} + b_k^l \right) \tag{36}$$

where w_{jk}^l is the weighted edge connecting the kth neuron in the $l-1th$ layer to the $j-th$ neuron in the lth neuron, a_k^{l-1} is the k-th output of the $l-1$-th layer. So suppose a neural network g and L the output layer, the overall structure is given by

$$g = \sigma^L(\sigma^{L-1}(\dots\sigma^3(\sigma^2(a^1))\dots)) \tag{37}$$

where a^1 is the output vector of the first layer and we consider

$$a^l = \sigma^l(W \cdot a^{l-1} + b^l) \tag{38}$$

with W, the weight matrix.

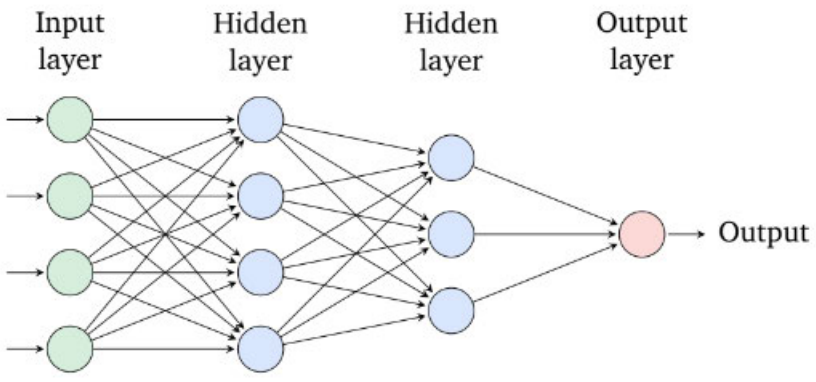


Figure 10: Figure from [127] of an ANN with two hidden layers, one input layer and one output layer with output to one neuron

.

These weights and biases are trained to minimize a loss function by gradient descent through a process of backpropagation. Backpropagation because, given the local losses, weights and biases of the different layers the gradient of the cost function is computed backwards that is. the attributes of the next layer is needed to train the weights of the previous layer. However, even though we might try to approximate an ANN's behaviour, ANNs are black box by nature i.e. we don't really know what accounts for specific aspects of classification. But we can approximate it's predictive tendencies with Counterfactual Explanations.

6.2 Robust Soft Learning Vector Quantization (RSLVQ)

RSLVQ by Seo et al. is a variant of **LVQ** classification model by Kohonen et al. based on Nearest Prototype Classification (NPC) with focus on maximizing correct classification under the assumption of an underlying Gaussian distribution.
Given $\{(x_n, y_n)|x_n \in \mathcal{X}, y_n \in \mathcal{Y}\}$, where $\mathcal{X}$ is a set of n_d d-dimensional data points and $\mathcal{Y}$ the set of labels of points of $\mathcal{X}$, RSLVQ classifies by maximizing of the following likelihood ratio:

$$L_r = \prod_{k=1}^{n} \frac{p(x_k, y_k \,|\, \mathcal{T})}{p(x_k \,|\, \mathcal{T})} \tag{39}$$

or for computational facility the $logL_r$

$$\log L_r = \sum_{k=1}^{n} \log \frac{p(x_k, y_k \,|\, \mathcal{T})}{p(x_k \,|\, \mathcal{T})} \stackrel{!}{=} \max \tag{40}$$

Where $0 \leq \frac{p(x_k, y_k \,|\, \mathcal{T})}{p(x_k \,|\, \mathcal{T})} \leq 1$. Here $\mathcal{T} = \{(\theta_j, c_j)\}_{j=1}^{M}$ is the set of prototypes θ_j with class c_j, $p(x_k, y_k \,|\, \mathcal{T})$ is the probability of correctly classifying x_k and $p(x_k \,|\, \mathcal{T})$ is the probability of correctly or incorrectly classifying x_k. We assume the probability p a Gaussian density.
Optimisation of the prototypes is done by Gradient Ascent on $logL_r$.

$$\theta_l(t+1) = \theta_l(t) + \alpha(t) \frac{\partial}{\partial \theta_l} \log \frac{p(x, y \mid \mathcal{T})}{p(x \mid \mathcal{T})} \tag{41}$$

We obtain the learning rule.

$$\begin{aligned} \theta_l(t+l) &= \theta_l(t) + \alpha(t) \begin{cases} (P_y(l \,|\, x) - P(l \,|\, x)) \frac{\partial f(x, \theta_l)}{\partial \theta_l}, & \text{if } c_l = y \\ -P(l \,|\, x)) \frac{\partial f(x, \theta_l)}{\partial \theta_l}, & \text{if } c_l \neq y \end{cases} \\ &= \theta_l(t) + \alpha(t) \begin{cases} (P_y(l \,|\, x) - P(l \,|\, x)) \frac{(x - \theta_l)}{\sigma^2}, & \text{if } c_l = y \\ -P(l \,|\, x) \frac{(x - \theta_l)}{\sigma^2}, & \text{if } c_l \neq y \end{cases} \end{aligned} \tag{42}$$

Where

$$\begin{aligned} P_y(l \,|\, x) &= \frac{p(l) \exp(f(x, \theta_l))}{\sum_{\{j:\, c_j = y\}} p(j) \exp(f(x, \theta_j))} \\ P(l \,|\, x) &= \frac{p(l) \exp(f(x, \theta_l))}{\sum_{j=1}^{M} p(j) \exp(f(x, \theta_j))} \end{aligned} \tag{43}$$

$P_y(l|x)$ describes the posterior probability that the data point x is assigned to the class l correctly classified.$P_y(l|x)$ describes the (posterior) probability that the data point x is assigned to the class l.
We assume $f(x, \theta_j) = \frac{-(x-\theta_j)^2}{\sigma^2}$ and $p(l) = \frac{1}{M}$ for each class, where M is the number of classes and σ, the width of the distribution.

6.3 Support Vector Machines (SVM)

The idea of a decision boundary or rather a hyperplane (or in 2D a line) separating different classes comes convenient when it comes to classification. But, for a classification case, there may exists infinitely many decision boundaries. So how do we choose a "best" boundary? Randomly choosing may cause extreme cases of missclassification. SVM, mainly used for classification but also used for regression, aims to find the best decision boundary such that the boundary width is maximised i.e. the decision boundary, where the distance between the points of opposite (different) classes, closest to the separating hyperplane (Support Vectors) is maximised.

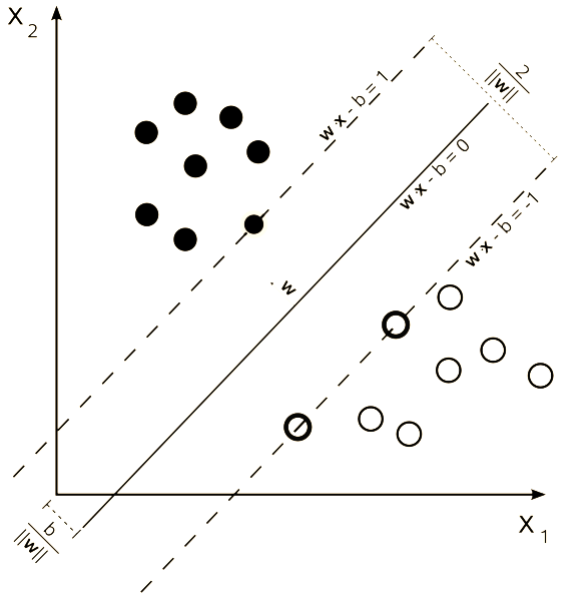


Figure 11: Figure representing a two class classification case with two feature. The points on the lower and upper dashed lines are the so-called support vectors. These dashed lines (margins) are of maximum width $\frac{2}{\|w\|}$

. This is done by optimising the parameters (weights corresponding to the features) such that we attain maximum boundary width. We try to mimimize

$$\frac{1}{2}\|w\|^2 \text{ such that } y_i(x_i \cdot w + b) \geq 1, \tag{44}$$

where, $w \in \mathrm{R}^d$ is the weight vector, b the bias and (x_i, y_i) a data point and it's label. 44

could be reformulated using Lagrange constraints. We minimize

$$\mathcal{L}_{\mathcal{G}} = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \left[y_i(x_i \cdot w + b) - 1\right], \tag{45}$$

where, $\alpha_i \geq 0$ are Lagrange multipliers.

By setting the derivatives of L_G w.r.t each parameter equals to 0, we then obtain the extended dual problem. (Proof in Appendix)

$$\mathcal{L}_{\mathcal{D}} = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} y_i, y_j \alpha_i \alpha_j \langle x_i, x_j \rangle, \tag{46}$$

So far, SVM looks good for linear separable or near linear separable cass. What about non-linear separable cases? Here comes the idea of kernels, projecting a linear plane unto a non-linear surface while conserving dimensions. For instance, a radial kernel for circular boundaries.

