CERTIFAI: A Common Framework to Provide Explanations and Analyse the Fairness and Robustness of Black-box Models

Shubham Sharma
University of Texas at Austin
Austin, Texas
shubham_sharma@utexas,edu

Jette Henderson
CognitiveScale
Austin, Texas
jhenderson@cognitivescale.com

Joydeep Ghosh
CognitiveScale
Austin, Texas
jghosh@cognitivescale.com

ABSTRACT

Concerns within the machine learning community and external pressures from regulators over the vulnerabilities of machine learning algorithms have spurred on the fields of explainability, robustness, and fairness. Often, issues in explainability, robustness, and fairness are confined to their specific sub-fields and few tools exist for model developers to use to simultaneously build their modeling pipelines in a transparent, accountable, and fair way. This can lead to a bottleneck on the model developer's side as they must juggle multiple methods to evaluate their algorithms. In this paper, we present a single framework for analyzing the robustness, fairness, and explainability of a classifier. The framework, which is based on the generation of counterfactual explanations[1] through a custom genetic algorithm, is flexible, model-agnostic, and does not require access to model internals. The framework allows the user to calculate robustness and fairness scores for individual models and generate explanations for individual predictions which provide a means for actionable recourse (changes to an input to help get a desired outcome). This is the first time that a unified tool has been developed to address three key issues pertaining towards building a responsible artificial intelligence system.

CCS CONCEPTS

• **Social and professional topics** → **Computing / technology policy**; • **Applied computing** → **Law, social and behavioral sciences**; • **Computing methodologies** → *Machine learning*.

KEYWORDS

Responsible Artificial Intelligence, explainability, fairness, robustness, machine learning

[1]The concept of counterfactuals has a well-established meaning in the causality literature. However, we are using the term "counterfactual" in the *counterfactual explanation* sense, one that has been recently defined in the explainability literature [15] and [13] where the model implies a machine learning model and not a causal mode and where no causal assumptions about the world are made. A detailed discussion on this is presented in [15]'s section III-C.

AIES '20, February 7–8, 2020, New York, NY, USA

ACM ISBN 978-1-4503-7110-0/20/02…$15.00
https://doi.org/10.1145/3375627.3375812

ACM Reference Format:
Shubham Sharma, Jette Henderson, and Joydeep Ghosh. 2020. CERTIFAI: A Common Framework to Provide Explanations and Analyse the Fairness and Robustness of Black-box Models. In *Proceedings of the 2020 AAAI/ACM Conference on AI, Ethics, and Society (AIES '20), February 7–8, 2020, New York, NY, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3375627.3375812

1 INTRODUCTION

Within the machine learning community, researchers are actively building approaches for explainability, fairness, and robustness of models. However, these existing approaches function in isolation from the other concerns, and within these one-off solutions, there are limitations that may make them less feasible to use in the real world. For example, many explainability approaches require making assumptions on the type of model [14] and the type of data [15] for which they can be used. Many methods that provide explainability [2] and robustness [18] require access to the model weights and internals. A company may not wish to divulge model internal processes but still needs to comply with regulations. Meanwhile, there exist a wide variety fairness metrics and viewpoints [3], and the lack of consensus is just a reflection of the complexity of this concept.

To address these shortcomings and provide a single framework with which to evaluate robustness and fairness as well as provide explainability, this paper introduces a unified approach called Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models (CERTIFAI). CERTIFAI's capabilities are built on the generation of *counterfactual explanations*. Counterfactual explanations for machine learning models were first introduced by [15] to help people understand machine-learning-generated decisions. Given an input data point and a classifier model, a *counterfactual explanation* is defined as a generated data point (i.e. a point found in the input space that might not necessarily be a training point) that is as close to the input data point but for which the model gives a different outcome. For example, if a user was denied a loan by a machine learning model, an example counterfactual explanation could be: "Had your income been $5000 greater per year and your credit score been 30 points higher, your loan would be approved." [15] argue that counterfactuals are a way of explaining model results to users such that they can identify actionable ways, called *recourse*, of changing their behaviors to obtain favorable outcomes.

