Producing Diverse Rashomon Sets of Counterfactual Explanations with Niching Particle Swarm Optimisation

Hayden Andersen
andershayd@ecs.vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Andrew Lensen
andrew.lensen@ecs.vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Will N. Browne
will.browne@qut.edu.au
Queensland University of Technology
Brisbane, Australia

Yi Mei
yi.mei@ecs.vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

ABSTRACT

Counterfactual explanation is a popular eXplainable AI technique, that gives contrastive explanations to answer potential "what-if" questions about the workings of machine learning models. However, research into how explanations are understood by human beings has shown that an optimal explanation should be both *selected* and *social*, providing multiple varying explanations for the same event that allow a user to select specific explanations based on prior beliefs and cognitive biases. In order to provide such explanations, a Rashomon set of explanations can be created: a set of explanations utilising different features in the data. Current work to generate counterfactual explanations does not take this need into account, only focusing on producing a single optimal counterfactual.

This work presents a novel method for generating a diverse Rashomon set of counterfactual explanations using the final population from a Particle Swarm Optimisation (PSO) algorithm. It explores a selection of PSO niching algorithms for PSO and evaluates the best algorithm to produce these sets. Finally, the ability of this method to be implemented and trusted by users is discussed.

KEYWORDS

Machine learning, Particle swarm optimisation, explainable AI

ACM Reference Format:
Hayden Andersen, Andrew Lensen, Will N. Browne, and Yi Mei. 2023. Producing Diverse Rashomon Sets of Counterfactual Explanations with Niching Particle Swarm Optimisation. In *Genetic and Evolutionary Computation Conference (GECCO '23), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3583131.3590444

1 INTRODUCTION

Explainable AI [29] (XAI) is a field of AI that explores the idea of producing explanations for predictions made by AI systems. There are many forms that these explanations can take, whether they are explaining the workings of the model, approximations of how a prediction was made, or describing the use of individual data features. One such method of producing explanations is Counterfactual explanation. Counterfactual explanation is a post-hoc explanation technique, explaining a specific prediction *after* the prediction has been made by a machine learning (ML) model. They take the form of a counterfactual argument [19], stating what changes would need to be made to the original input to produce a different, more desired output [10]. Such explanations have many uses, from explaining credit application decisions [26] to ensuring regulations such as the GDPR are followed [10]. Counterfactual explanations can be difficult to produce when explaining black-box models because the decision boundaries of these models can be very complex [1].

One major shortcoming of existing counterfactual algorithms [18, 31] is that they produce a single alternative to the original input. However, when considering explanation at the human level, it is known that an optimal explanation is both selected and social [27]. These real-world requirements for the adoption and use of explanation techniques cannot be met by current algorithms. The challenge of meeting these requirements is inherently a multi-modal optimisation problem, as many different contrasting explanations could sufficiently explain the prediction to a user at the same level of fidelity. From a collection of explanations, each user should be provided with the most appropriate explanation for their individual experiences and beliefs. Borrowing from the field of statistics [2], the full set of optimal explanations is known as a *Rashomon set* [9].

Previous research [36] has shown that population-based algorithms are strong contenders for counterfactual generation. A key benefit of these algorithms is that they can treat the underlying model as a complete black-box, unlike counterfactual algorithms that rely on specific model types [18, 23]. This allows for much more flexibility in applying the explanation algorithm, allowing a user or organisation to use it without needing to change the overall ML framework they are already working with.

Particle Swarm Optimisation (PSO) [17] is one such population-based optimisation method. It consists of a number of particles that represent solutions, and these particles explore the search space to find the most optimal solutions. It is suitable for many tasks as it makes no assumptions about the underlying data landscape; it has been shown to be a highly performant algorithm for the task of counterfactual production [1]. However, previous research has only explored the production of a single counterfactual with PSO,

GECCO '23, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0119-1/23/07…$15.00
https://doi.org/10.1145/3583131.3590444

which does not meet the requirements for real-world adoption of the explanation method.

Niching effects are often used in PSO to solve multi-modal problems [24]. Due to the population-based nature of the algorithm, different areas of the particle swarm can focus on and optimise different solutions to the problem. There are many different approaches adopted, from implicitly encouraging population diversity due to the structure of the swarm to explicitly defining niches. As PSO has previously been shown to produce strong counterfactual explanations, applying niching to the process should provide a strong Rashomon set of explanations.

There are many considerations and challenges in producing a Rashomon set of counterfactual explanations. First, a Rashomon set, by definition, requires there to be diverse and conflicting explanations contained within it. However, many optimisation algorithms will not preserve this diversity, producing a Rashomon set with minimal variation. In addition, the Rashomon set must be generated so that there is strong diversity in the set, but is not overly bloated with similar explanations. Finally, as counterfactual explanations usually aim to produce a specified output, a large portion of the potential search space consists of invalid solutions to the optimisation problem. In order to focus on these challenges, this paper will only consider classification datasets with real-valued features. However, future work will explore other types of tasks.

This paper introduces a novel technique to generate diverse Rashomon sets of counterfactual explanations for predictions made by black-box models, integrating multiple different niching PSO algorithms into the method to identify the best possible algorithm. The specific goals of this research are to:

- propose a new approach to generate Rashomon sets of counterfactual explanations from the final PSO population;
- evaluate the proposed approach using a selection of niching PSO algorithms on various datasets to find the best approach for producing a diverse Rashomon set;
- evaluate and discuss the proposed approach in terms of application in the real-world.

2 BACKGROUND

2.1 Optimal Explanations

In supervised ML, it can be simple to determine an optimal prediction. For example, many optimisation problems seek to achieve the lowest or highest possible result on a given function or task [35]. The closest equivalent to this in XAI, especially post-hoc explanations, is fidelity. This is a measure of how closely an explanation matches the actual prediction made by an ML model [33].

However, this is challenging to measure for explanation methods, as explanations must be made with the end user in mind [29]. To this end, Miller et. al. [28] introduced a number of general considerations that should be taken in mind when designing an explanation algorithm [27]. To quickly summarise these considerations:

- Explanations are *contrastive*. An explanation should describe why an outcome happened *instead of* another outcome, rather than just explaining the outcome [11].
- Explanations are *selected*. An explanation will be unlikely to be accepted if it describes every possible cause of an outcome, as it is too much for a person to understand properly. Instead, it should select only a few causes that explain most of the outcome [13]. In addition, the selected causes should be in line with a person's cognitive biases and prior beliefs.
- Probabilities often do not matter. People are often not looking for the "most correct" explanation, instead preferring one that is consistent with prior beliefs. This is true even if they are told that the probability of the cause is lower [13, 25].
- Explanations are social. In the hypothetical perfect scenario, the explanation should be provided as part of a conversation or interaction [12]. This allows the explanation to be tailored to the person who the explanation is created for.