7 Experiments and Results

7.1 Datasets

To detailly (at least to some level) explore the differences between CFEs and AEs, we consider a diverse set of datasets i.e. tabular data and image data with different properties (absence or presence of categorical values, number of clusters per class, dimensionality etc. Below is a table with a brief description of the datasets used for experiment.

Datasets	n_samples	n_features	n_classes	n_cat	Source
SyntheticData1	409	3	4	0	HS-Mittweida (ML course 2022/2023)
Breast Cancer Dataset	569	30	2	0	`www.kaggle.com/datasets`
Adult Income Dataset	48842	14	2	13	`www.kaggle.com/datasets`
Dim100 Dataset	100	200	2	0	Artificial
Dim1000 Dataset	200	1000	2	0	Artificial
Dim10000 Dataset	200	10000	2	0	Artificial
MNISTS Dataset	100	784	2	0	Tensorflow Datasets

Categorical columns were one-hot-encoded or ordinally encoded, for all but the Adult Income Dataset and the MNISTs dataset. the numerical columns were scaled to fit the interval $[0, 1]$ for computational ease. And after preprocessing and basic data cleaning, we obtain the following.

Datasets	n_samples	n_features	n_cat
SyntheticData1	409	3	0
Breast Cancer Dataset	569	30	0
Adult Income Dataset	41292	17	17
Dim100 Dataset	100	200	0
Dim1000 Dataset	200	1000	0
Dim10000 Dataset	200	10000	0
MNISTS Dataset	100	784	0

7.2 Classifiers, Parameters and Optimization

As classifiers, we considered the models described in section 5 above. To better differentiate between cases and for computational ease and efficiency, different models were considered for

different datasets. The classifiers RSLVQ, SVM[4], a simple MLP[5] were used for the tabular data. Convolutional Neural Networks was implemented on the MNIST dataset. The stated classifiers were optimised according to 1 and their accuracies and F1 Scores were calculated.

Table 1: Datasets, Classifiers and their Hyperparameters

Datasets	Classsifiers	Hyperparameters
SyntheticData1	RSLVQ	Number of Prototypes per Class: 3 sigma = 1 learning rate = 0.5 number of iterations: 100
	SVM	kernel = Polynomial
	MLP	Number of neurons on Input Layer: 3 Number of hidden layers: 6 Number of Neurons per hidden layer:5 Random state: 42 Learning rate: 0.05
Breast Cancer Dataset	RSLVQ	Number of Prototypes per Class: 1 sigma = 1 learning rate = 0.05 number of iterations: 100
	SVM	kernel = Linear
	MLP	Number of neurons on Input Layer: 30 Number of hidden layers: 6 Number of Neurons per hidden layer:9 Random state: 42 Learning rate: 0.01
Adult Income Dataset	RSLVQ	Number of Prototypes per Class: 1 sigma = 1 learning rate = 0.05 number of iterations: 100
	SVM	kernel = Linear
	MLP	Number of neurons on Input Layer: 3 Number of hidden layers: 6 Number of Neurons per hidden layer:6 Random state: 42 Learning rate: 0.05
Dim10 Dataset	RSLVQ	Number of Prototypes per Class: 10 sigma = 1 learning rate = 0.05 number of iterations: 100
	SVM	kernel = Linear
	MLP	Number of neurons on Input Layer: 10 Number of hidden layers: 6 Number of Neurons per hidden layer:6 Random state: 42 Learning rate: 0.05

Continued on next page

[4] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[5] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

Table 1 – continued from previous page

Datasets	Classsifiers	Hyperparameters
Dim100 Dataset	RSLVQ	Number of Prototypes per Class: 1 sigma = 1 learning rate = 0.05 number of iterations: 100
	SVM	kernel = Linear
	MLP	Number of neurons on Input Layer: 100 Number of hidden layers: 100 Number of Neurons per hidden layer: - Random state: 42 Learning rate: 0.05
Dim1000 Dataset	RSLVQ	Number of Prototypes per Class: 10 sigma = 1 learning rate = 0.05 number of iterations: 100
	SVM	kernel = Linear
	MLP	Number of neurons on Input Layer: 1000 Number of hidden layers: 150 Number of Neurons per hidden layer: - Random state: 42 Learning rate: 0.05
Dim10000 Dataset	RSLVQ	Number of Prototypes per Class: 15 sigma = 1 learning rate = 0.05 number of iterations: 100
	SVM	kernel = Linear
	MLP	Number of neurons on Input Layer: 10000 Number of hidden layers: 100 Number of Neurons per hidden layer: - Random state: 42 Learning rate: 0.05
MNIST Dataset	CNN	Number of neurons on Input Layer: 784 Number of hidden layers: 100 Number of Neurons per hidden layer: -

7.3 Metrics

To evaluate classification performance, the accuracy, recall and F1 scores were measured. All datasets were split into a 70:30 ratio, the models were trained on 70% of the datasets and tested on 30%. Accuracy on test set is the number of correctly predicted number divided by the total number of elements in the test set.

$$\text{Accuracy} = \frac{TP}{TP + FP} \tag{47}$$

where, TP are the true positives (correctly predicted) and FP, false positives (the incorrectly predicted). And,

$$\text{Recall} = \frac{TP}{TP + FN} \tag{48}$$

With FN being the false negatives. This is relevant to identify if the model in fact does correctly predict with respect to each class and to avoid bias in case of class inbalance. And our F1 score:

$$2 \cdot \frac{\text{Accuracy} \cdot \text{Recall}}{\text{Accuracy} + \text{Recall}} \tag{49}$$

To measure the success rate of our adversarial attacks, we use the Attack Success Rate (ASR) given by:

$$\text{ASR} = \frac{\text{Number of Successfully attacked Samples}}{\text{Number of Samples}} \tag{50}$$

Also, to evaluate the robustness of CFEs, we calculate the percentage of stable counterfactuals with respect to all counterfactuals generated. In this case, a stable counterfactual as defined as per Section 5.3 above.

7.4 CFE and AE Generation Settings

For CFE generation we proceeded considering two methodologies: (1) CFEs should be classified by a minimum of 90% (predicitve power) and (2) CFEs should be generated such that they are within a 90% probabibility of being within data domain (model domain closeness). Counterfactual Explanations produced by **CLEAR** were produced for tabular data such that their CFEs were classified with atleast 90% probability. Plausible CFEs were generated by **DICE** such that they exist substantially within counterfactual class domain. To do this, we ensured that the probability of classification for the target class was atleast 90 - 95% and the density measured using a GMM metric. In the case of RSLVQ, to measure this probability, we used the softmax function of the distance to closest prototypes. We also ensure meaningful distance to nearest neighbours of counterfactual by fitting some must-hold proximity value to class mean. Synthetic data was generated through univariate sampling[6] over the feature columns depending on their distributions. Categorical distances were measured with Manhattan distance or L0-norm on column values and the distance as defined per Wachter et al. for numerical columns. As internal explainable model, we tested over SVM, Logistic Regression, or Polynomial Regression. In most cases, Logistic regression was used on the balanced neighbourhood of atleast 100 instances around the original input. Even though DICE is originally defined as 'gradient-based', an agnostic version was made by Microsoft[7]. Slightly modified so it fits RSLVQ. Only DICE was used for image data with the same settings as in tabular data. CFE stability was evaluated with an $l\infty$-norm constraint on the CFEs of similar instances i.e.

$$\|x_{cf} - x_{cf}^{'}\|_\infty \leq \|x - x^{'}\|_\infty + \alpha, \quad \alpha \in \mathrm{R}$$

Here α is considered the bias indicator, where the greater α the more biased the model.
For the **Boundary Attack** and the **HopSkipJumpAttack**, instances were perturbed with the simple L2 norm constraint such that the norm of the perturbation vector didn't exceed

[6] univariate sampling often leads to loss of correlation and/or causality, but in our case minimal enough for us to ignore

[7] `https://github.com/interpretml/DiCE`

10^{-1}. We use **Boundary Attack** and **FGSM** attacks from the ART library[8] with it's already predefined settings for image data.

7.5 Results

7.5.1 Two Class Datasets

We consider here, the binary classification cases. First, the predictive and attack efficiency of the models were evaluated using the above defined scores and the results are as seen in Table 2.

