Multi-Granular Evaluation of Diverse Counterfactual Explanations

Yining Yuan¹, Kevin McAreavey¹, Shujun Li² and Weiru Liu¹

¹School of Engineering Mathematics and Technology, University of Bristol, U.K.
²School of Computing, University of Kent, U.K.

Keywords: Counterfactual Explanations, Explainable AI.

Abstract: As a popular approach in Explainable AI (XAI), an increasing number of counterfactual explanation algorithms have been proposed in the context of making machine learning classifiers more trustworthy and transparent. This paper reports our evaluations of algorithms that can output diverse counterfactuals for one instance. We first evaluate the performance of DiCE-Random, DiCE-KDTree, DiCE-Genetic and Alibi-CFRL, taking XGBoost as the machine learning model for binary classification problems. Then, we compare their suggested feature changes with feature importance by SHAP. Moreover, our study highlights that synthetic counterfactuals, drawn from the input domain but not necessarily the training data, outperform native counterfactuals from the training data regarding data privacy and validity. This research aims to guide practitioners in choosing the most suitable algorithm for generating diverse counterfactual explanations.

1 INTRODUCTION

With machine learning models being deployed widely across various sectors in decision-making, the predictions and decisions made by these models are growing in influence and impact. Explanations for these models’ outputs are crucial for domains where users need to build trust in the model and prefer more transparency in decision-making, such as healthcare, credit loans, etc. Explaining decisions made by AI and machine learning models can also help ensure compliance with laws such as the GDPR regulating automated decision-making (Wachter et al., 2017).

Background: Explainability can be realized through inherently interpretable models like linear and logistic regression, etc., or via post-hoc explanations, which also work for black-box machine learning models like random forests and neural networks. Post-hoc explanation approaches can be model-specific, including visual explanations and model simplification, or model-agnostic, including feature importance and local explanations (Verma et al., 2020). Feature importance can be calculated via SHAP (Lundberg and Lee, 2017). Local explanations can be further divided into two main types: approximation methods that aim to mimic the behaviour of the model locally, such as LIME (Ribeiro et al., 2016), and example-based methods that return nearby data points with differing predictions, such as Counterfactual Explanations (CFEs) (Poyiadzi et al., 2020).

Counterfactual explanations are a popular approach in XAI. These are sometimes said to offer actionable insights by suggesting modifications to a data point or, alternatively, an instance to alter its classification outcome. For instance, consider a loan application case where an individual is denied a loan based on a machine learning model’s prediction. That individual would naturally want to understand the changes they could make to secure approval. A CFE could inform this applicant that increasing their income by a certain amount or acquiring two additional years of education would have led to loan approval (Verma et al., 2020).

Understanding the importance of features in a model’s prediction is a crucial aspect of XAI. Tools like SHAP provide consistent measures of feature importance, which affect the outcome of a prediction. While SHAP values highlight the importance, the outcome of changing these feature values is crucial for CFEs. Even if a feature is deemed important, changing it may not lead to a different prediction. DiCE methods consider feature weight either by using distances or manual entries (Mothilal et al., 2020). The interplay between feature importance and actionability can be pivotal in ensuring that the recommendations provided by CFEs are both influential and feasible for end-users.

Challenges of CFEs: In real-world applications,
CFEs often need to adhere to specific constraints, ensuring that the recommended changes are not only actionable but also realistic and aligned with commonsense understanding. Using counterfactuals taken from the training data can significantly reduce the run time and enhance the plausibility of the explanations, where plausibility refers to how realistic the counterfactual explanation is concerning the data manifold (Goethals et al., 2023). CFEs that return data points that existed in the original dataset as explanations are known as native counterfactuals (Goethals et al., 2023). CFEs not relying on specific examples from the training set are known as synthetic counterfactuals (Keane and Smyth, 2020). However, there’s a privacy risk associated with counterfactual algorithms that use native instance-based strategies. They can potentially reveal private information about other decision subjects, such as someone’s grades in educational decisions or income in credit scoring decisions (Vo et al., 2023). While native counterfactuals can reveal records from the training dataset, synthetic counterfactuals do not have this risk. However, techniques that generate synthetic counterfactuals are also vulnerable to privacy attacks, such as model extraction (Goethals et al., 2023).

The motivation behind this research is to study the advantages and limitations of CFE algorithms currently available. There is a growing consensus on the advantages of producing multiple CFE alternatives rather than a single CFE, where a higher diversity value is desirable (Molnar, 2022). Such diversity provides decision-makers with distinct alternatives to reach the desired outcome.

