Information Sciences 701 (2025) 121844

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Overlap Number of Balls Model-Agnostic CounterFactuals (ONB-MACF): A data-morphology-based counterfactual generation method for trustworthy artificial intelligence

José Daniel Pascual-Triana [a,*], Alberto Fernández [a,*], Javier Del Ser [b,c,a], Francisco Herrera [a]

[a] Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Granada, 18071, Spain
[b] TECNALIA, Basque Research & Technology Alliance (BRTA), Derio, 48160, Spain
[c] University of the Basque Country (UPV/EHU), Leioa, 48940, Spain

ARTICLE INFO

Keywords:
Explainable artificial intelligence
Model-agnostic explanations
Counterfactual analysis
Data morphology
Trustworthy artificial intelligence

ABSTRACT

Explainable Artificial Intelligence (XAI) is a pivotal research domain aimed at clarifying AI systems, particularly those considered "black boxes" due to their complex, opaque nature. XAI seeks to make these AI systems more understandable and trustworthy, providing insight into their decision-making processes. By producing clear and comprehensible explanations, XAI enables users, practitioners, and stakeholders to trust a model's decisions. This work analyses the value of data morphology strategies in generating counterfactual explanations. It introduces the Overlap Number of Balls Model-Agnostic CounterFactuals (ONB-MACF) method, a model-agnostic counterfactual generator that leverages data morphology to estimate a model's decision boundaries. The ONB-MACF method constructs hyperspheres in the data space whose covered points share a class, mapping the decision boundary. Counterfactuals are then generated by incrementally adjusting an instance's attributes towards the nearest alternate-class hypersphere, crossing the decision boundary with minimal modifications. By design, the ONB-MACF method generates feasible and sparse counterfactuals that follow the data distribution. Our comprehensive benchmark from a double perspective (quantitative and qualitative) shows that the ONB-MACF method outperforms existing state-of-the-art counterfactual generation methods across multiple quality metrics on diverse tabular datasets. This supports our hypothesis, showcasing the potential of data-morphology-based explainability strategies for trustworthy AI.

1. Introduction

The expansive application of AI systems underscores the necessity for these systems to not only perform with exceptional accuracy but to do so with transparency, auditability, and accountability at every stage of their life-cycle [14]—from data collection and model training to deployment and ongoing validation. Regulatory frameworks like the European Union Regulation 2024/1689[1] and

* Corresponding authors.
E-mail addresses: jdpascualt@ugr.es (J.D. Pascual-Triana), alfh@ugr.es (A. Fernández).

[1] Regulation (EU) 2024/1689 of the European Parliament and of the Council, http://data.europa.eu/eli/reg/2024/1689/oj.

https://doi.org/10.1016/j.ins.2024.121844
Received 11 May 2024; Received in revised form 7 November 2024; Accepted 29 December 2024

Available online 8 January 2025

initiatives such as the United States National Institute of Standards and Technology's (NIST) framework for AI,[2] emphasise the critical need for AI technologies that are innovative, explainable, responsible and reliable.

In this context, Explainable AI (XAI) is an essential research area whose purpose is to ease the understanding of complex AI systems to a given audience [4,21]. With the clarification of AI decision-making processes, XAI seeks to improve their transparency and understandability [12]. XAI encompasses a broad spectrum of methodologies, ranging from inherently transparent models that allow for straightforward interpretation of their processes, to post-hoc explanatory tools designed to elucidate the workings of otherwise opaque models [23,1]. Moreover, the field distinguishes between model-specific explainers, tailored for specific types of models [17], but also model-agnostic ones, which provide explanations across different types of classification models.

To elucidate the decision-making processes, post-hoc explanations through factual, semifactual, and counterfactual instances stand out for their ability to validate models [31]. Factual explanations delineate reasons behind a model's decision for a specific case [15]. Semifactual explanations explore hypothetical scenarios where a sample's class remains unchanged despite feasible modifications, offering understanding of the model's response to input changes [3,30]. Counterfactual explanations pivot on the minimal feasible changes needed to alter a model's decision, providing dual utility: they empower users seeking to understand or challenge decisions and facilitate model auditing by revealing feature importance and potential biases or unrealistic feature values [16].

Counterfactual explanations hinge on a suite of features that collectively determine their quality and utility [10]. Primarily, a counterfactual must belong to a different class, providing an alternative to the model's decision. Minimalistic intervention advocates for altering a concise set of features to avoid unnecessary complexity and enhance interpretability. The counterfactual must be close to the original instance (involving minimal yet effective changes), realistic and representative of the data distribution. It should only alter features that can be realistically modified, avoiding immutable characteristics. Moreover, offering multiple pathways for action or understanding in generated counterfactuals allows users to evaluate their preferred course of action.

Despite their utility, current approaches to generating counterfactuals face limitations, as discussed in [33]. Many rely on model-specific methodologies, restricting their applicability [8]. Even when the design is model-agnostic, it does not always properly estimate the class boundaries [19], which complicates the generation of close, minimal-change counterfactuals that adhere to actual data distributions. Furthermore, existing techniques often overlook the treatment of protected or immutable features, rendering some counterfactuals impractical or unethical [18]. Finally, the inability to provide sets of diverse counterfactuals when needed restricts the depth of insight into the model's behaviour across possible outcomes.

Our hypothesis is based on the potential usefulness of data-morphology-based explainability strategies for trustworthy AI and, in particular, in the generation of counterfactual explanations. This hypothesis is associated with a method that embodies the ONB methodology [27], derived from geometry-based mathematical tools [22] and the morphology-based complexity metric, which facilitates coverings of tabular datasets by encapsulating data clusters into "balls" (the interior of hyper-spheres).

This work proposes the Overlap Number of Balls Model Agnostic CounterFactual method (ONB-MACF), a novel explanation strategy leveraging data morphology to delineate a model's class boundaries.[3] It is based on the construction of the aforementioned balls and involves a mapping strategy, where each ball encompasses instances from a single class based on the classifier's predictions and a predefined distance function. This mapping not only approximates complex class boundaries but also identifies pathways for counterfactual transitions, leveraging the proximity and geometry of balls associated with alternative class labels. When a counterfactual is sought for a specific input data instance, our approach harnesses this geometrically enriched map to navigate towards the most plausible counterfactual candidates, starting from the boundary closest to the original instance and moving towards the projected centre of an adjacent class's ball.

By construction, ONB-MACF is a model-agnostic counterfactual explainer adaptable to class boundary irregularities. Utilising the ball boundaries as a basis, the methodology can simultaneously maximise different requirements. Specifically, we ensure the feasibility and foster minimal feature modification and low distance from the original sample. This is achieved when carrying out small modifications towards the projected ball centre, obtained after enforcing immutable feature constraints, while mitigating unnecessary feature alterations. Finally, the methodology's prowess extends to generating plausible semifactuals, offering an additional layer of explainability.

To analyse the good capabilities of the ONB-MACF method, we conduct a comprehensive evaluation from a double perspective, both quantitatively and qualitatively. Our study employs pre-trained neural networks as baseline classification models for 8 classification problems, each with distinct characteristics and widely recognised in counterfactual studies. Comparison methodologies include a wide selection of explainers renowned for their efficacy in counterfactual generation [28]. Amongst them, we highlight the use of Growing Spheres [20] and NICE [6], methods that share similarities with the ONB-MACF method in their reliance on hyperspheres and prototypes, respectively. Despite their points in common, ONB-MACF makes use of hyperspheres' morphology, instead of employing them as a limit for the random candidate generation of Growing Spheres, and creates its own (closer) prototypes in addition to existing ones, while NICE only uses the latter. Quantitative performance is measured across a set of well-established metrics [34], each tailored to assess critical aspects of counterfactual quality; the results indicate that counterfactuals generated using the ONB-MACF method are close and in distribution, involve few feature changes and avoid changes in immutable features. Likewise, a complementary qualitative analysis is carried out to show the good behaviour of our approach and the usefulness of the provided explanations, such as using the appropriate features for the class changes and avoiding the protected features.

[2] NIST Framework for AI, https://www.nist.gov/artificial-intelligence, last accessed 12 April 2024.

[3] The code concerning the ONB-MACF method can be provided upon request.

The rest of this paper is structured in the following manner. Firstly, the preliminaries and state of the art on counterfactual explanations will be introduced in Section 2. Then, the proposed ONB-MACF method will be detailed in Section 3. Next, the experimental framework for the comparison against the state of the art will be described in Section 4. Afterwards, the quantitative analysis from the comparison of ONB-MACF with other benchmark methods will be explained in Section 5 (per-dataset results tables are shown in Appendix A). The qualitative analysis of the ONB-MACF method and its comparison to the aforementioned methods will be provided in Section 6. Finally, the concluding remarks and some paths for future work will be presented in Section 7.

2. Preliminaries on counterfactual explanations

The structure of this section, which serves as an introduction to counterfactual explanations, is the following. Firstly, the basics of counterfactual explanations, their concept, benefits and expected qualities, are explained in Section 2.1; then, a reference to the related works is included in Section 2.2.

2.1. Fundamentals of counterfactual explanations

Counterfactual explanations are a type of post-hoc technique aimed to explain the decisions issued by a classifier. In general, a counterfactual explanation (or counterfactual) for a given instance is another instance from the same data distribution that results from the injection of some small, plausible changes into the original data instance, but for which the classifier predicts a different class. The name “counterfactual explanation” can also be used to simply refer to the differences between the initial data instance and its given counterpart.

Counterfactuals began to take importance for machine learning as an explanation technique due to, amongst other characteristics, their aforementioned aptitudes for challenging decisions and closeness to human thinking.

In fact, this type of explanation is usually considered understandable by humans, as it is contrastive (it gives information about the critical features for the decision) and can be expressed in a few terms involving data features and values, which, most often, use natural language. Moreover, not only does it give insight into how the decision was made, but it also indicates which kinds of small changes would lead to a different outcome in an intuitive way. Thus, users know what they should modify in order to challenge the decision that, for example, denies them a loan. Furthermore, they can be a precious tool in auditing models, as a developer can easily read the explanation and see whether the model uses the right features to decide the class change. This is particularly important from the fairness and ethics standpoints, as the presence of sensitive/protected features in a counterfactual explanation can explicitly point out classification bias, which could lead to a need for accountability.[4]

For the effectiveness of counterfactuals as high-quality, understandable explanation techniques, there are several recommended or even mandatory characteristics to fulfil [16], [34], [8], from which the objective ones, and therefore those that can mostly be measured, are enumerated below.