2.2 Rashomon Sets

While the idea of the Rashomon effect exists in many fields, it was first introduced in the context of statistics by Breiman in 2001 [2]. They defined the Rashomon effect as the existence of multiple different functions $f(x)$ such that each function gives about the same minimum error rate on a given task. Despite each function using different variables in the data and therefore conflicting with each other, they must all be taken as correct in the absence of extra information. This concept is easily transferred to the context of XAI: given a model to predict a result on a task, there can exist many different and equally correct explanations for that model.

In 2019, Fisher et al. extended the Rashomon effect to the idea of a Rashomon set [9]. Given the set of all possible models M, the Rashomon set is the subset of models $R \subseteq M$ such that each element $r \in R$ performs similarly to the best-performing model in M.

While the idea of the Rashomon effect has been criticised as a weakness of a badly defined problem [4], it has also been shown that the Rashomon effect can be used to discover unknown information patterns in the data [9] and to navigate trade-offs between desired model properties such as interpretability [34].

As discussed in Section 2.1, explanations should be selected and social to be able to be accepted by a user. A Rashomon set of explanations allows meets these criteria, as each explanation in the set can utilise different features in the data, allowing the correct explanation to be chosen by the user based on their prior beliefs. This is reinforced by the fact that probabilities often do not matter for explanation — even if one model in the set has a higher fidelity than the others, the other explanations may still be valid.

2.3 Counterfactual Explanations

Counterfactual explanations, first proposed by Watcher et al. [39], are statements of what changes would need to be made to an original input instance to produce a different output from an ML model. In terms of real-world use of counterfactual explanations, this different output is one that would be more preferred by a user — for example, a user could want to know what changes they could make in their lifestyle in order for a rejected loan application to be accepted. As with many explanation algorithms, the fidelity of the explanation is important as an optimal counterfactual should be as close as possible to the original instance. Figure 1 shows an example of different counterfactuals produced for a data point, changing the predicted class from A to B. It can be observed that there are multiple possible counterfactual explanations for a single prediction. These may modify different subsets of all the possible features

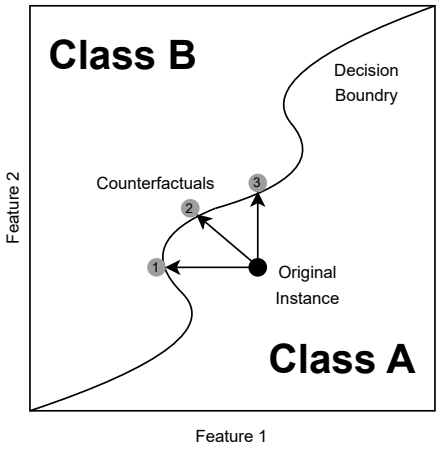

Figure 1: Counterfactual explanation example. A counterfactual can be produced by modifying one of each of the features or by modifying both of them. Modified from [1]

(dimensions in the search space). In this example, the set of explanations $\{1, 2, 3\}$ represents a Rashomon set of explanations where explanations 1 and 3 each only modify a single feature and explanation 2 modifies both features, but each by a lesser magnitude.

Counterfactual explanation is a useful post-hoc explanation method as it can provide a contrastive explanation, which, as discussed above, has been shown to be more effective and understandable to users [27]. In addition to this, it has been shown to fulfil the 'right to explanation' of the European GDPR data regulations [39].

In counterfactual generation, there are different definitions for "as close as possible to the original instance". This is usually taken as either the Euclidean distance or the Manhattan distance [10]. However, it is generally believed that the fewer features modified, the more likely the counterfactual is to be accepted by a user [27]. This means that the Manhattan distance, also known as the L_1 norm, is usually used. This is due to the sparsity-inducing properties of the L_1 norm [3, 6], which will push the individual differences of as many features as possible to be zero. Eq. (1) gives the Manhattan distance d between two points F_1 and F_2.

$$d = \sum_{i=1}^{dim(F)} |F_{1i} - F_{2i}| \tag{1}$$

Counterfactual explanations are often considered a full explanation of a prediction [5, 16], although they can also be used as a tool to discover more information about specific data or models [36, 41].

To clearly formulate the task of counterfactual explanation as an optimisation task, the goal is to find the optimal point c in the data landscape such that using it as an input to a trained classification model m produces a desired prediction class t. An optimal explanation is defined as the valid point that minimises d between the original point o and c. This problem is formulated as given in Eq. (2), where c^* is the optimal counterfactual explanation.

$$c^* = \underset{c}{argmin}\, d(o, c) | m(c) = t \tag{2}$$

While there is no current standardised method of evaluating counterfactual algorithms in the literature, a common approach, using a classification dataset, is to first train a classification model on the data and then select a number of random instances in the data to produce explanations for. These explanations can be evaluated on a number of metrics, including but not limited to the L_1, L_2, and L_∞ norms, the number of features modified, and the feasibility of the counterfactual as a real data point [1, 5, 16].

2.4 Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) [17] is a form of evolutionary computation, representing solutions as individual particles in a swarm. Each particle is represented as a vector of values, where the value in position i represents the position of the particle in the i^{th} data dimension. Similarly, each particle has a velocity through the data space, which is another vector of values. Through these vectors, the full representation of a particle is given as $([x_1, x_2, x_3, ..., x_D], [v_1, v_2, v_3, ..., v_D])$ where x_i and v_i give the position and velocity along dimension i, and D is the total number of dimensions in the data.

At each algorithmic step, each particle moves through the search space, being pulled towards both the best position that the particle has found so far in the data (*pbest*) and the best position found by the full swarm in the data (*gbest*). At each step, every particle i is updated according to Equations (3) and (4), where t represents the t^{th} step in the search process, r_1 and r_2 are random vectors sampled from U(0, 1), and w, c_1, and c_2 are provided constants.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{3}$$

$$v_i^{t+1} = w \cdot v_i^t + c_1 r_1 (pbest_i^t - x_i^t) + c_2 r_2 (gbest^t - x_i^t) \tag{4}$$

The constant w refers to the inertia weight of the algorithm. w is used to balance the exploration of new solutions with the exploitation of already discovered good solutions. The constants c_1 and c_2 refer to acceleration constants and are used to balance the contributions of pbest and gbest.