In this research, we selected DiCE and Alibi, the popular Python libraries on GitHub capable of generating diverse counterfactuals for a single instance for comparison. Notably, DiCE-Random, DiCE-Genetic and DiCE-KDTree are under the Diverse Counterfactual Explanations (DiCE) framework, which offers a unique opportunity to explore the different algorithms (random sampling, genetic algorithm, and k-dimension trees) within a unified framework. Alibi-CFRL is a reinforcement learning-based method integrated into Alibi. Here, DiCE-KDTree only outputs native instances in the training set, while other algorithms return synthetic data points. The overall workflow of our evaluation is shown in Figure 1. Our comparison stands to benefit end-users who leverage these explanations for informed decision-making and data scientists who can harness the derived insights to build models that are both more robust and more explainable.

Based on the above background, this research has the following contributions:

- We compared SHAP local explanation and other CFE algorithms on the central data point and the outlier data point within the dataset. Our central data point is the one closest to the mean centre of a dataset, while an outlier is the data point that deviates the most from the mean centre. Our comparison shows a disconnect between SHAP feature importance and the feature alterations in the CFEs by the four algorithms we examined. Specifically, the features identified as most important by SHAP do not consistently match those altered in the CFEs.

- We evaluated the actionability of four CFE algorithms on the unfavourable class. In our context, the unfavourable class means being rejected for a loan and having a lower income in census data. We discovered that Alibi-CFRL and DiCE-Random outperform other algorithms in different metrics.

- We emphasized the importance of synthetic counterfactuals instead of native counterfactuals in CFEs for data privacy.

Figure 1: Overview of the evaluation process.
2 RELATED WORK

2.1 CFEs Methods

Native Counterfactual Methods: CFEs that return data points that existed in the original dataset as explanations are known as native counterfactuals (Goethals et al., 2023). Poyiadzi et al. (2020) proposed FACE, which employs Dijkstra’s algorithm to identify the most direct route connecting close data points based on density-weighted distances and identifies counterfactuals using model predictions and density thresholds (Poyiadzi et al., 2020). As a result, this technique does not create new data points.

KDTree prototype (Van Looveren and Klaise, 2021) uses a k-dimension tree to partition training data based on feature values, identifying nearest neighbours to the input instance and sourcing counterfactuals directly from these neighbours, ensuring that they reflect the original feature distributions and meet specified constraints.

Genetic-based Methods: Genetic-based CFE methods generate synthetic CFEs. The genetic algorithms use mutation and crossover to evolve potential solutions, optimizing the predicted probability based on certain constraints and cost functions to provide synthetic CFEs. GIC (Lash et al., 2017) introduces real-valued heuristic-based methods, including hill-climbing, local search, and genetic algorithms. CERTIFAI (Sharma et al., 2019) is a meta-heuristic evolutionary algorithm that begins by producing a random set of data points that do not share the same prediction as the input data point. MOC (Dandl et al., 2020) employs mixed integer evolutionary strategies to handle both discrete and continuous search spaces for generating CFEs. Compared with previous methods, GeCO achieves real-time performance and a complete search space by incorporating two novel optimizations. δ-representation groups candidates by differing features, using compact tables for memory and performance gains. Classifier Specialization via Partial Evaluation streamlines classifiers to only assess performance gains. Classifier Specialization via Partial Evaluation

Prototype-based Methods: Both prototype-based methods and reinforcement learning-based methods below generate CFEs. Based on the original loss term by (Wachter et al., 2017), Van Looveren and Klaise (2021) used a prototype loss term to guide the perturbations towards an interpretable counterfactual. The prototype for each class can be defined using an encoder, where the prototype is the average encoding of instances belonging to that class. Duong et al. (2021) proposed a prototype-based method to ensure that the counterfactual instance respects the constraints and is interpretable.

Reinforcement Learning-based Methods: Samoilescu et al. (2021) introduced a deep reinforcement learning approach. The significant advantages of this method are its fast counterfactual generation process and its flexibility in adapting to other data modalities like images. Verma et al. (2022) proposed FASTAR, another approach that translates the sequential algorithmic recourses problem into a Markov Decision Process that uses reinforcement learning to generate amortized CFEs.

To summarize, the methods outlined above mostly generate synthetic CFEs rather than finding native CFEs. Our analysis indicates a prevailing trend among recent CFE algorithms, which lean more toward synthetic than native. However, the reasons driving this shift are seldom explored in the existing literature.

2.2 Quantitative Properties

In this subsection, we look into several quantitative properties, including validity, sparsity, proximity, data manifold, actionability and diversity.