- Validity: the counterfactual’s class must differ from the sample’s.
- Sparsity: a good counterfactual should change few features. There is no definitive consensus on the optimum number of changes, but it is recommended to change only 1-3 features [19].
- Similarity: a counterfactual must be close to the sample, and this closeness could be measured using different distance metrics.
- Actionability: a good counterfactual must not change features that cannot be acted upon, such as immutable features.
- Plausibility: a counterfactual should be in-distribution according to the observable data and it should not be considered an outlier.
- Causality: a good counterfactual should respect the causal relations from the dataset, as otherwise, it might not be plausible.
- Diversity: if multiple counterfactuals are given for one sample, they should be as diverse as possible, changing different features so that distinct actions can be taken.

2.2. Related work

This section presents a brief reference to the state of the art on counterfactual explanations. The counterfactual methods are organised with respect to different key aspects such as model specificity, technique, information requirements (from data or model) and target data type. Should the reader want to know more about individual topics, there have been multiple surveys on the matter in recent years [33,16,34,8,19].

According to model specificity, some techniques require a particular type (such as a decision tree or an ensemble [17]) due to using the model’s structure and parameters. Others accept any differentiable model, like Diverse Counterfactual Explanations (DiCE) [25], which applies optimisation techniques based on cost functions to generate sets of feasible and diverse feature changes to swap an instance’s class. Other methods work on any model (model-agnostic approaches), like Feasible and Actionable Counterfactual Explanations (FACE) [29], which finds viable paths connecting instances from different classes in graphs generated using kNN or ϵ-graphs. Model-specific approaches harness the intrinsic characteristics of the model to their advantage in exchange for being useful in fewer situations. Lately, model-agnostic approaches have been favoured due to their universality [16].

[4] Regulation (EU) 2024/1689 of the European Parliament and of the Council, http://data.europa.eu/eli/reg/2024/1689/oj.

Regarding the techniques for their implementation, several strategies have been used. Most strategies are based on optimisation. Some of them are cost-based, like Contrastive Explanations Method (CEM) [11], which uses iterative algorithms to detect which attribute values are positively and negatively related to a class. Other optimisation techniques are restriction-based, like the aforementioned DiCE [25], or probability-based, like Counterfactual Latent Uncertainty Explanations (CLUE) [2], which employs probabilistic differentiable models to study how instance changes (respecting data distributions) increase the probability of it belonging to a different class. There are also heuristic approaches like Growing Spheres (GS) [20], which generates uniformly distributed new instances on growing hyperspheres centred on the given instance. Less-explored strategies are instance-based. These can involve prototypes, like NICE [6], which swaps attribute values between the instance and members of the opposite class until the class changes. Other instance-based strategies use distances or graphs [29]) or involve surrogate models, like in [26], where they approximate model boundaries by learning a generative adversarial network and generate counterfactuals linearly in the boundary's direction.

Distinct counterfactual explainers also need access to different information. Some models might require access to training data (for example, instance-based explainers [6]). While all counterfactual explainers need access to parts of the prediction model, some of them require all information within the model [17], whereas others, such as the one proposed in [35], use optimisation techniques with weights and the restrictions and, thus, need access to the gradients. Some model-agnostic methods only require the outputs from the prediction function, like Counterfactual Recourse Using Disentangled Subspaces (CRUDS) [13], which uses autoencoders to obtain feasible and interpretable feature changes for an instance that modify its class while respecting causality and other restrictions.

Besides that, not all counterfactuals are created with the same target data. Counterfactual generators have generally been designed to explain models that work on tabular data [6]. Some methods explain images, such as in [10], where a generative adversarial network is used to produce counterfactuals with a focus on plausibility, the intensity of changes and adversarial power, or in [26], where the disentangled latent space of a generative adversarial network is employed to modify the instance towards the other class linearly. In a less explored manner, some methods have been designed for texts, such as in [36], where sentences can be modified in diverse ways using GPT-2, or in [24], where neural network characteristics and feature importance are used to generate counterfactual words in sentences. While some explainers work for any of them [29,9], most have been created for specific types to preserve their simplicity.

Since the quality of counterfactual explanations can be gauged in terms of many different metrics, it is very hard for a counterfactual generator to be considered superior to any other in all aspects simultaneously. While DiCE [25] can obtain multiple counterfactuals that are different enough from each other, sometimes the boundary directions it chooses might not be optimal for single explanations. FACE [29] uses graphs to find feasible changes, but it happens to the detriment of counterfactual distance. CEM [11] does not include restrictions regarding which features can be changed, giving rise to unfeasible counterfactuals. Despite considering them, CLUE [2] does not correctly enforce actionability constraints. The randomness of Growing Spheres affects the method's robustness and can lead to counterfactuals that change all actionable attributes, which is unfeasible for the user. While NICE [6] is fast, due to simply swapping attribute values between instances, it could get lower distances by employing the class boundary between both points. Wachter [35] optimises the distance without limiting the number of attributes to change, which can lead to unfeasible counterfactuals. CRUDS [13] disregards the geometry of the class boundaries, which can lead to counterfactuals further from the sample.

3. Overlap Number of Balls Model-Agnostic CounterFactuals

This section presents "Overlap Number of Balls Model-Agnostic CounterFactual" (ONB-MACF), a novel, model-agnostic, prototype-based counterfactual method developed for usage on tabular data. Unlike other counterfactual methods, ONB-MACF employs data morphology, which helps delineate the boundaries amongst the classes. This fact helps the ONB-MACF method generate close, sparse counterfactuals that follow the data distribution.

Firstly, the algorithm's inner workings are presented in Section 3.1; then, the advantages of the process followed are stated in Section 3.2.

3.1. Structure of the ONB-MACF method

The ONB-MACF method generates counterfactual explanations using a data-covering strategy from the Pure Class Cover Catch Digraph (PCCC-D) and Overlap Number of Balls (ONB) algorithms [22,27], so it requires access to the classification dataset, a distance function and the classifier's prediction function. For categorical features, the distance calculation involves using their one-hot encodings unless they are ordinal.

This strategy generates a ball coverage of the data where only elements from one class are present in each ball. ONB has proven useful when evaluating the complexity of datasets and their degrees of class overlap. This characterisation of the areas with class overlap (borderline areas) allows for the mapping of the approximate class boundaries, which might otherwise be very complex depending on the classification model.

Once a counterfactual is required for a given instance, the mapping helps to detect the directions in which better paths lie towards different classes. For that purpose, the ball boundaries and the centres of balls that are close to the instance but have the target classes are harnessed to generate close and feasible counterfactual candidates.

The ONB-MACF method can be structured in 3 stages:

1. dataset mapping via the ONB algorithm for the characterisation of the classification data space (Algorithm 1);

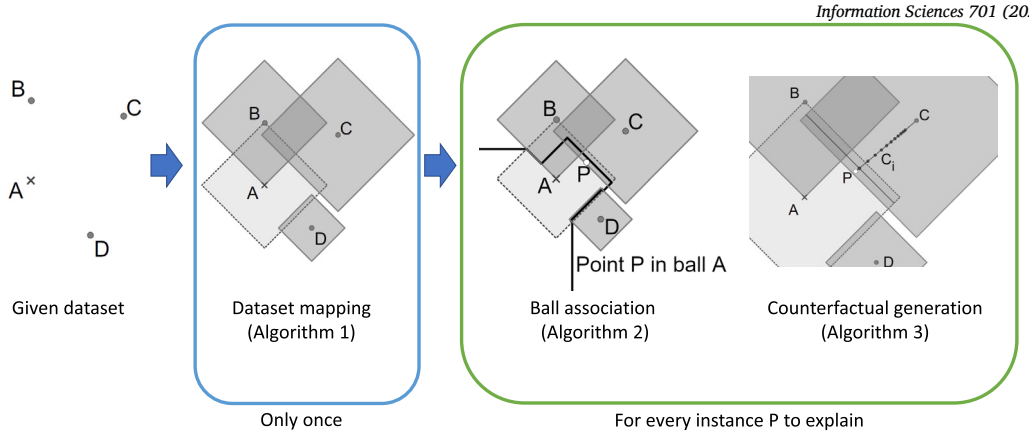

Fig. 1. Given a dataset (represented by points A to D), a class coverage mapping is obtained for a given distance metric (i.e. Manhattan); then, for each instance whose counterfactual is requested (P), said instance is assigned to an appropriate ball from the coverage (A), which helps select the closest class boundary ($A - C$), and counterfactual candidates (C_i) are generated between the boundary and the projected centre of the chosen ball of opposing class.

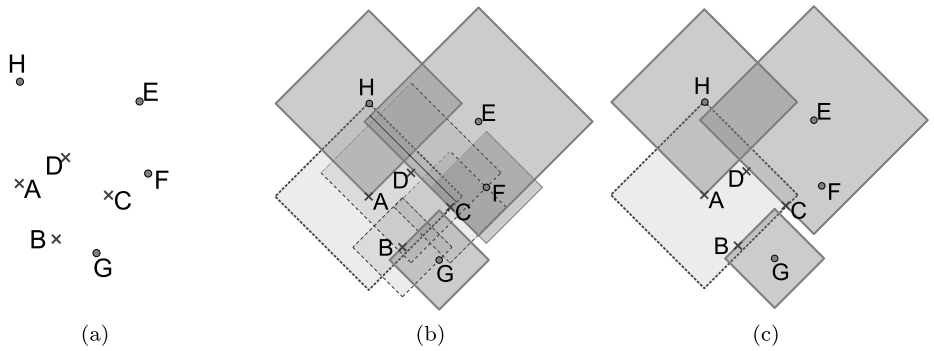

Fig. 2. Example of the class coverage using balls generated with the Manhattan distance, where the data for each class are symbolised by circles and crosses respectively. First, for all points (a), the balls are generated with their maximum radii that excludes instances of the opposing class (b); then, those including the most instances (those in thick lines) are iteratively selected until they are all covered (c).

