IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 February 2024, accepted 14 May 2024, date of publication 22 May 2024, date of current version 3 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3404043

RESEARCH ARTICLE

Counterfactual Explanation of AI Models Using an Adaptive Genetic Algorithm With Embedded Feature Weights

EBTISAM ALJALAUD[1,2] **AND MANAR HOSNY**[2]

[1]Computer Science Department, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11564, Saudi Arabia
[2]Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia

Corresponding author: Ebtisam AlJalaud (e.aljalaud@gmail.com)

ABSTRACT Explainable Artificial Intelligence (XAI) is a cutting-edge AI development motivated by the need for transparency of black-box models in AI systems. This transparency enhances user trust, facilitates accountability, and enables a better understanding of AI systems decisions, especially in critical applications where insights into decision processes are essential. These benefits have increased XAI research interest, aiming to provide techniques for interpreting and understanding the behavior of intelligent models. Counterfactual explanation is a popular technique for model interpretation based on updating a few features such that the outcome of an AI model is changed. Users can gain insights into the critical features or factors influencing the AI system's decision by analyzing these counterfactuals. However, most counterfactual techniques require more qualifications, such as simplicity, robustness, and coherence. In this research, we propose a novel approach, Adaptive Feature Weight Genetic Explanation (AFWGE), for generating counterfactual explanations of AI models, where a custom genetic algorithm (GA) is employed, incorporating adaptive feature weights to enhance the algorithm's performance. Experimental results on four benchmark datasets show that AFWGE allows for the adaptation of feature weights during the evolutionary process, producing more effective counterfactual explanations with superior proximity, sparsity, plausibility, and actionability. Furthermore, it emphasizes feature weights as reliable indicators of the significance of the model's features, providing valuable insights for interpreting the model. AFWGE not only advances the field of counterfactual explanation generation but also establishes a robust framework for assessing feature importance in machine learning models.

INDEX TERMS Explainable artificial intelligence, counterfactual explanation, genetic algorithm, machine learning, artificial neural network.

I. INTRODUCTION

The widespread adoption of artificial intelligence (AI) systems has hugely impacted human lives and society. Most recent powerful AI models are based on 'black box' machine learning (ML) approaches, meaning the rationale underlying their decision-making mechanism is challenging to understand and interpret [1]. This ambiguity makes these systems difficult for end-users to trust. At the same time, the process of inferring a classification model from examples cannot be controlled step by step because the size of the training data and the complexity of the learned model are usually too vast for humans [2]. These challenges have spurred research interest in explainable AI (XAI). This research field seeks to enable end-users to understand, trust, and effectively manage their intelligent models by providing techniques for interpreting and understanding the model's behavior [3].

Some characteristics of good explanations include shortness/simplicity, robustness, and coherence. One popular explanation technique involves updating features to identify how the decisions change, called counterfactual explanation [4], [5]. Counterfactual explanations can provide simple

The associate editor coordinating the review of this manuscript and approving it for publication was Nikhil Padhi.

explanations that reflect human cognition, eliciting spontaneous causal thinking about what might have been the case regarding a particular model's output. A good counterfactual explanation should be sparse; they need to adjust the fewest or most important features [6], [7]. However, to the best of our knowledge, none of the current counterfactual explanation studies has considered these criteria in the explanation generation itself.

Motivated by the limitations of existing techniques in ML model explainability, this research aims to develop a model that uses a custom genetic algorithm (GA) to generate accurate counterfactual explanations. This model takes into account feature characteristics such as correlations, importance, and weights. To assess the importance of each feature, an adaptive GA method, inspired by [8], is proposed as an enhancement of the custom GA. The model extends feature weights as part of the solution (counterfactual), aiming to adapt them automatically during the evolutionary process. The final optimized solution, thus, will embed feature importance for analysis and consideration by the decision maker, thereby enhancing the performance and explainability of the AI model. By incorporating adaptive feature weights, our GA significantly improves the productivity of the algorithm, leading to the generation of more effective counterfactual explanations. This not only enhances the generated counterfactual explanations but also provides a robust framework for assessing the significance of features in machine learning models, thereby aiding their interpretations.

The remainder of this paper is structured as follows: Section II describes the problem formulation, while Section III overviews some related work. Section IV explains our proposed method. Then, Section V presents the detailed experimental results. Finally, Section VI concludes this study with a summary and future research directions.

II. PROBLEM FORMULATION

In this work, analyzing explainability of a model builds upon and enhances the work in [4] named CERTIFAI (Counterfactual Explanations for the Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence). Our proposed model is based on a model-agnostic custom GA that aims to solve the following problem:

Given a black-box classifier f and an input instance x. Let the counterfactual be a feasible generated point c. Then the problem can be formulated as:

$$min\ d(x, c) \quad s.t\ f(c) \neq f(x) \tag{1}$$

The objective is to solve (1) where $d(x,c)$ is the distance between x and c. Each individual c of the GA population that has a different prediction from x is a candidate counterfactual. The goal is to find the fittest possible c^* to x using the following definition:

$$fitness = 1/d(x, c) \tag{2}$$

The possible set of individuals $c \in I$ are defined by:

$$I = W \setminus P \quad s.t\ P = \{p | f(p) = f(x), p \in W\} \tag{3}$$

where W represent the space from which individuals can be generated and P is the set of points with the same prediction as x. If a user wants the counterfactual c to belong to a particular class j, we define I as:

$$I = (W \setminus P) \cap Q \ \ s.t\ Q = \{q | f(q) = j, q \in W\} \tag{4}$$

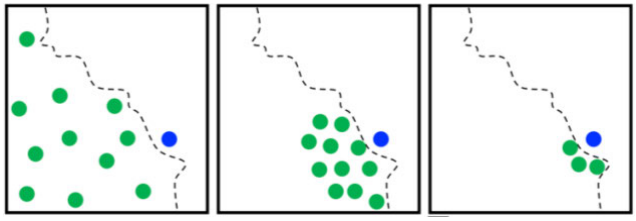


FIGURE 1. The proposed model counterfactual generation process [4].

The proposed model relies on counterfactuals generated by the GA. This generation process is based on fitness measurement, where higher fitness (i.e., shorter distance) is preferred (2). For example, Fig.1 shows the decision boundary for a binary classifier with the input instance x (in blue points). In (a), the model samples a set of points in the feature space with the condition that they should be on the other side of the decision boundary in (green points). Then, the GA in (b), evolves these samples to generate individuals c that are closer to the input point, but lie on the other side of the decision boundary. Finally, in (c), a closer and smaller set of counterfactuals, c^*, whose size is user-defined, is generated which form the fittest possible counterfactuals.

III. RELATED WORK

In the last few years, a thriving field of study has been devoted to the interpretability of machine learning models. There are two primary techniques for ostensibly interpretable models. One approach is to develop basic, memorized models by storing the input samples explicitly or by identifying the concept behind the input data and memorizing their general rules, such as decision trees and scoring systems [9], [10]. Scoring systems are linear classification models that only require users to do simple addition, subtraction, and multiplication operations on a few small integers to predict without focusing on machine learning methods to learn from data. Although these models are widely used in medicine, using them to learn from data is challenging because they must be precise and sparse, contain co-prime integer coefficients, and follow several operational restrictions [9].

For that, a more efficient approach, explainability, was introduced, offering post-hoc explanations for possibly sophisticated black-box models. These methods seek to explain how a fixed model leads to a particular prediction, either by fitting simpler, local approximations of a model around a particular decision or data points, which is called local approximation, or by perturbing variables to measure how the prediction changes, which is called counterfactual

explanation [11]. Numerous research threads are devoted to post-hoc explanations that examine methods to explain individual predictions. In the following subsections, we review some local approximation methods, followed by counterfactual ones.

A. LOCAL APPROXIMATION

The basic concept of local approximation is to focus on a single instance and attempt to comprehend how the model reached its prediction. This could be accomplished using a simpler interpretable model to approximate a specific region of interest in a black box model. It is described as local because the prediction may depend linearly on some features rather than having a complex dependence on them [12].

In [13], they provide explanations for individual predictions as a solution to ''trusting a prediction'' and ''trusting the model'' problems, where if the users do not trust a model or a prediction, they will not use it. These problems directly relate to how a human understands a model's behavior rather than viewing it as a black box. They proposed a method called Local Interpretable Model-agnostic Explanations (LIME), which fits a sparse linear model to approximate non-linear models locally. They tested simulated users and human participants using two sentiment analysis datasets (books and DVDs, 2000 occurrences each), where the task was to classify product reviews as positive or negative. Their research indicated that explanations benefit a range of models in trust-related tasks in the text and image areas, with expert and non-expert users choosing between models, rating trust, improving untrustworthy models, and gaining insights into forecasts.