	Breast Cancer			Adult Income Dataset		
	RSLVQ	**SVM**	**MLP**	**RSLVQ**	**SVM**	**MLP**
Accuracy	94.74%	98.25%	98.25%	76.92%	80.71	82.31%
Recall	98.15%	99.07%	99.01%	48.60%	56.20%	58.20%
F1 Score	95.92%	98.62%	98.54%	55.86%	61.00%	62.80%
Bound. ASR	15.29%	5.10%	6.34%	52.00%	16.30%	15.90%
HSJA ASR	4.75%	2.82%	6.15%	33.70%	28.10%	25.10%

Table 2: Accuracies and Attack Success Rate on the Binary Class Datasets

Of particular interest is determining the size of perturbation from moving from positive class to negative class and vice-versa, in our case, the average l2-norm of the difference between perturbed instances and original instances. For this, we introduce the following scores:

- **Average Cost from Positive to Negative (ACPN)**

$$\text{ACPN} = \frac{1}{|C_{+1}|} \sum_{x \in C_{+1}} \|x - x_{per}\|_2 \tag{51}$$

  where, x_{per} is the perturbed instance (explanation or adversarial).

- **Average Cost from Negative to Positive (ACNP)**

$$\text{ACNP} = \frac{1}{|C_{-1}|} \sum_{x \in C_{-1}} \|x - x_{per}\|_2 \tag{52}$$

For CFE and AE generation we considered, a 1000 samples from the adult data set.

[8] `https://adversarial-robustness-toolbox.readthedocs.io/en/latest/modules/attacks/evasion.html`

		Breast Cancer			Adult Income Dataset		
		RSLVQ	**SVM**	**MLP**	**RSLVQ**	**SVM**	**MLP**
CFEs	Clear ACNP	0.72×10^{1}	1.50×10^{1}	1.49×10^{1}	0.57×10^{1}	0.50×10^{1}	0.91×10^{1}
	Clear ACPN	0.93×10^{1}	1.19×10^{1}	1.24×10^{1}	1.19×10^{1}	1.22×10^{1}	2.07×10^{1}
	Dice ACNP	7.82×10^{1}	8.16×10^{1}	8.52×10^{1}	0.32×10^{1}	0.27×10^{1}	0.24×10^{1}
	Dice ACPN	4.18×10^{2}	4.42×10^{2}	5.03×10^{2}	0.28×10^{1}	0.22×10^{1}	0.20×10^{1}
		RSLVQ	**SVM**	**MLP**	**RSLVQ**	**SVM**	**MLP**
AE	Bound. ACNP	8.2×10^{-2}	1.3×10^{-1}	9.4×10^{-2}	5.2×10^{-2}	1.22×10^{-1}	1.10×10^{-1}
	Bound. ACPN	2.0×10^{-2}	5.8×10^{-2}	8.7×10^{-2}	4.2×10^{-2}	1.07×10^{-1}	1.04×10^{-1}
	HSJA ACNP	1.6×10^{-2}	1.2×10^{-2}	6.1×10^{-2}	3.5×10^{-5}	2.5×10^{-3}	2.4×10^{-3}
	HSJA ACPN	2.1×10^{-2}	1.0×10^{-1}	1.1×10^{-1}	3.0×10^{-2}	4.7×10^{-2}	4.4×10^{-2}

Table 3: Average L2 norm on perturbations for CFEs and AEs

To measure the magnitude of perturbation of categorical variables, we consider the above stated scores with regards to the L0 norm.

		Adult Income Dataset		
		RSLVQ	**SVM**	**MLP**
CFEs	Clear ACNP	3.02	2.00	4.38
	Clear ACPN	5.86	2.66	3.33
	Dice ACNP	1.41	1.67	1.81
	Dice ACPN	1.68	1.54	1.54
		RSLVQ	**SVM**	**MLP**
AE	Bound. ACNP	17.00	16.97	16.89
	Bound. ACPN	17.00	16.97	16.66
	HSJA ACNP	6.11	6.09	6.29
	HSJA ACPN	16.97	14.83	14.50

Table 4: Average L0 norm on perturbations for CFEs and AEs

From the CFEs and AEs generated, the transferability from model to model was investigated.

		Breast Cancer			Adult Income Dataset		
		RSLVQ	**SVM**	**MLP**	**RSLVQ**	**SVM**	**MLP**
CLEAR CFEs	**RSLVQ**	*	0.65%	0.66%	*	0.12%	0.19%
	SVM	0.75%	*	1.09%	0.88%	*	1.07%
	MLP	0.74%	1.09%	*	0.94%	1.14%	*
DICE CFEs	**RSLVQ**	*	36.41%	66.89%		81.56%	67.49%
	SVM	57.29%	*	67.49%	47.30%	*	00.00%
	MLP	58.00%	75.04%	*	43.80%	50.40%	*
		RSLVQ	**SVM**	**MLP**	**RSLVQ**	**SVM**	**MLP**
BOUND. AEs	**RSLVQ**	*	5.41%	8.11%	*	21.00 %	19.19%
	SVM	31.11%	*	33.33%	48.57%	*	33.33%
	MLP	33.33%	21.17%	*	44.36%	41.11%	*
HSJA AEs	**RSLVQ**	*	0.00%	11.11%	*	29.83%	29.60%
	SVM	61.54%	*	38.46%	40.00%	*	40.00%
	MLP	44.12%	29.41%	*	44.44%	77.78%	*

Table 5: Model Transferability rate for CFEs and AEs

7.5.2 Multi-Class Dataset

As per usual, we calculate the accuracy, recall, F1 score and ASR for SyntheticData1

	SyntheticData1		
	RSLVQ	**SVM**	**MLP**
Accuracy	89.04%	86.15 %	92.00%
Recall	81.30%	82.93%	86.99%
F1 Score	82.20%	83.50%	86.99%
Bound. ASR	34.72%	47.19%	42.54%
HSJA ASR	43.03%	71.12%	52.32%

Table 6: Accuracies and ASR for SyntheticData1

However, contrary to the binary classification case, the Score introduced is just the average l2-norm overall all inter-class perturbations i.e. the average cost of a point crossing to every class for every point.

$$\text{Score} = \frac{1}{n_c} \sum_{j=1}^{n_c} \frac{1}{N_j} \sum_{i=1}^{N_j} \|x_i - x_{per}^j\|_2 \tag{53}$$

where, N_j is the number of samples of class j, x_{per}^j is the perturbated instance in class j, n_c is the number of classes.

	SyntheticData1		
	RSLVQ	**SVM**	**MLP**
Clear Score	1.11×10^{1}	0.57×10^{1}	1.10×10^{1}
Dice Score	0.19×10^{1}	0.39×10^{1}	0.30×10^{1}
Boundary Score	4.4×10^{-2}	5.2×10^{-2}	8.3×10^{-2}
HSJA. Score	7.3×10^{-2}	5.8×10^{-2}	8.0×10^{-2}

Table 7: CFE and AE Scores for SyntheticData1 over RSLVQ, SVM and MLP

and the transfer rates

		SyntheticData1		
		RSLVQ	**SVM**	**MLP**
CLEAR CFEs	**RSLVQ**	*	0.38%	0.12%
	SVM	1.49%	*	1.66%
	MLP	0.74%	1.09%	*
DICE CFEs	**RSLVQ**	*	68.01%	41.58%
	SVM	41.16%	*	34.39%
	MLP	55.26%	49.96%	*
		RSLVQ	**SVM**	**MLP**
BOUND. AEs	**RSLVQ**	*	55.66%	35.75%
	SVM	44.64%	*	42.86%
	MLP	41.56%	43.72%	*
HSJA AEs	**RSLVQ**	*	59.66%	51.71%
	SVM	33.33%	*	48.07%
	MLP	51.61%	54.84%	*

Table 8: Model Transferability rate for CFEs and AEs

7.5.3 Artificial Created Datasets of Varying Dimensions

To investigate the generation of CFEs and AEs with respect to increasing dimensionality, for the respective datasets we calculate, the adversarial perturbation ASR and the percentage of stable counterfactuals.

