CausalBioCF: Causal Counterfactuals for Machine Learning Interpretability

Gabriel Covello Furlanetto[1(✉)], Alexandro Baldassin[1], and Aleardo Manacero[2]

[1] Department of Statistics, Applied Mathematics and Computer Science, Universidade Estadual Paulista (UNESP), Avenida 24A, 1515 - Jardim Bela Vista, Rio Claro, SP 13506-900, Brazil
{gabriel.furlanetto,alexandro.baldassin}@unesp.br

[2] Department of Computer Science and Statistics, Universidade Estadual Paulista (UNESP), Rua Cristóvão Colombo, 2265 - Jardim Nazareth, São José do Rio Preto, SP 15054-000, Brazil
aleardo.manacero@unesp.br

Abstract. Machine learning methods have been widely used to support decision-making, but most of the time, decisions cannot be easily explained. Therefore, providing explanations about the results generated by them becomes important. This is particularly relevant in high-risk decision scenarios in order to protect all the participants, as occurs, for example, in financial applications such as credit analysis. This work proposes CausalBioCF, a new method of explainability of machine learning algorithms based on counterfactuals. CausalBioCF combines a low-cost bioinspired optimization technique with domain-knowledge causal relationships to find more viable counterfactuals. The main novelty of CausalBioCF is the integration of counterfactual generation methods and traditional causality analysis. Compared to state-of-the-art systems such as DiCE, our experimental evaluation shows that CausalBioCF performs favorably, particularly when metrics such as sparsity and distance are considered. Furthermore, we show that existing systems such as DiCE can be improved by taking into account causal relationships.

Keywords: Counterfactuals · Causal Inference · Genetic Algorithm

1 Introduction

Machine learning algorithms are gradually being used in many decision-making contexts [16]. Their applications are found in low-risk scenarios, such as in the indication of media content [11,42], and high-risk contexts, such as in medical [9] and financial fields [2]. Therefore, it is important that the choices based on the data generated by these systems are reliable and allow the understanding of how the system is making its predictions, requiring transparency in the algorithms developed.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
O. Gervasi et al. (Eds.): ICCSA 2025, LNCS 15649, pp. 200–217, 2025.
https://doi.org/10.1007/978-3-031-96997-3_13

In this context, the concept of counterfactual interpretability becomes relevant. According to Wachter et al. [48], a counterfactual is a slight modification in a set of facts that could lead to a different conclusion in a given analysis. Thus, providing an alternate factual scenario would help users understand why a specific result was reached and how it can be modified to a desired situation. As an example, consider the case of a 20-year-old single bank customer whose income is \$1000/month with high school education and whose application was denied (factual); it can be said that a counterfactual for this scenario would indicate that if his income was \$1500/month and education level was college, he would have his application approved.

Many counterfactual generation algorithms have been proposed since the work presented by Watcher et al. [48]. Among their main challenges are [50]: the *inefficiency*, given that proposed algorithms are dependent on optimizations and/or deep neural networks, which can be extremely slow; the necessity of *high-quality approximation* for a better explanation, which often ends up generating non-feasible disturbances of the original data; the necessity of taking into account the *relationship between variables*, such as the causal dependence between attributes, since it is essential for ensuring feasible counterfactuals; and the necessity for *insertion of domain knowledge* by experts to generate restrictions that ensure the feasibility of hypothetical situations.

The proposed system described in this paper, CausalBioCF, leverages techniques in the literature so that domain knowledge can be obtained and utilized in the generation of counterfactuals by means of bioinspired optimization algorithms. This approach is specifically designed for tabular data, ensuring that the counterfactuals generated are both meaningful and applicable within structured datasets. The domain knowledge generation algorithm uses statistical analysis to obtain data intervals/distribution and possible causal relationships between them [23]. This work serves as a foundation for using causal relationships to impose constraints that enhance the quality of the counterfactuals generated. The results obtained indicate that bioinspired optimization algorithms can be adapted to produce more feasible and contextually appropriate counterfactuals within the framework of the applied model. CausalBioCF tackles the aforementioned challenges, particularly the difficult task of finding causal relationships without requiring expert input.

CausalBioCF presents two main features. Firstly, it improves the task of obtaining domain knowledge for the application, thus eliminating or reducing human interference as much as possible. Secondly, it relies on bioinspired optimization methods, as they are simple to implement, more adaptable, and require less computational processing than *auto-encoders* that are commonly used to generate counterfactuals [29,52]. The main contributions of this work are:

- The use of causality as a key constraint for the generation of more feasible counterfactuals. Furthermore, we also show how causality can also improve the counterfactuals generated by other systems such as DiCE;
- An innovative approach that combines generation of counterfactuals through bioinspired algorithms and the definition of constraints through domain knowledge obtained with existing statistical techniques to improve the counterfactuals quality.

This work is organized as follows. Section 2 presents the background and related works. Section 3 describes the methodology used to develop this work. The setup for the experimental evaluation is described in Sect. 4, followed by the results and analysis in Sect. 5. Conclusions are presented in Sect. 6.

2 Background and Related Work

The field of counterfactual generation has gained increasing attention in recent years, driven by the growing demand for transparent and interpretable machine learning models [12,47]. Several approaches have been developed to provide users with actionable feedback on model decisions, particularly in high-risk domains such as healthcare and finance [2,9].

Among the works in the literature, Guidotti et al. [12] propose a classification of counterfactual generation methods into algorithm families, highlighting different approaches to generation. On the other hand, Verma et al. [47] take a broader perspective, analyzing counterfactual explanations in the context of interpretability and algorithmic recourse [22,46,47].