2. instance association to the closest ball from the mapping for the selection of relevant boundaries for each instance (Algorithm 2) and
3. counterfactual candidate generation for the obtaining of the chosen counterfactuals (Algorithm 3).

To better understand this structure, a diagram is presented in Fig. 1.
In Sections 3.1.1 to 3.1.3, the three stages will be described, along with their associated algorithms.

3.1.1. Dataset mapping via the ONB algorithm (Algorithm 1)

In this stage, the ONB algorithm is used to cover all samples of the dataset using class-dependent open balls centred on instances from the dataset (see Algorithm 1).

The coverage strategy is geometrically intuitive. For each class, its instances are first identified (lines 4 to 5) and, then, a coverage set that includes them is generated. Said coverage set will be formed by balls (and their covered points), which are iteratively added until all instances for the current class are covered. In each iteration, the biggest ball centred on each non-covered instance that excludes instances of a different class is generated (line 9), and the still uncovered instances that would be covered by that ball are noted (line 10). After they have all been checked, the first ball that included the maximum number of instances (lines 11 to 14) is added to the coverage set (line 16), and those instances are marked as covered, updating the list of uncovered points (line 17). Since, as said before, this process happens for each class until all its instances are covered, a mapping for the whole dataset is obtained.

Relevant data from the chosen balls, such as the central instance or the radius, are saved in data structure D for further use. An example can be seen in Fig. 2. The structure of the algorithm can be seen in Algorithm 1, where $d_{i,j}$ is the distance between instances i and j, $c(i)$ is the class of element i (according to the classifier) and K is the set of classes from the given dataset.

3.1.2. Instance association to the closest ball from the mapping (Algorithm 2)

In this stage, the instance whose counterfactual is needed is associated to one of the balls from the mapping via Algorithm 2. This will help decide which part of the boundary must be used for the counterfactual generation.

Algorithm 1 Pseudocode outline of the adaptation of the P-CCCD and ONB algorithms with the maximum radius for each single-class open ball for the ONB-MACF dataset mapping.

1: **input** $d_{i,j} : i, j = 1...n$; K; c
2: $D = \{\}$
3: **for** $k \in K$
4: $U = \{i / c(i) = k\}$
5: $V = \{i / c(i) \neq k\}$
6: **while** $|U| > 0$
7: $G = \{\}$
8: **for** $i \in U$
9: $d = min_{j \in V}(d_{i,j})$
10: $P = \{u \in U / d_{i,u} < d\}$
11: **if** $|P| > |G|$
12: $G = P$
13: $l = i$
14: $r = d$
15: **end for**
16: $D = D \cup \{\{l, c(l), r, G\}\}$
17: $U = U \setminus G$
18: **end while**
19: **end for**
20: $L = \cup_{(l,c,r,G) \in D}\{l\}$
21: **return** D, L
22: **end**

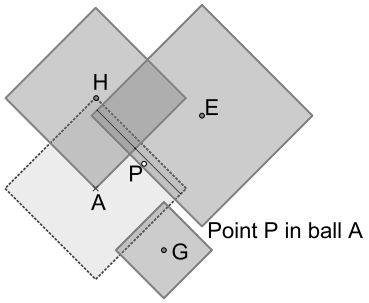

Fig. 3. For an instance's ball association using balls generated with the Manhattan distance, the boundaries amongst balls are studied to select the most appropriate one for the studied instance. In this example, point P is inside the balls centred on A and E, and is associated to A since it falls on its side of the boundary against E.

Given a sample whose counterfactual should be studied, the previously generated mapping is used to associate said sample to a ball, so that the appropriate boundaries can be checked. For this purpose, the set of balls that cover the sample is obtained (line 2). Depending on how many balls cover the instance, the association is performed as follows:

- if no ball covers the point, it is associated to the closest ball (lines 3 to 4);
- if only one ball covers the instance, it becomes associated to it (lines 5 to 6);
- if multiple balls cover the instance, it becomes associated to the ball that has the instance on its side of all boundaries with the other balls that include it (lines 7 to 15).

A simple example of this association to a ball from the coverage set for the latter case (which is the most complex of the three) is given in Fig. 3.

3.1.3. Counterfactual candidate generation (Algorithm 3)

This is the stage where counterfactuals are generated for the given instance. The process, whose pseudocode is shown in Algorithm 3, can be divided into 2 stages: firstly, the set of initial candidates in the boundaries of a previously decided number of balls m (which is the number of counterfactuals to be given for a single instance to explain) is chosen and saved (lines 2 to 18); then, assuming a convex domain (for the validity of points in a segment) and using both those initial counterfactual candidates and the projected centres used in their obtaining, an iterative process proposes further counterfactual candidates moving from the former towards the latter until the class changes, reaching the desired counterfactuals (lines 19 to 34). The process will now be detailed.

Obtaining the initial set of candidates To begin this stage, the 2 possible situations stemming from the results of Algorithm 2 need to be acknowledged: either the ball associated to the instance shares its class, or it does not.

In the latter case, the mapping in data structure D should be updated, as a mapping that better approximates the classification model can be obtained by including the explored instance. To do this, Algorithm 1 will be used on the both the instance whose counterfactual is required and all instances included in its assigned ball, obtaining a new coverage for that specific part of the data

Algorithm 2 Pseudocode outline of the ball association process for instance i, where L is the set of balls from Algorithm 1, with each ball centred in l, and where $r(l)$ is the radius of the ball centred in l.

```
1: input L; d_{j,k} : j,k ∈ L; i; d_{i,l} : l ∈ L
2: E = {l ∈ L : d_{i,l} < r(l)}
3: if |E| = 0
4:   a = argmin_{l∈L} d_{i,l}
5: elif |E| = 1
6:   a = l ∈ E
7: else
8:   for l ∈ E
9:     if a is undefined
10:      a = l
11:    else
12:      Make the boundary between balls centred in a and l
13:      if i falls on the side of the boundary closer to l
14:        a = l
15:  end for
16: return a
17: end
```

Algorithm 3 Pseudocode outline of the counterfactual generation process for instance i, where D and L are the data structure that includes the mapping information and the set of balls from Algorithm 1 respectively, with each ball centred in l, a is the associated ball for i from Algorithm 2, C_{target} is the set of target classes for instance i, m is the number of counterfactuals to be given, r is the ratio for the decreasing intervals and BCL is the best candidate list.

```
1: input D; L; d_{j,k} : j,k ∈ L; i; a; d_{i,l} : l ∈ L; C_target; m; r
2: BCL= { }                                   # Beginning of Stage 1: Obtaining the initial set of candidates
3: if c(a) ≠ c(i)
4:   Repeat Algorithm 1 on the distance matrix of G(a) ∪ {i}, obtaining D'
5:   D = D \ {a, c(a), r(a), G(a)}
6:   D = DU D'
7:   a = i
8: for l ∈ L_{|C_target}
9:   if the projection of l falls inside the ball, keeps the class and complies with immutability
10:    Reduce changes between said projection and i
11:    Take cf candidate on the boundary towards it
12:    if ∃j ∈ BCL : d_{i,cand} < d_{i,j}, or |BCL| < m
13:      update BCL with cand (keeping |BCL| ≤ m)
14: end for
15: if |BCL| = 0
16:   Same as 8-14, relaxing the projection validity restriction
17: if |BCL| = 0
18:   Same as 8-14 also withholding immutability restrictions
19: for cand in BCL                           # Beginning of Stage 2: Obtaining the counterfactual explanation set
20:   found = false
21:   t = 0
22:   V is the vector from cand to the associated projection
23:   while found = false and t < 10
24:     cand = cand + r^t V
25:     if c(cand)=c(proj)
26:       found = true
27:     t = t + 1
28:   end while
29:   if found = false
30:     cand = proj
31:   save cand as cf on list of counterfactuals for i
32: end for
33: return list of counterfactuals for i
34: end
```

space that will substitute the original ball (lines 3 to 6). Since all points of that coverage belong to a different class, the studied instance will be the centre of its own ball, which will become its assigned ball (line 7).

Now that the assigned ball shares the class with the studied instance, the closest balls from an opposing class are searched for using the mapping (lines 8 to 14), for which the boundaries between the assigned ball and the opposing ones are calculated.

So that the provided explanations are useful and fair, when the dataset includes immutable features, the projections of opposing balls and their centres on the subspace where said features are restricted to having the same values as the studied instance are used whenever possible. Furthermore, to adhere to the calculated mapping, and thus to add further interpretability to the process, these projections are considered viable when they fall inside their balls and maintain the ball's class (line 9). This way, balls that would not be able to provide useful and actionable counterfactuals for the particular instance can be discarded, and proposed candidates would share those immutable values.

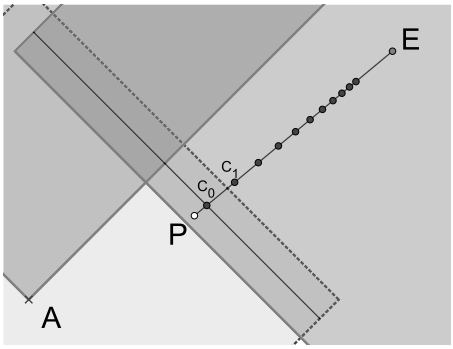

Fig. 4. For an instance's candidate counterfactual generation with a chosen opposing ball in a dataset with no immutable nor discrete features, a first candidate C_0 is given at the intersection of the boundary between the instance P's associated ball and the chosen opposing ball E and the segment that goes from the instance to the centre of the selected opposing ball. If that candidate's class is the given instance's, subsequent candidates C_i are generated in the said segment in intervals with decreasing length until the class changes or a check limit is reached (in which case the counterfactual would be E).

Once a valid projection is reached, the absolute differences in each feature between the sample and the projected centre are ordered and, starting with the feature with the lowest difference, some features of the latter are modified to take the instance's values, as long as the class is kept intact (line 10). Since counterfactual candidates will be generated in a straight line from the instance to the adapted and projected centre, no changes will be proposed features whose values coincide, which helps improve counterfactual sparsity.

After obtaining the valid adapted and projected centre of a ball, an initial counterfactual candidate is created at the intersection of the boundary and the straight line connecting the sample and the projected centre (line 11). The closest of these initial candidates, up to the desired m, as well as their adapted and projected centres, are saved (lines 12 to 13).