2.5 Niching PSO

Multi-modal optimisation is the task of solving problems for which there are more than one global or local optima, where it is often desired to find all of these optima. This is consistent with the idea of selected explanations: each potential explanation — that utilises different features — is a different optimum that must be found.

PSO has been shown [8] to be a competitive approach on multi-modal problems, with many niching algorithms proposed to improve performance in this task. Four commonly used niching PSO algorithms [22] are SPSO [30], E-SPSO [22], FER-PSO [20], and PSO with a ring topology [21]. Each of these algorithms replaces the *gbest* in Eq. (4) with a different particle to encourage unique niches of particles optimising in different areas of the search space.

2.5.1 SPSO. Species-based PSO (SPSO) [30] incorporates *speciation* of particles in the PSO algorithm. It introduces the species radius (r_s) as a hyperparameter. At every algorithmic step, a set of species is identified. First, the population is sorted by fitness. Then, a set of seeds for each species is initialised, and each particle is considered in turn: a particle is added to the first species for which it is less than r_s away from the corresponding seed. If it is not added to any species, it is added to the set of seeds and forms a new species. Once each species seed has been found, the global best in Eq. (4) is replaced with the seed from the species that particle i belongs to.

In addition to this, SPSO introduces a mechanism for replacing redundant particles in the population. Any particles in the population with the same fitness as their species' seed are removed, and random new particles are generated to replace them.

2.5.2 E-SPSO. E-SPSO [22] is an extension of SPSO that incorporates an equilibrium factor into the algorithm. It first performs the same process as SPSO in order to find each species seed, then finds the largest species LN and the smallest species SN. It then takes the average size DS of these two species and adds an extra term in Eq. (4) to the DS worst particles in LN that encourages the velocity of each particle towards the seed of SN. This helps to prevent the largest species from becoming too large.

E-SPSO also utilises the local search algorithm for PSO suggested by Qu et al. [32]. At each algorithmic step, the nearest personal best *pbest_nearest* to the *pbest* for each particle is found. A new random temporary particle is created that is slightly closer to *pbest_nearest* from *pbest*. If the fitness of this new particle is better than that of *pbest*, it replaces *pbest* as the personal best of that particle.

2.5.3 FER-PSO. PSO based on Fitness Euclidean-distance Ratio (FER) [20] considers the personal best particles as a *memory swarm* and the current positions of the particles as an *explorer swarm*. *gbest* is replaced by the *fittest and closest* neighbourhood best of particle i in the swarm. This neighbourhood best is the particle in the memory swarm which has the highest FER value. The FER value between i and each other particle j is the ratio of the difference in fitness between i and j and the Euclidean distance between i and j.

Unlike the SPSO methods that define a niche per particle, the replacement for *gbest* in FER-PSO is more fluid. Each particle instead finds its own individual replacement based on the FER values.

2.5.4 Ring Topology. PSO with a ring topology [21] is a simpler method than the SPSO and FER-PSO methods, simply restricting the connections between particles. Despite this, it has been shown to perform competitively with the more complex methods. The ring topology restricts each particle so it only knows of two neighbouring particles in the swarm. The global best *gbest* is then replaced in Eq. (4) by a local best *lbest*, the best particle found by either the particle or its two neighbours. This creates implicit niching properties without the explicit niching provided by the other methods.

2.6 Related Work

There is, to our knowledge, no existing work to produce sets of counterfactual explanations using PSO. We previously [1] showed that PSO is a competitive method for producing counterfactual explanations; however, this only focused on producing a single counterfactual explanation and did not utilise the innate ability of PSO to produce multiple diverse solutions.

Dandl et al. [5] explored the generation of sets of counterfactual explanations using the NSGA-II algorithm, treating the explanation as a multi-objective problem. They treated the objectives as the explanation fidelity, the number of features modified, and how well the counterfactual would fit the original data distribution. While this does provide a good set of explanations, this is only explored in the context of trade-offs between these multiple objectives and does not consider that for the purposes of explanation, it is desired to have multiple different but similarly performing explanations.

3 PROPOSED METHOD

The proposed approach is to use niching PSO algorithms to generate a population of PSO particles representing counterfactual explanations for a provided black-box model m. While a multi-objective solution was considered, these only provide trade-offs between objectives and do not allow for the creation of multiple different solutions in the final solution set that exhibit similar results for each objective. Instead, the final population of particles from a niching PSO algorithm is used to allow for these similar trade-offs.

Each counterfactual explanation (solution) is represented as a real-valued feature vector, with the same dimensionality as the data used to train m. In order to enforce that each feature is considered equally, and simplify some equations, every feature is scaled to be within the range $[0, 1]$. If m is trained without this scaling applied, the data is reverted to the original range before it is passed into m.

The fitness for an individual counterfactual is measured as the Manhattan distance d between the counterfactual and the original input data. This is consistent with prior work [1, 10] and encourages the algorithm to significantly change as few features as possible.

The optimisation goal of the algorithm is to produce a set of valid counterfactuals. A valid counterfactual is defined as one that modifies the prediction of m from the original class to a specified class t and is within the specified range of $[0, 1]$. Mathematically this can be given by Eq. (5), where C is the set of all possible valid counterfactual explanations and o is the original input instance.

$$C = \{c_{1..n} | n = dim(o), m(c) = t, \forall c_i : c_i \in [0, 1]\} \tag{5}$$

As the PSO algorithm performs its own bounds checking by moving any particle outside the defined search space back to the edge of the search space, the only consideration needed in this custom algorithm is how to handle a solution for which $m(c)$ predicts a class other than the target class t. As the PSO algorithm does not require a differentiable fitness landscape and only particularly depends on the best position found by each particle, an invalid particle can be assigned with a marker for invalid fitness. In this paper, that marker is an arbitrarily large value, denoted as ∞. The fitness of a given particle c is calculated according to Eq. (6). This is the function which the PSO algorithms aim to minimise.

$$f(c) = \begin{cases} \sum_{i=1}^{dim(o)} |o_i - c_i| & m(c) = t \\ \infty & m(c) \neq t \end{cases} \tag{6}$$

Algorithm 1 is used to generate a final population of counterfactual particles. This algorithm takes a pre-trained model m, a desired counterfactual class t, and an original input o (from the data). For a specified number of iterations, each particle is updated according to the best position it has found so far and a global/neighbourhood best based on the niching technique used. Lines marked with α are used for normal PSO, with β for SPSO, with γ for E-SPSO, with δ for FER-PSO, and lines marked with ϵ are used for a ring topology.