Based on these previous works, we can generally classify counterfactual generation methods in the literature into four main categories: optimization-based, perturbation-based, generative-based, and causal-based methods. This classification provides a structured way to analyze different approaches while highlighting the distinct strategies each category employs to ensure actionable and feasible counterfactuals. Each of these methods defines constraints differently. Optimization-based approaches enforce constraints through minimization limits of a loss function. Perturbation-based methods define constraints through functions that generate controlled modifications to the input data. Generative-based methods leverage deep learning models, such as transformers, Generative Adversarial Networks (GANs), or Autoencoders, to synthesize counterfactuals. Lastly, causal-based methods incorporate cause-effect relationships as intrinsic constraints to maintain causal validity.

This classification aligns with the perspectives of Guidotti et al. [12], who emphasize structural differences among counterfactual generation approaches. While their work categorizes methods into optimization-based, heuristic-based, instance-based, and decision tree-based, our classification extends this perspective by integrating perturbation and generative techniques, as well as explicitly considering causal constraints. Furthermore, Verma et al. [47] highlight the importance of evaluating counterfactuals beyond their generation process, advocating assessment criteria such as validity, sparsity, and minimal action. Rather than proposing a taxonomy, their work focuses on comparative evaluation frameworks. By incorporating these principles, our classification not only organizes counterfactual generation methods but also ensures that quality assessment remains integral to the discussion.

Thus, we structure the presentation of related works based on these four categories while using Verma et al.'s evaluation framework to guide the validation of the proposed approach.

2.1 Optimization-Based

This family is a union of optimization and heuristic-based families proposed by Guidotti et al.. Counterfactual generation is obtained through optimization strategies and uses a loss function to minimize the distance to the factual point as a minimization problem. These methods rely on mathematical equations to find optimal or near-optimal solutions. Another option of optimization is to find counterfactuals through local heuristic optimization, such as a genetic algorithm, that iteratively minimize a cost (loss) function. These methods are usually more efficient than the purely mathematical algorithms, but generally this efficiency comes at the cost of local solutions rather than global solutions. Some of the main works that exemplify this approach are **D**istribution-**A**ware **C**ounterfactual **E**xplanation (DACE) [20], **Di**verse **C**ounterfactual **E**xplanations (DiCE) [28], **C**ounterfactual **E**xplanations for **R**obustness, **T**ransparency, **I**nterpretability, and **Fai**rness (CERTIFAI) [44], and SkylineCF [49].

For linear programming algorithms [6], challenges include the need for high computational processing costs, especially in high-dimensional space problem and convergence to suboptimal counterfactuals, since algorithms get stuck in local minima. These algorithms also face challenges such as lack of plausibility, the generation of unrealistic counterfactuals, and difficulties in selecting the optimal objective function to satisfy all quality criteria, including proximity, plausibility, sparsity, and others. In contrast, heuristic-based optimization algorithms face fewer issues, are flexible and more time efficient but are still subject to provide less accurate or optimal solutions. Furthermore, it is necessary to define hyperparameters such as mutation rate or population size, which can significantly impact performance [12].

2.2 Perturbation-Based

Perturbation-based methods explore small variations in input attributes in order to identify plausible counterfactuals. The perturbations are generally generated by methods like a gradient or, in some cases, randomly. As an example of this approach, we can cite **N**earest **I**nstance **C**ountefactual **E**xplanations (NICE) [4], CounterFactuals Now (CFNOW) [31], **F**easible and **A**ctionable **C**ounterfactual **E**xplanations (FACE) [36], **G**enerating **RA**ndom **C**ontrastive sampl**E**s (GRACE) [26], and Growing Spheres [25]. The main challenge in these algorithms is to define the limitations of the generated perturbations, ensuring that the generated counterfactuals are plausible and feasible.

2.3 Generative

Generative approaches leverage neural networks or probabilistic models to synthesize realistic counterfactuals. Examples of generative methods are CounteR-GAN [29], and **C**ounterfactual **C**onditional **H**eterogeneous **A**utoencoder (CCH-VAE) [33]. The main challenges of these methods are ensuring the feasibility of counterfactuals, as generative models (such as GANs or VAEs) often produce

examples that do not fully align with real data distributions [47]. Furthermore, the high computational costs of training generative models and their complexity pose significant difficulties.

2.4 Causal

To address causal analysis, the concept of causality is defined in the sense that it must be based on premises rather than only on statistical associations or correlations, which can lead to erroneous conclusions. As suggested by Judea Pearl [35], associative and causal relationships can be distinguished as follows:

- **Associative relationship:** consists of any relationship of distributions observed together among a set of variables. Examples include correlation, dependence, conditional independence, etc.;
- **Causal relationship:** consists of relationships that cannot be defined only through a distribution. Examples include influence, effect, confusion, disturbance, etc.

It can be seen that causality is a complex concept but important for several areas of research, such as statistics, economics, epidemiology, computer science, and philosophy. According to Nogueira et al. [30], causality can be defined as the influence by which an event contributes to the production of other events. Thus, causality seeks to explain relationships between distinct events of which different variables are part. Other authors such as Pearce and Lawlor [34] also seek to define causality in a simplified way. According to them, in a case in which a variable X is a cause of a variable Y, Y depends on X to determine its value, that is, the variable Y uses the information provided by X to obtain the assumed value. This definition suggests that the variable X plays a causal role in determining the variable Y and that a change in X can result in a corresponding change in Y, indicating a cause-and-effect relationship between the variables. It is important to emphasize that this definition is based on a causal and dependent perspective between variables.

In this scenario, causality has been widely used to improve the performance of Machine Learning algorithms [35] as well. By capturing causal relationships between variables, mainly through graph-based methods such as Structural Causal Models (SCM) [35], some authors have proposed causal classifiers [32] that outperform traditional models.