In very complex situations, or those where inherent biases make it so no viable projections are possible, no candidates might be found after checking all balls of the target opposing class. In these cases, the process is repeated after relaxing the condition for viable projections to just having the correct class (lines 15 to 16), which would solve situations where the mapping' fit is not exact. If the process still provides no candidates, the immutability restrictions are withheld (lines 17 to 18); this last relaxation is enough to ensure the existence of candidates (as, with no projection being necessary, the closest ball of the target class would be eligible for a potential initial candidate).

Obtaining the counterfactual explanation set Given the initial counterfactual candidate set, each of those candidates will provide one counterfactual explanation in this stage. If the class of one of these initial candidates is not the opposing ball's, further counterfactual candidates are sequentially created on that straight line in decreasing intervals (each r times the previous interval, $0 < r < 0.5$) towards the appropriate adapted and projected centre until a counterfactual with the target class is reached (lines 19 to 28, and an example of this is included in Fig. 4). If after generating 10 candidates the class has not changed to the centre's, the proposed counterfactual will be the adapted and projected centre, which was a valid counterfactual since the beginning (lines 29 to 30). These final counterfactuals are saved as explanations for the model's decisions on the studied instance (lines 31 to 33). When this process does not find a counterfactual immediately (that is, when the initial candidate does not have the opposing class), the last candidate with the same class as the original instance is a valid semifactual.

If the dataset includes discrete features (such as features taking integer or binary values, as nominal or categorical features are one-hot-encoded), their values on the candidates are always rounded to the nearest integer throughout the process to maintain such property.

3.2. Advantages of the ONB-MACF method

The design of the proposed ONB-MACF method poses multiple advantages. Due to being model-agnostic, it only requires access to the prediction function from the model, furthering its versatility. While it was designed for tabular data, it might be able to be used on text or image datasets after some data transformations [9]. Furthermore, unlike other methods whose strategies involve random modifications of the instance's features [20] or swapping feature values [6], ONB-MACF yields stable results amongst multiple executions with the same set-up. The identification and utilisation of class boundaries instead of simply other instances also allows for closer counterfactuals. Additionally, unlike most methods, ONB-MACF can propose semifactuals, further bolstering explainability.

The methodology followed for the generation of counterfactuals by the ONB-MACF method also has several advantages.

Firstly, it must be noted that there is a single computationally intensive step, namely the generation of the dataset mapping (Algorithm 1), which is $O(|K|n^3)$, where $|K|$ is the number of classes and n is the number of instances in the dataset. Considering the usefulness of this process, we may highlight two clear advantages associated with this computation.

- For repeated use of a dataset and its associated classification model, which is very common in real-life situations where the outcome of the model itself needs to be audited, this step only has to be performed once over the whole dataset, thus avoiding

unnecessary computations. While in some cases Algorithm 1 can be triggered from Algorithm 3 to update the data structure D, only a small subset of the dataset is employed, so its impact on the overall computation time should be small.
- The mapping reduces the search space size for other stages (from the number of dataset instances to the number of balls in the coverage). Thus, fewer comparisons are necessary for counterfactual generation than other instance-based counterfactual strategies [29].

Secondly, for the generation of counterfactual candidates, the procedure stated in Algorithm 3 includes several features that allow it to be customised to different cases:

- counterfactuals for multiple samples can be obtained in a single run;
- for each sample, a target number of counterfactuals can be generated, using different opposing balls for each case in order to foster better counterfactual diversity;
- in multiclass cases, target counterfactual classes can be specified;
- immutability of features is maintained; and
- discrete and categorical features maintain their properties.

4. Experimental framework

To analyse the sound capabilities of the proposed ONB-MACF method, in this section, we provide the required material for a fair comparison with respect to the state of the art. In order to ease the compilation of the necessary resources, the CARLA[5] benchmarking package [28] was used. It includes implementations of counterfactual explanation and recourse methods from literature, as well as datasets, pre-trained models and performance metrics that evaluate the methods from different perspectives. For further proof of the capabilities of the ONB-MACF method, a qualitative analysis of its counterfactuals compared to those of Growing Spheres and NICE was performed.

This experimental framework section comprises the following structure. Firstly, a short description of the datasets is included in Section 4.1. Secondly, the structure of the classifier for each dataset and the different methods against which our proposal will be tested are indicated in Section 4.2. Finally, the metrics indicating how their performance will be measured are detailed in Section 4.3.

4.1. Datasets

This subsection describes the characteristics of the datasets used in the experimental study.

The benchmarking study includes eight well-known binary datasets: “adult”, “COMPAS”, “Give Me Some Credit”, “HELOC”, “Irish”, “Saheart”, “Titanic” and “Wine”.

- The “adult” dataset evaluates whether a person will earn over 50,000 dollars a year according to census data. While this dataset initially included 48,832 instances, its size was halved to 24,416 samples by stratified undersampling to reduce execution times while maintaining the structure and representativity of the dataset. It contains 13 features and the binary class, of which 7 are categorical (binary) attributes and 2 are considered immutable (“age” and “sex”).
- The “COMPAS” dataset predicts whether convicts will re-offend in the following two years based on their personal data and criminal record. It includes 6,172 samples, with 7 features and the binary class, of which 3 are categorical (binary) and 3 are immutable (“age”, “race”, “sex”).
- The “Give Me Some Credit” dataset evaluates whether a person will have financial problems in the next two years, to give or deny them a loan. While originally including 115,527 instances, its size was reduced to a fifth by stratified undersampling (23,105) to reduce execution times while maintaining the structure and representativity of the dataset. It has 10 variables and the binary class, all of which take continuous values and where only 1 feature is immutable (“age”).
- The “HELOC” dataset is a credit dataset that predicts whether people will be able to pay their loans in two years. It includes 9871 samples, with 21 features (all of them numeric and none of them immutable) and the binary class.
- The “Irish” dataset evaluates whether Irish schoolchildren aged 11 in 1967 would take their Leaving Certificate of their studies (which implied passing their final examinations). This dataset includes 500 samples, with 5 features and the binary class, of which 3 are categorical (1 is binary, 2 are one-hot-encoded due to having multiple options) and 1 is immutable (“sex”).
- The “Saheart” dataset predicts whether males in a high-risk heart-disease region of the Western Cape in South Africa have a coronary heart disease. This dataset includes 462 samples, 9 features and the binary class, of which 1 is categorical (binary) and 1 is immutable (“age”).
- The “Titanic” dataset includes data from the passengers of the Titanic and whether they survived the accident. It includes 2099 instances, 8 features and the binary class, of which 4 are categorical (1 is binary, 3 are one-hot-encoded due to their multiple options) and 3 are immutable (“gender”, “age” and “country”).
- The “Wine” (“Wine quality white”) dataset evaluates the quality of Portuguese white wines according to their chemical composition and the grades given by experts. While this dataset is often used for regression, as the grades are given out of 10, the target

[5] CARLA Benchmark, https://github.com/carla-recourse/CARLA, last accessed 2 April 2024.

was transformed to binary (passing grade or not). It includes 4898 instances, 11 attributes, all of them numeric, and the binary class.

From each dataset, a sample of 200 instances (enough to provide some variety of inputs) will be separated. Their classes will be predicted using the benchmarking classifier and they will serve as factuals whose predictions want to be explained.

4.2. Classifier configuration and explanation methods

Counterfactual methods aim to explain and give further insight into the decisions of a classifier. The chosen model was a neural network, which is considered a black-box.

For each dataset, a pre-trained neural network with two hidden fully-connected layers with 32 and 16 neurons respectively, and a ReLU activation function was used, akin to CARLA's benchmarking. All counterfactual algorithms must work on the same model for fair comparison on each dataset.

Regarding the counterfactual explanation algorithms from the state of the art, CEM [11] (on its base and variational autoencoders versions), CLUE [2], CRUDS [13], DiCE [25], FACE [29] (on its knn and ϵ-graph versions), Growing Spheres [20], Wachter [35] and NICE [6], whose particularities were explained in Section 2.2, were included in the quantitative benchmarking.

4.3. Counterfactual performance metrics and evaluation approach

We evaluate the performance of the generated counterfactuals using several metrics, which are enumerated below. Each of these metrics is related to one or more properties of interest for counterfactuals, as described in Section 2.1.

- **L0 norm**: it measures the L0 norm between the studied sample and its counterfactual; that is, how many attributes change between them. It measures **sparsity**.
- **L1 norm**: it measures the Manhattan distance between the studied sample and its counterfactual, which evaluates **similarity/closeness**.
- **L2 norm**: it measures the squared Euclidean distance between the studied sample and its counterfactual, which also gauges **similarity/closeness**.
- **L∞ norm**: it measures the L∞-norm between the studied sample and its counterfactual, which evaluates **similarity/closeness**, too.
- **Constraint violation**: it measures how many immutability restrictions have failed in an explanation. It is a metric of **actionability**.
- **Redundancy**: it checks how many of the changed attributes between the sample and its counterfactual could be reverted without modifying the counterfactual's class (that is, how many changes were unnecessary). This also measures **sparsity**.
- **Y-NN**: it checks the ratio of same-class elements amongst the Y (5) nearest neighbours of each counterfactual, which measures **plausibility** but is related to **similarity**.
- **Success rate**: it indicates the ratio of samples whose counterfactuals were successfully found, which is a matter of **validity** of counterfactuals.

The presented metric for the evaluation of each method is the arithmetic mean of the results for all the studied samples of each dataset. By default, the y-NN and success rate metrics are better for values closer to 1, while the rest are better for values closer to 0. It must also be noted that the metrics from CARLA are unable to evaluate counterfactual diversity (possibly due to not many methods are explicitly designed to produce different counterfactuals over which diversity can be measured) and, even though the L1, L2 and L∞ norms measure the same properties, they do so from different points of view: the Manhattan distance can correlate to the Euclidean one (but not always, depending on the distribution of feature changes), and the L∞ norm gives information on the extremeness of changes in individual features.

5. Quantitative performance analysis of the ONB-MACF method

This section shows the results obtained with each dataset and counterfactual method, carrying out a thorough study of the good behaviour of the ONB-MACF method. The results are evaluated following the characteristics of their generated counterfactuals according to the metrics enumerated in Section 4.3.