Although developed to help explain model decisions to users, we show how counterfactual explanations can be used to evaluate issues in fairness and robustness in addition to providing explainability. However, some important issues must be resolved before

Figure 1: The CERTIFAI counterfactual generation process. The decision boundary for a binary classifier is shown, with the input instance in black. We sample a set of points (left) in the feature space with a constraint that they must lie on the other side of the decision boundary (green points). The algorithm then evolves these samples (middle) to generate individuals that lie closer to the input point but on the other side of the decision boundary. Finally, a smaller set, the size of which is user-defined, of counterfactuals is generated (right).

counterfactual explanations can be used in realistic settings for explainability as well as for deeper analysis of model characteristics. As promising as the original method [15] and subsequent methods of generating counterfactuals [13, 14] are, they are also limited in that some only work for linear models, while others cannot deal with different data types. To resolve these limitations, CERTIFAI generates counterfactual explanations via a custom genetic algorithm. The meta-heuristic evolutionary algorithm starts by generating a random set of points such that they do not have the same prediction as the input point. A subsequent evolutionary process results in a set of points close to the input that maintain the prediction constraint. Figure 1 shows an example of three counterfactuals (green points) generated for a given input (black point). A major advantage of using the genetic algorithm to generate counterfactual explanations is that it is model agnostic and works with a variety of data types (from mixed tabular data to image data) without any approximations to or assumptions for the model. Additionally, constraints on the form of the resulting counterfactual explanations can be included, making it adaptable to a users needs.

The major contributions of this paper include:

- Counterfactual explanations are generated using a custom genetic algorithm, which is model-agnostic, flexible, and can be used to provide explanations and recourse to users subjected to the decisions of the classifier.
- Counterfactual explanations are used to generate a normalized score(NCERScore), which can be used to compare the robustness of different models.
- Counterfactual explanations can be used to evaluate fairness with respect to a given user as well as the fairness of the model towards groups of individuals. We define burden, a notion of group fairness that is more understandable and explainable than presently used group-fairness metrics [4]

Using CERTIFAI, an end-user (i.e., a person subject to a machine learning model's decision, a model developer, or a third-party regulator), can understand the issues most relevant to their situations. Because of limited space, we include the following in the appendix [2]: detailed related work, details of the genetic algorithm, counterfactuals for images (adversarial examples), additional experiments on explanations with constraints including sparse explanations and notions of individual fairness. We encourage readers to go through the appendix for a detailed analysis.

2 RELATED WORK

Research on the explainability, fairness, and robustness of machine learning models and the ethical, moral, and legal consequences of using AI has been growing rapidly. General surveys on explainability, fairness, and robustness have been described by [10],[5], and [1] respectively. In this section, we discuss and compare the literature for analyzing the robustness, fairness, and interpretability of machine learning models with a particular focus on counterfactual explanations. Table 1 summarizes the key features of CERTIFAI and the work most related to ours. CERTIFAI uniquely covers all six desirable features, as indicated by the table.

3 THE CERTIFAI FRAMEWORK

Figure 2 shows the CERTIFAI framework at a high level. An overview of the framework is described below and the details of each aspect of the framework are provided in subsequent sections. The only required inputs to CERTIFAI to generate explanations are a black-box classifier and one instance or a set of instances appropriate for the model (first box on left in Figure 2). Optionally, the end-user or model developer can supply a set of constraints to which the counterfactual explanations will adhere (dotted box on left in Figure 2). These optional constraints are 1) not allowing certain features to change, 2) specifying a range of values that the features can take and the data-type of features, and 3) k, the number of counterfactual explanations per input instance. The constraints provided by a model developer are chosen as default which can be over-ridden by the user's preferences. CERTIFAI then generates k counterfactual explanations for each input instance using a custom genetic algorithm (two middle columns of Figure 2) and these explanations are then used to perform the robustness, fairness, and explainability analysis in addition to providing counterfactual explanations for individual instances (far right of Figure 2).