In [2], they solved the challenge of explaining the algorithm's decision-making outcome by offering meaningful explanations when automated decision-making occurs by presenting Local Rule-Based Explanations (LORE). LORE is an agnostic technique that uses GAs to create interpretable and truthful explanations. They enhanced the LIME technique in [13] by matching a decision-tree classifier to approximate the nonlinear model and then tracing the decision-tree routes to create explanations with rules for which input attributes might differ and result in various outcomes. To assess the mimicry performance of the decision tree inferred by LORE, they compared it to LIME and Anchors [14] using three real-world tabular datasets: Adult and German[1] datasets from the UCI Machine Learning Repository and Compas dataset [48]. They used the following predictors as black boxes: support vector machines (SVM), random forests with 100 trees (RF), and neural networks (NN). The findings demonstrated the efficacy of the genetic neighborhood technique, which enables LORE to outperform other methods.

[1]D.DuaandC.Graff.(2017).UCIMachineLearningRepository. http://archive.ics.uci.edu/ml.

B. COUNTERFACTUAL EXPLANATION

In counterfactual explanations, actionable feedback is obtained by producing counterfactuals that explain alternative scenarios. For example, a profile with changes to a candidate's salary or skills is a ''counterfactual'' to the original profile. The counterfactual explanation describes the change to an input data point that would change a model's prediction for that point.

The counterfactual explanation was introduced by [15] as a novel way for explaining automated decisions. It addresses various issues raised by previous research on algorithmic interpretability and accountability. Recently, new regulations were enacted to ensure the verifiability, accountability, and, most crucially, complete transparency of algorithmic decisions. A major example is the new European General Data Protection Regulation (GDPR and ISO/IEC 27001) [16], which became effective in May 2018 and grants data subjects the right to an explanation of algorithmic decisions [17]. However, in [15] it was demonstrated that the GDPR needs more support to achieve the stated objectives. They proposed three objectives for explanations to assist data subjects: (1) to inform and assist the subject in comprehending why a particular decision was made, (2) to provide foundation for contesting adverse decisions, and (3) to comprehend what could be modified in the future to achieve the desired outcome, using the current decision-making model. They introduced the concept of unconditional counterfactual explanations as a novel approach to explaining automated decisions that solve several difficulties encountered in existing research on algorithmic interpretability and accountability. They used the LSAT [18] and Diabetes [19] datasets to demonstrate their approach to the problem of law school admissions and risk variables that enhance a patient's likelihood of developing diabetes. They concluded that unconditional counterfactual explanations could find a middle ground between data subjects' and controllers' interests, which would otherwise function as an obstacle to a legally obligatory right to explanation.

Producing counterfactuals does not imply that changes have to be actionable or feasible. An actionable change does not change immutable features nor mutable features in an infeasible way (e.g., has_phd from true to false). The mutable features are changeable attributes that can change over time, such as tastes, preference for a specific genre of music, support for a particular football team, or style of dress. The immutable features represent those attributes that cannot change, such as race or gender [20]. So, if the employment system rejects a person, he would like to know how he can get the job or what could be changed in his application to get the job. In such a case, actionable changes are mandatory. To this end, a person's ability to change a model's decision by altering feasible input variables was proposed by [21]. They used an integer programming routine that produces two methods. The first method evaluates the counterfactual feasibility and difficulty of a linear classifier over its population.

In contrast, the second one generates a list called a *flipset* of the actionable changes for the person. They used these methods in credit scoring problems with three experiments, each using a different real dataset: Credit dataset [22], GiveMeCredit,[2] and German dataset. Their experimental study aims to evaluate the disparity in the counterfactual explanation of a classifier and to demonstrate how common practices in the development and deployment of machine learning models affect actionable changes, such as feature selection and parameter tuning. They found variance in the complexity of gaining the provided counterfactual state across genders, demonstrating a significant difference in the cost of counterfactual generation.

For linear models, a recent paper by Russell [23] mainly focuses on the problem of explaining financial decisions where the classifier is linear. They used integer programming to construct an efficient technique for discovering varied counterfactuals. They validated it by creating varied counterfactuals for mixed data (FICO dataset) on various of situations, where all produced explanations are human-readable text that demonstrates the minimal adjustments required. Additionally, when using the LSAT dataset [18] to examine inequality, the variety of explanations presented highlights the racial bias.

Evaluating models' explainability, accountability, and fairness is usually done separately using multiple tools. However, in [4], analyzing the robustness, fairness, and explainability of a classifier was done in one single framework named CERTIFAI (Counterfactual Explanations for the Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence). CERTIFAI uses a custom GA, which is flexible, model-agnostic, and does not need access to model internals. It relies on counterfactuals generated by the GA given a black-box classifier and an input instance. CERTIFAI uses the generated counterfactuals to analyze models' robustness, fairness, and explainability. In the experiments, the CERTIFAI framework was tested to evaluate the robustness of classic classifiers (decision trees (DT), SVM, and multilayer perceptron (MLP)) using three data sets (Diabetes, Breast Cancer, and Iris). On the other hand, explainability and fairness were evaluated using the UCI Adult dataset. The results demonstrate the flexibility of the GA in providing plausible counterfactual explanations and how to use them to comprehend critical features, as well as how robustness and fairness can be quantified using fitness values obtained during the counterfactual generation process.

The explainability of a model usually comes as a response to people's concerns. People may feel that they do not understand a model, what information the model relies on, and how this information is being used. Also, people may be worried that models might behave in unfair ways. Most works, as shown above, focus primarily on explainability and fairness (except for CERTIFAI [4], which adds robustness to these concerns). However, some applications focus only on robustness, particularly within neural networks. For example, the explainability of computer vision models poses an especially compelling challenge since extremely tiny modifications to the input image can easily trick a neural network, even when the benign case is correctly classified and the shift is unnoticeable to the human eye. Apart from the obvious security implications, such incidents reveal that our existing models cannot robustly learn the fundamental concepts. Consequently, an important question arises: *How can we train deep neural networks (DNNs) that are robust to adversarial inputs?*

To tackle this challenge, a defensive distillation approach was proposed by [24]. *Distillation* is a training method originally developed to train a DNN with knowledge from another DNN. Instead of transferring knowledge between different architectures, defensive distillation employs the knowledge acquired from a DNN to improve its resilience to adversarial samples. It is used to train an arbitrary DNN and increase its robustness, reducing the success rate of the current attacks' ability to find adversarial examples.

In [25], they presented GeCo, which relies on a customized GA to favor searching counterfactual explanations with the smallest number of changes (plausible counterfactual). The primary performance limitation in GeCo is the repeated calls to selection and mutation operations, so two optimization procedures were introduced. They introduced a lossless, compressed data representation of candidate counterfactuals (generated population during the genetic algorithm) to optimize mutation. Also, to optimize selection, they used a partial evaluation technique to optimize the evaluation of the classifier. In the experiments, they compared GeCo against five other systems: MACE [26], DiCE [27], WIT (Google's What-if Tool) [28], CERTIFAI [4], and SimCF [29]. They used four real datasets: Credit [22] and Adult (from the UCI repository), Allstate,[3] and Yelp.[4] GeCo finds a valid counterfactual explanation close to the distance of the optimal explanation in linear runtime.

In [30], they introduce the DisCERN algorithm, a case-based counterfactual explanation generator. The counterfactuals are generated by changing feature values from the nearest unlike neighbor (NUN) until an actionable change is found. The DisCERN algorithm utilizes explainers based on feature relevance, such as LIME and SHAP, to determine the minimum feature changes required to provide a counterfactual explanation from the returned NUN. The DisCERN algorithm is evaluated on five datasets, compared to the commonly used counterfactual approach DiCE [27]. The performance of both algorithms is measured using the number and amount of changes made to the features. The results indicate that DisCERN has either surpassed or attained performance similar to DiCE.

[2]Kaggle Give Me Some Credit. http://www.kaggle.com/c/GiveMeSomeCredit/.

[3]Allstate. 2011. Allstate Claim Prediction Challenge. https://www.kaggle.com/c/ClaimPredictionChallenge.

[4]Yelp. 2017. Yelp Dataset Challenge. https://www.yelp.com/dataset/challenge/.

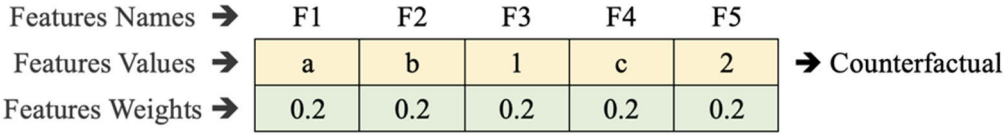


FIGURE 2. Initial solution representation example.