Counterfactuals are also frequently used as a means of assessing causality or estimating outcomes over time. Methods for estimating time-varying outcomes were initially introduced in epidemiology and are heavily based on simple linear models [27]. The goal here is to estimate the average (non-individual) effects of treatment variations in time using counterfactuals.

Examples of research in this family that leverage causality to generate counterfactuals and enhance explainability in Machine Learning algorithms include the work of Shao et al. [43]. In this study, Structural Causal Models are utilized to measure causality between variables, producing counterfactuals with superior quantitative and qualitative evaluation metrics. However, as cited by the

authors, the primary limitation of this approach lies in its reliance on expert knowledge to generate SCM graphs.

2.5 CausalBioCF's Approach

Counterfactual generation existing methods often perform well in one area (e.g. proximity, sparsity, or feasibility) while compromising another (e.g., execution time or interpretability). Furthermore, bioinspired optimization techniques remain underexplored, presenting opportunities for more efficient algorithms. Our approach seeks to address these limitations by integrating bioinspired optimization techniques and obtaining domain knowledge from a causal model to generate feasible counterfactuals while maintaining computational efficiency. Unlike other works, CausalBioCF uses causality to replace expert-provided domain knowledge, employing the results of causal analysis to define mutable and immutable features to generate high-quality counterfactuals. The approach is part of the family of causal counterfactual generation algorithms.

3 CausalBioCF

The proposed approach, CausalBioCF, considers statistical and causality analysis techniques to automate the acquisition of domain knowledge. This knowledge is used as constraints, improving the quality of counterfactuals generated without the need for specialist knowledge.

Figure 1 shows the overall structure of the proposed approach. The main information that must be fed to CausalBioCF is as follows:

- **Classifier model and Model input:** classifier model is a black-box machine learning model or statistical function that takes input data (model input), such as a tuple, composed of numeric, categorical, or other data formats, to determine the class or the probability of belonging to a given class. In CausalBioCF, these components determine the class of the factual instance and check if the individuals generated by the genetic algorithm belong to the opposite class, forming a population of valid counterfactuals. For example, a credit analysis classifier takes inputs such as age, education, profession, and income and classifies the credit application as approved or denied based on learned patterns;
- **Data Sample:** A set of real labeled data that exemplifies the input and output scenarios for a machine learning model. This data will serve as the basis for calculating causal inference, ranges of continuous variables, possible categories for attributes, among other domain knowledge information.

From these inputs, CausalBioCF can infer the domain knowledge required by the classifier, through its causal and statistical analysis module, generate counterfactuals (counterfactual generating component) and deliver them, in the desired quantity, to the end user with model result, aiming to provide explanations for the model's decision.

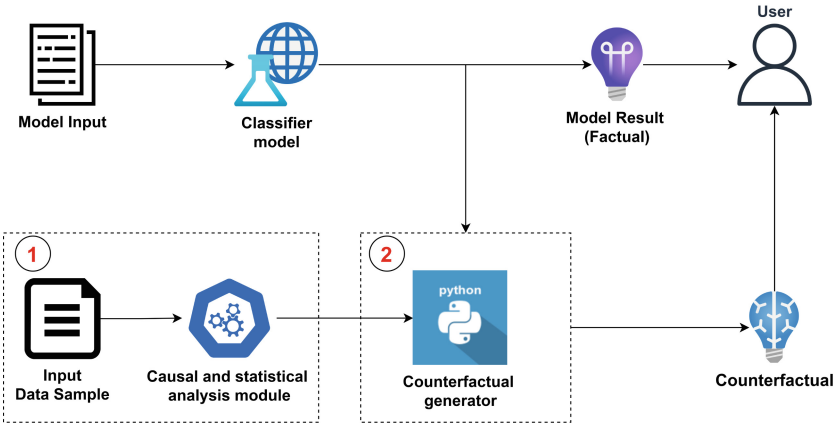

Fig. 1. CausalBioCF architecture.

The two main modules of CausalBioCF are presented using two numbered dashed boxes in Fig. 1. Module 1 includes the actions responsible for analyzing the features (input variables for the classification model) and the relationship between them through data samples. When we refer to relationships, this includes causal inference, variation intervals, possible categories for each feature, and the correlation between them. A more consistent sample will return better results in the form of constraints that will be used in the generation of counterfactuals. Module 2 is responsible for using the domain knowledge obtained from the previous module in order to generate viable, unique, diverse, and valid counterfactuals.

Thus, it can be noted that CausalBioCF addresses the challenge of manual intervention of specialists by using generated domain knowledge and ensuring that viable counterfactuals, closer to the factual, can be generated. Furthermore, this approach performs sufficiently fast with the use of a genetic algorithm.

In the following subsections, we describe in more detail each of the two main modules of CausalBioCF.

3.1 Causal Inference

In order to identify a good technique to select the best causal relationships among predictors and target variables, without the need for subjective definitions by experts, we evaluated some methods found in the literature. From this evaluation, we discarded the use of graphs such as the Structural Causal Model (SCM) [35], due to their high level of subjectivity, and the Inverse Propensity Weight (IPW) [41], due to the difficulty in defining confounders that may affect the problem.

Our choice was to adapt a method called Propensity Score Matching (PSM) [5,38] to guide the generation of counterfactuals in CausalBioCF. Propensity Score Matching (PSM) is a method used to reduce selection bias and estimate the

causal effect of a treatment. Four key steps are followed to assess causality. First, the propensity score is estimated for each individual using a logistic regression model. Individuals are then matched into treatment and control groups based on score similarity, using techniques such as Nearest-Neighbor or Caliper Matching. The balance assessment then ensures that both groups are comparable in terms of observable characteristics. Finally, the average treatment effect (ATE) is estimated by comparing outcomes between the matched groups [5,21].