Validity: CFE methods are unsound in that they may return data points that do not have the desired target label. A counterfactual correctly classified into the desired class while satisfying hard constraints is considered a valid counterfactual. Popular hard constraints specify immutable features, feature ranges (Mothilal et al., 2020) and the direction of feature value change (Samoilescu et al., 2021). Validity measures the proportion of instances within the test set for which sound CFEs can be generated. A higher validity ratio indicates better performance because a larger proportion of generated counterfactuals have the opposite class label to the original (Samoilescu et al., 2021).

Sparsity: Sparsity refers to the number of features changed to obtain a counterfactual. Additionally, a CFE \( x' \) is minimally sparse given input \( x \) if it minimizes the Hamming distance \( d(x, x') = |\{i : i = 1, 2, ..., n : x_i \neq x'_i\}|. \) This is considered because shorter answers have been argued to be easier for people to understand (Naumann and Ntoutsi, 2021). Sparsity is typically enforced by methods using solvers (Karimi et al., 2020) or by constraining the \( L_0 \) norm for black-box methods (Dandl et al., 2020). \( L_0 \) norm counts the number of non-zero elements in a vector. Gradient-based methods often use the \( L_1 \) norm between counterfactuals and input data points. The \( L_1 \) norm of a vector is the sum of the absolute values of its components. Some approaches change a fixed number of features (Keane and Smyth, 2020), adjust features iteratively (Le et al., 2020), or flip the min-
imal possible split nodes in a decision tree (Guidotti et al., 2018) to induce sparsity.

**Proximity:** Proximity refers to the closeness between a counterfactual and the original instance (Brughmans et al., 2023). To be useful, counterfactuals should ideally be close to the original data point. The smaller the distance between the counterfactual and the original data point, the better. Distance metrics such as the Euclidean distance or Mahalanobis distance are often used for this purpose and are often measured separately for numerical and categorical features (Verma et al., 2020).

**Data Manifold Closeness:** Data manifold means the closeness to the training data distribution (Verma et al., 2020; Verma et al., 2022). Several approaches address data manifold adherence using techniques such as training VAEs on the data distribution (Mahanjan et al., 2019), constraining the counterfactual distance from the nearest training data points (Dandl et al., 2020), sampling points from the latent space of a VAE (Pawelczyk et al., 2020), using an ensemble model to capture predictive entropy (Schut et al., 2021), or applying Kernel Density Estimation (Förster et al., 2021) and Gaussian Mixture Models (Arteit and Hammer, 2021b). Some methods use cycle consistency loss in GANs (Van Looveren et al., 2021), feature correlations (Arteit and Hammer, 2021a), or restrict to using existing data points (Poyiadzi et al., 2020).

**Action Sequence:** Most approaches often overlook the sequential nature of actions and their consequences, leading to explanations that may lack realism and applicability in real-world scenarios. Consequence-aware sequential counterfactual generation by Naumann and Ntoutsi (2021) exemplifies this approach, employing a genetic algorithm and a consequence-aware cost model to generate sequential counterfactuals. However, as critiqued (De Toni et al., 2022), the incorporation of sequence also introduces challenges in terms of computational complexity and the need for explicit cost modelling.

**Diversity:** Diverse counterfactuals for a single input increase the likelihood for users to achieve a desired outcome, with diverse sets presenting a broader spectrum of choices (Guidotti, 2022). While each counterfactual should be close to the original instance, the collective set should emphasize differences among its constituents, offering varied actionable insights (Verma et al., 2020). To achieve diversity, methodologies range from using determinantal point processes (Mothilal et al., 2020) to enforcing hard constraints (Karimi et al., 2020).

### 2.3 Qualitative Properties

**Stability:** For an explainer to be stable, it should provide similar CFEs for instances that are alike and receive the same classification. Stability can also be referred to as robustness (Guidotti, 2022). Virgolin and Fracaros (2023) explored the trade-off between robustness and simplicity in CFEs. They concluded that measuring robustness on prescribed mutable features is more efficient and reliable than on immutable features.

**Fairness:** Lack of stability in explanations can undermine fairness and erode trust in a system (Molnar, 2022). If two financially similar individuals are denied loans but receive vastly different CFEs, one needing a slight income change boost whilst the other a significant one plus other changes, it raises fairness concerns. Hence, the consistency of CFEs is crucial for ensuring individual fairness. The fairness of counterfactuals, especially concerning non-actionable features, is elaborated upon in works by Von Kügelgen et al. (2022).