Once the final PSO population has been evaluated, a Rashomon set of explanations is created from the results, as shown in Figure 2. First, each particle in the population is replaced by their *pbest*, the best position that particular particle has found. For each particle, the specific features that are significantly modified to generate the counterfactual, here defined as a feature that is more than 0.01 away from the value in the original instance, are found. This represents a

Algorithm 1: PSO algorithm for counterfactual generation, with modifications for various niching algorithms

Parameters: w, c_1, c_2
Data: m model to be explained, o point to be explained, t desired prediction

generate population randomly, sampled from U(0, 1);
$gbest \leftarrow \infty$;
for *particle in population* **do**
  $pbest \leftarrow \infty$;
  $v \leftarrow$ random values from U(0, 0.5);
end
for *desired number of iterations* **do**
  for *particle in population* **do**
    if $m(particle)$ *is* t **then**
      $fitness \leftarrow d(particle, o)$ (Eq. (1));
    else
      $fitness \leftarrow \infty$;
    end
    if *fitness* < *pbest* **then**
      $pbest \leftarrow$ particle;
    end
    if *fitness* < *gbest* **then**
      $gbest \leftarrow$ particle;
    end
  end
  β, γ: find species seed for each particle based on fitness and position;
  for *particle in population* **do**
    α: update v by Eq. (4);
    β: update v by Eq. (4) replacing *gbest* with species seed;
    γ: update v by Eq. (4) replacing *gbest* with species seed and including equilibrium factor;
    δ: $ferbest \leftarrow pbest$ of different particle that maximises *FER* value;
    δ: update v by Eq. (4) replacing *gbest* with $ferbest$;
    ϵ: $lbest \leftarrow$ best of *pbest* of particle and neighbours;
    ϵ: update v by Eq. (4) replacing *gbest* with *lbest*;
    update particle according to Eq. (3);
  end
  β: replace redundant particles in the population;
  γ: perform local search for each particle;
end
return full population;

change of more than 1% of the range of the data. This is represented in Fig. 2 as a bit vector, where a 1 represents a significant change and a 0 represents no significant change. For every unique combination of features modified, the particle with the best fitness is found. The Rashomon set is then taken as each of these best particles. This means that the Rashomon set is made of the explanations that are most similar to the original prediction while also balancing the ability to produce a diverse set of solutions.

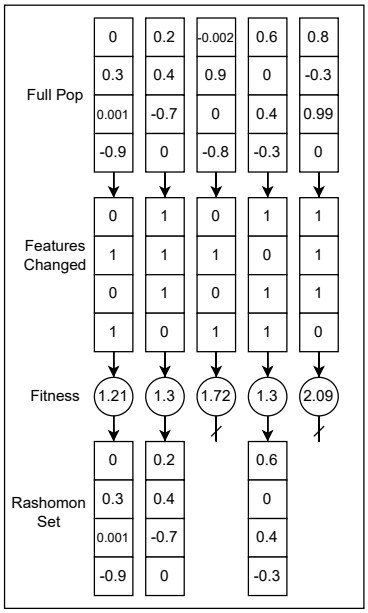

Figure 2: An example of reduction from a final population of results to a Rashomon set.

4 EXPERIMENTAL DESIGN

4.1 Datasets

Nine datasets were selected to evaluate the proposed methods. Seven were sourced from the UCI machine learning repository [7], one (kc2) from the OpenML platform [38], and the final dataset (Penguins) is from an online R repository [14]. These datasets each present classification tasks that operate over a continuous data domain, providing a good analogue for a real-world task requiring counterfactual explanations. Table 1 shows the selected datasets, ordered by the (approximate) complexity of the task.

Two particular datasets that are worth highlighting are the Segmentation dataset and the Ionosphere dataset. Both of these datasets initially contained one feature for which every instance had the same value. As it would be problematic for a counterfactual to have the option to modify these features, they have been removed.

4.2 Experiment Setup

To explore the ability of various niching PSO algorithms in creating diverse Rashomon sets of counterfactual explanations, we compared with basic PSO. For each algorithm, standard PSO parameters of $w = 0.7298$, $c_1 = 1.4962$, and $c_2 = 1.4962$ are used [37]. For the two species-based algorithms, the species radius r_s is set to 1.0. Our method is designed to be easily implemented in an existing ML setup, and so we did not tune the parameters for each problem.

For each niching PSO algorithm, the following steps are taken:

(1) Scale the input data so that each feature is in the range $[0, 1]$.
(2) Train a black-box model on the provided training data. While a random forest classifier is used for these experiments, any black-box model could be used.
(3) Select a single random instance from the data.
(4) Use the black-box model to predict the class of the instance.

Table 1: Chosen datasets

Name	Features	Instances	Classes
Penguins	4	333	3
Breast Cancer Wisconsin (BCW)	10	699	2
Wine	13	178	3
Segmentation	18	2310	7
kc2	21	522	2
Steel Plates Faults (SPF)	27	1941	2
Ionosphere	33	351	2
Dermatology	34	366	6
Madelon	500	4400	2

(5) Choose a class in the data that was not predicted by the original instance.
(6) Apply the PSO algorithm in Algorithm 1 to find a final population, treating t as the class selected in Step 5.
(7) Create a Rashomon set of explanations as shown in Fig. 2.
(8) Record statistics on the Rashomon set size, the size of the explanations, and the L_1 and L_2 norms of the explanations.
(9) Repeat from Step 5 until a Rashomon set has been produced for each class apart from the original prediction.
(10) Repeat from Step 3 20 times, using the same black-box model but a different chosen instance.

Thus, each instance has a Rashomon set created for each other possible class. While this means that more sets will be created for datasets with greater numbers of potential classes, it is important as it lessens the statistical impact of any false-positive results where strong performance is reported simply because two classes are represented by very similar data in the overall dataset. Likewise, each black-box model is used multiple times to ensure results are not biased by the original instances chosen. Both the black-box model training and the instance selection use a specified randomisation seed. This ensures that the same instances in the same models are explained between each algorithm to provide a fair evaluation.

To provide strong statistical significance to the results in a non-deterministic environment, each algorithm and dataset undergoes the above steps 30 times with 30 different random seeds.