3.1 Custom genetic algorithm

In this section, we formulate a custom genetic algorithm to find counterfactual(s) that CERTIFAI uses for its analysis. Consider a black-box classifier f and an input instance $\mathbf{x}$. Let the counterfactual be a feasible generated point $\mathbf{c}$. Then the problem can be formulated as:

$$\begin{aligned} &\min_{c} d(\mathbf{x}, \mathbf{c}) \\ &\text{s.t.} f(\mathbf{c}) \neq f(\mathbf{x}) \end{aligned} \tag{1}$$

where $d(\mathbf{x}, \mathbf{c})$ is the distance between $\boldsymbol{x}$ and $\boldsymbol{c}$. To avoid using any approximations to or assumptions for the model, we use a genetic algorithm to solve Equation 1. The custom genetic algorithm works for any black-box model and input data type and it is model-agnostic.

[2]The appendix can be found at:
https://drive.google.com/open?id=1AXURcktJmUx0gSd6OVOLNIU6i9$_HJNGJ$

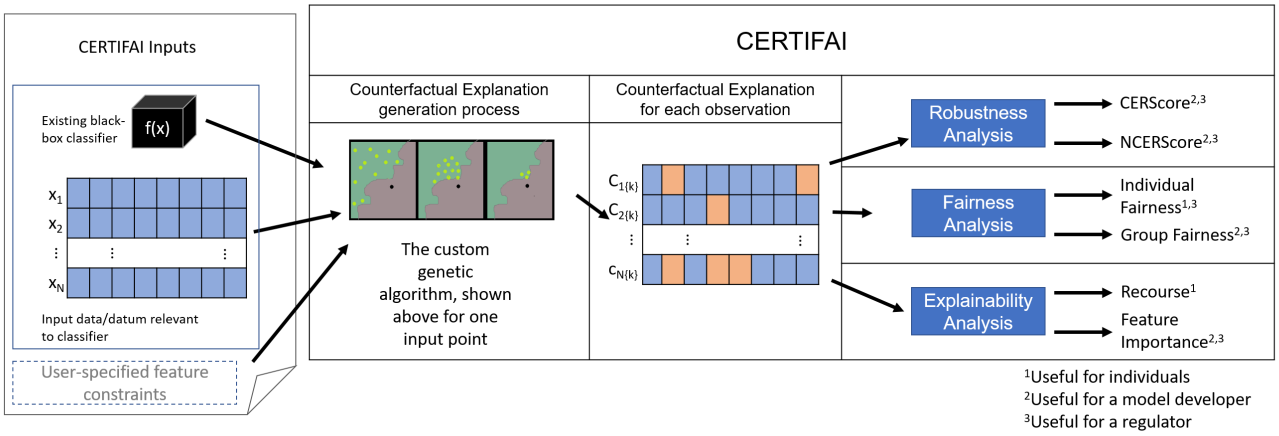

Figure 2: The CERTIFAI framework. Given a black-box ML model and input data along with optional user-specified feature constraints (such as feature type, range, etc.), the method generates counterfactual explanations using a genetic algorithm. The explanations can then be used for three purposes: explainability, fairness and robustness. k represents the number of explanations per input which can be set by the user for recourse purposes and is set to 1 for the feature importance, fairness and robustness analysis. On the right, we show how each of CERTIFAI's attributes is useful for different stakeholders using the tool

Method	Black-box	Model-Agnostic	Mixed-data	Exp.	Fairness	Robustness
CERTIFAI	✓	✓	✓	✓	✓	✓
[14]	✓		✓	✓	✓	
[15]	✓	✓		✓	✓	
[13]	✓		✓	✓	✓	
[12]	✓	✓	✓	✓		
[9]	✓	✓	✓	✓		
[6]		✓				✓
[18]		✓				✓

Table 1: Related work. Exp. represents explainability. We consider the approaches most similar to ours. Mixed-data means the method can work with both discrete and continuous data, without any discretization or assumptions.

Additionally, it provides a great deal of flexibility in counterfactual generation.