**Privacy:** Barocas et al. (2020) stressed the inherent challenges and tensions between the need for detailed, tailored explanations and the preservation of individual privacy. Vo et al. (2023) emphasized that generalizing the data by employing discretization of continuous features, as done in their L2C method, is useful to prevent inference attacks. Goethals et al. (2023) proposed solutions like k-anonymous CFEs to mitigate privacy risks.

### 3 METHODOLOGY

Our evaluation methodology outlines the initial steps taken for executing CFE experiments. We will analyse how different methods perform relative to each other, which can guide the selection of the most suitable method for a given application.

#### 3.1 Preliminaries

**Data Preprocessing:** We consider two tabular datasets, Adult Census \(^1\) and German Credit \(^2\). Their features are provided in Table 2 and Table 4, respectively. We then specify numeric and categorical columns according to the description of the datasets. Adult Census consists of 6 numerical and 8 categorical attributes for 48,842 instances, while German Credit provides 7 numerical and 13 categorical credit-related attributes for 1,000 individuals.

---

\(^1\)https://doi.org/10.24432/CSWX20

\(^2\)https://doi.org/10.24432/CSNC77
To ensure the accuracy and consistency of the model, the data preprocessing steps employed should align with those used during the model’s training phase. In the context of this study, we opted for the OneHotEncoder for categorical data and the MinMaxScaler for numerical data. Firstly, the reinforcement learning-based algorithm we have selected, Alibi-CFRL, only accommodates data encoded using OneHotEncoder. Secondly, the use of MinMaxScaler is informed by our literature analyses on proximity. When numerical features vary significantly in magnitude, discrepancies in scale can introduce errors in computing the proximity and distance of counterfactuals for different features.

**XGBoost Classifier:** In the initial phase of our machine learning system, we employed XGBoost (Chen and Guestrin, 2016), an open-source library that stands for Extreme Gradient Boosting. Unlike traditional gradient-boosting decision trees that build trees sequentially, XGBoost constructs trees in parallel, following a level-wise strategy. One of the key features of XGBoost is its ability to handle sparse data and missing values, making it robust for OneHotEncoder.

### 3.2 SHAP Local Explanation

The SHAP explanation approach derives its computations from the Shapley values found in coalitional game theory (Lundberg and Lee, 2017). In this context, the feature values of a data instance are likened to players forming a coalition. Specifically, Shapley values are articulated as the average marginal contribution of a feature value, encompassing all conceivable combinations of features. To garner a more granular understanding of feature contributions for individual predictions, we employed the force plot visualization from the SHAP Python package. This plot offers an intuitive representation, where each feature’s contribution is denoted by an arrow, the magnitude and direction of which signify the feature’s impact on the prediction, e.g., as illustrated in Figure 2a in Section 4. The cumulative effect of these arrows, or forces, demonstrates how the prediction deviates from a base value.

### 3.3 Evaluation Setup

**Auto Encoder:** Instead of directly modelling perturbations in the potentially high-dimensional input space, we create an autoencoder following the algorithm by Samoilescu et al. (2021) and apply perturbations to the latent space, leveraging its compact representations. The pre-trained encoder then maps these perturbed embeddings back to the input feature space, ensuring that the counterfactuals are coherent and interpretable.

**Counterfactual Explainers:** The four algorithms we compare include approaches that return native or synthetic CFEs. DiCE-KDTree is a Python package within the DiCE framework. It is inspired by a counterfactual search with k-dimension trees, which discovers native CFEs (Van Looveren and Klaise, 2021). The following algorithms are for synthetic CFEs: DiCE-Random generates CFEs through random sampling. DiCE-Genetic is inspired by the GeCQ (Schleich et al., 2021). It employs a genetic algorithm to generate counterfactuals. Alibi-CFRL is based on reinforcement learning (Samoilescu et al., 2021). It employs deep deterministic policy gradient to train a generative model that directly models counterfactual instances. We set the hyperparameters, including feature constraints and diversity, to ensure all the explainers output 5 CFEs for an input instance.

### 3.4 Evaluation Metrics

**Validity:** Validity quantifies the proportion of data points for which valid counterfactuals were found. We generate five diverse CFEs for each instance. If at least one of those is valid, then the method is deemed capable of generating a valid CFE for that instance.

**Sparsity:** Sparsity is computed separately for categorical and continuous features and then summed (Verma et al., 2022). A lower sparsity is better. Categorical features only count the number of non-zero differences. In our context, a zero value in the vector indicates similarity, while a value of 1 indicates dissimilarity. Continuous features vary within a continuous range, which might consider each feature’s scale or range, allowing precise differences to be calculated. We calculate the differences and normalize them by the number of features. Some measurements calculate sparsity in a reversed way (Vo et al., 2023), so in their discussion, the larger the sparsity, the better.