This paper only evaluates PSO-based methods of producing Rashomon sets of counterfactual explanations, as no other algorithms in the literature focus on producing a Rashomon set of counterfactual explanations. Any EC algorithm such as the GA method used by Sharma et al. [36] could be used to generate Rashomon sets, however, we build on our prior work that showed PSO outperforms other EC algorithms for counterfactual production.

5 RESULTS AND DISCUSSION

As a simple example of a counterfactual explanation produced: on the Penguins dataset it was discovered that if the bill length of an Adelie penguin was increased by 5.95cm it would instead be classified as a Chinstrap penguin by the prediction model.

The quantitative results of the experiments are shown in Table 2. For each combination of dataset and algorithm, the table first shows the mean size of the produced Rashomon sets. Additionally, it shows the mean of the average, best, and worst counts of features modified within each set, and the same for the Manhattan (L_1) and the Euclidean (L_2) distances from the original instance to the counterfactual explanations. While Euclidean distance is not utilised in the algorithm itself, it represents the most natural human understanding of distance [40]. The Manhattan distance can also be heavily influenced by having too many features. Thus, the Euclidean distance gives a more pure metric for distance from the original instance.

The set size of the Rashomon set is taken as a measure of the diversity of the final solution. This is because each solution in the final Rashomon set utilizes a different selection of features from the full feature set. Each of these unique combinations of features can be taken as a distinct local optima in the optimisation task of producing counterfactual explanations.

A Friedman test followed by post-hoc Nemenyi tests is performed for each of the metrics, with a α of 0.05. For each metric, a $\uparrow$ is shown, and the result is **bolded** if it is statistically significantly better than each other algorithm, and a $\downarrow$ is shown, and the result is *italicised* if the result is statistically significantly worse than each other algorithm. For the purposes of these, a larger final Rashomon set is considered better, as it gives more variety in solutions. For all other metrics, a lower value is considered better.

5.1 Analysis

Table 3 shows the number of wins and losses each algorithm has across each metric, where wins are bolded and losses are italicised to match Table 2. A win or a loss is only recorded if it is statistically significantly better or worse than all four other algorithms.

From this table, some clear patterns begin to emerge. In terms of producing the largest possible Rashomon set, and therefore the most diverse set of solutions, SPSO is a clear winner, winning by a large margin in all but MADELON, the largest and most complex dataset. In addition to this, SPSO only has three clear losses across the entire suite of experiments. This means that in cases where strong diversity is desired, SPSO is potentially the best choice of algorithm. However, closer analysis shows that while SPSO has very few losses, in many cases, it performs significantly worse than the best algorithm for that metric and generally has performance closer to that of the worst-performing algorithm.

FER-PSO has 22 wins across all metrics, whereas the ring topology has 23 wins; neither method has any losses. This means that for creating Rashomon sets of high-fidelity explanations, these are the two best algorithms to use. Between these two algorithms, FER-PSO creates less diverse Rashomon sets with smaller sizes in almost all datasets; however, it beats the ring topology more often in terms of the worst solutions in the Rashomon set. Essentially, this means that FER-PSO creates less diverse sets but with the trade-off of a better performance floor within the sets. Additionally, FER-PSO performs better than the ring topology on the two more complex datasets, dermatology and MADELON.

E-SPSO has the worst performance, losing on a majority of metrics. In this problem, especially on multi-class datasets, there tends to be a large area of the data space that produces invalid solutions due to $m(x)$ producing a different class than the target class t. As the equilibrium factor in E-SPSO does not check where the locations of the smallest and largest species are, it is very likely that it simply pushes a number of particles within the invalid portion of the data space. This slows down the algorithm and loses some of the benefits provided by the speciation mechanism of SPSO.

Table 2: Experimental results. The best algorithm for each metric is marked with a ↑ and the worst is marked with a ↓