CERTIFAI's genetic algorithm solves the optimization problem in Equation 1 through a process of natural selection. The only mandatory inputs for the genetic algorithm are the black-box classifier f and an input instance $\mathbf{x}$. Generally, for an n-dimensional input vector $\mathbf{x}$, let $W \in \mathbb{R}^n$ represent the space from which individuals can be generated and P be the set of points with the same prediction as $\mathbf{x}$:

$$P = \{\mathbf{p} | f(\mathbf{p}) = f(\mathbf{x}), \mathbf{p} \in W\}. \tag{2}$$

The possible set of individuals $\mathbf{c} \in I$ are defined such that

$$I = W \setminus P. \tag{3}$$

Each individual $\mathbf{c} \in I$ is a candidate counterfactual. The goal is to find the fittest possible $\boldsymbol{c}^*$ to $\mathbf{x}$ constrained on $\boldsymbol{c}^* \in I$. The fitness for an individual $\mathbf{c}$ is defined as:

$$fitness = \frac{1}{d(\mathbf{x}, \mathbf{c})}. \tag{4}$$

Here $\boldsymbol{c}^*$ will then be the point closest to $\mathbf{x}$ such that $\boldsymbol{c}^* \in I$. For a multi-class case, if a user wants the counterfactual $\mathbf{c}$ to be belong to a particular class j, we define Q as:

$$Q = \{\mathbf{q} | f(\mathbf{q}) = j, \mathbf{q} \in W\}. \tag{5}$$

Then Equation 3 becomes:

$$I = (W \setminus P) \cap Q. \tag{6}$$

The algorithm is carried out as follows: first, a set I_c is built by randomly generating points such that they belong to I. Individuals $\boldsymbol{c} \in I_c$ are then evolved through three processes: selection, mutation, and crossovers. Selection chooses individuals that have the best fitness scores (Equation 4). A proportion of these individuals (dependent on p_m, the probability of mutation) are then subjected to mutation, which involves arbitrarily changing some feature values. A proportion of individuals (dependent on p_c, the probability of crossover) are then subjected to crossover, which

involves randomly interchanging some feature values between individuals. The population is then restricted to the individuals that meet the required constraint (Equation 3 or Equation 6), and the fitness scores of the new individuals are calculated. This is repeated until the maximum number of generations is reached. Finally, the individual(s) $\boldsymbol{c}^*$ with the best fitness score(s) is/are chosen as the desired counterfactual(s)[3].

Note that while there is no guarantee that the globally maximum fitness point will always be obtained due to the nature of meta-heuristics, this will also be the case for any practical approach that needs to cater to arbitrary, non-convex black-box models and arbitrarily complex decision-surfaces in high-dimensional feature space. In fact alternative approaches to finding counterfactuals further compromise on quality in the quest for simplicity. For example, Google's what-if tool [8] restricts a counterfactual to be an actual data-point in the training set, which also runs the risk of privacy violation since the counterfactual cannot be a simulated point.

3.2 Choice of distance function

The choice of distance function used in Equation 1 depends on the details provided by the model creator and the type of data being considered. If the data is tabular, [15] demonstrated how the L_1 norm normalized by the median absolute deviation (MAD) is better than using the L_1 or L_2 norm for counterfactual generation. For tabular data, the L_1 norm for continuous features (NormAbs) and a simple matching distance for categorical features (SimpMat) are chosen as default. In the absence of training data, normalization using MAD is not possible. However in model development and our experiments where there is access to training data, normalization is possible. The distance metric used is:

$$d(\mathbf{x}, \mathbf{c}) = \text{NormAbs}(\mathbf{x}, \mathbf{c}) + \text{SimpMat}(\mathbf{x}, \mathbf{c}) \tag{7}$$

For image data, the Euclidean distance and absolute distance between two images are not good measures of image similarity [16]. Hence, we use SSIM (Structural Similarity Index Measure) [17], which has been shown to be a better measure of what humans consider to be similar images [16]. SSIM values lie between 0 and 1, where a higher SSIM value means that two images look more similar to each other. For the input image $\mathbf{x}$ and counterfactual image $\mathbf{c}$, the distance is:

$$d(\mathbf{x}, \mathbf{c}) = \frac{1}{\text{SSIM}(\mathbf{x}, \mathbf{c})}. \tag{8}$$

3.3 Improving counterfactuals with constraints

Optionally, a CERTIFAI user can provide three different kinds of constraints that help make the counterfactuals more realistic: 1) *Muting features*: For example, if a user cannot change their education level, they can mute that feature; 2) *Feature range*: For example, it might be difficult for a user to drastically increase their income to be approved for a loan, so an income range can be specified; 3) *Number of explanations*: CERTIFAI can generate multiple counterfactual explanations in a single run of the algorithm. To receive a set of recourse options, a user can specify how many such explanations to generate.