**Proximity:** A lower proximity is better for both numerical and categorical features. The proximity for continuous features is computed by the $L_1$ norm. This involves calculating the absolute difference between each continuous feature of the counterfactual and the original data point and normalizing it by the feature’s median absolute deviation (MAD). The proximity for categorical features is computed by a sparse version of the $L_0$ norm. This metric calculates the normalized mismatches between the counterfactual and the original data point across the categorical features.

**Manifold Distance:** Manifold distance quantifies
how close a counterfactual is to the data manifold of the original dataset. A lower manifold distance is better. It indicates that the counterfactuals are closer to the data distribution. In our study, first, a 1-Nearest Neighbor model is trained on the dataset. This model allows us to find the nearest neighbours of a given data point within the dataset. For each counterfactual, the distance to its nearest neighbour in the dataset is computed. Then, the average of the nearest neighbours’ distances for all the CFEs for a given original datapoint is taken as the manifold distance.

**Constraint Satisfaction:** Constraints determine whether a counterfactual is logically consistent with domain knowledge. In our evaluation, we use the following domain constraints as a proxy for actionability.

The first test is **Immutable Features.** These are features that their values should not change in the counterfactual. All tested CFE methods are aware of immutable features and feature ranges. These are set as hard constraints. The second test is **Non-decreasing Features.** These are features that should not decrease in the counterfactual. Only Alibi-CFRl is aware of increasing and decreasing feature values, and other methods follow desiderata by specifying value ranges. There are other preferred conditions for a counterfactual to be actionable. The third test is **Correlated Features.** These constraints ensure that if one feature changes in some way (e.g., increase), then another feature’s value shall change in a certain way (e.g., decrease or increase). Table 1 shows our setting of Immutable Features and preferred constraints. Here, our preferred constraints include Non-decreasing Features and Correlated Features. These are considered in papers about algorithmic recourse (Verma et al., 2022). However, in our tested CFEs, these preferred constraints are often not set as user-configured. Techniques for achieving correlation between these variables, and thus making counterfactuals actionable, may be realized by optimizing Mahalanobis’ distance (Kanamori et al., 2020). Therefore, besides evaluating whether the algorithms conform to the hard constraints, we hope to evaluate whether there are algorithms that can consider the increasing or decreasing nature of the domain-specific variables and the trends in the correlations between them.

For a given instance with diverse counterfactuals, the average constraint satisfaction can be computed by combining the above three tests. These constraints are tested in a function that returns 1 if the counterfactual satisfies all constraints and 0 otherwise. The function can be called individually for each counterfactual for our diverse counterfactuals and calculate the mean satisfaction value.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Immutable features</th>
<th>Preferred constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>German Credit</td>
<td>Age, Sex, Country</td>
<td>Age and Education cannot decrease.</td>
</tr>
<tr>
<td>Adult Census</td>
<td>Marital-status, Race, Age</td>
<td>Age and Education cannot decrease. Increasing Education increases Age.</td>
</tr>
</tbody>
</table>

**Time:** We measure the time taken to generate a set of 5 CFEs for each input instance by `time` package. The average of these times across all instances is then used to determine the time of the method. A shorter duration indicates faster counterfactual production.

## 4 RESULTS

### 4.1 Instance Level Performance

We summarize the companions of CFEs for data points with some characteristics, such as a central data point or an outlier. In our analysis, we employed the mean centre approach to determine the central tendency of the dataset. The mean centre is calculated by averaging the values of all data points in each dimension. We then computed the Euclidean distance for each data point to the mean centre. A central data point in a dataset is the closest to the mean centre, and an outlier data point is the one that deviates the most from the mean centre. It’s worth noting that while we used the mean centre in this analysis, other methods, such as the median centre or geometric centre, could yield different combinations of central points and outliers. Analyzing multiple centres and outliers could provide a broader perspective, but for the scope of this study, we focused on the most pronounced outlier. We then compared the four CFE algorithms with respect to these two example data points.

**Adult Census Dataset:** Instance 766 is the closest to the mean centre of the Adult Census dataset. This instance and its counterfactuals are shown in Table 2. Notably, DiCE-KDTree failed to find valid counterfactuals. This means that, within the training set, there are no subsets that can fit in all constraints. For preferred conditions, although ‘Age’ is configured only
Table 2: A centre of Adult Census and its counterfactuals.