Dataset	Method	Set Size	Features modified			L_1			L_2		
			Average	Best	Worst	Average	Best	Worst	Average	Best	Worst
Penguins	PSO	*2.25*↓	1.99	1.61	2.37	0.343	0.274	**0.449**↑	0.274	0.228	**0.343**↑
	SPSO	**7.35**↑	2.69	1.44	4.0	0.374	0.277	0.635	0.292	0.224	0.46
	E-SPSO	3.06	*3.38*↓	*3.0*↓	4.0	*0.849*↓	*0.674*↓	*1.06*↓	*0.56*↓	*0.443*↓	*0.693*↓
	FER-PSO	2.7	1.97	1.45	2.5	0.345	0.271	0.462	0.276	0.221	0.356
	Ring Topology	3.68	2.02	1.4	2.72	0.346	0.273	0.477	0.277	**0.22**↑	0.37
BCW	PSO	*5.63*↓	5.09	3.14	5.74	1.31	0.659	1.8	0.631	0.413	0.798
	SPSO	**71.8**↑	7.42	4.06	10.0	2.86	1.07	5.34	1.29	0.557	2.1
	E-SPSO	28.5	*7.88*↓	*6.21*↓	9.83	*4.39*↓	*2.59*↓	6.41	1.8	*1.27*↓	2.29
	FER-PSO	6.09	5.0	**2.93**↑	5.8	1.31	0.629	1.81	0.632	0.4	0.808
	Ring Topology	14.2	5.43	3.51	7.22	1.41	0.856	2.07	0.68	0.432	0.953
Wine	PSO	5.61	4.99	2.89	6.02	1.02	0.476	1.6	0.478	0.328	0.649
	SPSO	**57.6**↑	11.0	8.91	13.0	3.42	1.95	5.24	1.25	0.729	1.79
	E-SPSO	12.2	*12.0*↓	*11.2*↓	13.0	*4.47*↓	*3.47*↓	5.64	1.49	*1.17*↓	*1.82*↓
	FER-PSO	5.69	4.83	2.7	5.86	0.986	0.44	1.57	0.461	0.309	0.632
	Ring Topology	13.5	4.97	**2.31**↑	7.71	**0.961**↑	0.453	1.85	0.457	**0.287**↑	0.727
Segmentation	PSO	*12.6*↓	10.5	6.22	12.2	3.15	1.2	4.69	1.06	0.573	1.43
	SPSO	**88.4**↑	15.5	12.3	18.0	5.76	3.26	9.0	1.8	1.1	2.56
	E-SPSO	21.8	*16.7*↓	*15.4*↓	18.0	*7.36*↓	*5.57*↓	*9.34*↓	*2.11*↓	*1.66*↓	*2.56*↓
	FER-PSO	12.9	10.3	**5.87**↑	12.0	3.09	**1.11**↑	**4.69**↑	1.03	**0.53**↑	**1.42**↑
	Ring Topology	25.9	9.92	6.08	13.5	**2.91**↑	1.43	5.13	0.999	0.584	1.54
kc2	PSO	2.76	6.08	5.11	6.75	0.86	0.377	1.31	0.279	0.179	0.377
	SPSO	**148.0**↑	16.8	11.7	20.8	6.28	2.4	13.2	1.98	0.926	3.37
	E-SPSO	43.7	*18.3*↓	*14.7*↓	20.9	*14.0*↓	*9.13*↓	*17.2*↓	*3.52*↓	*2.78*↓	*3.96*↓
	FER-PSO	2.69	5.66	4.65	6.38	0.835	0.351	1.3	0.262	0.161	0.364
	Ring Topology	13.3	4.7	**2.49**↑	6.7	1.11	**0.33**↑	2.04	0.321	**0.136**↑	0.538
SPF	PSO	1.18	4.81	4.66	4.96	0.569	0.523	0.634	0.316	0.305	0.331
	SPSO	**329.0**↑	23.7	18.8	*27.0*↓	9.7	5.45	15.3	2.43	1.48	3.46
	E-SPSO	41.2	23.9	*21.6*↓	26.4	*14.6*↓	*11.6*↓	*18.2*↓	*3.33*↓	*2.82*↓	*3.86*↓
	FER-PSO	1.25	4.19	4.02	4.36	0.387	0.346	0.439	0.211	0.201	0.223
	Ring Topology	22.9	**2.18**↑	**1.583**↑	4.52	**0.109**↑	**0.0417**↑	**0.317**↑	**0.0525**↑	**0.0218**↑	**0.136**↑
Ionosphere	PSO	1.34	7.51	6.85	7.94	1.53	1.3	1.72	0.704	0.663	0.737
	SPSO	**342.0**↑	*28.3*↓	22.4	*33.0*↓	9.74	6.28	13.5	2.22	1.52	2.87
	E-SPSO	29.2	28.0	*26.3*↓	30.5	*12.8*↓	*10.3*↓	*16.6*↓	*2.76*↓	*2.31*↓	*3.37*↓
	FER-PSO	1.52	6.65	5.95	**7.05**↑	1.2	0.959	1.39	0.568	0.521	0.605
	Ring Topology	60.6	**5.12**↑	**1.85**↑	11.5	**0.53**↑	**0.317**↑	1.12	**0.297**↑	**0.194**↑	0.504
Dermatology	PSO	43.8	21.0	14.5	23.4	8.46	5.01	11.6	1.97	1.43	2.49
	SPSO	**314.0**↑	28.5	17.9	34.0	12.7	7.63	18.0	2.82	1.94	3.65
	E-SPSO	79.6	*31.7*↓	*28.3*↓	33.8	*15.0*↓	*10.9*↓	*19.5*↓	*3.09*↓	*2.4*↓	3.82
	FER-PSO	43.8	20.9	**14.4**↑	**23.2**↑	**8.36**↑	**4.86**↑	**11.5**↑	**1.92**↑	**1.36**↑	**2.45**↑
	Ring Topology	59.8	22.0	16.9	25.2	8.71	5.89	11.8	1.96	1.45	2.51
MADELON	PSO	500.0	347.0	334.0	370.0	60.1	59.6	64.9	4.81	4.8	4.96
	SPSO	500.0	490.0	478.0	498.0	153.0	123.0	226.0	8.01	6.7	10.9
	E-SPSO	*184.0*↓	*496.0*↓	*491.0*↓	*500.0*↓	*212.0*↓	*192.0*↓	*247.0*↓	*10.2*↓	*9.35*↓	*11.6*↓
	FER-PSO	500.0	**340.0**↑	**323.0**↑	372.0	**45.7**↑	**44.7**↑	**53.8**↑	**3.93**↑	3.9	**4.23**↑
	Ring Topology	500.0	463.0	444.0	481.0	65.4	59.8	76.5	4.05	**3.67**↑	4.62

Table 3: Wins and Losses by Method and Metric

Method	Set Size	Features modified			L_1			L_2		
		Average	Best	Worst	Average	Best	Worst	Average	Best	Worst
PSO	**0**/*3*	**0**/*0*	**0**/*0*	**0**/*0*	**0**/*0*	**0**/*0*	**1**/*0*	**0**/*0*	**0**/*0*	**1**/*0*
SPSO	**8**/*0*	**0**/*1*	**0**/*0*	**0**/*2*	**0**/*0*	**0**/*0*	**0**/*0*	**0**/*0*	**0**/*0*	**0**/*0*
E-SPSO	**0**/*1*	**0**/*7*	**0**/*9*	**0**/*1*	**0**/*9*	**0**/*9*	**0**/*7*	**0**/*7*	**0**/*9*	**0**/*7*
FER-PSO	**0**/*0*	**1**/*0*	**4**/*0*	**2**/*0*	**2**/*0*	**3**/*0*	**3**/*0*	**2**/*0*	**2**/*0*	**3**/*0*
Ring Topology	**0**/*0*	**2**/*0*	**4**/*0*	**0**/*0*	**4**/*0*	**3**/*0*	**1**/*0*	**2**/*0*	**6**/*0*	**1**/*0*

The base PSO algorithm exhibits average performance compared with the other algorithms, winning and losing on almost no metrics. It also tends to place in the middle range of the metrics. This is interesting, as it is the basic PSO algorithm without any extra niching mechanisms included. This is the same algorithm used to create counterfactual explanations by Andersen et al. [1]. While some niching algorithms can produce better results, this shows that PSO can still produce usable Rashomon sets of explanations without explicit niching control.

The MADELON dataset was artificially created as a feature selection challenge, with only 20 features that are informative of the overall class. Interestingly, the lowest number of features modified among any Rashomon set from any algorithm was 323 features by FER-PSO. Thus, even in the best case, 300 redundant features were used as part of the explanation. However, despite this, the overall fidelity (L_1 and L_2) of the solutions is quite low, especially the L_2 metric. This means that despite modifying many features, it is only modifying them by a very small amount each. Further research should focus on placing more explicit limits on the number of features modified, cf work by Dandl et al. [5].

5.2 Further Discussion

A very important consideration for XAI methods is how well they provide explanations and how well they can be implemented in a real-world context. Many companies and end users will not implement XAI methods unless they can be done with very little effort, so this is an important consideration when designing these methods.