	Dim10			Dim100		
	RSLVQ	**SVM**	**MLP**	**RSLVQ**	**SVM**	**MLP**
Accuracy	90.00%	90.00%	83.67%	96.67%	90.00%	86.67%
Recall	88.00%	99.07%	99.01%	96.77%	86.96%	76.93%
F1 Score	88.89%	98.62%	98.54%	93.75%	78.57%	86.62%
	Dim1000			Dim10000		
	RSLVQ	**SVM**	**MLP**	**RSLVQ**	**SVM**	**MLP**
Accuracy	70.00%	73.34%	70.00%	60.00%	80.71	82.31%
Recall	50.00%	73.34%	60.00%	50.00%	56.20%	58.20%
F1 Score	50.00%	73.34%	66.67%	50.00%	61.00%	62.80%

Table 9: Accuracies and Attack Success Rates

	Dim10		Dim100	
	SVM	**MLP**	**SVM**	**MLP**
Dice Stable CFEs	00.00%	00.00%	45.00%	44.00%
HSJA ASR	41.00%	91.00%	24.00%	27.00%
Average HSJA L∞ distance	6.2×10^{-1}	6.8×10^{-2}	4.5×10^{-2}	8.8×10^{-3}
	Dim1000		Dim10000	
	SVM	**MLP**	**SVM**	**MLP**
Dice Score	39.00%	21.00%	11.00%	9.00%
HSJA ASR	29.00%	56.00%	52.00%	66.00%
Average HSJA L∞ distance	8.5×10^{-3}	7.4×10^{-2}	5.9×10^{-5}	1.6×10^{-5}

Table 10: Percentage of stable CFEs vs attack Success Rate for increasing Dimensions

7.5.4 MNISTS Dataset

We trained a Convolutional Neural Network on 80% of the data and tested our accuracy and ASR on the test set.

	MNISTS
	CNN
Accuracy	92.39%
Bound. ASR	84.00%
Bound. Average Pertubation	1.9×10^{-1}
FGSM ASR	91.00%
FGSM Average Pertubation	2.0×10^{-2}

Table 11: Accuracies and ASR for CNN performed on MNISTs

7.6 Observations

We summarize the above results:

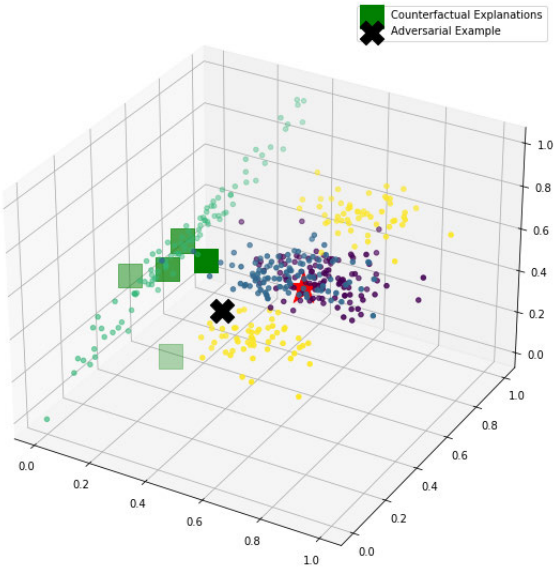


Figure 12: Counterfatuals (green squares) produced by DICE for the red point (star). And the Adversarial Example from HSJA (black cross) for SyntheticData1. Even though the CFEs and the AE are of the same class, AE ignores class manifold restrictions and lie closer to the original point

Higher Recall improves Robustness, but CFEs do not care: For all instances, there exist an alternative, but not always an ϵ-alternative. For all datasets, irrespective of the nature of their decision boundaries, CFE production rate was 100%, not making it a plausible CFE or stable CFE, but a CFE nonetheless. However, the higher the recall, the harder it was to produce AEs of order less than 10^{-1}.

The cost of AEs generation is considerably lower than that of CFEs: Our initial argument was the constraint of proximity on AEs being stricter than that on CFEs. We observe from the overall performance of the CLEAR, DICE, Boundary Attack, HopSkipJumpAttack, an overwhelming difference in their respective l2 norms noting average order of magnitude of 10^{-2} for AEs and 10^{1} for CFEs, with these norms being directly proportional (in general) to the precision and recall of the models.

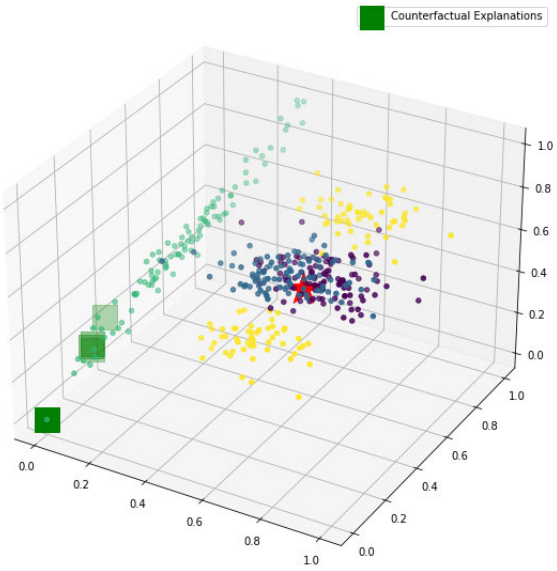


Figure 13: Counterfatuals (green squares) produced by CLEAR for the red point (star). Predicted with atleast 90% but do not lie in area of high density

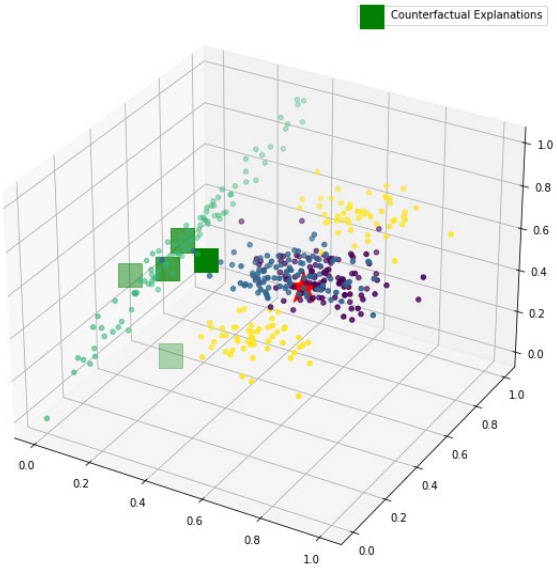


Figure 14: Counterfatuals (green squares) produced by DiCE for the red point (star). Predicted with atleast 90% and lie in area of high density

A high Within-Class-Density ensures high transferability: Of highest l2 norms are plausible CFEs from DiCE as the stricter the constraint of model closeness the higher the cost of CFE generation. This results translates into transferability. Even though, transferability in AEs is more common and somewhat stable, Plausible CFEs are more transferable. There comes a certain ignorance of decision boundary of target model during transfer when the

CFE is dense within class domain. CFEs and AEs from higher precision models tend to be more transferable to models of lower precision.Contrary to our previous assertions, similarity in decision boundaries doesn't play as much of a role in transferability for plausible CFEs as much as it did for AEs. The boundaries given by SVM and RSLVQ throughout our experiments appeared to be somewhat different but yet the transferability rate for plausible CFEs was similar.

High Cost but fewer changes vs Low cost but more changes: To make sense of the sparse CFEs vs imperceptible AEs' argument, we considered a dataset of categorical values to evaluate the average number of perturbed features using the l0 norm. Our Observation: imperceptibility demands perturbation on (almost) all features whereas sparsity doesn't. From the plausible CFEs (taking into account immutable features such as age) from DiCE, a maximum of two changes was recorded, which is the goal of plausibility, very different from AEs whose only goal is proximity.

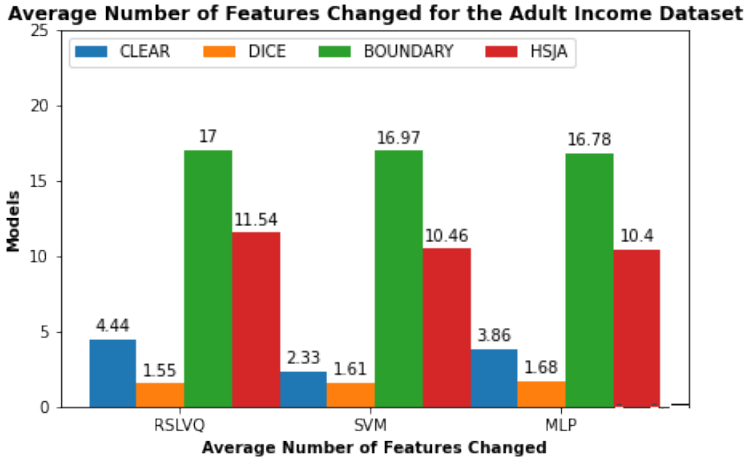


Figure 15: Average l0 norm of the perturbation vector for CLEAR, DICE, Boundary attack and HSJA on the adult dataset. CFEs generators (CLEAR and DICE) have maximum 4.44 changes in feature, AE changes almost every feature (17 out of 17).