For each dataset, the counterfactual explainers from Section 4.2 are compared to the proposed ONB-MACF method (where $r = 0.15$ is the ratio for the intervals in Algorithm 3). Given the differences in dataset characteristics, the metrics are obtained and transformed separately for each dataset (using the ratio between the metric's value for each method and the maximum for the metric in that dataset when higher numbers are preferable, or 1 minus that ratio otherwise), so that all metrics take values between 0 and 1, and higher values indicate better performance. The results are then aggregated. While the goodness of a method according to a particular metric can be easily seen, its general goodness is evidenced by the mean of the metrics. These means are presented in Table 1, along with the averages of the results (scaled per dataset) for each metric and method. The raw metric results per dataset are presented in Tables A.3 to A.10 of Appendix A.

Table 1
Average of the performance results, scaled per dataset, with an overall mean over the metrics.

Method	L0 norm	L1 norm	L2 norm	L-∞ norm	Const. Vio.	Redun.	YNN	Succ. Rate	Overall mean
CEM	0.287	0.799	0.815	0.474	0.628	0.428	0.603	**1.000**	0.629
CEM-VAE	0.286	0.798	0.814	0.474	0.623	0.430	0.606	**1.000**	0.629
CLUE	0.019	0.138	0.252	0.178	0.353	0.032	0.713	0.663	0.293
CRUDS	0.015	0.259	0.354	0.229	0.395	0.045	0.886	0.641	0.353
DICE	0.765	0.703	0.702	0.357	0.804	0.928	0.759	**1.000**	0.752
FACE-EPSILON	0.131	0.445	0.553	0.375	0.324	0.315	0.849	0.946	0.492
FACE-KNN	0.133	0.448	0.556	0.374	0.336	0.319	0.838	0.999	0.501
GS	0.205	0.777	0.799	0.471	0.970	0.520	0.488	**1.000**	0.654
WACHTER	0.082	0.474	0.512	0.305	0.424	0.480	0.450	0.860	0.448
NICE	0.729	**0.873**	**0.917**	**0.664**	0.776	**0.939**	0.561	**1.000**	0.808
ONB-MACF	**0.756**	0.861	0.906	0.635	**0.989**	0.915	**0.879**	**1.000**	**0.868**

For the sake of a better understanding, results are graphically summarised in the form of radial plots. In these radial plots, all metrics from Section 4.3 are associated with the vertices of a regular polygon, following the same order from that list in a clockwise fashion. For radial plot generation, and similarly to Table 1, values in all metrics were transformed so that the scale went from 0 to 1 and so higher values indicated better performance. Each method's resulting scaled metric values are located in the segment between the centre and that metric's vertex and joined into another polygon. Results for every individual dataset are shown in Fig. 5, while the averaged comparison of the best-performing methods can be found in Fig. 6.

To properly evaluate the general quality of the counterfactuals obtained using the ONB-MACF method, the results from the 8 cases of study will be summarised from the multiple fronts recommended in Section 2.1 and reflected by the metrics.

- **Sparsity**. The ONB-MACF method consistently achieves good results, as measured by the L0 norm, with changes in generally 1-3 features as recommended, so the counterfactuals should be easy for humans to grasp. This point is further proven by the low (sparsity-related) redundancy values, and happens due to the reduction of attribute changes in stage 1 of Algorithm 3.
- **Similarity**. Evaluated using the L1, L2 and L-∞ norms, the ONB-MACF method presents excellent results in this regard, with close counterfactuals to the given instances and without lopsided feature changes (evidenced by low L-∞ norm values). This stems from the good estimation of class boundaries and their close crossing.
- **Actionability**. The method obtains very few constraint violations, which is highly positive. Due to ONB-MACF's design, which employs projections, the immutability restrictions are always followed for an instance unless there are no suitable coverage balls (that is, there are no balls whose centre's projection onto the values of the immutable features of the instance maintains that ball's class).
- **Plausibility**. The YNN values show that the ONB-MACF method can achieve some of the best results on datasets without a substantial percentage of immutable and discrete features. This makes sense, as the counterfactual is obtained inside a ball opposite class points.
- **Validity**. As measured by the Success Rate, the method found counterfactuals for all samples, just like most other studied counterfactual explainers.
- **Diversity**. While the metrics from CARLA cannot evaluate this property, the ONB-MACF method promotes it. When multiple counterfactuals are required for a sample, the counterfactual candidates are generated using different balls. Thus, counterfactuals are created in different directions and using different combinations of features.

On another note, regarding computational speed, the ONB-MACF method is generally comparable to the other slow counterfactual methods when the distance matrix and the mapping have been obtained beforehand (as they only need to be obtained the first time a dataset is evaluated and can be saved). The speed is adequate on curated datasets with limited data to be used, and using it on big datasets can be achieved after an instance reduction step (which, if carefully done, can preserve most of the interesting characteristics of the base dataset [5]). A similarly scaled ranking of the time results can be seen in Table A.11 of Appendix A, both excluding and including the time to generate the mapping.

6. Qualitative analysis of the ONB-MACF method

This section qualitatively assesses the capabilities of the ONB-MACF method. For this qualitative analysis, 200 counterfactuals were obtained on the neural network classifiers for each of the four chosen datasets ("Irish", "Saheart", "Titanic" and "Wine quality white"). The counterfactuals given by the ONB-MACF method are scrutinised and compared to those obtained with two methods that share similarities with it:

- Growing Spheres [20] uses hyperspheres centred on the instances, which may seem similar to the ONB-MACF method. However, the use of multiple balls simultaneously in our proposal and the better exploitation of their morphology, compared to the random instance generation in growing spheres, sets both methods apart.

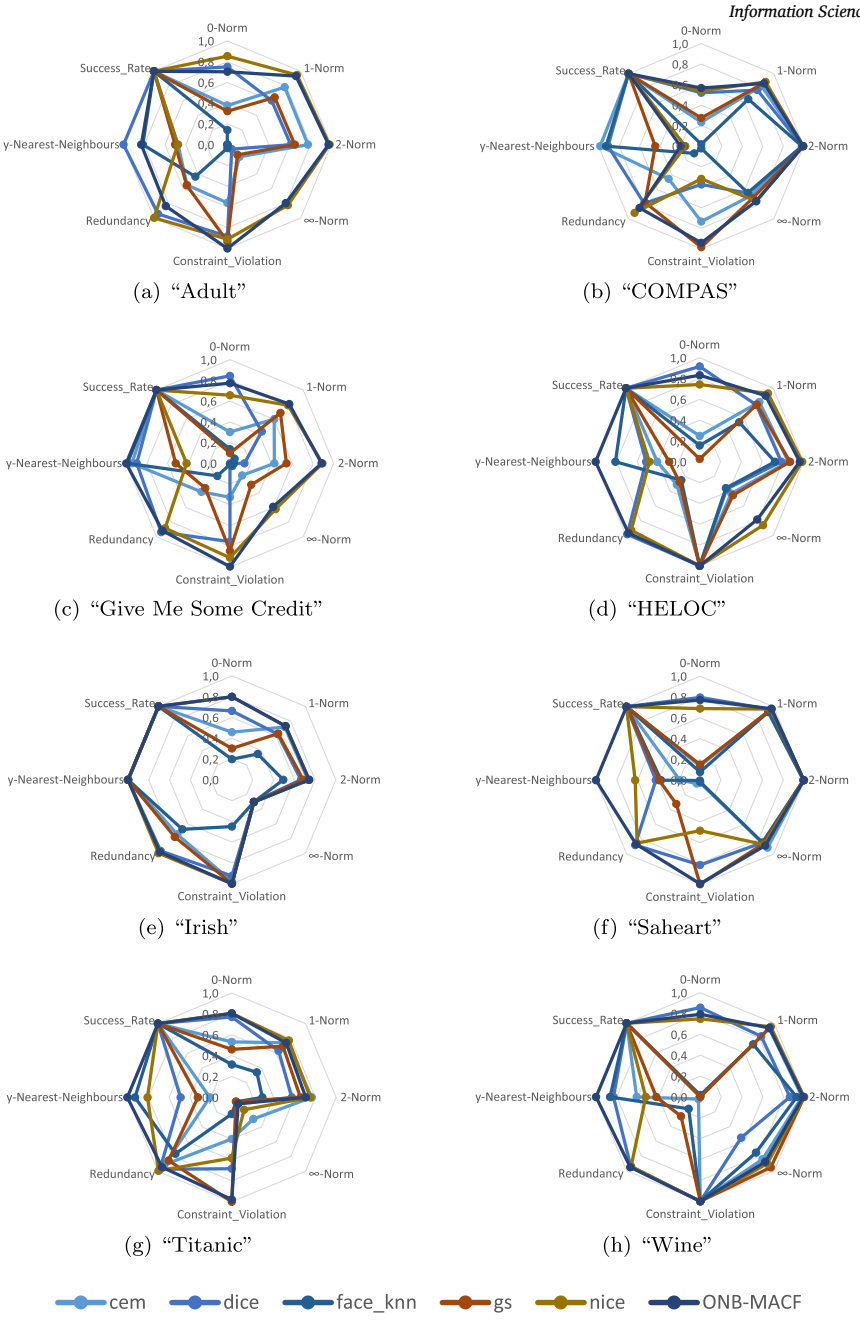

Fig. 5. Comparison of the best-performing methods for the "Adult" (a), "COMPAS" (b), "Give Me Some Credit" (c), "HELOC" (d), "Irish" (e), "Saheart" (f), "Titanic" (g) and "Wine" (h) datasets.

- NICE [6] is an instance-based counterfactual method that uses opposite-class members as prototypes, like the ONB-MACF method. NICE's strategy is to iteratively swap feature values of the instance to explain and those of the dataset's prototype, while the ONB-MACF method goes a step beyond by generating its own candidates.

The differences amongst counterfactual strategies can lead to very different types of counterfactuals, which can affect the auditability of the classification models. In broad terms, counterfactuals can be used to check whether the recommended changes make sense in particular cases, but when you generate enough of them, you can also aggregate the results and see which variables are generally causing the class change.