These auxiliary constraints are incorporated by restricting the space defined by the set W: the space from which individuals can be generated, to ensure feasible solutions. For an n-dimensional input, let W be the Cartesian product of the sets W_1,W_2,...,W_n. For continuous features, W_i can be constrained as $W_i \in [W_{imin}, W_{imax}]$, and categorical features can be constrained as $W_i \in \{W_1, W_2, ..., W_j\}$. However, certain variables might be immutable (e.g., race). In these cases, a feature i for an input $\mathbf{x}$ can be muted by setting $W_i = x_i$. If the constraints are too tight, CERTIFAI generates no solution and asks the user to expand the possible range values for features.

3.4 Robustness Analysis

Machine learning models are prone to attacks and threats. For example, deep learning models have performed exceedingly well for image recognition tasks, but it has been widely shown [6, 11] that these networks are prone to adversarial attacks. Two images may look the same to a human, but when presented to a model, they can produce different outcomes. A counterfactual is a generated point close to an input that changes the prediction and can, therefore, be considered an adversarial example. Using this notion of counterfactuals as adversarial examples, we define CERScore and NCERScore, which are the first ever black-box model robustness scores. These scores are a direct consequence of the distance between the input and counterfactual points.

Specifically, given two black-box models, if the counterfactuals across classes are farther away from the input instances on average for one model as compared to the other model, that first model would be harder to fool. Since CERTIFAI directly gives a measure of distance d($\mathbf{x}$,$\mathbf{c}$), this can be used to define the robustness score for a classifier. Using this distance, we introduce Counterfactual Explanation-based Robustness Score (CERScore), the first ever black-box model robustness score. Given a model, the CERScore is defined as the expected distance between the input instances and their corresponding counterfactuals:

$$CERScore(model) = \mathop{\mathbb{E}}_{X}[d(\mathbf{x}, \mathbf{c}^*)]. \tag{9}$$

To be able to better compare models trained on different data sets, the CERScore can be normalized by the expected value of the distance between data points in each class over all classes k, and hence we get the normalized CERScore NCERScore (abbreviated as NC) as:

$$NC = \frac{\mathbb{E}_X[d(\mathbf{x}, \mathbf{c}^*)].}{\sum_{k=1}^{K} P(x \in \text{class}_k)\mathbb{E}[d(x_i, x_j); x_i, x_j \in \text{class}_k]} \tag{10}$$

(i.e., we normalize by dividing by the expected distance between two datapoints drawn from the same class). A higher CERScore implies that the model is more robust. Note that the normalized CERScore can be greater than 1. Unlike [18], CERTIFAI only needs model predictions and not the model internals.

3.5 Fairness Analysis

The fitness measure (Equation 4) and CERScore can additionally be used to investigate fairness from individual and group perspectives, respectively. CERTIFAI can be used by model developers to

[3] p_m=0.2 and p_c=0.5, which is standard in literature. The population size is the square of the input feature size with a maximum cap of 30,000 for the datasets we experimented on. Grid-search is used to find the number of generations

Model	CERScore	CI	CLEVER
Inception-v3	1.17	1.09-1.25	0.229
Resnet-50	1.06	1.05-1.08	0.137
MobileNet	1.08	1.06-1.09	0.151

Table 2: Robustness score and 95 percent confidence intervals (CI) for those scores for 3 deep learning models and the corresponding CLEVER scores.

audit the fairness for different groups of observations. If the fitness measure is markedly different for counterfactuals generated for the different partitions of a feature's domain value, this could be an indication the model is biased towards one of the partitions or groups. For example, if the gender feature is partitioned into two values (men and women), and the average fitness values of generated counterfactuals are lower for women than for men, this could be used as evidence that the model is not treating females fairly. Using counterfactuals and the distance function, we can calculate the overall burden for a group, measured as:

$$Burden(g) = \mathop{\mathbb{E}}_{g}[d(\mathbf{x}, \mathbf{c}^*)] \tag{11}$$

where g is a partition defined by the distinct values for a specified feature set. Note that burden is related to CERScore as it is the expected value over a group. Burden can be considered to be a nuanced version of other fairness measures (such as demographic parity), as with burden, the score assigned to every group is a weighted version of the proportion of individuals in the negative class of that group, where the weight is dependent on the distance to the boundary.