Targeted Attacks are not CFEs : In a binary class case, there is no difference between a targeted and an untargeted attack, there is only one alternative class. SyntheticData1 has four classes, the average l2 norm of perturbations of the targeted attacks is still noticably less than that of CFEs both plausible or not. To further validate our argument, even though the values in transfer rate seem within same range for AEs and CFEs, we still fall to the same conclusion, plausible CFEs are more transferable than AEs.

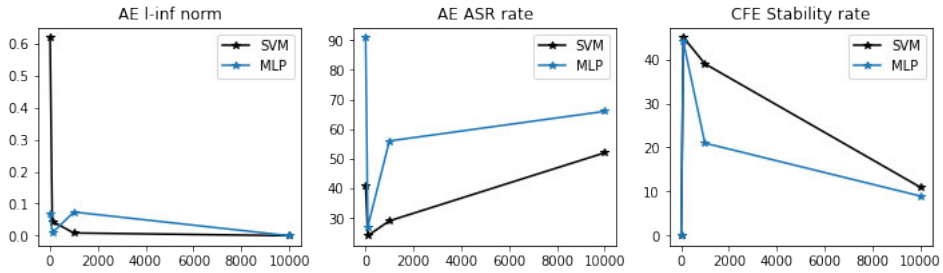


Figure 16: On the left is a comparison of l∞ norm of perturbation vectors of AEs with increasing dimensions. For incresaing dimensions there is a decrease in the norm. In the middle we see that the ASR is proportional to the dimensions with the exception of dimension 10 where ASR is actually at its highest. On the right, the higher the dimensions (in general), the lower the stability rate

Dimensionality favours AEs but handicaps CFEs: A rather inconclusive statement, as from experimentation arises some anomalies (and the size and nature if datasets were overly manipulated, so it is unwise to generalise). But from previous experiment, and from the results above, not only was computationally taxing to produce CFEs for increasing dimensions, but with increasing number of features came instability of CFEs and decrease in the l∞-norm of AE perturbation vectors. This doesn't really affect the nature of CFEs as CFE with regard to algorithmic recourse, CFEs do not work well with higher dimensions (a long explanation is not really an explanation, rather a complication). Concerning AEs, most works on dimensionality were mostly done with a dimensionality reduction approach on the same dataset, but in this paper, we focused more on rather similar datasets of different dimensions.

Mainstream CFE techniques are not very good with images: In theory, it should be simple to look for the closest point in a different class, but would we really call that a valid CFE without including properties like sparsity When experimenting with MNISTs handwritten digits dataset, while for some of the inputs, there were valid CFEs, most of them looked like Adversarial Example.

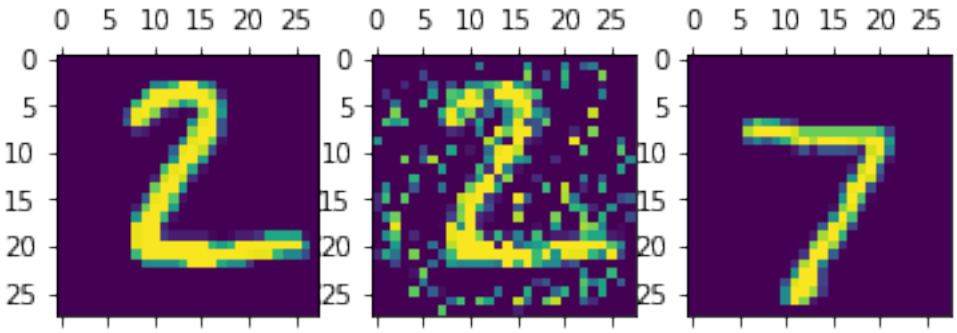


Figure 17: On the left the original input. A handwritten digit 2 classified as 2 by our ML classifier and human classifier. In the middle is a failure while attempting to produce a CFE of output 7. While the ML classifier classifies it as 7, the human eye does not agree. On the right is a successful attempt of producing a CFE of 7, where both ML and human classifier agree on output 7.

Boundary attacks and PGD achieved their goal of imperceptibility

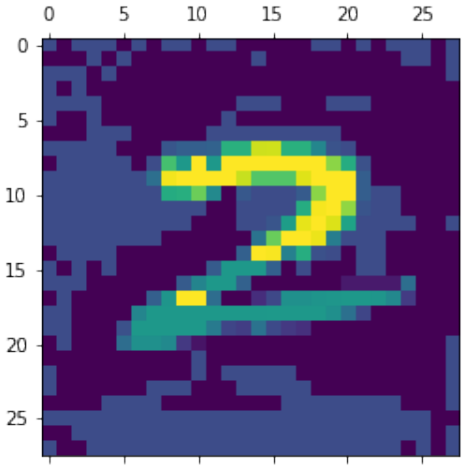


Figure 18: Successful attack on the handwritten digit 2. Seen by the human eye as 2 but classified by the ML clasifier as 7

8 Conclusion

8.1 Summary

This thesis explores the differences between Counterfactual Explanations and Adversarial Examples. Two similar concepts with similar framework with different objectives. The question as to if one was just a reformulation of the other was investigated. Our research in combination with previous work proposes that there is no reformulation rather the concepts are just not the same. CFEs highlight sparsity and class domain properties, properties which AEs ignore and aim only for imperceptibility.

We investigated the effects of domain constraints on transferability and how it favoured plausible CFEs, explored their behaviours in higher dimensions and their effectiveness in diverse datasets. From the results, their difference became clearer: the effect of rising dimensionality in CFEs is reversed in AEs, the sparsity vs imperceptibility argument with categorical data when the l0 norm was used, and the disfavour of CFE generating algorithms when images are involved.

8.2 Discussion and Future Work

So far, we explored some of the mathematical properties that differentiate Counterfactual Explanations and Adversarial Examples, trying to lay bare these differences through simple definitions and basic experimentations. While our results prove our aforementioned claims and theories, we admit that we laid some restrictions like limiting ourselves to just the l2-norm and l∞ norm to measure perturbations or using (for the most part) black box approaches to CFE or AE generation. What we aim is further investigation between these two concepts, so prospective works might entail the use of more generation methods and more complex datasets (especially those with non-linear boundaries), further research on dimensionality and test for stability for different $\alpha's$. Overall, just more testing to further ascertain our claims

References

[1] Miller, Tim. “Explanation in artificial intelligence: Insights from the social sciences.” arXiv Preprint arXiv:1706.07269. (2017).

[2] Kim, Been, Rajiv Khanna, and Oluwasanmi O. Koyejo. “Examples are not enough, learn to criticize! Criticism for interpretability.” Advances in Neural Information Processing Systems (2016).

[3] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4669963/

[4] https://en.wikipedia.org/wiki/Ignorability

[5] https://plato.stanford.edu/entries/causation-counterfactual/

[6] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti, Local rule-based explanations of black box decision systems (2018). arXiv:1805.10820.

[7] M. T. Ribeiro, S. Singh, C. Guestrin, Nothing else matters: Model-agnostic explanations by identifying prediction invariance (2016). arXiv:1611.05817.

[8] S. Krishnan, E. Wu, Palm: Machine learning explanations for iterative debugging, in: Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics, ACM, 2017, p. 4

[9] J. Krause, A. Perer, K. Ng, Interacting with predictions: Visual inspection of black-box machine learning models, in: CHI Conference on Human Factors in Computing Systems, ACM, 2016, pp. 5686–5697.

[10] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, 2017, pp. 4765–4774.

[11] A. Palczewska, J. Palczewski, R. M. Robinson, D. Neagu, Interpreting random forest classification models using a feature contribution method, in: Integration of Reusable Systems, Springer, 2014, pp. 193–218.

[12] SH. F. Tan, G. Hooker, M. T. Wells, Tree space prototypes: Another look at making tree ensembles interpretable (2016). arXiv:1611.07115.

[13] S. H. Welling, H. H. Refsgaard, P. B. Brockhoff, L. H. Clemmensen, Forest floor visualizations of random forests (2016). arXiv:1605.09196.

[14] P. Sollich, Probabilistic methods for support vector machines, in: Advances in neural information processing systems, 2000, pp. 349–355.

[15] B. Haasdonk, Feature space interpretation of SVMs with indefinite kernels, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (4) (2005) 482–492.

[16] L. Rosenbaum, G. Hinselmann, A. Jahn, A. Zell, Interpreting linear support vector machine models with heat map molecule coloring, Journal of Cheminformatics 3 (1) (2011) 11.

[17] W. Landecker, M. D. Thomure, L. M. Bettencourt, M. Mitchell, G. T. Kenyon, S. P. Brumby, Interpreting individual classifications of hierarchical networks, in: 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2013, pp. 32–38.

[18] M. G. Augasta, T. Kathirvalavakumar, Reverse engineering the neural networks for rule extraction in classification problems, Neural Processing Letters 35 (2) (2012) 131–150.