Among the advantages of PSM are the reduction of selection bias and its flexibility, allowing it to be used in different research contexts. PSM was chosen because of your simplicity of the implementation, the possibility to automate causal analysis, and the corresponding run-time performance, since it tries to generate equivalent treatment and control groups (without bias or confounders) in order to calculate the Average Treatment Effect (ATE) and consequently the causality between the variables. Although we acknowledge that the model has limitations, such as only controlling observable variables and influencing the estimation of the propensity score, we understand that in the intended application scenario, these limitations are not as impactful compared to the benefit of achieving causal analysis independent of human-provided knowledge.

To explore PSM during counterfactual generation, the approach of Kline et al. [23] was adapted to calculate the ATE between model variables and target variable, discarding (as confounders) those that did not have a high causal relationship and using it to define what variables are mutable and what are immutable, decreasing computational cost, and reducing the risk of generating an unfeasible counterfactual.

3.2 Counterfactual Generation Algorithms

In order to use all the domain knowledge provided by causal analysis and overcome the challenge of using causality in counterfactual generation, we decided to work with optimization by bioinspired algorithms, due to their simplicity of implementation and computational efficiency compared to generative algorithms and traditional mathematical optimization. A Genetic Algorithm (GA) [15] approach was chosen.

Considering a genetic algorithm with steps such as population initialization, fitness evaluation, selection, crossover, and mutation [10,15], some modifications were necessary to generate causal counterfactuals. Among these changes, an important part was the modification of the initialization step. Instead of randomly generating the entire population, the initialization uses data based on samples required for the causal inference module. This adjustment aims to reduce the number of iterations needed to reach an optimal solution.

To make this possible, only individuals labeled as the counterfactual class are filtered from the data sample. They are then ordered according to the implemented objective function. If the total number of records from the desired class is smaller than the desired size for the population, the initial population is completed using random individuals of the desired class.

In addition, the other necessary change was in relation to the objective function used by the algorithm. The objective function is central to the method and directly depends on the definitions of sparsity and distance, given by Eq. 1 and Eq. 2, respectively.

$$sparsity = \sum_{i=1}^{n} 1(x_i \neq x'_i) \tag{1}$$

where n is the number of features in the dataset; x_i is the *i-th* feature of the factual instance; x'_i is the *i-th* feature of the counterfactual instance; $x_i \neq x'_i$ is the comparison that returns 0 if the features are different and returns 1 if the features of factual and counterfactual are the same.

For the distance definition, we used the Euclidean distance [8] as described by Eq. 2.

$$distance = \sqrt{\sum_{i=1}^{n} (x_i - x'_i)^2} \tag{2}$$

where x_i is the *i-th* feature of the factual instance and x'_i is the *i-th* feature of the counterfactual instance.

Finally, the objective function is given by Eq. 3.

$$\text{obj_function} = (w_1 \times \text{sparsity}) + (w_2 \times \text{distance}) \tag{3}$$

where w_1 and w_2 are weights used in the calculation. These weights consist of parameters passed to the genetic algorithm and start with equal values (1 for both) to ensure there is no distinction between the parameters initially. This initialization is done because the algorithm aims to improve not just one validation metric, but all of them. However, these parameters exist to allow the user to increase the emphasis on one metric over the other if desired, by adjusting their values.

The most critical challenge in adapting genetic algorithms to generate counterfactuals was defining the optimal method for generating the initial population, a key factor in ensuring the convergence of the objective function to an optimal solution within a reasonable execution time. CausalBioCF main algorithm is presented in Fig. 2.

The algorithm consists of two methods (Lines 5–10 and 13–36). The first, named "knowledgeDomain", processes the data sample provided by the user to extract domain knowledge for the given problem. This includes analyzing the intervals of numerical variables, identifying possible values of categorical variables, and performing the most complex task, the causal analysis. The second method, named "geneticAlgorithm", leverages the constraints established by the first method to execute the genetic algorithm while ensuring that these constraints are respected to generate feasible counterfactuals. Between them, causal analysis constraints define mutable and immutable features. Finally, Line 39 shows the invocation of the first module to acquire the knowledge domain (kd), followed by the counterfactual generation in Line 40.

```
Input: data_sample, factual_data, blackbox_model
Output: counterfactual_set

// Function for knowledge domain extraction
def knowledgeDomain(data_sample):
    Perform causal evaluation using PSM
    Calculate value ranges for continuous variables
    Calculate possible values for categorical variables
    Compute ATE and define mutables/immutables
    return kd

// Genetic Algorithm function
def geneticAlgorithm(data_sample, factual_data, knowledge_domain):
    actual_class = blackbox_model(factual_data)
    counterfactual_class = 1 - actual_class

    // Filter and initialize population
    Filter data_sample for counterfactual examples
    if no examples found:
        Generate initial population randomly
    if insufficient examples found:
        Use found examples and generate additional ones
    else:
        Use selected points as initial population

    // Genetic algorithm loop
    Initialize population
    Evaluate fitness: (w1 * sparsity) + (w2 * distance)
    for each generation:
        Select parents based on fitness
        Apply crossover to create descendants
        Mutate descendants using domain constraints
        Evaluate descendants  fitness
        descendants_class = blackbox_model(factual_data)
        Select best valid individuals for next generation
    return best counterfactual valid set

// Main execution
kd = knowledgeDomain(data_sample)
counterfactual_set = geneticAlgorithm(data_sample, factual_data, kd)
```

Fig. 2. CausalBioCF's main algorithm.