Thus, methods with fewer hyperparameters are often preferred: less hyperparameter tuning is required, so the turnaround time to implement is much quicker. From the algorithms explored in this paper, neither FER-PSO nor the ring topology requires extra tuning, while SPSO and E-SPSO both require a niching parameter to be set. This aligns with the results found, where the algorithms without extra parameters performed consistently better across the metrics.

As per Section 2.1, an optimal explanation algorithm should provide explanations that are contrastive, selected, social, and do not only focus on probabilities. Our method proposed in this paper meets each of these considerations, as discussed in turn below.

Contrastive: Counterfactual explanations are inherently contrastive as they provide an explanation of how an instance would need to be changed to produce a different prediction. This is the same as explaining why the given prediction was reached instead of a preferred one, simply framed from the other direction. *Selected:* A selected explanation is, simply put, one using as few features as possible while still being correct. Observing the results produced in the experiments by the best-performing algorithms, every dataset except for Dermatology and MADELON has an average explanation size across the Rashomon set of at most half of the total number of features in the data, with many of them being well under this number. For the best explanation in each set, only MADELON produces explanations that use more than half of the features. This makes the explanations more likely to be accepted, as they are much simpler than one that would use every feature.

Social: For an explanation method to be social, it has to have the capability to provide multiple varying explanations. While not fully providing this mechanism, in this work, this ability is represented by the size of the Rashomon set. As each explanation in the Rashomon set uses a unique combination of features, this means that a larger set has more ability to provide a social explanation. In these experiments, the sets produced by SPSO most closely follow this idea, producing very large Rashomon sets allowing for many variations of explanations. However, the ring topology has the strongest trade-off between Rashomon set size and fidelity, producing high-quality explanations but still having reasonably large sets. While a large Rashomon set does not provide social explanations on its own, it provides the opportunity for these explanations.

Not focused on probabilities: An important part of this work is that the entire Rashomon set is evaluated, including the worst explanation in the final set. Despite this, once the final set is produced, each explanation in the Rashomon set should be treated as an equally valid explanation. This is consistent with the idea that probabilities do not matter in explanations. If a user prefers an explanation that uses a specific combination of features, it should be treated as better than an explanation with a higher fidelity that uses different features. In producing Rashomon sets, the focus should be on increasing the floor of the fidelity within the set rather than continuing to improve the ceiling like many algorithms focus on.

6 CONCLUSIONS

This paper introduced a new mechanism for producing Rashomon sets of counterfactual explanations of predictions in black-box ML models. To our knowledge, this is the first work to explore the synthesis of such sets; previous work focused only on trade-offs between different explanations. We showed that both the FER-PSO and ring topology niching methods are especially appropriate for this task, producing high-quality Rashomon sets with good fidelity to the original instance while still retaining strong variety in the features that are used. We showed that the ring topology should be used when a more diverse Rashomon set is desired, and FER-PSO should be used when the overall fidelity of the Rashomon set must be as high as possible. The suitability of this mechanism for real-world adoption was also discussed, showing that the algorithms can be integrated into an existing ML ecosystem with very little tuning and that the Rashomon sets of explanations allow for explanations backed by psychology that users are more likely to accept.

Future work will focus on stronger mechanisms for social explanations, exploring how a diverse Rashomon set can be appropriately presented to a user to provide individual users explanations that they are more likely to accept. While this was briefly touched on in this paper, it requires more practical work to show the benefits of having this larger set. In addition to this, future work should focus on introducing a more complex fitness function that considers each feature differently based on the abnormality of the feature values. Studies have shown that explanations are strongly preferred when they focus on abnormal causes [15], and so this should be utilised and encouraged in the mechanism to produce explanations. Finally, this paper has only explored the production of Rashomon sets of counterfactual explanations for continuous classification problems. Future work will explore different types of data to make this technique more widely applicable to different domains.

REFERENCES

[1] Hayden Andersen, Andrew Lensen, Will N Browne, and Yi Mei. 2022. Evolving Counterfactual Explanations with Particle Swarm Optimization and Differential

Evolution. In *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE Press, 01–08. https://doi.org/10.1109/CEC55065.2022.9870283

[2] Leo Breiman. 2001. Statistical Modeling: The Two Cultures. *Statist. Sci.* 16, 3 (2001), 199–231. https://doi.org/10.1214/ss/1009213726

[3] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. 2006. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics* 59, 8 (aug 2006), 1207–1223. https://doi.org/10.1002/CPA.20124 arXiv:0503066 [math]

[4] Alexander D'Amour. 2021. Revisiting Rashomon: A Comment on "The Two Cultures". *ArXiv* abs/2104.0 (2021).

[5] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. 2020. Multi-Objective Counterfactual Explanations. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12269 LNCS (apr 2020), 448–469. https://doi.org/10.1007/978-3-030-58112-1_31 arXiv:2004.11165

[6] David L. Donoho. 2006. For most large underdetermined systems of linear equations the minimal l-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics* 59, 6 (jun 2006), 797–829. https://doi.org/10.1002/CPA.20132

[7] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[8] A. P. Engelbrecht, B. S. Masiye, and G. Pampará. 2005. Niching ability of basic particle swarm optimization algorithms. *Proceedings - 2005 IEEE Swarm Intelligence Symposium, SIS 2005* 2005 (2005), 407–410. https://doi.org/10.1109/SIS.2005.1501650

[9] Aaron J Fisher, Cynthia Rudin, and Francesca Dominici. 2019. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *Journal of machine learning research : JMLR* 20 (2019).

[10] Bryce Goodman and Seth Flaxman. 2017. European union regulations on algorithmic decision making and a "right to explanation". *AI Magazine* 38, 3 (2017), 50–57. https://doi.org/10.1609/aimag.v38i3.2741 arXiv:1606.08813

[11] Germund Hesslow. 1988. The problem of causal selection. *Contemporary science and natural explanation: Commonsense conceptions of causality* (1988), 11–32.