[19] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network (2015). arXiv: 1503.02531.

[20] J.Li, X. Chen, E. Hovy, D. Jurafsky, Visualizing and understanding neural models in NLP (2015). arXiv:1506.01066.

[21] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 427–436.

[22] T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net (2014). arXiv:1412.6806.

[23] A. Mahendran, A. Vedaldi, Understanding deep image representations by inverting them, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 5188– 5196.

[24] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer, 2014, pp. 818–833

[25] A. Henelius, K. Puolam¨aki, A. Ukkonen, Interpreting classifiers through attribute interactions in datasets (2017). arXiv:1707.07576.

[26] P. Dabkowski, Y. Gal, Real time image saliency for black box classifiers, in: Advances in Neural Information Processing Systems, 2017, pp. 6967–6976.

[27] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, 2017, pp. 4765–4774.

[28] Z. Che, S. Purushotham, R. Khemani, Y. Liu, Interpretable deep models for ICU outcome prediction, in: AMIA Annual Symposium Proceedings, Vol. 2016, American Medical Informatics Association, 2016, p. 371.

[29] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti, Local rule-based explanations of black box decision systems (2018). arXiv:1805.10820

[30] D. W. Apley, Visualizing the effects of predictor variables in black box supervised learning models (2016). arXiv:1612.08468.

[31] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer, 2014, pp. 818–833

[32] M. T. Ribeiro, S. Singh, C. Guestrin, Why should I trust you?: Explaining the predictions of any classifier, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1135–1144.

[33] Timo Freiesleben, Counterfactual Explanations & Adversarial Examples Common Grounds, Essential Differences, and Potential Transfers, 2020

[34] The Logic of Counterfactuals in Causal Inference (Discussion of 'Causal Inference without Counterfactuals' by A.P. Dawid) Judea Pearl

[35] https://yzhu.io/courses/core/reading/04.causality.pdf

[36] Alexander Balke, Judea Pearl 2004 :Counterfactual Probabilities: Computational Methods, Bounds and Applications https://arxiv.org/abs/1302.6784

[37] Statistical Mechanics and the Asymmetry of Counterfactual Dependence Adam Elga

[38] http://causality.cs.ucla.edu/blog/index.php/2014/11/29/on-the-first-law-of-causal-inference/

[39] https://youtu.be/Q6JbmGQstDM

[40] https://www.tandfonline.com/doi/abs/10.1080/00048409512346441

[41] Timo Freiesleben, The Intriguing Relation Between Counterfactual Explanations and Adversarial Examples, 2020

[42] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable Recourse in Linear Classification. In Proceedings of the Conference on Fairness, Accountability, and Transparency (FAccT) (FAT* '19). Association for Computing Machinery, New York, NY, USA, 10. https://doi.org/10.1145/3287560.3287566

[43] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. SSRN Electronic Journal 31, 2 (2017). https://doi.org/10.2139/ssrn.3063289

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

[45] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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 (KDD '16). Association for Computing Machinery, New York, NY, USA, 10. https://doi.org/10.1145/2939672.2939778

[46] Adam White and Artur d'Avila Garcez. 2019. Measurable Counterfactual Local Explanations for Any Classifier. http://arxiv.org/abs/1908.03020

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

[48] Michael T. Lash, Qihang Lin, William Nick Street, Jennifer G. Robinson, and Jeffrey W. Ohlmann. 2017. Generalized Inverse Classification. In SDM. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 162–170.

[49] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. 2019. CERTIFAI: Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models. http://arxiv.org/abs/1905.07857

[50] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. FACE: Feasible and Actionable Counterfactual Explanations. , 344–350 pages. https://doi.org/10.1145/3375627.3375850 arXiv: 1909.09369.

[51] Wenzhuo Yang, Jia Li, Caiming Xiong, Steven C.H. Hoi. 2022. MACE: An Efficient Model-Agnostic Framework for Counterfactual Explanation https://arxiv.org/abs/2205.15540

[52] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020. From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence 2, 1 (1 2020), 56–67.

[53] Emanuele Albini, Jason Long, Danial Dervovic, Daniele Magazzeni. (2022). Counterfactual Shapley Additive Explanations https://arxiv.org/abs/2110.14270

[54] GitHub Issues. 2018. Interpretation of Kernel SHAP and Its Hyperparameters - Issue #23 https://github.com/slundberg/shap.

[55] GitHub Issues. 2019. Choosing the Background Set · Issue #391 · https://github.com/slundberg/shap.

[56] GitHub Issues. 2019. Interpretation of SHAP Values Away from the Mean · Issue #435 · https://github.com/slundberg/shap.

[57] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. 2020. DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization. In Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI. 2855–2862.

[58] Miguel Á. Carreira-Perpiñán and Suryabhan Singh Hada. 2021. Counterfactual Explanations for Oblique Decision Trees: Exact, Efficient Algorithms. Proceedings of the AAAI Conference on Artificial Intelligence 35 (May 2021), 6903–6911. https://doi.org/10.1609/aaai.v35i8.16851.

[59] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. 2020. Multi-Objective Counterfactual Explanations. In Parallel Problem Solving from Nature – PPSN XVI. Springer International Publishing, Cham, 448–469.

[60] Rubén R. Fernández, Isaac Martín de Diego, Víctor Aceña, Alberto Fernández- Isabel, and Javier M. Moguerza. 2020. Random forest explainability using counterfactual sets. Information Fusion 63 (2020), 196–207. https://doi.org/10.1016/j.inffus.2020.07.001

[61] Guillaume Jeanneret, Loic Simon, Fredéric Jurie, Adversarial Counterfactual Visual Explanations, 2023 https://arxiv.org/abs/2303.09962

[62] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2018. Local Rule-Based Explanations of Black Box Decision Systems. http://arxiv.org/abs/1805.10820

[63] A.-H. Karimi, G. Barthe, B. Balle, and I. Valera. 2020. Model-Agnostic Counterfactual Explanations for Consequential Decisions. http://arxiv.org/abs/1905.11190

[64] Divyat Mahajan, Chenhao Tan, and Amit Sharma. 2020. Preserving Causal Constraints in Counterfactual Explanations for Machine Learning Classifiers. http://arxiv.org/abs/1912.03277

[65] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. In Proceedings of the Conference on Fairness, Accountability, and Transparency (FAccT) (FAT* '20). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3351095.3372850

[66] Pearl, Judea, and Dana Mackenzie. The Book of Why. Penguin Books, 2019.

[67] Chris Russell. 2019. Efficient Search for Diverse Coherent Explanations. In Proceedings of the Conference on Fairness, Accountability, and Transparency (FAccT) (FAT* '19). Association for Computing Machinery, New York, NY, USA, 20–28. https://doi.org/10.1145/3287560.3287569

[68] Maximilian Schleich, Zixuan Geng, Yihong Zhang, and Dan Suciu. 2021. GeCo: Quality Counterfactual Explanations in Real Time. arXiv:cs.LG/2101.01292

[69] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. 2019. CERTIFAI: Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models. http://arxiv.org/abs/1905.07857

[70] Mark T. Keane and Barry Smyth. 2020. Good Counterfactuals and Where to Find Them: A Case-Based Technique for Generating Counterfactuals for Explainable AI (XAI). arXiv:cs.AI/2005.13997

[71] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp), pages 39–57. IEEE, 2017.

[72] Shubham Rathi. 2019. Generating Counterfactual and Contrastive Explanations using SHAP. http://arxiv.org/abs/1906.09293 arXiv: 1906.09293.

[73] Karthikeyan Shanmugam, and Payel Das. 2018. Explanations Based on the Missing: Towards Contrastive Explanations with Pertinent Negatives. In Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18). Curran Associates Inc., Red Hook, NY, USA, 590–601.