4 Experimental Setup

In this part, we evaluate our approach to counterfactual generation and compare it with Microsoft Research approach named DiCE [28]. DiCE, a Microsoft research project[1], was chosen because it is one of the most complete libraries found to date and compatible with the causal method proposed in CausalBioCF. In this way, we also have injected our causality method to define the constraints, such as mutable and immutable variables and ranges of numerical variables, for the generation of counterfactuals in Dice, which we refer to as *Causal DiCE*. It is important to note that in all tests DiCE was configured to generate counterfactuals through its own genetic algorithm, made available by the authors themselves.

There are other counterfactual generation systems that were also considered, but they were discarded because we did not have access to the source code,

[1] DiCE source code: https://github.com/interpretml/DiCE.

the code could not be easily adapted into our experimental framework, or they are from another family of generation algorithms. Among the systems we have considered are:

- **GeCO** [40]**:** developed in Julia, which made comparison with Python code difficult;
- **JUICE** [24]**:** uses a commercial and paid library called Gurobi to implement optimizers;
- **KAOGExp** [45]: The source code did not run as presented in the documentation;
- **DECE** [7]: Uses Node.js programming language to run an API, without batch explanations;
- **Certifai** [44]: The source code is not available;
- **GRACE** [26]: it only works with the explainability of neural networks and its original code generates only one counterfactual per execution. This execution format becomes incompatible with the test scenarios that we use, since we do not evaluate only the best counterfactual generated, but rather the set of the N best, as will be described in Sect. 4.2.

4.1 Datasets

The chosen datasets are among the most well-known datasets for counterfactual evaluation and have been used by most of the works in the area [47], including Credit Card Default [51], German Credit [14], Adult [1], Churn [18], and Compas Scores [37]. These datasets are related to various fields, such as finance, census, and healthcare, demonstrating the library's versatility across domains. The Diabetes [19] and Titanic [17] datasets were excluded due to insufficient records and features, making their results unreliable. It is important to note that the selected dataset are used to evaluate CausalBioCF only with binary classifiers.

4.2 Metrics

To evaluate the counterfactuals generated, quantitative metrics such as distance, execution time, sparsity, and validity are used. They are described below:

- **Distance or Proximity:** metric used to understand whether, through the generated restrictions, the counterfactuals remain close to the factual points. For this purpose, Euclidean (L2) [8] and Manhattan (L1) [3] distances are used;
- **Execution time:** running time in seconds of the counterfactual generation method;
- **Sparsity:** metric used to evaluate the number of features changed to produce counterfactuals. Better counterfactuals have fewer features changed;
- **Valid:** metric that evaluate whether a generated point is a counterfactual, or in other words, whether it is of the complementary or contrary class to the original point (factual) [47].

These metrics were compared together, in a combined manner, in order to ensure that the evaluation of the counterfactuals generated was as complete as possible.

All tests were performed on a machine with a 64-bit Windows 11 operating system, 16.0 GB DRAM, Intel(R) Core(TM) i7-1165G7, 2.80GHz. The black-box models evaluated in all experiments are a Random Forest (RF) [13] and a Multilayer Perceptron (MLP) [39]. It is important to emphasize that the training and testing of the models were done prior to the evaluation of the counterfactuals by the authors themselves, not using previously trained models. The entire source code developed to achieve this goal was written in Python 3.9.

To perform the validations, 10 different counterfactuals were generated for each of the 10 factuals randomly selected from a test dataset (10 repetitions of execution). In total, 100 counterfactuals per dataset per model were generated, or a total of 1000 counterfactuals. The results generated by this set of experiments are presented in the form of Average (Avg) and Standard Deviation. To plan test scenarios, our proposed methodology followed the schemes of Kuratomi et al. [24] and Brughmans et al. [4].

5 Results

In this section, we present the results obtained with CausalBioCF and DiCE from the analysis of the black-box models (MLP) and (RF) on the test data. We will analyze model accuracy, compare distances (Euclidean and Manhattan), evaluate valid counterfactuals, and the sparsity metric, concluding with the analysis of execution time. It is important to note that the best results for each row are highlighted in bold in Tables 1 and 2, which form the basis for the discussion we present next.

5.1 Model Accuracy

To evaluate counterfactual generation approaches, as mentioned previously, MLP and RF models are used. In Table 1 it is possible to check the accuracy of each of the models trained per dataset (last two columns).

From the analysis of the data, it is possible to see that all models had an accuracy above 60%. It is important to highlight that for the Churn and German Credit datasets, the Multilayer Perceptron model had performances that apparently refer to the overfitting condition. However, in order to confirm the model-agnostic behavior that counterfactuals have, these models were maintained, allowing readers to observe whether there are differences in the evaluated metrics due to this condition.

5.2 Distance

Considering Euclidean or Manhattan distance metrics, from the perspective that the closer a counterfactual outcome is to the factual, the better the result achieved, Table 1 shows that CausalBioCF presented better results compared to DiCE and Causal DiCE in most cases. It is important to note here that the causality restrictions inserted into DiCE, generating the Causal DiCE test version, ensure that its distance metrics are superior (smaller) to those of the original version of the algorithm. However, even with causal restrictions, DiCE does not outperform CausalBioCF due to factors such as the objective function, which optimizes sparsity and proximity for GA convergence, and the distinct initialization approach.