<table>
<thead>
<tr>
<th>Method</th>
<th>Age</th>
<th>Workclass</th>
<th>Education</th>
<th>Marital Status</th>
<th>Occupation</th>
<th>Relationship</th>
<th>Race</th>
<th>Sex</th>
<th>Capital Gain</th>
<th>Capital Loss</th>
<th>Hours per week</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiCE-Random</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DiCE-Genetic</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

upon analyzing the metrics from Table 6, DiCE-Random is the most time-efficient method, averaging 0.161 seconds per instance for 5 CFEs. Alibi-CFRL, however, excels in several metrics, boasting a superior validity score of 0.965 and a minimal average sparsity of 3.125, signifying its counterfactuals have fewer feature changes to original data points. Its constraint satisfaction score of 0.035 further establishes its abil-
### Table 3: An outlier of Adult Census and its counterfactuals.

<table>
<thead>
<tr>
<th>Method</th>
<th>Age</th>
<th>Workclass</th>
<th>Education</th>
<th>Marital Status</th>
<th>Occupation</th>
<th>Relationship</th>
<th>Race</th>
<th>Sex</th>
<th>Capital Gain</th>
<th>Capital Loss</th>
<th>Hours per week</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>27</td>
<td>Animals</td>
<td>Bachelor's</td>
<td>Never Married</td>
<td>White-Collar</td>
<td>Not-in-family</td>
<td>White</td>
<td>Male</td>
<td>0</td>
<td>2258</td>
<td>70</td>
<td>United-States</td>
</tr>
<tr>
<td>Alibi-CFRL</td>
<td>26</td>
<td>Bachelors</td>
<td>Bachelor’s</td>
<td>-</td>
<td>Blue-Collar</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>8165</td>
<td>2264</td>
<td>69</td>
<td>-</td>
</tr>
<tr>
<td>DiCE-Random</td>
<td>78</td>
<td>Associates</td>
<td>Bachelor’s</td>
<td>-</td>
<td>Blue-Collar</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>9526</td>
<td>2268</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DiCE-Genetic</td>
<td>59</td>
<td>Dropout</td>
<td>Bachelor’s</td>
<td>-</td>
<td>Blue-Collar</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>9666</td>
<td>2287</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DiCE-KDTree</td>
<td>36</td>
<td>Bachelors</td>
<td>Bachelor’s</td>
<td>-</td>
<td>Blue-Collar</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>9860</td>
<td>2264</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table 4: A centre of German Credit and its counterfactuals.

<table>
<thead>
<tr>
<th>Method</th>
<th>Credit History</th>
<th>Purpose</th>
<th>Credit Amount</th>
<th>Savings Account</th>
<th>Employment Size</th>
<th>TotalAnnualRate</th>
<th>Personal Status</th>
<th>Other Debts</th>
<th>Personal Residence</th>
<th>Property</th>
<th>Age</th>
<th>Other head's marital status</th>
<th>Rating</th>
<th>Term</th>
<th>People living in household</th>
<th>Telephone</th>
<th>Foreign Worker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>27</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>28</td>
<td>-</td>
<td>-</td>
<td>2205</td>
<td>24</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Alibi-CFRL</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>27</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DiCE-Random</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>24</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DiCE-Genetic</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DiCE-KDTree</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>39</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

5 DISCUSSION

5.1 Model Comparison

Comparing the results from the German Credit with Adult Census, the metrics show noticeable differences. Despite having fewer samples (1,000 VS 30,000) and more attributes (20 VS 12), the German Credit exhibits faster processing times for Alibi-CFRL. This could be attributed to the inherent complexities and relationships within the dataset. With its larger size, the Adult Census dataset might have being the fastest, stands out in following constraints with a score of 0.913. However, for those seeking minimal deviation from the original data in terms of sparsity, DiCE-Random would be the better choice.
Table 5: An outlier of German Credit and its counterfactuals.