[74] Amit Dhurandhar, Tejaswini Pedapati, Avinash Balakrishnan, Pin-Yu Chen, Karthikeyan Shanmugam, and Ruchir Puri. 2019. Model Agnostic Contrastive Explanations for Structured Data. http://arxiv.org/abs/1906.00117

[75] Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. 2019. Towards Realistic Individual Recourse and Actionable Explanations in Black-Box Decision Making Systems. http://arxiv.org/abs/1907.09615

[76] Goutham Ramakrishnan, Y. C. Lee, and Aws Albarghouthi. 2020. Synthesizing Action Sequences for Modifying Model Decisions. In Conference on Artificial Intelligence (AAAI). AAAI press, California, USA, 16. http://arxiv.org/abs/1910.00057

[77] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, Francisco Herrera 2019. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI https://arxiv.org/abs/1910.10045

[78] Arnaud Van Looveren and Janis Klaise. 2020. Interpretable Counterfactual Explanations Guided by Prototypes. http://arxiv.org/abs/1907.02584

[79] Martin Pawelczyk, Johannes Haug, Klaus Broelemann, and Gjergji Kasneci. 2020. Learning Model-Agnostic Counterfactual Explanations for Tabular Data. , 3126–3132 pages. https://doi.org/10.1145/3366423.3380087

[80] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. 2021. Algorithmic Recourse: From Counterfactual Explanations to Interventions. In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21). Association for Computing Machinery, New York, NY, USA, 10. https://doi.org/10.1145/3442188.3445899

[81] Amir-Hossein Karimi, Julius von Kügelgen, Bernhard Schölkopf, and Isabel Valera. 2020. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. http://arxiv.org/abs/2006.06831

[82] Thai Le, Suhang Wang, and Dongwon Lee. 2019. GRACE: Generating Concise and Informative Contrastive Sample to Explain Neural Network Model's Prediction. arXiv:cs.LG/1911.02042

[83] C. M. Bishop. Pattern recognition and machine learning. springer, 2006.

[84] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, Yuichi Ike, Kento Uemura, and Hiroki Arimura. 2021. Ordered Counterfactual Explanation by Mixed- Integer Linear

Optimization. Proceedings of the AAAI Conference on Artificial Intelligence 35, 13 (2021), 11. https://doi.org/10.1609/aaai.v35i13.17376

[85] Kiarash Mohammadi, Amir-Hossein Karimi, Gilles Barthe, and Isabel Valera. 2021. Scaling Guarantees for Nearest Counterfactual Explanations. In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society. Association for Computing Machinery, New York, NY, USA, 177–187. https://doi.org/10.1145/3461702.3462514

[86] Fan Yang, Sahan Suresh Alva, Jiahao Chen, and Xia Hu. 2021. Model-Based Counterfactual Synthesizer for Interpretation. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21). Association for Computing Machinery, New York, NY, USA, 1964–1974. https://doi.org/10.1145/3447548.3467333

[87] André Artelt and Barbara Hammer. 2021. Convex optimization for actionable & plausible counterfactual explanations. https://doi.org/10.48550/ARXIV.2105.07630 3126–3132 pages. https://doi.org/10.1145/3366423.3380087

[88] Guillermo Navas-Palencia. 2021. Optimal Counterfactual Explanations for Scorecard modelling. https://arxiv.org/abs/2104.08619

[89] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. 2020. DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization. In International Joint Conference on Artificial Intelligence (IJCAI). California, USA. https://doi.org/10.24963/ijcai.2020/395

[90] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable Recourse in Linear Classification. In Proceedings of the Conference on Fairness, Accountability, and Transparency (FAccT) (FAT* '19). Association for Computing Machinery, New York, NY, USA, 10 https://doi.org/10.1145/3287560.3287566

[91] Andrea Ferrario Michele Loi 2021 The Robustness of Counterfactual Explanations Over Time https://www.researchgate.net/publication/362513984_The_Robustness_of_Counterfactual_Explanations_over_Time

[92] Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas. 2017. Interpretable Predictions of Tree-Based Ensembles via Actionable Feature Tweaking. In International Conference on Knowledge Discovery and Data Mining (KDD) (KDD '17). Association for Computing Machinery, New York, NY, USA, 10.https://doi.org/10.1145/3097983.3098039

[93] David Warde-Farley and Ian Goodfellow. Adversarial perturbations of deep neural networks. In Advanced Structured Prediction. 2016.

[94] Ana Lucic, Harrie Oosterhuis, Hinda Haned, and Maarten de Rijke. 2019. FOCUS: Flexible Optimizable Counterfactual Explanations for Tree Ensembles. https://doi.org/10.48550/ARXIV.1911.12199

[95] Ana Lucic, Harrie Oosterhuis, Hinda Haned, and Maarten de Rijke. 2020. Actionable Interpretability through Optimizable Counterfactual Explanations for Tree Ensembles.

http://arxiv.org/abs/1911.12199

[96] Suryabhan Singh Hada and Miguel Á. Carreira-Perpiñán. 2021. Exploring Counterfactual Explanations for Classification and Regression Trees. In Machine Learning and Principles and Practice of Knowledge Discovery in Databases. Springer International Publishing, Cham, 489–504

[97] Emily Black, ZifanWang, and Matt Fredrikson. 2022. Consistent Counterfactuals for Deep Models. In International Conference on Learning Representations. https://arxiv.org/abs/2110.03109

[98] Axel Parmentier and Thibaut Vidal. 2021. Optimal Counterfactual Explanations in Tree Ensembles. https://arxiv.org/abs/2106.06631

[99] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. 2018. Comparison-Based Inverse Classification for Interpretability in Machine Learning. In Information Processing and Management of Uncertainty in Knowledge-Based Systems, Theory and Foundations (IPMU). Springer International Publishing. https://doi.org/10.1007/978-3-319-91473-2_9

[100] Adam White and Artur d'Avila Garcez. 2021. Counterfactual Instances Explain Little. https://doi.org/10.48550/ARXIV.2109.09809

[101] Philip Naumann and Eirini Ntoutsi. 2021. Consequence-aware Sequential Counterfactual Generation. arXiv:cs.LG/2104.05592

[102] Dieter Brughmans and David Martens. 2021. NICE: An Algorithm for Nearest Instance Counterfactual Explanations. https://doi.org/10.48550/ARXIV.2104.07411

[103] Tri Dung Duong, Qian Li, and Guandong Xu. 2021. Prototype-based Counterfactual Explanation for Causal Classification.https://doi.org/10.48550/ARXIV.2105.00703

[104] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi- objective genetic algorithm: Nsga-ii, IEEE transactions on evolutionary

computation 6 (2) (2002) 182–197.

[105] S. Shimizu, Lingam: Non-gaussian methods for estimating causal struc- tures, Behaviormetrika 41 (1) (2014) 65–98.

[106] Robert-Florian Samoilescu, Arnaud Van Looveren, and Janis Klaise. 2021. Model-agnostic and Scalable Counterfactual Explanations via Reinforcement Learning. https://doi.org/10.48550/ARXIV.2106.02597

[107] Sahil Verma, Keegan Hines, and John P. Dickerson. 2021. Amortized Generation of Sequential Counterfactual Explanations for Black-box Models. arXiv:cs.LG/2106.03962

[108] https://www.wyden.senate.gov/news/press-releases/wyden-booker-and-clarke-introduce-algorithmic-accountability-act-of-2022-to-require-new-transparency-and-accountability-for-automated-decision-systems

[109] https://www.fdic.gov/resources/supervision-and-examinations/consumer-compliance-examination-manual/documents/5/v-7-1.pdf

[110] Fraunhofer IOSB, Maximilian Becker, Nadia Burkart, Pascal Birnstill, and Jürgen Beyerer. 2021. A Step Towards Global Counterfactual Explanations: Approximating the Feature Space Through Hierarchical Division and Graph Search. Advances in Artificial Intelligence and Machine Learning (2021), 90–110. https://doi.org/10.54364/aaiml.2021.1107

[111] Maximilian Förster, Philipp Hühn, Mathias Klier, and Kilian Kluge. 2021. Capturing Users' Reality: A Novel Approach to Generate Coherent Counterfactual Explanations. https://doi.org/10.24251/HICSS.2021.155

[112] André Artelt, Valerie Vaquet, Riza Velioglu, Fabian Hinder, Johannes Brinkrolf, Malte Schilling, Barbara Hammer .2021. Evaluating Robustness of Counterfactual Explanations https://arxiv.org/abs/2103.02354

[113] Annabelle Redelmeier, Martin Jullum, Kjersti Aas, and Anders Løland. 2021. MCCE: Monte Carlo sampling of realistic counterfactual explanations. https://doi.org/10.48550/ARXIV.2111.09790

[114] Prateek Yadav, Peter Hase, and Mohit Bansal. 2021. Low-Cost Algorithmic Recourse for Users With Uncertain Cost Functions. https://doi.org/10.48550/ARXIV.2111.01235

[115] Rory Mc Grath, Luca Costabello, Chan Le Van, Paul Sweeney, Farbod Kamiab, Zhao Shen, and Freddy Lecue. 2018. Interpretable Credit Application Predictions With Counterfactual Explanations. http://arxiv.org/abs/1811.05245

[116] Rory Mc Grath, Luca Costabello, Chan Le Van, Paul Sweeney, Farbod Kamiab, Zhao Shen, and Freddy Lecue. 2018. Interpretable Credit Application Predictions With Counterfactual Explanations. http://arxiv.org/abs/1811.05245

[117] Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E. Hines, John P. Dickerson, Chirag Shah. 2022: Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review https://arxiv.org/abs/2010.10596

[118] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2574–2582, 2016.