4 EXPERIMENTS

4.0.1 Evaluating Deep Networks. In this section, we evaluate how well CERScore can give an informative measure of robustness. We consider the same networks as in [18] (Inception-v3, ResNet-50 and MobileNet, pre-trained on ImageNet) where they define the CLEVER score for robustness. Unlike CLEVER, we consider the model to be a black-box (only relying on its predictions).

Ideally, to derive a measure of robustness for a model, all images from all classes should be considered, their counterfactuals should be generated, and the CERScore should then be calculated. However, since the number of training samples for a deep network is in the order of millions, it is not computationally feasible to calculate the score for each example. Hence, we consider a subset of classes and images to calculate the CERScore. We sampled n=50 random images from every class across k=100 random classes. We generate the counterfactuals for all 5,000 images such that the counterfactual gives a prediction of the second most likely class (by generating individuals constrained on belonging to that class as in Equation 6) and empirically estimate the CERScore as:

$$CERScore = \frac{1}{nk}\sum_{i=1}^{k}\sum_{j=1}^{n} d(\mathbf{x}_{ij}, \mathbf{c}^*_{ij}) \tag{12}$$

where $\mathbf{x}_{ij}$ is the j^{th} input instance belonging to predicted class i, and $\mathbf{c}^*_{ij}$ is the corresponding counterfactual. The CERScores along with the 95% confidence intervals, where we have assumed the distribution of distances between the images and their counterfactuals follows a normal distribution, are shown in Table 2. One way to interpret the score is that on average, the SSIM score for Inception-v3 is 1/1.17 = 0.85, where an SSIM score of 1 means the images look exactly the same and an SSIM score of 0 means the images are highly different. Hence, adversarial attacks for Inception-v3 could be more easily identified than for the other models. The confidence intervals are tight around the CERScores.

Table 2 also shows the CLEVER scores [18] for the same images, considering the top-2 class attack. The CERScore implies that Inception-v3 is most robust and Resnet-50 is least robust, which is similar to what the CLEVER scores suggest. Hence, even though CERTIFAI does not access any model weights, it is able to evaluate a model's robustness to adversarial attacks.

4.0.2 Robustness of Classic Classifiers. Next, we use NCERScore (Equation 10) to compare the robustness of different models trained on different data sets. We train three models (decision trees (DT), Support Vector Machines with RBF kernel (SVM), and multilayer perceptrons (MLP)) on the three data sets listed in Table 3. We report the NCERScore and the accuracy on the test set in Table 3. Across all data sets, the neural network has the highest NCERScore and is therefore the most robust of the classifiers for these data sets. In the Pima diabetes data set, the accuracy of the decision tree is much lower than the other models, which suggests this simple model cannot adequately capture the class separation. Hence, more points would be concentrated near the decision boundaries, resulting in a lower NCERScore. For the Iris data set, all models have similar accuracy, but the decision tree has the lowest NCERScore while the scores for SVM and MLP are similar.

4.1 Explainability

Counterfactuals are used to provide explanations and transparency to a user on how much change is needed for them to obtain a favorable prediction. We show example counterfactual explanations, the effect of constraints, and how they can be used to measure feature importance. Some other approaches would not be viable to perform similar experiments: since the UCI Adult dataset contains many categorical variables, finding a counterfactual using [15] would not be feasible and using [14] would only allow for linear models, while we consider neural networks as well. Additional experiments on explainability are given in the appendix.

4.1.1 Multiple Counterfactual Explanations. Multiple explanations are helpful to a user so that they can receive a diverse set of changes that could be made to achieve a desired outcome. The UCI adult dataset is considered and features such as native-country are muted and a set range is given for features like hours-per-week (based on the min-max of the dataset). We run the genetic algorithm for the input instance and select the best two individuals that have different changes in feature indices. The advantage of our approach is that we only need to run the algorithm once to generate many explanations, as opposed to [13] where the solver needs to be run multiple times to generate many explanations.