In [31], they propose a novel optimization formulation that generates sparse counterfactual explanations via another custom genetic algorithm to explain the black-box model's predictions. They provide a novel formulation of the optimization problem that leads to a single sparse counterfactual explanation using GA, where the normalization of continuous features and selection of predictive features achieves a computationally efficient generation of counterfactuals. They evaluated the efficacy of the proposed method on two credit scoring datasets -German and Home Loan Equity (HMEQ) - by comparing the generated counterfactual explanations with explanations from credit scoring experts. The experimental results indicate that the proposed approach efficiently generates sparse counterfactuals compared to a similar method, CERTIFAI [4].

None of the proposed explanation techniques provides counterfactuals that include the importance of features or groups of features in the explanation. They may be able to predict the importance of some features by analyzing the results produced by the model, for example, by noticing which features change more frequently in the produced counterfactuals [4]. However, none of them produce counterfactual explanations that include weights for features to indicate their importance. Embedding feature weights in the counterfactual explanation itself is helpful for individuals, model developers, and regulators to understand the model's behavior and take more informed decisions that can help resolve the current situation. For example, this is done by focusing on the features (or groups of features) that are more important in reversing the decision and producing the desired outcome. Our proposed method attempts to bridge this gap by embedding feature weights as part of the counterfactual solution to optimize feature values and weights alongside each other during the evolutionary process, as explained next.

IV. PROPOSED ALGORITHM

Natural genetics served as the foundation for the widely used optimization methods known as evolutionary algorithms, where GA (Genetic Algorithm) is the most popular variant. To find a solution that is almost optimal, GA produces a set of solutions through selection and merging. A population is a collection of chromosomes, where each chromosome represents an individual problem solution. In the first stage, a population is generated through a random process, followed by the assignment of a fitness score to each individual chromosome based on a predetermined fitness function. The following stages involve the selection of the most promising individuals, based on their fitness, to undergo genetic operations, such as crossover and mutation, to generate a new population. These operations are iteratively performed until a predetermined number of generations is attained or a termination condition is met.

It has been found in the literature that the search performance of evolutionary algorithms is usually improved with adaptive strategies or dynamic control parameters such as adaptive GA [32]. Thus, our proposed GA algorithm: Adaptive Feature Weight Genetic Explanation (AFWGE) makes use of features weights, which are experimentally determined. These weights take specific values in the algorithm, where they are equally initialized and then evolved adaptatively during the search. Consequently, search accuracy, exploration, exploitation, convergence speed, and overall algorithm behavior are guided by these adaptive weights. In fact, these weights have a significant impact on the behavior of the method since they are also optimized together with the main optimization of the counterfactual solution. The general framework of AFWGE algorithm is shown in Algorithm 1 and the flowchart in Fig.3, where the general structure of CERTIFAI [4] is followed, but with key modifications which are pointed out in the following subsections that explain in detail the behavior of AFWGE.

A. POPULATION GENERATION

For a given instance x the counterfactual explanation c^* is generated using a GA. Like CERTIFAI, the proposed algorithm starts with an initial population N of randomly generated k chromosomes, where k is the population size, and each chromosome is an individual solution. As shown in Fig.2 a solution in AFWGE is represented as a chromosome composed of two main parts: counterfactual and weights. The counterfactual part is the list of n features F_x with their values (top row of the chromosome in Fig.2), while the second part (lower row of the chromosome) is the features weights. In AFWGE, each feature is assigned an initial weight $w=1/n$ where the sum of all the feature weights $w_0 + w_1 + \ldots + w_n$ is equal to 1. Both features' values and their weights are the evolved elements of the solution during the processing of the AFWGE algorithm. In the features values part of the chromosome, a feature is either categorial or numerical feature. A categorial feature value indicates one category of a categories' group and is represented in a textual format. On the other hand, a numerical feature value, as the name implies, is represented in the form of a number. In Fig.2, the features F1, F2, and F4 are categorical features, while F3, and F5 are numerical ones. To generate realistic chromosomes, the features values are generated based on the values found

in all data instances (line 8 in Algorithm 1). For categorial features, the value is generated by choosing one of the feature categories randomly over all instances, whole for numerical features, the value is generated randomly between the minimum and the maximum values of the feature over all instances. However, this feature values generation procedure follows the constraints that consider mutable and immutable features. Features values generated for mutable features can be changed, such as *body weight, blood pressure, salary,* and *working hours*. On the other hand, there are no values generated for immutable features, such as sex, race, and age. In other words, immutable features have fixed values all over the generated chromosomes since they can't be changed. Moreover, some feature values can be generated following partial constraints, where a feature can be changed according to a certain limitation. For example, age can increase but not decrease (i.e., obviously, people can get older, but none can get younger).

After the population N, consisting of the k chromosomes, has been generated, the chromosomes are evaluated based on their objective function score (1), where the chromosomes with minimum distance are the best, and the ones with the maximum distance are the worst (line 9 in Algorithm 1). Then, selection *Select* as refining process is performed, where a predetermined number Z of the best chromosomes are selected to undergo crossover, followed by mutation (line 10 in Algorithm 1).

B. CROSSOVER

In crossover (line 11 in Algorithm 1), some of the chromosomes Z' are selected from Z using binomial randomization with crossover probability P_c and a random list of features Y are selected. Then, from Z', multiple couples of chromosomes are randomly selected and crossed by exchanging their Y features values. Fig.4 shows a crossover example where chromosomes A and B are selected to be crossed-over by exchanging a list of Y features values, which are the second and fifth feature. The new produced chromosomes are the children called offsprings O. Contrary to CERTIFAI, where offsprings directly replace their parents in the population, in AFWGE, if the offsprings have higher fitness -closer to the instance x- than the parents, they replace their parents in the population as new chromosomes; thus, the population N is updated forming N', otherwise these offsprings are discarded and the parents are retained. We also note that only feature values are involved in the crossover operator while feature weights remain intact in the chromosome during crossover. Feature weights are evolved using mutation as will be explained next.

C. MUTATION

After crossover, the new chromosomes O are mutated by updating both feature values and weights (line 12 in Algorithm 1). Feature values are mutated by updating a few numbers of features, which are selected by binomial randomization with a certain mutation probability.

Algorithm 1 AFWGE

```
1: procedure findCounterfactuals(Model, Dataset, Select,
                  k, Generations, Constraints, q, pc, pm)
2:   counterfactualsList ← empty list
3:   for x ∈ Dataset do // x is one instance of Dataset
4:     bestFitness ← large number
5:     terminationCounter ←0
6:     populationList ← empty list
7:     for i ← 0 to Generations do
8:       N ← generate(x, k, populationList, Constraints)
9:       populationList ← evaluate(N)
10:        Z ← select(N, Select)
11:        N′, O ← crossover(N, x, Z, pc)
12:        N″ ← mutate(N′, x, O, pm)
13:        N‴ ← filter(Model, N″, x)
14:        populationList ← evaluate(N‴)
15:        if (populationList = k)
            and (i ≠ Generations) then
16:          j ← randomInteger(k/2, k-2)
17:          populationList ← replace(populationList, j)
18:        end if
19:        fitness ← getFitness(populationList[0])
            //best counterfactual fitness at index[0]
20:        if fitness < bestFitness then
21:          bestFitness ←fitness
22:          terminationCounter ← 0
            //reset terminationCounter countdown
23:        else
24:          terminationCounter ← terminationCounter+1
25:        end if
26:        if terminationCounter = 5 then
27:          break
28:        end if
29:      end for
30:      counterfactualsList ← getCF(populationList, q)
          //get best q counterfactuals
31:    end for
32:    return counterfactualsList
33: end procedure
```

For each chromosome in O, following the approach in CERTIFAI, a categorial feature is mutated by copying the same feature value from another randomly selected chromosome in the N' population, while numerical features are mutated by the taking the average of the same feature values of two randomly selected chromosomes from N'. For example, in Fig.5, chromosome A is selected for mutation, and the first and third features are selected to be mutated. Since the first feature value a is a categorial feature, it is mutated by copying the feature value from any other randomly selected chromosome. In this example, B is selected which has the value d for the first feature. Thus, the first feature of chromosome A is updated from a to d by copying the value of chromosome B's first feature. On the other hand, the third feature, which is numerical, is mutated using the average