In this qualitative study, the focus will be on four questions:

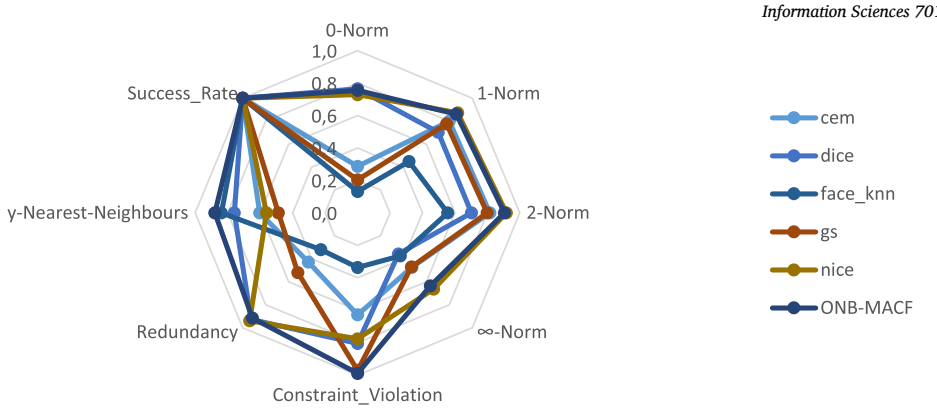

Fig. 6. Comparison of the best performing methods, averaged over the datasets.

- which features are most often used for class change,
- which features are often used for class change together,
- whether the change in class according to those features would make sense in the eyes of an expert, and
- whether the features most used for class change differ amongst counterfactual methods.

The number of feature changes per method and dataset are compiled into Table 2.

The first part of Table 2 shows the feature changes on the counterfactuals for the Irish dataset, where the most important feature is Education level, as all methods suggest. In this case, both ONB-MACF and NICE properly point to changes in that feature, while Growing Spheres also includes changes in most other non-immutable features. Thus, the auditability according to ONB-MACF and NICE is very good (the proper feature is the one used), and the quality is diluted while using GS.

Regarding Saheart, in the second part of Table 2, Growing Spheres proposes many changes in all non-immutable features, which does not allow for any auditability. As for ONB-MACF and NICE, they give a reasonable amount of changes per instance, and both methods agree on the importance of the cumulative tobacco consumption and the amounts of low-density lipoprotein cholesterol, and to some extent the family history of such heart diseases. These changes make sense. NICE also gives much importance to Age (which is one of the immutable attributes) and Obesity. The model seems to pass the auditability test here too. No pairs of features are too common (ldl and tobacco being the most common with 13/200), and single changes in ldl and tobacco are the most common in both NICE and ONB-MACF (up to 31/200) but are still unusual.

As for Titanic, in the third part of Table 2, we again see that Growing Spheres proposes changes in most non-immutable attributes most of the time. This time, ONB-MACF and NICE differ in the most relevant features, since ONB-MACF selects Class, while NICE points to Gender and Age. This is a relevant situation, since both Gender and Age are immutable features (and, in this case, bias-inducing) that the model must not use; thus, this model cannot pass the auditability test. This is so important that even the ONB-MACF method, which includes immutability clauses, could not find any prototypes with feasible changes that did not include changes in those features for some samples. In the case of Class, as selected by ONB, it makes sense that it would be an important feature while maintaining Gender and Age (as bribing or preference might have saved wealthy people). The most common pairs were age and gender on NICE (23/200) and class and embarked (13/200) and class and sibsp (12/200) on ONB-MACF, which were not too common. Single changes in gender on NICE and class on ONB-MACF were very common (up to 105/200).

Finally, the last part of Table 2 shows the proposed changes on the Wine dataset. While Growing Spheres shows its inability to help regarding auditability, the feature changes are mainly mixed for ONB-MACF and NICE. Both prefer Volatile acidity, Free Sulphur Dioxide and Alcohol contents, but it is not as clear as in the other datasets. In this situation, checking the plausibility of the obtained counterfactuals would be useful. Counterfactuals obtained using the ONB-MACF method are plausible (as evidenced by the YNN metric). However, counterfactuals obtained using NICE are often too close to the boundary. 83/200 of the counterfactuals were borderline (2 or 3 of their neighbours were from the original instance's class), and 49/200 had 4 or 5 of their nearest neighbours with the original instance's class (which could be more comparable to adversarial examples). On the remaining 68 counterfactuals, the same three features remain the most important and in similar amounts, with a slight increase in the importance of Volatile acidity, so it is mostly a matter of distance rather than of overall counterfactual direction. This means that, in this dataset, counterfactuals follow a more case-by-case situation, without general rules for feature changes. There are no common pairs of features (the most common was alcohol + volatile acidity on NICE, which happened 8/200 times). Single changes in alcohol, volatile acidity or free sulfur dioxide were slightly common (up to 43/200) depending on the method.

Overall, the results above show the good qualities of the ONB-MACF method from a user's standpoint. The proposed changes avoid immutable features, fostering the actionability and, thus, the feasibility of counterfactuals. Moreover, the ONB-MACF method proposes modifications in features that would likely imply a class change in the respective ambits, so they make sense for the user and this, together with the lower average number of proposed feature alterations, improves the understandability of its counterfactual explanations.

Table 2
Number of changes per feature on each dataset for each counterfactual method.

Dataset	Feature	Feature changes per method		
		GS	NICE	ONB-MACF
Irish	Sex	0	0	0
	DVRT	200	0	0
	Education level	200	200	200
	Prestige score	200	1	0
	Type school	98	0	0
	Mean changes	3.49	1.005	**1**
Saheart	Sbp	200	8	37
	Tobacco	200	132	94
	Ldl	200	138	103
	Adiposity	200	1	34
	Famhist	129	83	59
	Typea	200	12	33
	Obesity	200	80	24
	Alcohol	200	2	29
	Age	0	103	0
	Mean changes	7.645	2.795	**2.065**
Titanic	Gender	0	135	8
	Age	0	82	5
	Class	186	12	165
	Embarked	78	2	44
	Country	0	34	0
	Fare	200	19	25
	Sibsp	199	19	32
	Parch	200	8	32
	Mean changes	4.315	**1.555**	**1.555**
Wine	Free acidity	200	34	47
	Volatile acidity	200	108	64
	Citric acid	200	38	26
	Residual sugar	200	64	30
	Chlorides	200	43	16
	Free sulphur dioxide	200	93	59
	Total sulphur dioxide	200	24	40
	Density	200	52	15
	PH	200	10	34
	Sulphates	200	8	35
	Alcohol	200	77	89
	Mean changes	11	2.755	**2.275**

7. Concluding remarks

This paper has presented ONB-MACF, a novel model-agnostic method to provide counterfactual explanations that simultaneously grasps data space coverage and class boundaries using data morphology. This method is designed to obtain close counterfactuals with few feature changes in a way that is adaptable to the characteristics of the dataset (feature restrictions, number of classes, etc.).

As exposed by the qualitative and quantitative analyses of the obtained benchmarking results against multiple counterfactual methods, we have verified the usefulness of data-morphology-based strategies to provide counterfactual explanations. The ONB-MACF method has shown excellent results on multiple fronts simultaneously, such as closeness, sparsity, redundancy, actionability and validity, while also providing sensible counterfactuals that the audience of the model can understand and reason about. This was possible due to a solid model class boundaries estimation based on data morphology properties and the careful design of the ONB-MACF method's three algorithms. Particularly:

- the class mapping step using Algorithm 1 obtains the morphology of boundaries amongst classes;
- the ball assignment step using Algorithm 2 helps locate good counterfactual directions and improve class boundary estimation;
- and the counterfactual generation step in Algorithm 3 fosters compliance with immutability (via projections) and data type restrictions, including feature change minimisation and the possibility of generating multiple explanations for a single instance. Furthermore, due to the step-by-step counterfactual candidate generation from the well-estimated boundary and towards a group of data of opposing class, it favours both closeness to the initial instance and belonging to the opposing class.

This allows the ONB-MACF method to foster explainability while complying with feature immutability and existing data distributions.

Moreover, the general strategy the ONB-MACF method follows for counterfactual generation is geometrically intuitive: for a given instance, it takes candidates towards the closest suitable group of points of the target class (covered by a ball), until the class changes.

Additionally, the use of projections, which the user can also find intuitive as it involves leaving some features constant, lets them see that the method takes special care in situations prone to discrimination. This way, it provides insightful and useful explanations that people can use to comprehend and challenge a classifier's decisions. It lets users understand the mechanism that led to the explanation in broad terms. This is a relevant advantage since a sensible criticism regarding model explanations is that they may not be explainable themselves.

The ONB-MACF method works better on datasets with few categorical variables, as observed on the "Give Me Some Credit" dataset, since this makes it easier for the algorithm to find candidates using straight lines to the projected centres of opposing balls and to optimise the sparsity of the final counterfactuals. Datasets that simultaneously have a high percentage of categorical and immutable features can make it challenging to find feasible candidates that do not violate any constraints. Similarly, it is better used on curated datasets to reduce computational time.

Finally, regarding future work lines, the method could be applied to interpretable models and, in particular, to rule-based systems, to improve the configuration and granularity of critical features and, with that, the efficacy of the global model. Furthermore, the understandability of the ONB-MACF method as an explainer can also be connected with the recent studies on the use of explanations for Human-AI decision making, involving the concept of human intuition and its role [7]. Similarly, the usefulness of counterfactuals in the analysis of AI-assisted decision-making processes and how they can be combined with human intuition to avoid over-reliance in IA [32] are still open research fields.

CRediT authorship contribution statement

José Daniel Pascual-Triana: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Alberto Fernández:** Writing – review & editing, Writing – original draft, Supervision, Project administration. **Javier Del Ser:** Writing – review & editing, Supervision. **Francisco Herrera:** Writing – review & editing, Validation, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has received funding from the Spanish Ministry of Science and Technology under project PID2020-119478GB-I00, including European Regional Development Funds, and the "Cátedra Tecnalia en Inteligencia Artificial" programme. It was also partially supported by Knowledge Generation Projects, funded by the Spanish Ministry of Science, Innovation, and Universities of Spain under the project PID2023-150070NB-I00. It is also partially supported by the I+D+i project granted by C-ING-250-UGR23 co-funded by "Consejería de Universidad, Investigación e Innovación" and the European Union related to FEDER Andalucía Program 2021-27. J. Del Ser would like to thank the Basque Government for the funding support received through the EMAITEK and ELKARTEK programs (ref. KK-2023/00012), as well as the Consolidated Research Group MATHMODE (IT1456-22) granted by the Department of Education of this institution.