To underscore the benefits of suggesting alternative counterfactuals, Table 4 shows two sets of explanations that are generated by

Data set	Num. obs.	Num. features	DT NCERS.	DT Acc.	SVM NCERS.	SVM Acc.	MLP NCERS.	MLP Acc.
Pima Diabetes	768	8	0.074	73.25	0.387	81.42	**0.486**	98.61
Breast Cancer	569	32	0.081	95.80	**0.121**	96.50	**0.124**	96.50
Iris	150	4	0.132	95.67	0.235	95.67	**0.241**	95.67

Table 3: Descriptions of data sets, and NCERScore (NCERS.) and test set accuracy (Acc.) for three models: decision tree (DT), SVM with RBF kernel (SVM), and Multilayer Perceptron (MLP).

Person	Feature(s)	Original	Counterfactual
1	Education	12th	Bachelors
	Occupation	Tech-suppt	Exec-managerial
1	Hrs-per-week	50	70
	Workclass	Local-gov	Private

Table 4: Two explanations for the same person from the UCI adult dataset, with constraints on feature values.

Person	Feature(s)	Original	Counterfactual
1	Glucose (CWC)	115	71
	BMI (CUC)	35.3	10.1
2	Glucose (CWC)	168	89
	Age (CUC)	34	44

Table 5: Counterfactual explanations for the Pima Indian diabetes dataset. CWC: counterfactuals with constraints on feature values and CUC: unconstrained counterfactuals. Unconstrained features lead to infeasible solutions (BMI 10.1) or unchangeable features (age) being changed.

CERTIFAI for the same person. The number of explanations can be set by the user, and they can decide which counterfactual may be the most actionable for them.

4.1.2 Importance of constraints. We consider two cases of counterfactual generation, counterfactuals with constraints (CWC) and counterfactuals unconstrained (CUC) for two users with a prediction of high diabetes risk from the Pima Indian diabetes dataset. CWC corresponds to a user or model creator providing a range of values for features. CUC corresponds to a user only providing the black-box model and the input instance without any constraints on the feature values. We show features for which the values have changed (between the input and counterfactual), all other values remained constant.

As shown in Table 5, for person 1, when we provide constraints (CWC), the explanation is: *Had your glucose been less by 34, you wouldn't have been at the risk of diabetes.* All other feature values for the user remained constant. Without constraints, the explanation shows that the BMI would have to be decreased to 10.1. While this is a smaller change in magnitude as compared to changing the glucose level, achieving a BMI of 10.1 is not feasible, and hence it is important to use the flexibility of our approach to add additional constraints that ensure feasibility. Similarly, for person 2, age is suggested to be changed, which is not feasible.

4.1.3 Measuring Feature Importance. From a model developer's perspective, counterfactuals can show the importance of every feature value to the prediction and hence provide transparency. If CERTIFAI is changing a particular feature more often than another feature when comparing the input and counterfactual, it implies that that feature is more significant for a model.

For the Pima Indian diabetes dataset, we generate counterfactuals for all samples (irrespective of prediction) and analyze the number of times every feature value has changed, as shown in Figure 3. Interestingly, the importances are qualitatively similar to those

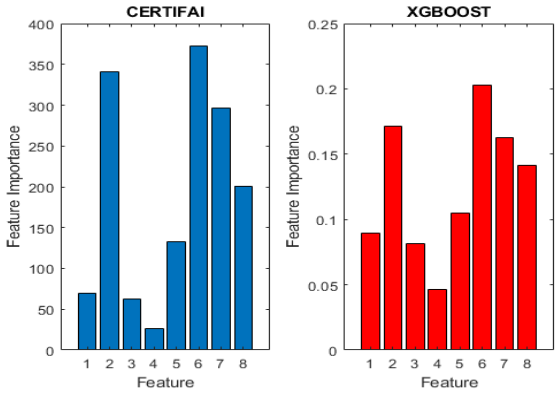

Figure 3: Feature importance for the model, trained on the Pima Indian diabetes dataset, measured by the number of times a feature changed to generate the counterfactual (left) and feature importance by XGBoost (right).

returned by Python's XGBoost [7] library (also shown in Figure 3). Specifically, feature 5 (BMI) and feature 2 (Glucose) are the most important in predicting diabetes risk. This analysis can be extended to the multi-class case by constraining sampled individuals such that they belong to a desired class (Equation 6).