<table>
<thead>
<tr>
<th>Method</th>
<th>Checking</th>
<th>Duration</th>
<th>Purpose</th>
<th>Credit Amount</th>
<th>Savings Account</th>
<th>Employment Status</th>
<th>Instalment Rate</th>
<th>Personal Rate</th>
<th>Amount</th>
<th>Account</th>
<th>Status</th>
<th>Age</th>
<th>Job</th>
<th>People Liable</th>
<th>Credit History</th>
<th>Existing Credit Score</th>
<th>People Liable</th>
<th>Employment Status</th>
<th>Purpose</th>
<th>Credit Amount</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>0-2000</td>
<td>42</td>
<td>all paid</td>
<td>5283</td>
<td>&lt;=1000 DM</td>
<td>unemployed</td>
<td>1 male single</td>
<td>none</td>
<td>2 unknown</td>
<td>51</td>
<td>bank free</td>
<td>free</td>
<td>yes</td>
<td>1 noaccount</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alibi-CFRL</td>
<td>-</td>
<td>46</td>
<td>critical</td>
<td>5225</td>
<td>100-500 DM</td>
<td>&gt;=7yrs</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>40</td>
<td>late pay</td>
<td>5213</td>
<td>500-1000 DM</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>43</td>
<td>early pay</td>
<td>5216</td>
<td>&gt;=7yrs</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DiCE-Random</td>
<td>-</td>
<td>11</td>
<td>critical</td>
<td>5283</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>10</td>
<td>critical</td>
<td>5283</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>25</td>
<td>critical</td>
<td>5283</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>15</td>
<td>critical</td>
<td>5283</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>36</td>
<td>critical</td>
<td>5283</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>28</td>
<td>critical</td>
<td>5283</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: SHAP force plots for the Adult Census dataset.

Table 6: Comparison of different methods on Adult Census.

<table>
<thead>
<tr>
<th>Method</th>
<th>Time(s)</th>
<th>Validity</th>
<th>Proximity_cont</th>
<th>Proximity_cat</th>
<th>Sparsity</th>
<th>Constraints</th>
<th>Manifold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alibi-CFRL</td>
<td>0.636</td>
<td>0.965</td>
<td>0.000</td>
<td>0.059</td>
<td>3.125</td>
<td>0.035</td>
<td>8.184</td>
</tr>
<tr>
<td>DiCE-Random</td>
<td>0.161</td>
<td>0.760</td>
<td>0.005</td>
<td>0.101</td>
<td>5.355</td>
<td>0.010</td>
<td>1.933</td>
</tr>
<tr>
<td>DiCE-Genetic</td>
<td>0.682</td>
<td>0.760</td>
<td>0.049</td>
<td>0.199</td>
<td>10.765</td>
<td>0.000</td>
<td>8.031</td>
</tr>
<tr>
<td>DiCE-KDTree</td>
<td>0.410</td>
<td>0.760</td>
<td>0.049</td>
<td>0.201</td>
<td>10.870</td>
<td>0.000</td>
<td>8.000</td>
</tr>
</tbody>
</table>

Table 7: Comparison of different methods on German Credit.

<table>
<thead>
<tr>
<th>Method</th>
<th>Time(s)</th>
<th>Validity</th>
<th>Proximity_cont</th>
<th>Proximity_cat</th>
<th>Sparsity</th>
<th>Constraints</th>
<th>Manifold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alibi-CFRL</td>
<td>0.073</td>
<td>1.000</td>
<td>0.000</td>
<td>0.204</td>
<td>2.652</td>
<td>0.913</td>
<td>13.784</td>
</tr>
<tr>
<td>DiCE-Random</td>
<td>0.224</td>
<td>1.000</td>
<td>0.028</td>
<td>0.127</td>
<td>1.792</td>
<td>0.761</td>
<td>2.681</td>
</tr>
<tr>
<td>DiCE-Genetic</td>
<td>2.609</td>
<td>1.000</td>
<td>0.186</td>
<td>0.841</td>
<td>11.857</td>
<td>0.000</td>
<td>13.429</td>
</tr>
<tr>
<td>DiCE-KDTree</td>
<td>1.882</td>
<td>1.000</td>
<td>0.186</td>
<td>0.869</td>
<td>12.236</td>
<td>0.000</td>
<td>13.000</td>
</tr>
</tbody>
</table>

more intricate data patterns, making counterfactual generation more time-consuming. Additionally, the increased number of attributes in the German Credit might lead to higher sparsity and proximity values, as more features can change in the generated counterfactuals. The differences in the datasets' nature, distribution, and inherent relationships likely contribute to the variations in the metrics observed.

When choosing CFE algorithms in practice, if the primary focus is constraint satisfaction and time efficiency, Alibi-CFRL is the best choice. It has a parameter ranges to set conditions for user-selected variables that can only be increased or decreased, whereas conditions in DiCE need to read the value of the instance and manually set the range of values for it. When inquiring CFEs of a single instance, giving con-
The constraints is not particularly troublesome. But when inquiring CFEs for many instances at once, a succinct way of setting conditions is worth considering. For those who value proximity to the original data distribution, DiCE-Random is likely to be the best all-rounder.