Appendix A. Experimental result tables for each dataset

The benchmarking results (per dataset) of the different counterfactual methods in CARLA and ONB-MACF are presented in Tables A.3 to A.10. In a few cases, no counterfactual was able to be generated for any of the 200 instances with a particular method, so the metrics could not be calculated - in these cases, their values are "Na". The mean of the scaled time results is presented in Table A.11.

Table A.3
Performance results of the studied counterfactual methods on the "adult" dataset, according to the benchmarking metrics.

Method	L0 norm	L1 norm	L2 norm	L-∞ norm	Const. Vio.	Redun.	YNN	Succ. Rate
CEM	5.785	0.925	0.840	0.831	0.645	3.960	0.311	**1.000**
CEM-VAE	5.785	0.925	0.840	0.831	0.645	3.960	0.312	**1.000**
CLUE	8.710	4.164	3.321	0.985	1.320	8.785	0.503	**1.000**
CRUDS	9.303	3.795	3.395	0.974	1.333	8.091	0.206	0.165
DICE	2.325	1.694	1.461	0.937	0.170	**0.490**	**0.656**	**1.000**
FACE-EPSILON	7.870	4.135	3.623	0.992	1.469	4.792	0.563	0.960
FACE-KNN	7.995	4.265	3.762	0.997	1.420	4.955	0.534	**1.000**
GS	6.300	1.513	1.315	0.859	0.100	3.920	0.330	**1.000**
WACHTER	6.660	0.864	0.669	0.668	0.840	4.415	0.211	0.530
NICE	**1.365**	**0.217**	**0.058**	**0.175**	0.130	**0.025**	0.314	**1.000**
ONB-MACF	2.765	0.274	0.091	0.203	**0.000**	1.425	0.544	**1.000**

Table A.4
Performance results of the studied counterfactual methods on the "COMPAS" dataset, according to the benchmarking metrics.

Method	L0 norm	L1 norm	L2 norm	L-∞ norm	Const. Vio.	Redun.	YNN	Succ. Rate
CEM	3.975	1.355	1.068	0.843	0.430	2.285	0.984	**1.000**
CEM-VAE	3.970	1.336	1.040	0.843	0.475	2.275	0.984	**1.000**
CLUE	4.820	1.984	1.453	0.891	1.050	3.740	0.979	**1.000**
CRUDS	5.045	2.057	1.512	0.903	1.175	4.155	**1.000**	**1.000**
DICE	2.490	1.805	1.590	0.937	1.015	0.885	0.908	**1.000**
FACE-EPSILON	5.185	2.860	2.528	0.940	1.620	3.895	0.932	**1.000**
FACE-KNN	5.090	2.787	2.451	0.950	1.585	3.745	0.928	**1.000**
GS	3.760	0.961	0.794	0.763	**0.025**	0.935	0.448	**1.000**
WACHTER	4.669	7.957	130.542	2.653	0.790	1.828	0.575	0.785
NICE	2.395	**0.896**	**0.681**	0.648	1.105	**0.340**	0.155	**1.000**
ONB-MACF	**2.245**	1.060	0.853	**0.642**	0.095	0.620	0.201	**1.000**

Table A.5
Performance results of the studied counterfactual methods on the "Give Me Some Credit" dataset, according to the benchmarking metrics.

Method	L0 norm	L1 norm	L2 norm	L-∞ norm	Const. Vio.	Redun.	YNN	Succ. Rate
CEM	6.950	0.956	0.724	0.688	0.670	5.990	0.945	**1.000**
CEM-VAE	6.950	0.958	0.725	0.689	0.670	5.990	0.946	**1.000**
CLUE	9.952	2.411	1.204	0.787	0.977	9.047	0.849	0.860
CRUDS	9.940	2.156	1.006	0.729	0.995	9.780	**1.000**	**1.000**
DICE	**1.550**	1.364	1.088	0.824	0.245	**0.610**	0.898	**1.000**
FACE-EPSILON	8.545	2.284	1.267	0.804	0.990	8.045	0.985	**1.000**
FACE-KNN	8.595	2.253	1.218	0.796	0.985	8.115	0.988	**1.000**
GS	8.965	0.748	0.577	0.584	0.155	6.490	0.522	**1.000**
WACHTER	9.735	1.032	0.594	0.563	0.750	6.785	0.691	**1.000**
NICE	3.410	0.497	**0.137**	**0.311**	0.090	1.150	0.417	**1.000**
ONB-MACF	2.255	**0.459**	0.147	0.337	**0.005**	0.795	**1.000**	**1.000**

Table A.6
Performance results of the studied counterfactual methods on the "HELOC" dataset, according to the benchmarking metrics.

Method	L0 norm	L1 norm	L2 norm	L-∞ norm	Const. Vio.	Redun.	YNN	Succ. Rate
CEM	16.235	1.414	0.959	0.832	**0.000**	14.300	0.408	**1.000**
CEM-VAE	16.195	1.410	0.956	0.833	**0.000**	14.250	0.423	**1.000**
CLUE	21.621	7.563	4.577	1.058	**0.000**	20.514	0.678	0.180
CRUDS	21.025	7.503	6.516	1.459	**0.000**	20.740	**1.000**	**1.000**
DICE	**1.805**	1.731	1.406	0.921	**0.000**	**0.335**	0.515	**1.000**
FACE-EPSILON	18.075	3.606	1.865	0.934	**0.000**	15.400	0.825	**1.000**
FACE-KNN	18.230	3.485	1.800	0.939	**0.000**	15.580	0.809	**1.000**
GS	21.025	1.684	0.858	0.797	**0.000**	15.080	0.292	**1.000**
WACHTER	21.025	1.752	0.835	0.784	**0.000**	15.980	0.252	**1.000**
NICE	5.555	**0.537**	**0.095**	**0.201**	**0.000**	1.435	0.483	**1.000**
ONB-MACF	3.62	0.773	0.248	0.314	**0.000**	0.655	**1.000**	**1.000**

Table A.7
Performance results of the studied counterfactual methods on the "Irish" dataset, according to the benchmarking metrics.

Method	L0 norm	L1 norm	L2 norm	L-∞ norm	Const. Vio.	Redun.	YNN	Succ. Rate
CEM	2.705	1.045	1.045	**1.000**	0.020	3.065	0.996	**1.000**
CEM-VAE	2.740	1.065	1.065	**1.000**	0.030	3.045	0.994	**1.000**
CLUE	5.000	3.001	2.679	1.069	1.000	11.905	**1.000**	**1.000**
CRUDS	4.828	3.729	3.895	1.428	1.000	9.630	**1.000**	0.960
DICE	1.680	1.410	1.301	1.005	0.075	0.370	**1.000**	**1.000**
FACE-EPSILON	4.092	2.509	2.062	**1.000**	0.563	4.134	**1.000**	0.710
FACE-KNN	4.010	2.418	1.969	**1.000**	0.550	3.885	**1.000**	**1.000**
GS	3.490	1.383	1.219	**1.000**	**0.000**	2.640	**1.000**	**1.000**
WACHTER	4.225	1.927	1.431	1.000	0.900	2.805	**1.000**	**1.000**
NICE	1.005	1.000	1.000	**1.000**	**0.000**	**0.005**	**1.000**	**1.000**
ONB-MACF	**1.000**	**1.000**	**1.000**	**1.000**	**0.000**	0.225	**1.000**	**1.000**

Table A.8
Performance results of the studied counterfactual methods on the "Saheart" dataset, according to the benchmarking metrics.

Method	L0 norm	L1 norm	L2 norm	L-∞ norm	Const. Vio.	Redun.	YNN	Succ. Rate
CEM	7.770	0.859	0.329	**0.414**	0.975	5.965	0.195	**1.000**
CEM-VAE	7.770	0.855	**0.328**	0.417	0.970	5.895	0.201	**1.000**
CLUE	Na	Na	Na	Na	Na	Na	Na	0.000
CRUDS	Na	Na	Na	Na	Na	Na	Na	0.000
DICE	**1.835**	1.073	0.791	0.762	0.185	**0.745**	0.425	**1.000**
FACE-EPSILON	8.29	2.147	1.099	0.767	0.995	6.245	0.394	**1.000**
FACE-KNN	8.288	2.143	1.096	0.755	0.985	6.226	0.375	0.990
GS	7.645	1.053	0.704	0.687	**0.000**	4.235	0.394	**1.000**
WACHTER	9.000	29.22	158.312	4.994	1.000	1.743	0.000	0.565
NICE	2.795	0.943	0.652	0.624	0.515	0.905	0.623	**1.000**
ONB-MACF	2.065	**0.838**	0.520	0.565	0.005	0.815	**1.000**	**1.000**

Table A.9
Performance results of the studied counterfactual methods on the "Titanic" dataset, according to the benchmarking metrics.

Method	L0 norm	L1 norm	L2 norm	L-∞ norm	Const. Vio.	Redun.	YNN	Succ. Rate
CEM	3.745	1.190	**0.902**	**0.747**	1.805	1.955	0.202	**1.000**
CEM-VAE	3.760	1.210	0.921	0.750	1.825	1.965	0.199	**1.000**
CLUE	8.000	4.602	3.877	1.054	3.000	22.267	0.124	**0.260**
CRUDS	Na	Na	Na	Na	Na	Na	Na	0.000
DICE	1.830	1.688	1.648	0.966	0.950	0.580	0.471	**1.000**
FACE-EPSILON	5.556	3.040	2.729	0.995	2.556	5.297	0.874	0.900
FACE-KNN	5.470	3.039	2.738	0.993	2.525	5.255	0.890	**1.000**
GS	4.320	1.410	1.326	0.996	**0.000**	3.205	0.312	**1.000**
WACHTER	7.430	3.451	2.948	1.000	2.690	8.595	0.194	**1.000**
NICE	**1.555**	**1.048**	0.983	0.879	1.255	**0.160**	0.775	**1.000**
ONB-MACF	**1.555**	1.205	1.128	0.962	0.065	1.225	**0.960**	**1.000**

Table A.10
Performance results of the studied counterfactual methods on the "Wine" dataset, according to the benchmarking metrics.