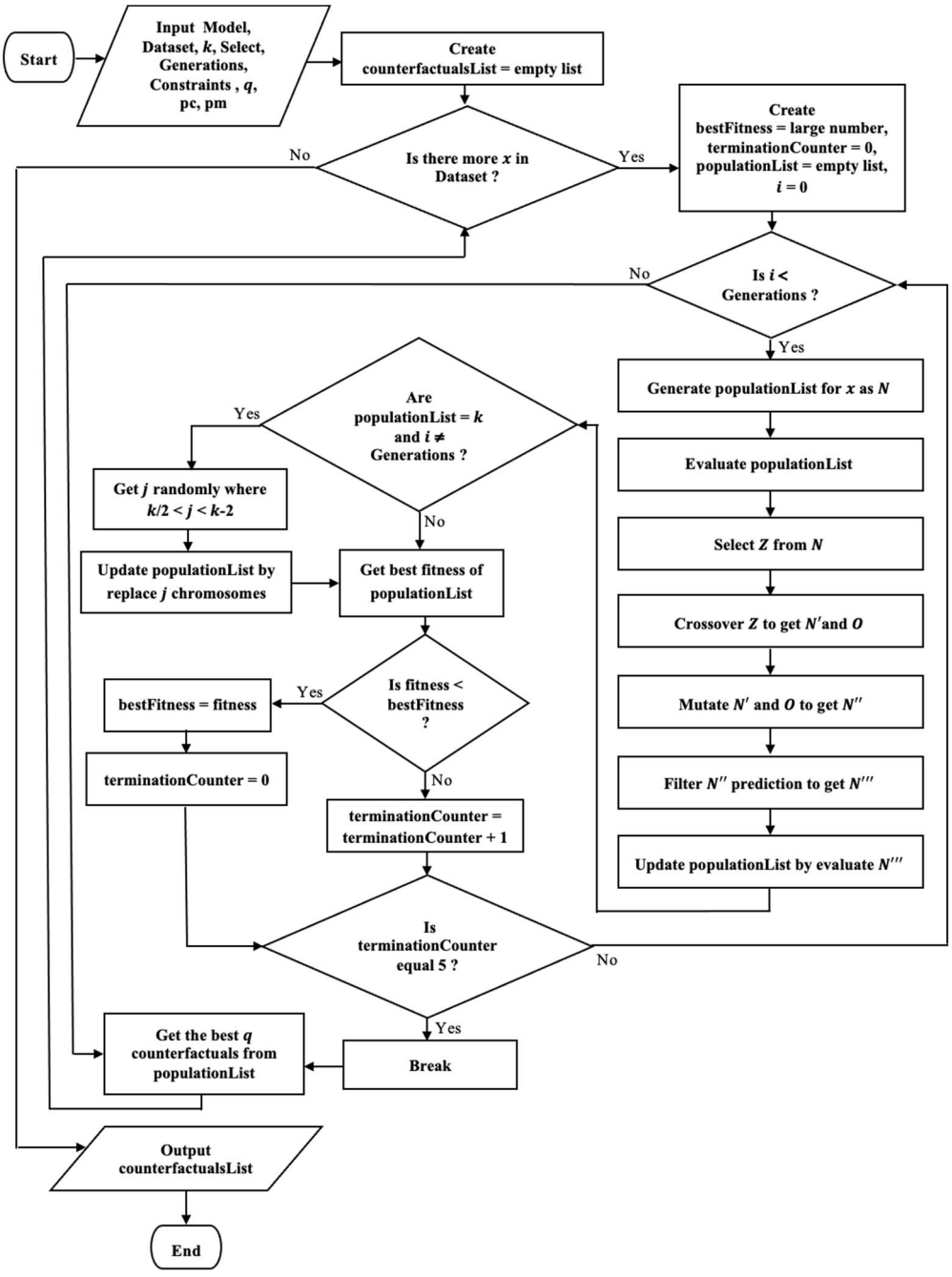


FIGURE 3. Flowchart of AFWGE algorithm to generate counterfactual explanations.

value of the third feature in chromosomes B and C that are selected randomly.

Additionally, in AFWGE, feature weights are mutated by updating two different randomly selected feature weights for chromosomes O. One feature weight will be incremented while the other is decremented with the same amount to preserve the features weights summation adding up to 1. In Fig.5, the second and fourth feature weights are selected for mutation. The second feature weight 0.2 is incremented by 0.1 (which is half of the initial weight 0.2) and becomes 0.3, while the fourth feature weight 0.2 is decremented by 0.1 and becomes 0.1. The amount of change is fixed to half of the initial weight $w=1/n$, as this value was found to provide reasonable impact on the chromosome's features' roles. 50% of the initial weight is considered a median value that is neither too small to be useless nor too large to potentially cause domination or vanishing problems for the feature's role. The dominating problem occurs when a chromosome's feature weight becomes the largest, with significant differences compared to other weights. Conversely,

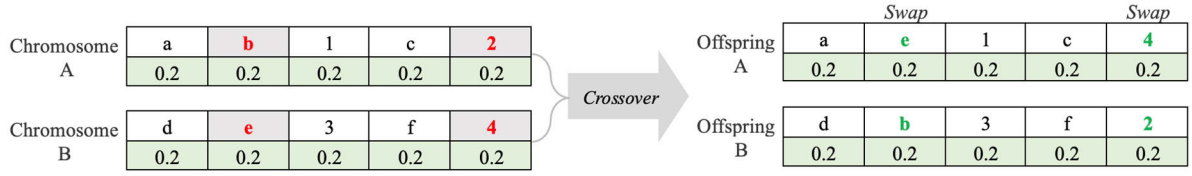


FIGURE 4. Crossover example.

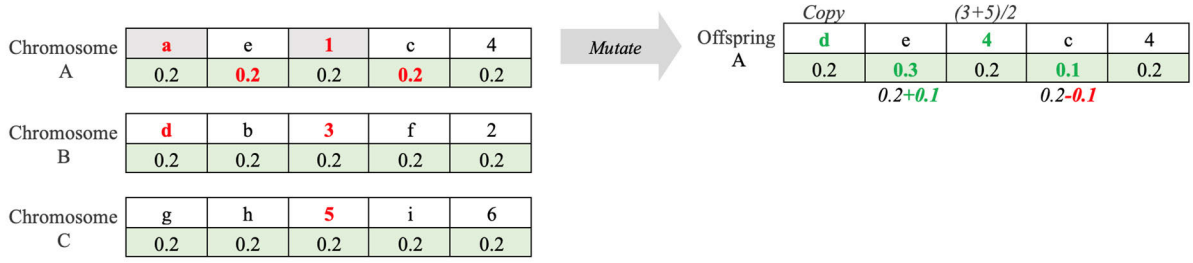


FIGURE 5. Mutation example.

the vanishing problem may occur when a chromosome's feature weight becomes the lowest, again with a significant difference from other weights. In fact, in case the same feature of the same chromosome has been accidentally incremented or decremented multiple times, a dominating/vanishing problem may occur. For that, after each weight mutation, the incremented weight should not exceed the maximum weight, which is set to double the initial weight, and not less than the minimum weight, which is set to the half the initial weight. As done in crossover (Section IV-B), if the mutated chromosomes have higher fitness (i.e., closer to the original instance x), their features' new weights are set and the population is updated to N'', otherwise the weights before mutations are restored.

D. FILTRATION

The mutated population N'' is then filtered by checking its chromosomes' predictions, using the same black box model used to generate the original predictions. Since the intention is to obtain counterfactual explanations having a different outcome than the original instance, the chromosomes that produce the same prediction as the original instance x are removed. Thus, the population N'' is updated to N''' by keeping only the chromosomes that provide a different prediction than the prediction of the instance x (line 13 in Algorithm 1).

E. RE-EVALUATION

Then, for the population N''' the chromosomes are re-evaluated and sorted ascendingly based on their objective function score in (1), (line 14 in Algorithm 1). If the maximum number of generations is not yet reached, nor the termination condition occurred, all the steps are repeated, starting with the generating stage. The population thus keeps evolving from one generation to the next, until the termination condition is satisfied, which is when the best fitness value remains unchanged for five consecutive generations.

F. REPLACEMENT

Contrary to CERTIFAI that appends the population over generations, with each generation's round beginning with a completely random population, AFWGE supplements the filtering process. After the population undergoes filtration, resulting in a decrease in the number of chromosomes from the original population size k, AFWGE replenishes the population list in the next round by producing enough chromosomes to reach the required population size k. However, generation after another, the population list may become full of k chromosomes that provide different predictions from the original instance prediction. In such a case, and if the termination condition has not yet been met, it will not be possible to proceed to the next round with a full population list. Additionally, there will be no benefit from the generation stage as there is no available space for any new chromosomes. For that, at the end of each round, the population list size is checked for being full. If it is full, a number j of fittest chromosomes is kept in the population and the rest are discarded. The j number is determined randomly to be a number between $k/2$ and k-2. This allows to update the population list over generations with different unknown number k-j of new randomly generated chromosomes. In other words, there is a chance to exchange only two chromosomes up to half of the best of the population list with new randomly generated chromosomes each generation. This approach is contrary to a steady convergence in the objective space, which may result in the elimination of converged sub-optimal solutions that may be helpful to improve the diversity of the population and avoid premature convergence [33]. To overcome this drawback, AFWGE attempts to obtain a balance between convergence and diversity over the search space.