4.2 Fairness

A model developer can use the idea of burden (Equation 11) to evaluate how fair a model is being to groups of individuals. To demonstrate the idea of burden, we consider the attribute race in the UCI adult dataset and take all training examples that have an

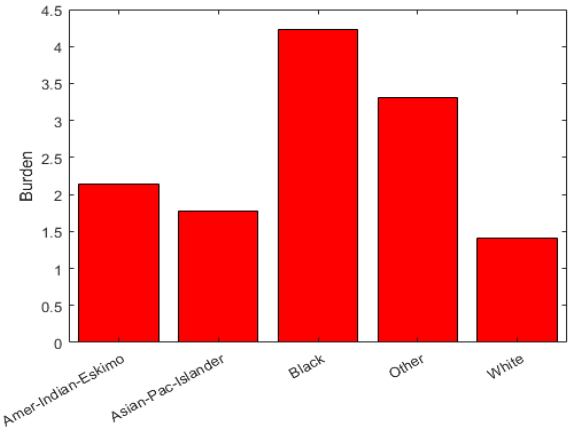

Figure 4: Burden on different groups belonging to a particular race in the UCI adult dataset, found using the distance between the input instances and counterfactuals (Equation 11)

unfavorable outcome. The results of our experiments are shown in Figure 4. As we can see, the burden when the race is Black and the race is Other is more than the other races. This means that on average, these groups would have to make more changes to achieve a desired prediction as compared to others. Hence the model imposes a greater burden on these groups, which could imply that the model has been unfair.

5 CONCLUSION AND FUTURE WORK

In this paper, we introduced CERTIFAI, a model-agnostic, flexible, and user-friendly technique that helps build responsible artificial intelligence systems by providing explainability to its users and evaluations of robustness and fairness. We demonstrate the flexibility that the genetic algorithm brings to provide feasible counterfactual explanations to a user and how to use them to understand important features. We show how fairness can be measured using the fitness values obtained during the counterfactual generation process. Finally, we define CERScore and NCERScore by drawing a relation between counterfactuals and adversarial examples, which can be used to compare the robustness of different models.

We have developed a User-Interface to CERTIFAI and are currently testing it. While the notion of burden is in keeping with other group fairness measures in terms of the results we obtained for the Adult dataset and the results in [4], a formal comparison between burden and previous group fairness metrics would also be useful. Moreover, we would like to compare the counterfactual explanations to other techniques of explainability by conducting a user-study.

REFERENCES

[1] Naveed Akhtar and Ajmal Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* 6 (2018), 14410–14430.
[2] Sebastian Bach, Alexander Binder, Gregoire Montavon, Frederick Klauschen, Klaus-Robert Muller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10, 7 (2015), e0130140.
[3] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. 2018. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943* (2018).
[4] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. 2018. AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias. https://arxiv.org/abs/1810.01943
[5] Reuben Binns. 2017. Fairness in machine learning: Lessons from political philosophy. *arXiv preprint arXiv:1712.03586* (2017).
[6] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 39–57.
[7] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785
[8] Google. 2019. Google what-if tool.
[9] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2018. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820* (2018).
[10] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* 51, 5 (2018), 93.
[11] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 427–436.
[12] 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*. ACM, 1135–1144.
[13] Chris Russell. 2019. Efficient Search for Diverse Coherent Explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, 10–19.
[14] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable Recourse in Linear Classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, 10–19.
[15] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harvard Journal of Law & Technology* 31, 2 (2017), 2018.
[16] Zhou Wang, Alan C Bovik, and Ligang Lu. 2002. Why is image quality assessment so difficult?. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (CASSP*. IEEE, IV–3313.
[17] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Vol. 2. Ieee, 1398–1402.
[18] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. 2018. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578* (2018).