Method	L0 norm	L1 norm	L2 norm	L-∞ norm	Const. Vio.	Redun.	YNN	Succ. Rate
CEM	10.955	0.259	0.067	0.185	**0.000**	9.875	0.610	**1.000**
CEM-VAE	10.955	0.253	0.064	0.184	**0.000**	9.900	0.613	**1.000**
CLUE	11.000	3.828	2.726	1.172	**0.000**	9.745	0.588	**1.000**
CRUDS	11.000	1.574	0.467	0.499	**0.000**	10.155	**1.000**	**1.000**
DICE	**1.565**	0.676	0.386	0.525	**0.000**	0.575	0.833	**1.000**
FACE-EPSILON	10.785	1.065	0.193	0.277	**0.000**	8.600	0.891	**1.000**
FACE-KNN	10.800	1.093	0.211	0.288	**0.000**	8.530	0.866	**1.000**
GS	11.000	0.242	**0.014**	0.055	**0.000**	7.500	0.422	**1.000**
WACHTER	11.000	0.313	0.014	**0.029**	**0.000**	8.645	0.560	**1.000**
NICE	2.755	**0.177**	0.023	0.111	**0.000**	0.620	0.524	**1.000**
ONB-MACF	2.275	0.242	0.039	0.142	**0.000**	**0.490**	**1.000**	**1.000**

Table A.11
Average of the time results over the 8 datasets, where times for each dataset were transformed so that the scale went from 0 to 1 and so higher values indicated better performance.

Method	Average Time	Average Time (with Algorithm 1)
CEM	0.880	0.913
CEM-VAE	0.876	0.911
CLUE	0.696	0.777
CRUDS	0.328	0.507
DICE	0.982	0.987
FACE-EPSILON	0.894	0.944
FACE-KNN	0.899	0.945
GS	0.998	0.999
WACHTER	0.645	0.724
NICE	0.999	0.999
ONB-MACF	0.504	0.415

Data availability

The datasets are publicly available (in the CARLA and UCI repositories); the code can be shared if requested.

References

[1] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J.M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, F. Herrera, Explainable artificial intelligence (XAI): what we know and what is left to attain trustworthy artificial intelligence, Inf. Fusion 99 (2023) 101805.

[2] J. Antorán, U. Bhatt, T. Adel, A. Weller, J.M. Hernández-Lobato, Getting a CLUE: a method for explaining uncertainty estimates, in: International Conference on Learning Representations, 2021, pp. 1–34.

[3] S. Aryal, M. Keane, Even if explanations: prior work, desiderata & benchmarks for semi-factual XAI, in: Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI-23), ISBN 978-1-956792-03-4, 2023, pp. 6526–6535.

[4] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion (ISSN 1566-2535) 58 (2020) 82–115.

[5] M.J. Basgall, M. Naiouf, A. Fernández, FDR2-BD: a fast data reduction recommendation tool for tabular big data classification problems, Electronics (ISSN 2079-9292) 10 (15) (2021) 1757:1–1757:12, https://doi.org/10.3390/electronics10151757.

[6] D. Brughmans, P. Leyman, D. Martens, NICE: an algorithm for nearest instance counterfactual explanations, Data Min. Knowl. Discov. (ISSN 1573-756X) 38 (2024) 2665–2703, https://doi.org/10.1007/s10618-023-00930-y.

[7] V. Chen, Q.V. Liao, J. Wortman Vaughan, G. Bansal, Understanding the role of human intuition on reliance in human-AI decision-making with explanations, Proc. ACM Hum.-Comput. Interact. 7 (CSCW2) (2023) 370:1–370:32.

[8] Y.-L. Chou, C. Moreira, P. Bruza, C. Ouyang, J. Jorge, Counterfactuals and causability in explainable artificial intelligence: theory, algorithms, and applications, Inf. Fusion (ISSN 1566-2535) 81 (2022) 59–83.

[9] R. de Oliveira, K. Sörensen, D. Martens, A model-agnostic and data-independent tabu search algorithm to generate counterfactuals for tabular, image, and text data, Eur. J. Oper. Res. (ISSN 0377-2217) 317 (2) (2024) 286–302.

[10] J. Del Ser, A. Barredo-Arrieta, N. Díaz-Rodríguez, F. Herrera, A. Saranti, A. Holzinger, On generating trustworthy counterfactual explanations, Inf. Sci. (ISSN 0377-2217) 655 (2024) 119898:1–119898:16, https://doi.org/10.1016/j.ins.2023.119898.

[11] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, P. Das, Explanations based on the missing: towards contrastive explanations with pertinent negatives, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, Curran Associates Inc., 2018, pp. 590–601.

[12] W. Ding, M. Abdel-Basset, H. Hawash, A.M. Ali, Explainability of artificial intelligence methods, applications and challenges: a comprehensive survey, Inf. Sci. 615 (2022) 238–292.

[13] M. Downs, J.L. Chu, Y. Yacoby, F. Doshi-Velez, W. Pan, CRUDS: counterfactual recourse using disentangled subspaces, in: ICML Workshop on Human Interpretability in Machine Learning, 2020, p. 23.

[14] N. Díaz-Rodríguez, J. Del Ser, M. Coeckelbergh, M. López de Prado, E. Herrera-Viedma, F. Herrera, Connecting the dots in trustworthy artificial intelligence: from AI principles, ethics, and key requirements to responsible AI systems and regulation, Inf. Fusion 99 (2023) 101896.

[15] G. Fernández, J.A. Aledo, J.A. Gamez, J.M. Puerta, Factual and counterfactual explanations in fuzzy classification trees, IEEE Trans. Fuzzy Syst. (ISSN 1941-0034) 30 (12) (2022) 5484–5495.

[16] R. Guidotti, Counterfactual explanations and how to find them: literature review and benchmarking, Data Min. Knowl. Discov. (ISSN 1573-756X) (2022) 1–55.

[17] A.-H. Karimi, G. Barthe, B. Balle, I. Valera, Model-agnostic counterfactual explanations for consequential decisions, in: Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 895–905.

[18] A. Kasirzadeh, A. Smart, The use and misuse of counterfactuals in ethical machine learning, in: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021, pp. 228–236.

[19] M.T. Keane, E.M. Kenny, E. Delaney, B. Smyth, If only we had better counterfactual explanations: five key deficits to rectify in the evaluation of counterfactual XAI techniques, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, ISBN 978-0-9992411-9-6, 2021, pp. 4466–4474.

[20] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, M. Detyniecki, Comparison-based inverse classification for interpretability in machine learning, in: Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations, in: Communications in Computer and Information Science, Springer International Publishing, Cham, ISBN 978-3-319-91473-2, 2018, pp. 100–111.

[21] L. Longo, M. Brcic, F. Cabitza, J. Choi, R. Confalonieri, J.D. Ser, R. Guidotti, Y. Hayashi, F. Herrera, A. Holzinger, R. Jiang, H. Khosravi, F. Lecue, G. Malgieri, A. Páez, W. Samek, J. Schneider, T. Speith, S. Stumpf, Explainable artificial intelligence (XAI) 2.0: a manifesto of open challenges and interdisciplinary research directions, Inf. Fusion (ISSN 1566-2535) 106 (2024) 102301.

[22] A. Manukyan, E. Ceyhan, Classification of imbalanced data with a geometric digraph family, J. Mach. Learn. Res. (ISSN 1532-4435) 17 (1) (2016) 6504–6543.

[23] D. Minh, H.X. Wang, Y.F. Li, T.N. Nguyen, Explainable artificial intelligence: a comprehensive review, Artif. Intell. Rev. 55 (5) (2022) 3503–3568.

[24] I. Mollas, N. Bassiliades, G. Tsoumakas, LioNets: a neural-specific local interpretation technique exploiting penultimate layer information, Appl. Intell. (ISSN 0924-669X) 53 (3) (2023) 2538–2563.

[25] R.K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 2020, pp. 607–617.

[26] S.-H. Na, W.-J. Nam, S.-W. Lee, Toward practical and plausible counterfactual explanation through latent adjustment in disentangled space, Expert Syst. Appl. (ISSN 0957-4174) 233 (2023) 120982.

[27] J.D. Pascual-Triana, D. Charte, M. Andrés Arroyo, A. Fernández, F. Herrera, Revisiting data complexity metrics based on morphology for overlap and imbalance: snapshot, new overlap number of balls metrics and singular problems prospect, Knowl. Inf. Syst. (ISSN 0219-3116) 63 (7) (2021) 1961–1989.

[28] M. Pawelczyk, S. Bielawski, J.v.d. Heuvel, T. Richter, G. Kasneci, CARLA: a Python library to benchmark algorithmic recourse and counterfactual explanation algorithms, in: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1 (NeurIPS Datasets and Benchmarks 2021), vol. 1, 2021, pp. 1–13.

[29] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, P. Flach, FACE: feasible and actionable counterfactual explanations, in: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, 2020, pp. 344–350.

[30] R.R. Fernández, I. Martín de Diego, J.M. Moguerza, F. Herrera, Explanation sets: a general framework for machine learning explainability, Inf. Sci. (ISSN 0020-0255) 617 (2022) 464–481.

[31] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 1135–1144.

[32] J. Schoeffer, M. De-Arteaga, N. Kühl, Explanations, fairness, and appropriate reliance in human-AI decision-making, in: Proceedings of the CHI Conference on Human Factors in Computing Systems, CHI'24, Association for Computing Machinery, New York, NY, USA, ISBN 9798400703300, 2024, pp. 1–18.

[33] I. Stepin, J.M. Alonso, A. Catala, M. Pereira-Fariña, A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence, IEEE Access (ISSN 2169-3536) 9 (2021) 11974–12001.

[34] S. Verma, J. Dickerson, K. Hines, Counterfactual explanations for machine learning: a review, in: NeurIPS 2020 Workshop: ML Retrospectives, Surveys & Meta-Analyses (ML-RSA), 2020, pp. 1–22.
[35] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: automated decisions and the GDPR, Harv. J. Law Technol. 31 (2) (2018) 841–887.
[36] T. Wu, M.T. Ribeiro, J. Heer, D. Weld, Polyjuice: generating counterfactuals for explaining, evaluating, and improving models, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, 2021, pp. 6707–6723.