In other words, keeping the chromosomes with best fitness increases the convergence speed to the best solutions, while adding new randomly generated ones increases the diversity of the solutions to reach better local optima. Moreover, the random changing ratio between the best and random chromosomes dynamically control the diversity-convergence balance (line 17 in Algorithm 1).

G. FITNESS

Each individual chromosome belonging to the population N is a candidate counterfactual c for instance x such that $f(c) \neq f(x)$. The goal is to find the fittest possible counterfactual c^* of the instance x. The fitness for an individual chromosome is defined as in (2). For distance calculation, the Manhattan Distance is used for numerical features, and a simple matching distance is used for categorical features. Both types of distances, numerical and categorical, are multiplied by their corresponding feature weights. For that, min-max normalization (usually called feature scaling) performs a linear transformation on the original data. This technique converts all the scaled data to the range 0 to 1 [34]. Normalization requires access to the training data. In our model development and experiments, the training data is accessible, so normalization is possible. The distance metric used where m is the list of numerical features, and u is the list of categorial ones is defined by (5):

$$d(x, c) = \sum_{i=1}^{m} [(|x_i - c_i|) \times w_i] + \sum_{i=1}^{u} [Match\,(x_i, c_i) \times w_i] \quad (5)$$

where for a categorical feature a, a simple matching distance is defined by (6):

$$Match\,(x_a, c_a) = \begin{cases} 0 & if\ x_a = c_a \\ 1 & if\ x_a \neq c_a \end{cases} \quad (6)$$

V. COMPUTATIONAL EXPERIMENTATION

The experiments aim to apply AFWGE and compare it with the state-of-the-art method in [4] called CERTIFAI. As previously motioned, the main difference of AFWGE lies in incorporating feature weights as part of the chromosome and dynamically evolving them alongside the feature values throughout the evolutionary process. AFWGE is experimentally tested to determine its effectiveness in adapting feature weights directly into the counterfactual explanation generation. Thus, the objective is to assess its utility for individuals, model developers, and regulators in enhancing their understanding of the model's behavior. In the following subsections we explain the details of the experimental setup and the implementation environment used to test our approach, followed by the detailed results obtained.

A. EXPERIMENTAL SETUP

1) DATASET

We considered the same four real datasets used in CERTIFAI [4]: (1) The Adult dataset [35], used to predict whether the income of an adult exceeds \$50K/year using US census data from 1994; (2) The Diagnostic Wisconsin Breast Cancer dataset [36] to diagnose whether a breast tumor is malignant or benign; (3) Pima Indians Diabetes [37] to predict whether or not a patient has diabetes, based on certain diagnostic measurements, where all patients are females at least 21 years old of Pima Indian heritage, and (4) The Iris dataset [38] to predict the type of the Iris plant from 3 types. Table. 1 provides the considered datasets details for the experiments.

In each experiment, all the dataset's instances, which is the dataset size, was fully used as a sample input in the experiment, with the exception of the Adult dataset. For this dataset, a sample was chosen randomly (5000 & 100 instances) because of the huge time needed to obtain the counterfactuals for the whole dataset consisting of 48842 instances. The 5000 Adult instances needed almost 17 to 21 days, so to run multiple tests, only 100 instances were chosen randomly and considered as the sample size.

2) RIVAL ALGORITHIMS

We benchmarked AFWGE against CERTIFAI reported in [4], where we reimplemented CERTIFAI using its publicly available source code[5] and used the same experimental parameters for both for the sake of a fair comparison.

TABLE 1. Characteristics of the datasets.

		Adult	Breast Cancer	Pima Indians Diabetes	Iris
Dataset Size		48842	569	768	150
Sample Size		100	569	768	150
Number of Features		14	30	8	4
Outcome Classes		2	2	2	3
Features Constraints	Only increase	Age	-	Age	-
	Cannot change	Race, Sex	-	-	-

3) EVALUATION METRICS

We used the following five metrics to evaluate the quality of the produced counterfactual explanation:

1. Proximity: The distance between counterfactual c and the original instance x. The higher the counterfactual explanation fitness, the closer it should be to the original instance x with respect to feature values [39]. We computed the distances in (5) based on the features including their weights. On the other hand, CERTIFAI is a non-adaptive genetic algorithm, so the features are assumed to have equal weights to ensure a fair distance comparison. For CERTIFAI, these weights have no role in the algorithm's behavior nor its decisions.

2. Sparsity: The number of features changes in a counterfactual c. An efficient counterfactual explanation should

[5]https://github.com/Ighina/CERTIFAI

minimize the number of changed features to enhance its understanding and effectiveness [40].

3. Number of objective function evaluations: this measure is used as an indication of the computational cost of the algorithm, as the objective function is the most time-consuming part of the algorithm.

4. Plausibility: A counterfactual c should come from a possible world. This implies that the feature values of the counterfactual should not exceed or fall below those that are observed in the data [39]. Additionally, the counterfactual should not be identified as an outlier to the instances in the data. The plausibility of an explanation is crucial for building trust. It is difficult to trust a counterfactual of an unrealistic feature combination that is incompatible with existing examples in the data. In contrast, a plausible counterfactual becomes "realistic" when it is close to the known dataset and follows the observed correlations between the features [39]. For example, if a counterfactual suggests increasing the education level to a Master's degree to obtain a different desired outcome, then the age of the person also needs to change as it is linked with the educational level over the data.

5. Processing time: The average CPU time it takes to generate an explanation for single instance x. To compare runtimes, both AFWGE and CERTIFAI return multiple counterfactuals for each instance. We only consider the highest fitness counterfactual in this evaluation.

4) THE CLASSIFICATION MODEL

We evaluated the performance of the algorithms on a multi-layered perceptron (MLP). For the comparison with existing methods, we should use the same classifier to guarantee a fair comparison. Thus, we considered the same classifier proposed by CERTIFAI [4], which is a four layers neural network with an input layer, two hidden layers of 20 neurons each, and an output layer with ReLU activation, trained on the four datasets listed above with an 80-20 training-testing split.

5) SETUP

We implemented AFWGE and CERTIFAI in Python 3.10. All experiments were run on MAC studio M1 Ultra with 128 GB RAM. We used the default hyperparameters as recommended in [2] and [41]. The parameters of the genetic procedure, namely probabilities of crossover and mutation, number of generations, and population size were set with the default values of 0.7, 0.2, 10 and 1000, respectively. The number of generations was set to 10 after multiple tests with different numbers of generations equal to 10, 15, and 20. The termination condition was set to no improvement for 5 consecutive iterations; however, it was noted that the fitness does not improve beyond the ninth generation in both AFWGE and CERTIFAI for all four datasets. For the selection, it is recommended to select 40% of the population to undergo the evolutionary process [42]; however, after multiple tests with selection values equal to 200, 400, and 600 individuals, we set the selection value to 400, which achieved best performance for all datasets.

B. RESULTS AND DISCUSSION

This section presents the outcomes of AFWGE in comparison to CERTIFAI [4] and analyzes them in various subsections according to the aforementioned evaluation criteria. At first, we assess and compare the distances and the number of feature changes. Next, we analyze the features importance and correlations from various perspectives. After that, we compare the number of objective function evaluations. Next, we analyze the counterfactual plausibility of AFWGE and CERTIFAI. Finally, we compare the computational time for both methods.

1) COMPARISON OF DISTANCES AND NUMBER OF FEATURE CHANGES

Table.2, 3, 4 and 5 present the results obtained from running each algorithm ten times. The tables display the mean and standard deviation values of the distance between the original instance and the generated counterfactuals, along with the number of feature changes for the Adult, Breast Cancer, Pima Indian Diabetes, and Iris datasets, respectively. What stands out in these tables is the significantly lower distance and number of feature changes observed in the counterfactuals generated by AFWGE. In comparison to CERTIFAI's counterfactuals; the mean distance of AFWGE is lower by 19.5%, 8.5%, 12%, and 37.2% for the Adult, Breast Cancer, Pima Indian Diabetes, and Iris datasets, respectively. Additionally, AFWGE reduces the mean number of feature changes in CERTIFAI's counterfactuals for the Adult, Breast Cancer, Pima Indian Diabetes, and Iris datasets by 13.7%, 2.3%, 10.1%, and 4.8% respectively. Moreover, AFWGE exhibits a very low standard deviation (STD), indicating its robustness, although slightly lower than CERTIFAI.