5.2 Failure of Finding Native CFEs

Viewing the failure of finding valid CFEs by DiCE-KDTree, relying solely on existing data points can inadvertently narrow the space of solutions. Existing data points might not capture the full spectrum of potential counterfactuals, leading to a constrained and potentially biased view. This bias is further exacerbated if the original datasets carry inherent prejudices based on their collection or curation methodologies.

Another concern is the potential for privacy breaches. Drawing counterfactuals from existing datasets might inadvertently expose sensitive or personal information (Goethals et al., 2023), especially if the datasets contain confidential data. This risk is accentuated in today’s data-driven world, where privacy preservation is paramount. For instance, a linkage attack is a malevolent effort that involves using background information to uncover the identity (i.e., re-identification) of a concealed entry in a released dataset (Vo et al., 2023).

On the other hand, although computationally intensive, synthetic CFEs offer a broader, more diverse exploration of scenarios without the associated privacy risks. Given these considerations, the inclination towards synthetic counterfactuals, free from existing dataset constraints and potential biases, appears prudent for comprehensive and ethical analysis. Therefore, when designing CFE algorithms, we need to consider pre-processing the data to ensure privacy and prevent the original data from being recognized or restored.

5.3 Comparison with Related Work

In our study, the performance of DiCE-KDTree is found to be the worst among the evaluated algorithms. However, unlike the complete failure to compute metrics as reported in (Verma et al., 2022) for the Adult Census and German Credit datasets, our evaluation showed that DiCE-KDTree can still output valid CFEs to a certain extent. Furthermore, while Verma et al. (2022) evaluated other DiCE algorithms on the Adult Census and German Credit datasets, they did not assess Alibi-CFRL. In contrast, Alibi-CFRL has been reported to exhibit outstanding performance in validity, sparsity, and manifold distance (Samoilescu et al., 2021). However, our findings extend this evaluation by considering proximity and constraint satisfaction, which were not addressed in previous papers. Lastly, even though we used an inverse measurement of proximity, our conclusion aligns with (Vo et al., 2023), indicating that DiCE-Random alters fewer features compared to DiCE-Genetic.

6 CONCLUSION

6.1 Contributions

In this work, we explored various CFE methods and compared their properties on the central data points, the outlier, and the unfavoured class of the dataset, which is ‘<$ 50k’ in Adult Census and ‘Bad’ in German Credit. For data preprocessing, we built an autoencoder using Artificial Neural Network to reduce the feature space. We then applied fine-tuned XGBoost for classification, reaching the weighted F1-score of 0.71 for German Credit and 0.87 for Adult Census.

We systematically compared DiCE-Random, DiCE-Genetic, DiCE-KDTree and Alibi-CFRL
across outliers and centres as sample instances in both datasets and compared their suggested changes with SHAP force plot. To have a broader insight, we further tested their time, validity, proximity, sparsity, constraint satisfaction and manifold distance on the unfavoured class. To conclude, we discovered
- Features of larger importance by SHAP are not the ones CFEs suggest to change in most the four CFE algorithms we evaluated.
- Alibi-CFRL has better performance on proximity and constraint satisfaction, while DiCE-Random has better performance on manifold distance. For datasets with more features, DiCE-Random uses less time to output the same amount of diverse CFEs.
- Synthetic CFEs perform better than native CFEs for data privacy and wider feature ranges because DiCE-KDTree fails to perform in various metrics.

6.2 Limitations and Future Work

However, our analysis comes with certain limitations listed as follows. Firstly, in terms of datasets, only tabular data have been considered. Secondly, our metric for assessing ordinal categorical proximity does not adequately capture the nuances of real-world scenarios. Specifically, while the measure can identify if a change has occurred, it does not differentiate between the magnitudes of different changes. For instance, the effort to be made in the real world altering the savings account from ‘unknown’ to ‘500-1kDM’ is substantially greater than transitioning to ‘<100 DM’. However, in our computations, where we adopted a simplified approach, these two transitions are treated as equivalent. This limitation warrants further refinement to ensure a more accurate representation of real-world implications in our analysis.

In future work, we have the following directions.
- Focus on causal constraints: More methods consider actionability as loss terms in the optimizing problem or reward function in the reinforcement learning process. Future work can design algorithms and metrics that can capture more causal constraint patterns. One possible direction is using Bayes Networks.
- Consider data privacy: Future research could delve into advanced data preprocessing techniques and the development of algorithms that inherently prioritize privacy.
- Conduct user studies: There could also be a possibility of building an interactive end-to-end XAI system where we display CFEs and let end-users decide which method fits them best.

ACKNOWLEDGEMENTS

This work is partially funded by the EPSRC CHAI project (EP/T026820/1).

REFERENCES