[12] Denis J Hilton. 1990. Conversational processes and causal explanation. *Psychological Bulletin* 107, 1 (1990), 65–81. https://doi.org/10.1037/0033-2909.107.1.65

[13] Denis J Hilton. 1996. Mental Models and Causal Explanation: Judgements of Probable Cause and Explanatory Relevance. *Thinking & Reasoning* 2, 4 (1996), 273–308. https://doi.org/10.1080/135467896394447

[14] Allison Marie Horst, Alison Presmanes Hill, and Kristen B Gorman. 2020. *palmerpenguins: Palmer Archipelago (Antarctica) penguin data.* https://doi.org/10.5281/zenodo.3960218

[15] Daniel Kahneman and Amos Tversky. 1982. *The simulation heuristic.* Cambridge University Press, 201–208. https://doi.org/10.1017/CBO9780511809477.015

[16] Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. 2019. Model-Agnostic Counterfactual Explanations for Consequential Decisions. *CoRR* abs/1905.1 (2019). http://arxiv.org/abs/1905.11190

[17] J. Kennedy and R. Eberhart. 1995. Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks* 4 (1995), 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

[18] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. 2017. Inverse Classification for Comparison-based Interpretability in Machine Learning. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations*. Springer International Publishing. arXiv:1712.08443 https://arxiv.org/abs/1712.08443v1

[19] David Lewis. 1973. Counterfactuals and Comparative Possibility. *Journal of Philosophical Logic* 2, 4 (1973), 418–446. http://www.jstor.org/stable/30226074

[20] Xiaodong Li. 2007. A Multimodal Particle Swarm Optimizer Based on Fitness Euclidean-Distance Ratio. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07)*. Association for Computing Machinery, New York, NY, USA, 78–85. https://doi.org/10.1145/1276958.1276970

[21] Xiaodong Li. 2010. Niching Without Niching Parameters: Particle Swarm Optimization Using a Ring Topology. *IEEE Trans. Evol. Comput.* 14, 1 (2010), 150–169. https://doi.org/10.1109/TEVC.2009.2026270

[22] Yikai Li, Yongliang Chen, Jinghui Zhong, and Zhixing Huang. 2019. Niching particle swarm optimization with equilibrium factor for multi-modal optimization. *Inf. Sci.* 494 (2019), 233–246. https://doi.org/10.1016/j.ins.2019.01.084

[23] Shusen Liu, Bhavya Kailkhura, Donald Loveland, and Yong Han. 2019. Generative Counterfactual Introspection for Explainable Deep Learning. In *2019 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2019, Ottawa, ON, Canada, November 11-14, 2019*. IEEE, 1–5. https://doi.org/10.1109/GlobalSIP45357.2019.8969491 arXiv:1907.03077

[24] Wenjian Luo, Yingying Qiao, Xin Lin, Peilan Xu, and Mike Preuss. 2022. Hybridizing Niching, Particle Swarm Optimization, and Evolution Strategy for Multimodal Optimization. *IEEE Trans. Cybern.* 52, 7 (2022), 6707–6720. https://doi.org/10.1109/TCYB.2020.3032995

[25] John McClure. 2002. Goal-based Explanations of Actions and Outcomes. *European Review of Social Psychology* 12, 1 (2002), 201–235. https://doi.org/10.1080/14792772143000067

[26] Rory McGrath, Luca Costabello, Chan Le Van, Paul Sweeney, Farbod Kamiab, Zhao Shen, and Freddy Lécué. 2018. Interpretable Credit Application Predictions With Counterfactual Explanations. *CoRR* abs/1811.0 (2018). http://arxiv.org/abs/1811.05245

[27] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (feb 2019), 1–38. https://doi.org/10.1016/j.artint.2018.07.007 arXiv:1706.07269

[28] Tim Miller, Piers Howe, and Liz Sonenberg. 2017. Explainable AI: Beware of Inmates Running the Asylum Or: How I Learnt to Stop Worrying and Love the Social and Behavioural Sciences. *CoRR* abs/1712.0 (2017). http://arxiv.org/abs/1712.00547

[29] Christoph Molnar. 2019. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Leanpub. https://christophm.github.io/interpretable-ml-book/

[30] Daniel Parrott and Xiaodong Li. 2006. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Trans. Evol. Comput.* 10, 4 (2006), 440–458. https://doi.org/10.1109/TEVC.2005.859468

[31] Rafael Poyiadzi, Kacper Sokol, Raúl Santos-Rodríguez, Tijl De Bie, and Peter A Flach. 2020. FACE: Feasible and Actionable Counterfactual Explanations. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (2020).

[32] B Y Qu, J J Liang, and P N Suganthan. 2012. Niching particle swarm optimization with local search for multi-modal optimization. *Information Sciences* 197 (2012), 131–143. https://doi.org/10.1016/j.ins.2012.02.011

[33] Marko Robnik-Šikonja and Marko Bohanec. 2018. *Perturbation-Based Explanations of Prediction Models*. Springer International Publishing, Cham, 159–175. https://doi.org/10.1007/978-3-319-90403-0_9

[34] Lesia Semenova and Cynthia Rudin. 2019. A study in Rashomon curves and volumes: A new perspective on generalization and model simplicity in machine learning. *CoRR* abs/1908.0 (2019). http://arxiv.org/abs/1908.01755

[35] Uday Shankar Shanthamallu, Andreas Spanias, Cihan Tepedelenlioglu, and Mike Stanley. 2017. A brief survey of machine learning methods and their sensor and IoT applications. In *2017 8th International Conference on Information, Intelligence, Systems Applications (IISA)*. 1–8. https://doi.org/10.1109/IISA.2017.8316459

[36] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. 2019. CERTIFAI: Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models. *AIES 2020 - Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (may 2019), 166–172. https://doi.org/10.1145/3375627.3375812 arXiv:1905.07857

[37] Frans Van Den Bergh and A P Engelbrecht. 2002. *An Analysis of Particle Swarm Optimizers*. Ph. D. Dissertation. University of Pretoria, ZAF.

[38] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. 2013. OpenML: networked science in machine learning. *SIGKDD Explorations* 15, 2 (2013), 49–60. https://doi.org/10.1145/2641190.2641198

[39] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *SSRN Electronic Journal* (nov 2017). https://doi.org/10.2139/ssrn.3063289 arXiv:1711.00399

[40] Janett Walters-Williams and Yan Li. 2008. Comparative Study of Distance Functions for Nearest Neighbors. In *Advanced Techniques in Computing Sciences and Software Engineering, Volume II of the proceedings of the 2008 International Conference on Systems, Computing Sciences and Software Engineering (SCSS), part of the International Joint Conferences on Computer*, Khaled M Elleithy (Ed.). Springer, 79–84. https://doi.org/10.1007/978-90-481-3660-5_14

[41] Adam White and Artur S d'Avila Garcez. 2020. Measurable Counterfactual Local Explanations for Any Classifier. In *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020) (Frontiers in Artificial Intelligence and Applications, Vol. 325)*. IOS Press, 2529–2535. https://doi.org/10.3233/FAIA200387