TABLE 2. Comparison between the distance and the number of features changes, for the counterfactual explanations of AFWGE and CERTIFAI on the Adult dataset.

	Distances Average		Number of Features Changes Average	
Run#	AFWGE	CERTIFAI	AFWGE	CERTIFAI
Run 1	0.27	0.314	554	607
Run 2	0.258	0.315	535	610
Run 3	0.265	0.322	534	601
Run 4	0.261	0.31	512	600
Run 5	0.264	0.313	540	611
Run 6	0.235	0.32	502	593
Run 7	0.209	0.291	480	577
Run 8	0.219	0.278	479	540
Run 9	0.235	0.303	471	590
Run 10	0.261	0.31	512	600
Mean	**0.2477**	0.3076	**511.9**	592.9
STD	0.0215	0.0136	28.7805876	21.1998428

TABLE 3. Comparison between the distance and the number of features changes for the counterfactual explanations of AFWGE and CERTIFAI on the Breast Cancer dataset.

	Distances Average		Number of Features Changes Average	
Run#	AFWGE	CERTIFAI	AFWGE	CERTIFAI
Run 1	0.238	0.26	12864	13132
Run 2	0.238	0.261	12774	13139
Run 3	0.237	0.261	12770	13093
Run 4	0.237	0.26	12749	13166
Run 5	0.239	0.26	12907	13133
Run 6	0.238	0.261	12707	13140
Run 7	0.238	0.259	12873	13149
Run 8	0.239	0.26	12853	13059
Run 9	0.239	0.26	12853	13091
Run 10	0.237	0.259	12846	13168
Mean	**0.238**	0.2601	**12819.6**	13127
STD	0.0008	0.0007	64.6291988	35.1757492

TABLE 4. Comparison between the distance and the number of features changes, for the counterfactual explanations of AFWGE and CERTIFAI on the Pima Indian Diabetes dataset.

	Distances Average		Number of Features Changes Average	
Run#	AFWGE	CERTIFAI	AFWGE	CERTIFAI
Run 1	0.104	0.118	1678	1837
Run 2	0.102	0.117	1593	1831
Run 3	0.105	0.118	1716	1844
Run 4	0.103	0.117	1687	1846
Run 5	0.104	0.119	1627	1885
Run 6	0.105	0.118	1673	1816
Run 7	0.103	0.117	1654	1829
Run 8	0.103	0.116	1632	1805
Run 9	0.104	0.117	1633	1845
Run 10	0.103	0.12	1666	1888
Mean	**0.1036**	0.1177	**1655.9**	1842.6
STD	0.001	0.0012	35.5166628	26.5631323

To facilitate a side-by-side comparison of the distance and number of feature changes results between AFWGE and CERTIFAI, box plots were used, as illustrated in Fig.6 and Fig.7 for the distance and the number of feature changes respectively. From both figures, it is evident that AFWGE exhibits significantly lower values for both distance and number of feature changes compared to CERTIFAI. Furthermore, a Wilcoxon Signed-Rank test was conducted on the results to ascertain the presence of any statistically significant difference between the two methods. The Wilcoxon test uses the following null h_0 and alternative h_A hypotheses (h_0 : The average distances and number of changes is equal between the two groups, h_A: The average distances and number of changes is not equal between the two groups). Interestingly,

TABLE 5. Comparison between the distance and the number of features changes, for the counterfactual explanations of AFWGE and CERTIFAI on the Iris Diabetes dataset.

	Distances Average		Number of Features Changes Average	
Run#	AFWGE	CERTIFAI	AFWGE	CERTIFAI
Run 1	0.036	0.057	136	147
Run 2	0.035	0.058	139	144
Run 3	0.036	0.057	139	146
Run 4	0.036	0.058	137	144
Run 5	0.036	0.056	136	145
Run 6	0.036	0.057	136	147
Run 7	0.037	0.057	139	144
Run 8	0.036	0.058	138	146
Run 9	0.036	0.057	140	145
Run 10	0.035	0.057	140	141
Mean	**0.0359**	0.0572	**138**	144.9
STD	0.0006	0.0006	1.63299316	1.79195734

for both measures and for all four datasets, there was a significant difference at a significance level of $\alpha = 0.05$; thus the null hypothesis was rejected. This is a sufficient evidence that the distances and number of feature changes produced by AFWGE are significantly less than those generated by CERTIFAI.

2) ANALYSIS OF FEATURES IMPORTANCE

Changing a particular feature more often than another when comparing the input and its counterfactuals implies that that feature is more significant for a model [4]. As reported in [4], CERTIFAI's number of feature changes represents features' importance, since it is similar to those returned by Python's XGBoost library [43]. On the other hand, AFWGE changes less features than CERTIFAI, which indicates sparser counterfactuals (refer to the definition of sparsity at the beginning of Section V.1, Evaluation Metrics). It is intriguing to investigate whether AFWGE alters features in a manner that contradicts their importance as reported in [4]. In other words, does the achieved sparsity by AFWGE impact the interpretation of feature importance as reported in CERTIFAI? To answer this question, we plot the number of feature changes distribution for both algorithm for all four datasets in Fig.8.

Looking at Fig.8, it appears that AFWGE changes the features in a similar manner to CERTIFAI. To confirm this observation, a Chi-Squared test was carried out. It uses the following null and alternative hypotheses (h_0: There is no difference in the distribution of the number of feature changes between AFWGE and CERTIFAI, h_A: There is a significant difference in the distribution of the number of feature changes between AFWGE and CERTIFAI). If the Chi-squared value is equal to or larger than the critical value, the null hypothesis rejected. Otherwise, the null hypothesis is accepted. In the results, for all four datasets, the Chi-squared value is smaller

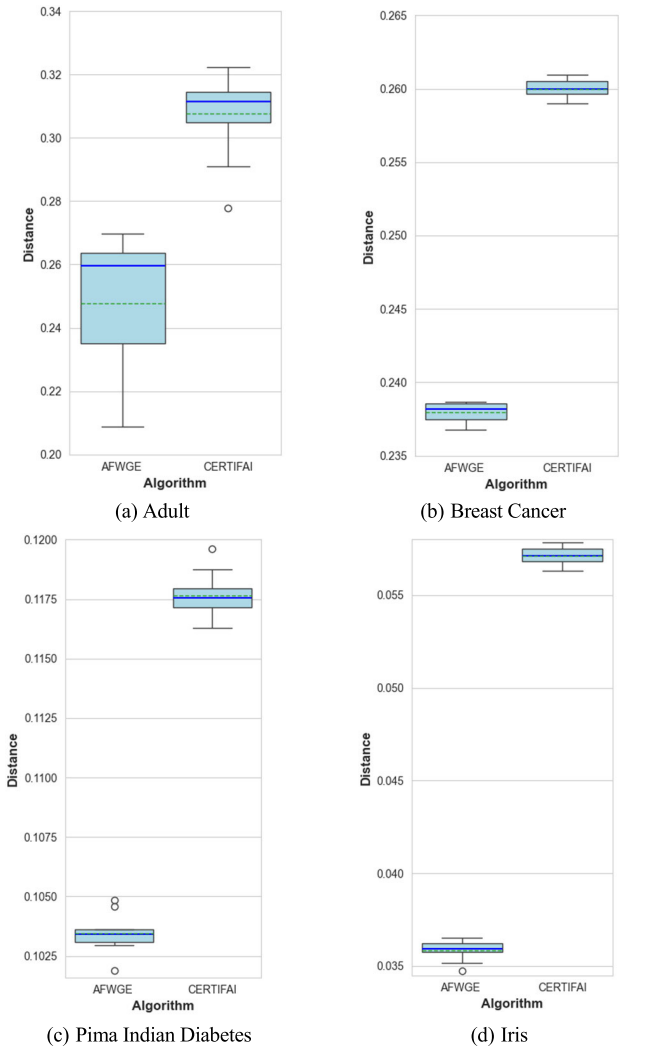


FIGURE 6. Comparison between the averages of the distances of the explanations produced by AFWGE and CERTIFAI on the four datasets. The green dashed line (_ _ _ _) represents the mean value, and the blue straight line (____) represents the median.