[119] Martin Arjovsky, Soumith Chintala, Léon Bottou, Wasserstein GAN,2017 https://arxiv.org/abs/1701.07875

[120] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In International Conference on Learning Representations, 2018.

[121] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In Proceedings of The Web Conference 2020, pages 3126–3132, 2020.

[122] https://www.researchgate.net/publication/271737760

[123] André Artelt, Roel Visser and Barbara Hammer 2022: Model Agnostic Local Explanations of Reject https://www.esann.org/sites/default/files/proceedings/2022/ES2022-34.pdf

[124] Andre Artelt and Barbara Hammer 2020: Convex Density Constraints for Computing Plausible Counterfactual Explanations

[125] Peyman Rasouli, Ingrid Chieh Yu, 2022: CARE: coherent actionable recourse based on sound counterfactual explanations

[126] Javier Del Sera, Alejandro Barredo-Arrietaa ,Natalia D Rodriguez,Francisco Herrera and Andreas Holzinger 2022: Exploring the Trade-off between Plausibility, Change Intensity and Adversarial Power in Counterfactual Explanations using Multi-objective Optimization

[127] Generative Adversarial Network-based Robustness Evaluation of Machine Learning Classification Algorithms for DDoS-Attacks, https://www.kom.tu-darmstadt.de/assets/4ac52330-f865-4e1b-8e6e-61700c74eed5.pdf

[128] Ruth M.J. Byrne 2002 Mental models and counterfactual thoughts about what might have been

[129] Nicole Van Hoeck, Patrick D. Watson and Aron K. Barbey,2015 Cognitive neuroscience of human counterfactual reasoning

[130] CounteRGAN: Generating Counterfactuals for Real-Time Recourse and Interpretability using Residual GANs, 2021, Daniel Nemirovsky, Nicolas Thiebaut, Ye Xu, Abhishek Gupta

[131] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXivpreprintarXiv:1412.6572, 2014.

[132] Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statistical science, 16(3):199–231, 2001.

[133] Predictive Multiplicity in Classification Charles T. Marx, Flavio du Pin Calmon, Berk Ustun

[134] SCGAN: Sparse CounterGAN for Counterfactual Explanations in Breast Cancer Prediction, 2021, Siqiong Zhou, Upala J. Islam, Nicholaus Pfeiffer, Imon Banerjee, Bhavika K. Patel, Ashif S. Iquebal

[135] Recent Advances in Adversarial Training for Adversarial Robustness Tao Bai, Jinqi Luo , Jun Zhao , BihanWen , QianWang https://arxiv.org/abs/2102.01356

[136] Generative Counterfactual Introspection for Explainable Deep Learning, 2019, Shusen Liu, Bhavya Kailkhura, Imon Banerjee, Donald Loveland, Yong Han

[137] Thomas Spooner, Danial Dervovic, Jason Long, Jon Shepard, Jiahao Chen, Daniele Magazzeni,2021 Counterfactual Explanations for Arbitrary Regression Models

[138] On Counterfactual Explanations under Predictive Multiplicity Martin Pawelczyk, Klaus Broelemann,Gjergji Kasneci

[139] Danilo Numeroso, Davide Bacciu, MEG: Generating Molecular Counterfactual Explanations for Deep Graph Networks

[140] Eoin Delaney, Derek Greene, and Mark T. Keane 2022 Instance-based Counterfactual Explanations for Time Series Classification

[141] The Space of Transferable Adversarial Examples

Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel

[142] Timo Freiesleben, 2020, Counterfactual Explanations & Adversarial Examples Common Grounds, Essential Differences, and Potential Transfers

[143] Nandish Chattopadhyay, Anupam Chattopadhyay , Sourav Sen Gupta and Michael Kasper Fraunhofer Singapore, Nanyang Technological University, Singapore, 2019, Curse of Dimensionality in Adversarial Examples

[144] J. Hopcroft and R. Kannan, "Foundations of data science," 2014.

[145] A.T. Tiembukong, 2023, Counterfactual Explanations as a tool for Model Explanation, its Properties and Relations.

[146] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In ICLR, 2014 https://arxiv.org/abs/1312.6199.

[147] Adversarial Training Methods for Deep Learning: A Systematic Review Weimin Zhao, Sanaa Alwidian and Qusay H. Mahmoud, 2022 https://www.mdpi.com/1999-4893/15/8/283

[148] Florian Tram'er, Alexey Kurakin, Nicolas Papernot, Ian J Goodfellow, Dan Boneh, and Patrick D McDaniel. Ensemble Adversarial Training: Attacks and Defenses. In ICLR, 2018., https://arxiv.org/abs/1705.07204

[149] Ruitong Huang, Bing Xu, Dale Schuurmans, and Csaba Szepesv´ari. Learning with a strong adversary. arXiv preprint arXiv:1511.03034, 2015. https://arxiv.org/abs/1511.03034

[150] Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. Neurocomputing, 307:195–204, 2018. https://arxiv.org/abs/1511.05432

[151] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In ICLR, 2015. https://arxiv.org/abs/1412.6572

[152] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial Robustness through Local Linearization. In NeurIPS. 2019.https://arxiv.org/abs/1907.02610

[153] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks Which Do Not Kill Training Make Adversarial Learning Stronger. In ICML, volume 119, 2020. https://arxiv.org/abs/2002.11242

[154] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving Adversarial Robustness Requires Revisiting Misclassified Examples. In ICLR, 2020. https://arxiv.org/pdf/2201.00148.pdf

[155] Harini Kannan, Alexey Kurakin, and Ian J Goodfellow. Adversarial Logit Pairing. 2018. https://arxiv.org/abs/1803.06373

[156] Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. Metric Learning for Adversarial Robustness. In NeurIPS. 2019. https://arxiv.org/abs/1909.00900

[157] Minhao Cheng, Qi Lei, Pin-Yu Chen, Inderjit Dhillon, and Cho-Jui Hsieh. Cat: Customized adversarial training for improved robustness. arXiv preprint arXiv:2002.06789, 2020., https://arxiv.org/abs/2002.06789

[158] Yogesh Balaji, Tom Goldstein, and Judy Hoffman. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. arXiv preprint arXiv:1910.08051, 2019. https://arxiv.org/abs/1910.08051

[159] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. MMA Training: Direct Input Space Margin Maximization through Adversarial Training. In ICLR, 2020. https://arxiv.org/abs/1812.02637

[160] Sanjay Kariyappa andMoinuddin K Qureshi. Improving adversarial robustness of ensembles with diversity training. arXiv preprint, 2019.https://arxiv.org/abs/1901.09981

[161] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving Adversarial Robustness via Promoting Ensemble Diversity. In ICML, 2019. https://arxiv.org/abs/1901.08846

[162] Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawhich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li. DVERGE: Diversifying Vulnerabilities for Enhanced Robust Generation of Ensembles. In NeurIPS, 2020. https://proceedings.neurips.cc//paper/2020/hash/3ad7c2ebb96fcba7cda0cf54a2e802f5-Abstract.html

[163] Qi-Zhi Cai, Chang Liu, and Dawn Song. Curriculum Adversarial Training. In IJCAI-18, 2018. https://arxiv.org/abs/1805.04807

[164] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the Convergence and Robustness of Adversarial Training. In ICML, pages 6586–6595, 2019. https://arxiv.org/abs/2112.08304

[165] Ian J. Goodfellow, Jonathon Shlens , Christian Szegedy, EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES, 2015 https://arxiv.org/pdf/1412.6572.pdf

[166] Alexey Kurakin, Ian J. Goodfellow, Samy Bengio, ADVERSARIAL EXAMPLES IN THE PHYSICAL WORLD, 2017

[167] Wieland Brendel, Jonas Rauber, Matthias Bethge, DECISION-BASED ADVERSARIAL ATTACKS: RELIABLE ATTACKS AGAINST BLACK-BOX MACHINE LEARNING MODELS, 2018 https://arxiv.org/pdf/1712.04248.pdf

[168] Jianbo Chen, Michael I. Jordan, Martin J. Wainwright, HopSkipJumpAttack: A Query-Efficient Decision-Based Attack, 2020 https://arxiv.org/abs/1904.02144

[169] Huanran Chen, Yichi Zhang, Yinpeng Dong, Jun Zhu, Rethinking Model Ensemble in Transfer-based Adversarial Attacks, 2023 https://arxiv.org/abs/2303.09105

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 27.09.2023

[redacted]

Amadeo Tunyi Tiembukong