Table 1. Distance Metric (normalized distance)

Metric	Model	Dataset	Avg CBioCF	Avg. DiCE	Avg. Causal DiCE	Train Accuracy	Test Accuracy
Euclidian	MLP	Adult	**0.39** $\pm$ 0.09	0.74 $\pm$ 0.25	0.65 $\pm$ 0.30	88.52	84.52
Euclidian	MLP	Churn	**0.55** $\pm$ 0.15	0.97 $\pm$ 0.21	0.76 $\pm$ 0.29	99.67	87.43
Euclidian	MLP	Compas	**0.36** $\pm$ 0.13	0.53 $\pm$ 0.25	0.52 $\pm$ 0.20	68.80	69.42
Euclidian	MLP	Default	**0.45** $\pm$ 0.12	0.81 $\pm$ 0.19	0.57 $\pm$ 0.21	82.50	81.97
Euclidian	MLP	German	**0.45** $\pm$ 0.24	1.42 $\pm$ 0.25	0.80 $\pm$ 0.17	100.00	72.00
Euclidian	RF	Adult	**0.74** $\pm$ 0.22	0.99 $\pm$ 0.35	0.94 $\pm$ 0.37	83.63	82.27
Euclidian	RF	Churn	**0.52** $\pm$ 0.38	1.00 $\pm$ 0.14	0.75 $\pm$ 0.39	89.63	87.72
Euclidian	RF	Compas	**0.26** $\pm$ 0.10	0.38 $\pm$ 0.17	0.36 $\pm$ 0.16	68.51	68.28
Euclidian	RF	Default	**0.46** $\pm$ 0.17	0.81 $\pm$ 0.20	0.51 $\pm$ 0.21	81.90	81.47
Euclidian	RF	German	**0.70** $\pm$ 0.26	1.62 $\pm$ 0.20	0.86 $\pm$ 0.24	79.56	76.00
Manhattan	MLP	Adult	**0.58** $\pm$ 0.12	1.36 $\pm$ 0.61	1.12 $\pm$ 0.70	88.52	84.52
Manhattan	MLP	Churn	**0.82** $\pm$ 0.24	2.57 $\pm$ 0.51	1.31 $\pm$ 0.51	99.67	87.43
Manhattan	MLP	Compas	**0.38** $\pm$ 0.14	0.65 $\pm$ 0.32	0.65 $\pm$ 0.28	68.80	69.42
Manhattan	MLP	Default	**0.56** $\pm$ 0.12	1.81 $\pm$ 0.60	0.89 $\pm$ 0.42	82.50	81.97
Manhattan	MLP	German	**0.52** $\pm$ 0.27	3.42 $\pm$ 0.93	1.08 $\pm$ 0.27	100.00	72.00
Manhattan	RF	Adult	**1.28** $\pm$ 0.46	2.13 $\pm$ 0.86	1.85 $\pm$ 0.85	83.63	82.27
Manhattan	RF	Churn	**0.67** $\pm$ 0.51	2.69 $\pm$ 0.35	1.09 $\pm$ 0.63	89.63	87.72
Manhattan	RF	Compas	**0.27** $\pm$ 0.13	0.44 $\pm$ 0.22	0.42 $\pm$ 0.22	68.51	68.28
Manhattan	RF	Default	**0.68** $\pm$ 0.30	1.81 $\pm$ 0.56	0.84 $\pm$ 0.41	81.90	81.47
Manhattan	RF	German	**1.00** $\pm$ 0.51	4.15 $\pm$ 0.84	1.44 $\pm$ 0.62	79.56	76.00

5.3 Valid Counterfactuals and Quantities

When considering the number of counterfactuals generated and how many were valid, we mention that for the CausalBioCF algorithm, ten counterfactuals were

generated in all executions and all of them were valid. Meanwhile, for the DiCE and Causal DiCE methods, although all counterfactuals were valid, there was a respective generation of 9.9 and 9.5 counterfactuals on average for all executions, regardless of the black-box model executed. This occurs because Dice's stopping criteria do not always ensure the generation of all counterfactuals.

5.4 Sparsity

Regarding sparsity (Table 2), we noticed that the results are similar to the distance ones. Again, DiCE shows lower performance than CausalBioCF method, though this is somewhat improved when causal constraints are incorporated into its execution (Causal DiCE).

Table 2. Sparsity Metric

Model	Dataset	Avg CBioCF	Avg. DiCE	Avg. Causal DiCE
MLP	Adult	**3.04** $\pm$ 0.41	5.56 $\pm$ 1.37	3.88 $\pm$ 1.41
MLP	Churn	**4.61** $\pm$ 0.47	14.95 $\pm$ 0.84	4.92 $\pm$ 0.33
MLP	Compas	**1.17** $\pm$ 0.31	2.06 $\pm$ 0.47	2.02 $\pm$ 0.46
MLP	Default	**1.99** $\pm$ 0.03	15.09 $\pm$ 2.96	2.95 $\pm$ 0.76
MLP	German	**1.77** $\pm$ 0.39	8.47 $\pm$ 1.44	2.96 $\pm$ 0.42
RF	Adult	**3.82** $\pm$ 0.78	7.42 $\pm$ 1.22	5.35 $\pm$ 1.20
RF	Churn	**2.47** $\pm$ 0.48	14.09 $\pm$ 0.56	3.37 $\pm$ 0.64
RF	Compas	**1.16** $\pm$ 0.31	1.82 $\pm$ 0.40	1.86 $\pm$ 0.40
RF	Default	**2.69** $\pm$ 0.62	16.77 $\pm$ 1.50	3.13 $\pm$ 0.76
RF	German	**2.31** $\pm$ 0.89	9.04 $\pm$ 1.20	3.37 $\pm$ 0.89

5.5 Execution Time

When comparing the execution times of the algorithms, excluding cases where the genetic algorithm does not reach the optimization criteria for its objective function in both counterfactual generators, we have noticed that counterfactuals generation in DiCE took around 1 s most of the times, making it 3.5x faster than CausalBioCF overall. This performance is further optimized when the causality information is incorporated. Specifically, Causal DiCE (on average 0.88 s of exeution to generate counterfactuals) is about 4 times faster than CausalBioCF and 1.15 times faster than the original version of DiCE.