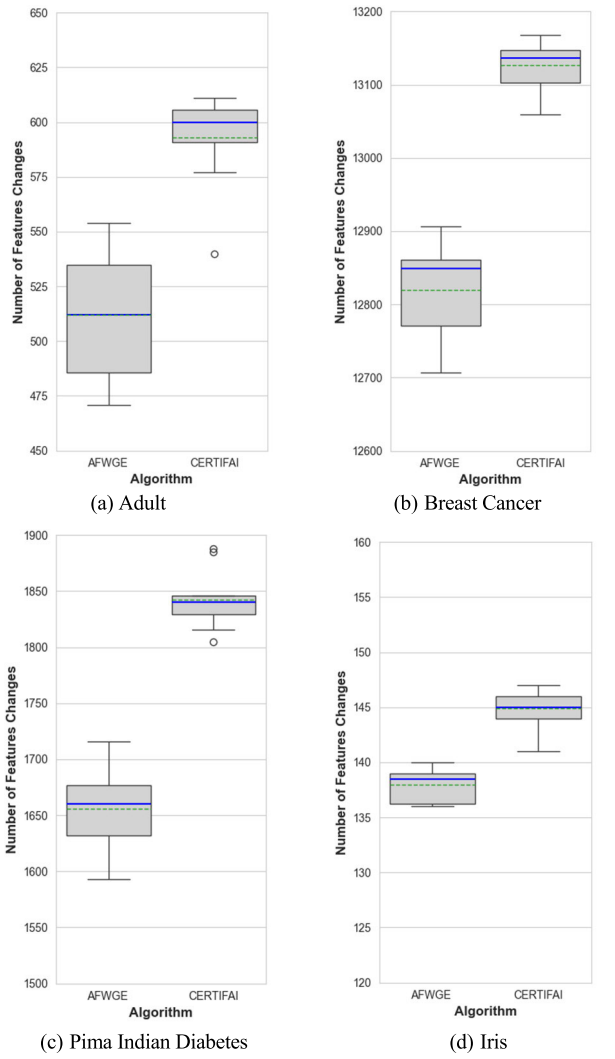


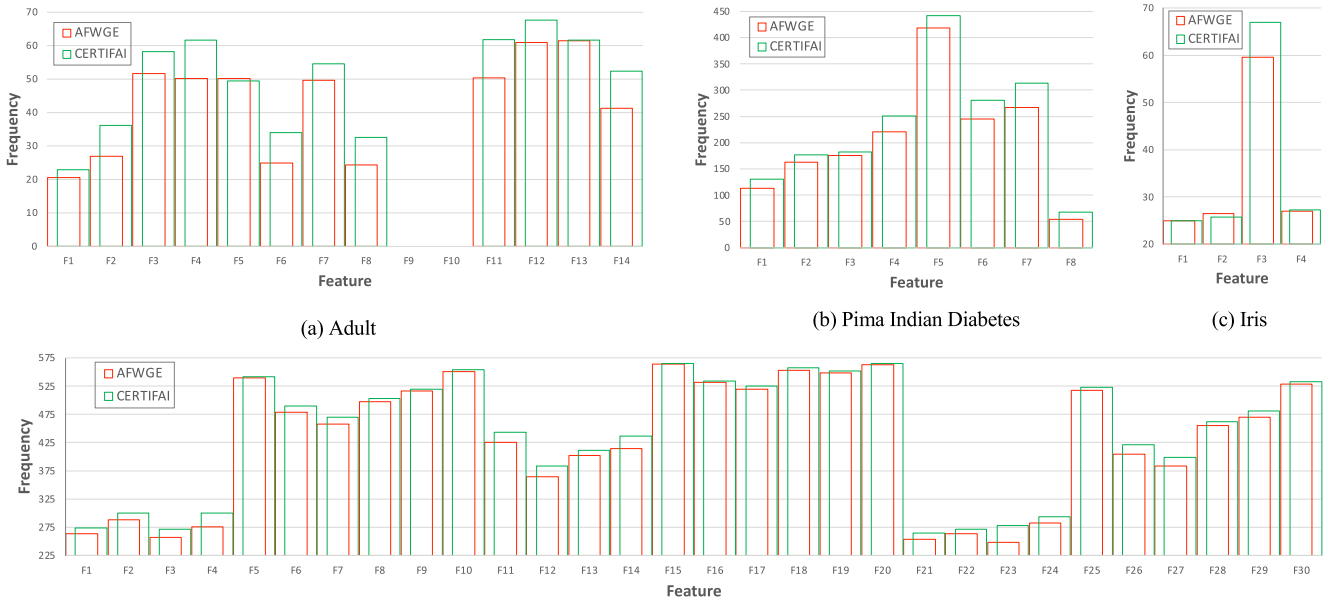
FIGURE 7. Comparison between the averages of the number of features changes for the explanations produced by AFWGE and CERTIFAI on the four datasets. The green dashed line (_ _ _ _) represents the mean value, and the blue straight line (____) represents the median.

than the critical value at $\alpha = 0.05$, so the null hypothesis can be accepted. Thus, there is no difference in the distribution of the number of feature changes between AFWGE and CERTIFAI. However, same distribution does not necessarily mean that they are identical. The distribution describes the feature changes behavior from a probabilistic standpoint [44]. This implies that the occurrence of each feature change is equally likely in both. In other words, although AFWGE changes less features than CERTIFAI, meaning that is sparser, it preserves the features significance as in CERTIFAI.

We also questioned whether feature weights correspond to features importance for the model. To investigate this, we compared the feature weights produced by AFWGE with the feature importance returned by Python's XGBoost [43] for the Pima Indians Diabetes dataset, which is also reported in CERTIFAI. For both XGBoost and CERTIFAI, BMI and Glucose were found to be the most important features in predicting diabetes risk. Interestingly, the feature weights produced by AFWGE, as depicted in the chart in Fig.9, provides almost the same feature ranking of feature importance as done by both XGBoost and CERTIFAI. BMI and Glucose have the highest weights, while skin thickness feature is the lowest.

3) COMPARISON OF NUMBER OF OBJECTIVE FUNCTION EVALUATIONS

In the next experiment, we examine the fitness values and the quantity of objective function evaluations for both algorithms, as depicted in Fig.10. Fig.10 (b) shows that AFWGE carries out a larger number of evaluations of the objective function than CERTIFAI, with a percentage difference of no more than 40% for generating counterfactuals. Indeed, this is somehow expected since AFWGE assesses each chromosome following crossover and mutation and compares it with



(a) Adult

(b) Pima Indian Diabetes

(c) Iris

(d) Breast Cancer

FIGURE 8. Distribution of the number of feature changes for the explanations generated by AFWGE and CERTIFAI on the four datasets.

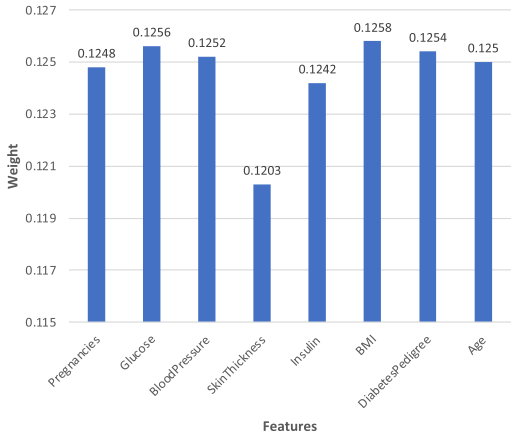
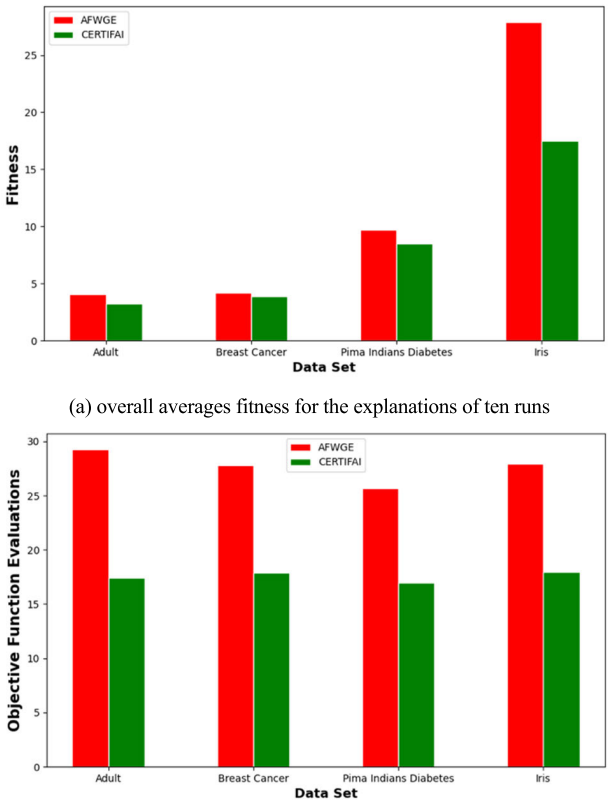


FIGURE 9. Feature importance for the model, trained on the Pima Indian diabetes dataset, measured by the weight of a feature to generate the counterfactual.

its parents to choose the best one, prior to updating the population. On the other hand, with respect to fitness, as computed by (2), Fig.10 (a) shows that the average fitness achieved by AFWGE exhibits a larger value compared to CERTIFAI's. Overall, upon examining Fig.10, it is evident that AFWGE yields superior fitness as the number of objective function evaluations increases. However, a direct comparison of the ratios of these AFWGE metrics reveals that they are not proportional. In other words, the number of objective function evaluations increases at a higher rate than the improvement in fitness, which is a side effect of the enhanced fitness achieved by AFWGE.