This better execution performance can be justified by the difference in stopping criteria between DiCE and the proposed work. Although DiCE employs

a criteria based on sparsity and proximity loss functions, along with the number of generations and the quantity of counterfactuals generated, CausalBioCF relies solely on the number of generations that are executed until completion, disregarding other criteria.

5.6 Discussions and Future Work

Among the analyses, sparsity and distance are important, ensuring that counterfactual results are close to the factual. Additionally, generating valid counterfactuals within acceptable execution times is crucial for practical use and must provide at least one viable option.

As mentioned, counterfactual explanations are model-agnostic. Thus, even though we have two datasets (Churn and German Credit) in which MLP-trained models showed a tendency of overfitting, the overall trends in counterfactual metrics remained unchanged.

Based on these conclusions and on the results presented, it is evident that the proposed work meets most of the criteria, often exceeding DiCE, an established algorithm in the literature. This is possible due to the restrictions generated by the knowledge domain acquisition module developed, which can be used not only in the proposed algorithm, but also expanded to other works that accept this type of constraints as input.

Other factors that contribute to this are the proposed objective function, which considers sparsity and proximity criteria for the convergence of the genetic algorithm, as well as the different way of initializing the GA, as previously mentioned.

Furthermore, although DiCE and Causal DiCE displayed better overall execution times, CausalBioCF still generates counterfactuals in acceptable times (about 3.5 s on average). Also note that we did not strive to optimize the Python code as this was not our main goal at this stage.

In general, we understand that there are still possible areas to explore and challenges to overcome. In order to achieve these goals, future work includes the following: expanding quantitative testing scenarios to take into account metrics related to the quality and feasibility of the counterfactuals generated; comparing the developed library with modules from other libraries that use another generation algorithms like NICE [4] and CFNOW [31]; and finally, expanding the possibility of generating counterfactuals to multiclass datasets.

The last point mentioned, which proposes using the algorithm in multiclass datasets, perhaps brings the greatest challenge as future work, which consists of adapting the causal inference strategy so that it can analyze relationships with more than 2 treatment groups.

6 Conclusions

In this paper we presented CausalBioFC, a bioinspired approach to counterfactuals generation that takes into account causality. The possibility of using

causal inference tends to generate better counterfactuals, resulting in less sparse results, closer to the factual. Moreover, our results compare favorably against DiCE when evaluating metrics such as distance and sparsity, showing the potential of CausalBioFC.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Becker, B., Kohavi, R.: Adult. UCI Machine Learning Repository (1996). https://doi.org/10.24432/C5XW20
2. Bin Sulaiman, R., Schetinin, V., Sant, P.: Review of machine learning approach on credit card fraud detection. Human-Centric Intell. Syst. **2**(1–2), 55–68 (2022)
3. Black, P.E.: Dictionary of algorithms and data structures. Paul E, Black (1998)
4. Brughmans, D., Leyman, P., Martens, D.: Nice: an algorithm for nearest instance counterfactual explanations. Data mining and knowledge discovery, pp. 1–39 (2023)
5. Caliendo, M., Kopeinig, S.: Some practical guidance for the implementation of propensity score matching. J. Econom. Surv. **22**(1), 31–72 (2008)
6. Carrizosa, E., Ramírez-Ayerbe, J., Morales, D.R.: Mathematical optimization modelling for group counterfactual explanations. Eur. J. Oper. Res. **319**(2), 399–412 (2024)
7. Cheng, F., Ming, Y., Qu, H.: Dece: Decision explorer with counterfactual explanations for machine learning models. IEEE Trans. Visual Comput. Graphics **27**(2), 1438–1447 (2020)
8. Danielsson, P.E.: Euclidean distance mapping. Comput. Graphics Image Process. **14**(3), 227–248 (1980)
9. Fatima, M., Pasha, M., et al.: Survey of machine learning algorithms for disease diagnostic. J. Intell. Learn. Syst. Appl. **9**(01), 1 (2017)
10. Golberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addion wesley **1989**(102), 36 (1989)
11. Gomez-Uribe, C.A., Hunt, N.: The Netflix recommender system: algorithms, business value, and innovation. ACM Trans. Manage. Inform. Syst. (TMIS) **6**(4), 1–19 (2015)
12. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. Data Min. Knowl. Disc. **38**(5), 2770–2824 (2024)
13. Ho, T.K.: Random decision forests. In: Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 1, pp. 278–282. IEEE (1995)
14. Hofmann, H.: Statlog (German Credit Data). UCI Machine Learning Repository (1994). https://doi.org/10.24432/C5NC77
15. Holland, J.H.: Genetic algorithms. Scientific American **267**(1), 66–73 (1992)
16. Jin, W.: Research on machine learning and its algorithms and development. J. Phys.: Conf. Series **1544**(1), 012003 (May 2020). https://doi.org/10.1088/1742-6596/1544/1/012003
17. Kaggle: Titanic - machine learning from disaster. https://www.kaggle.com/competitions/titanic (2012). Accessed 04 Oct 2024
18. Kaggle: Telco customer churn. https://www.kaggle.com/datasets/blastchar/telco-customer-churn (2020). Accessed 04 Oct 2024

19. Kahn, M.: Diabetes. UCI Machine Learning Repository (1994). https://doi.org/10.24432/C5T59G
20. Kanamori, K., Takagi, T., Kobayashi, K., Arimura, H.: DACE: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In: IJCAI, pp. 2855–2862 (2020)
21. Kane, L.T., et al.: Propensity score matching: a statistical method. Clin. Spine Surg. **33**(3), 120–122 (2020)
22. Karimi, A.H., Von Kügelgen, J., Schölkopf, B., Valera, I.: Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. Adv. Neural. Inf. Process. Syst. **33**, 265–277 (2020)
23. Kline, A., Luo, Y.: PSMPY: a package for retrospective cohort matching in Python. In: 2022 44th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1354–1357. IEEE (2022)
24. Kuratomi, A., Miliou, I., Lee, Z., Lindgren, T., Papapetrou, P.: Juice: Justified counterfactual explanations. In: Discovery Science: 25th International Conference, DS 2022, Montpellier, France, October 10–12, 2022, Proceedings, pp. 493–508. Springer (2022)
25. Laugel, T., Lesot, M.J., Marsala, C., Renard, X., Detyniecki, M.: Comparisonbased inverse classification for interpretability in machine learning. In: Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations: 17th International Conference, IPMU 2018, Cádiz, Spain, June 11-15, 2018, Proceedings, Part I 17, pp. 100–111. Springer (2018)
26. Le, T., Wang, S., Lee, D.: Grace: Generating concise and informative contrastive sample to explain neural network model's prediction. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 238–248 (2020)
27. Melnychuk, V., Frauen, D., Feuerriegel, S.: Causal transformer for estimating counterfactual outcomes. In: International Conference on Machine Learning, pp. 15293–15329. PMLR (2022)
28. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 607–617 (2020)
29. Nemirovsky, D., Thiebaut, N., Xu, Y., Gupta, A.: Countergan: generating counterfactuals for real-time recourse and interpretability using residual GANs. In: Uncertainty in Artificial Intelligence, pp. 1488–1497. PMLR (2022)
30. Nogueira, A.R., Pugnana, A., Ruggieri, S., Pedreschi, D., Gama, J.: Methods and tools for causal discovery and causal inference. Wiley Interdisc. Rev.: Data Mining Knowl. Discov. **12**(2), e1449 (2022)
31. de Oliveira, R., Sörensen, K., Martens, D.: A model-agnostic and data-independent tabu search algorithm to generate counterfactuals for tabular, image, and text data. Eur. J. Oper. Res. **317**(2), 286–302 (2024)
32. O'Brien, A., Kim, E., Weber, R.: Investigating causally augmented sparse learning as a tool for meaningful classification. In: 2023 IEEE Sixth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), pp. 33–37. IEEE (2023)
33. Pawelczyk, M., Broelemann, K., Kasneci, G.: Learning model-agnostic counterfactual explanations for tabular data. In: Proceedings of The Web Conference 2020, pp. 3126–3132 (2020)
34. Pearce, N., Lawlor, D.A.: Causal inference–so much more than statistics. Int. J. Epidemiol. **45**(6), 1895–1903 (2016)

35. Pearl, J.: Causality. Cambridge university press (2009)
36. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., Flach, P.: Face: feasible and actionable counterfactual explanations. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pp. 344–350 (2020)
37. ProPublica: Compas recidivism risk score data and analysis. https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis (2016). Accessed 04 Oct 2024
38. Rosenbaum, P.R., Rubin, D.B.: The central role of the propensity score in observational studies for causal effects. Biometrika **70**(1), 41–55 (1983)
39. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
40. Schleich, M., Geng, Z., Zhang, Y., Suciu, D.: GECO: Quality counterfactual explanations in real time. arXiv preprint arXiv:2101.01292 (2021)
41. Seaman, S.R., White, I.R.: Review of inverse probability weighting for dealing with missing data. Stat. Methods Med. Res. **22**(3), 278–295 (2013)
42. Shani, G., Gunawardana, A.: Evaluating recommendation Systems. In: Recommender Systems Handbook, pp. 257–297 (2011)
43. Shao, X., Wang, H., Chen, X., Zhu, X., Zhang, Y.: Cube: causal intervention-based counterfactual explanation for prediction models. IEEE Trans. Knowl. Data Eng. **36**(6), 2416–2429 (2023)
44. Sharma, S., Henderson, J., Ghosh, J.: Certifai: A common framework to provide explanations and analyse the fairness and robustness of black-box models. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pp. 166–172 (2020)
45. da Silva, A.T., Bertini, J.R.: Using the k-associated optimal graph to provide counterfactual explanations. In: 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–8 (2022). https://doi.org/10.1109/FUZZ-IEEE55066.2022.9882751
46. Spangher, A., Ustun, B., Liu, Y.: Actionable recourse in linear classification. In: Proceedings of the 5th Workshop on Fairness, Accountability and Transparency in Machine Learning (2018)
47. Verma, S., Boonsanong, V., Hoang, M., Hines, K., Dickerson, J., Shah, C.: Counterfactual explanations and algorithmic recourses for machine learning: A review. ACM Comput. Surv. **56**(12), 1–42 (2024)
48. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harv. JL Tech. **31**, 841 (2017)
49. Wang, Y., et al.: The skyline of counterfactual explanations for machine learning decision models. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 2030–2039 (2021)
50. Yang, F., Alva, S.S., Chen, J., Hu, X.: Model-based counterfactual synthesizer for interpretation. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 1964–1974 (2021)
51. Yeh, I.C.: Default of credit card clients. UCI Machine Learning Repository (2009). https://doi.org/10.24432/C55S3H
52. Zhang, W., Barr, B., Paisley, J.: An interpretable deep classifier for counterfactual generation. In: Proceedings of the Third ACM International Conference on AI in Finance, pp. 36–43 (2022)