(a) overall averages fitness for the explanations of ten runs

(b) overall averages objective function evaluations for the explanations of ten runs

FIGURE 10. Comparison of the overall averages fitness and objective function evaluations by AFWGE and CERTIFAI on the four datasets.

TABLE 6. AFWGE generated counterfactuals properties.

Counterfactual Explanation		AFWGE	
Property	Definition	Metric	Validation
Validity	Counterfactuals should lead to a different outcome than the original model's [39].	Model Prediction	Checking counterfactual predictions using the same black box model used to generate the original corresponding instance prediction.
Proximity	Counterfactual explanations should be as close as possible to the original instance with respect to feature values where fewer considerable changes are required.	Distance	Wilcoxon test shows that the distances produced by AFWGE are significantly less than those of CERTIFAI.
Sparsity	Counterfactual explanations should change as few features as possible to increase its understanding and effectiveness.	Number of feature changes	Wilcoxon test shows that the number of feature changes with AFWGE is significantly less than those of CERTIFAI.
Plausibility	a. Counterfactual values should not exceed or fall below those that are observed in the data.	a. Realistic generation	a. AFWGE generates realistic values by choosing randomly one of the feature categories over the data instances for categorial features and generates a random number between the minimum and the maximum numbers of the feature over the data instances for numerical features.
	b. Counterfactuals should combine realistic features that are compatible with existing examples in the data.	b. Correlation	b. AFWGE changes features according to weights, which shows closer correlations to the dataset features' correlation than the number of feature changes correlation exhibited by CERTIFAI.
Actionability	Counterfactual explanations must exclude value changes on non-actionable(or non-mutable) features [40].	Features Constraints	Restrict non-actionable features (e.g., Age value may only increase, while Race, and Sex features values are fixed)

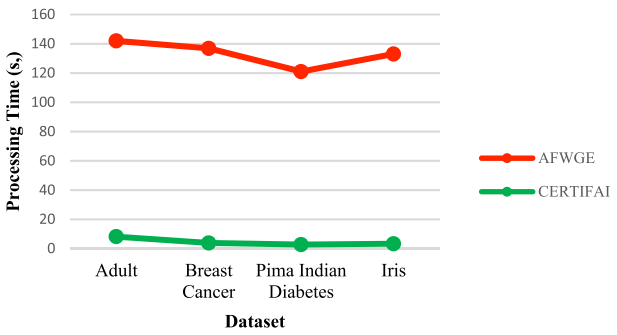


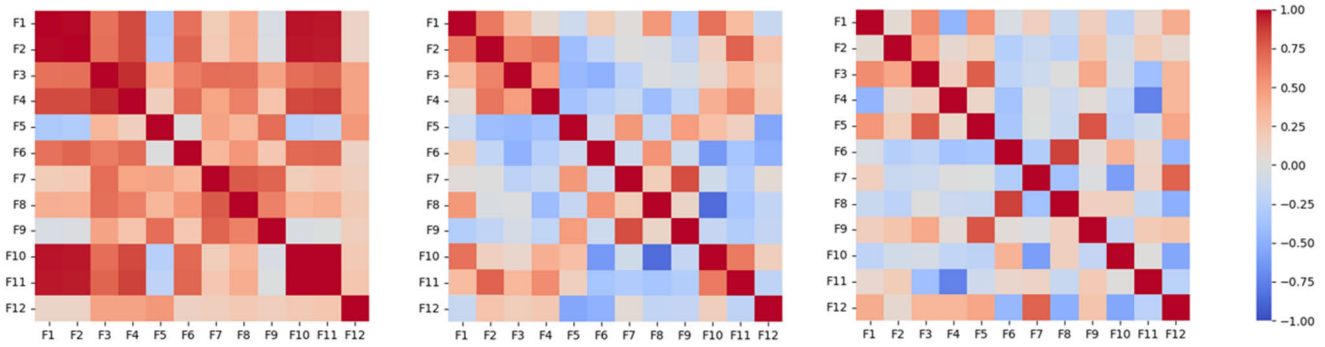
FIGURE 11. Comparison of the overall average processing time for the explanations of ten runs by AFWGE and CERTIFAI on the four datasets.

4) ANALYSIS OF PLAUSIBILITY

As previously mentioned, plausibility in counterfactuals is fundamentally a relationship concept that if one feature changes, we must consider how other features change alongside that feature [39]. To assess the plausibility achieved by both methods under consideration, we analyzed the correlation of features using Pearson's correlation on the Breast Cancer dataset. This dataset was chosen because it contains highly correlated features, as illustrated in Fig.11 (a). The correlation is calculated and represented using a heat map, where the more the features are correlated, the more the corresponding cell becomes red. To get an accurate perception of the correlations, we considered twelve features which are the most correlated features of the Breast Cancer dataset. Then, we compared which method, AFWGE or CERTIFAI, provides more plausible counterfactuals. Recall, though, that AFWGE's produces counterfactuals guided by feature weights, where the higher the weight of the feature, the more the probable the change in this feature. Thus, we calculated the feature weights correlation for AFWGE in Fig.11(b), as opposed to the number of feature changes correlation for CERTIFAI in Fig.11(c).

When analyzing correlations, the method that yields more plausible counterfactuals takes into account the dataset's feature correlations more effectively than the other method. In other words, the plausibility of the produced counterfactuals increases as they better preserve the correlation among the features in the dataset [4], [31]. So, in Fig.12, we compared the correlations exhibited by both AFWGE and CERTIFAI to the Breast Cancer's features correlation.

As can be seen in Fig.12 (a) and (b), the similarity between AFWGE's and Breast Cancer features correlations is remarkable, while CERTIFAI's correlation tends to move farther from the Breast Cancer features (Fig.12 (c)).

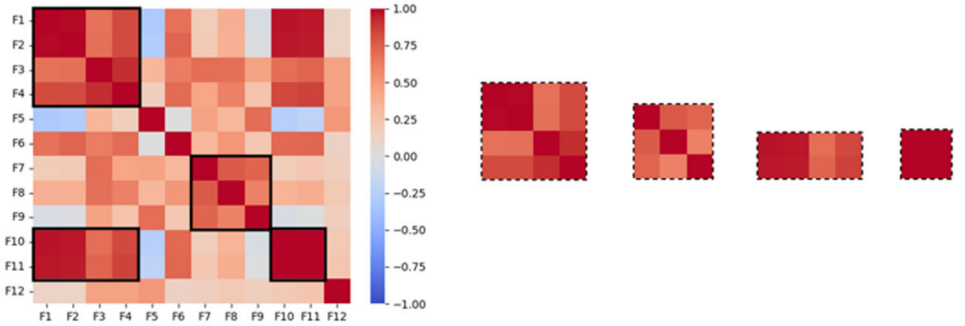


(a) Breast Cancer dataset features correlation

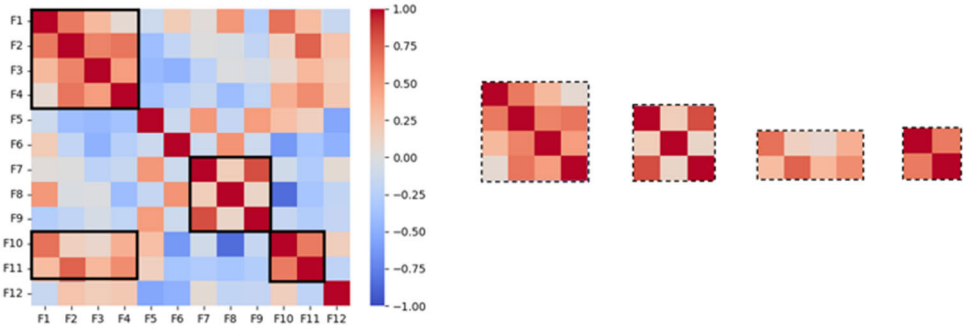
(b) AFWGE's features weights correlation

(c) CERTIFAI's number of features changes correlation

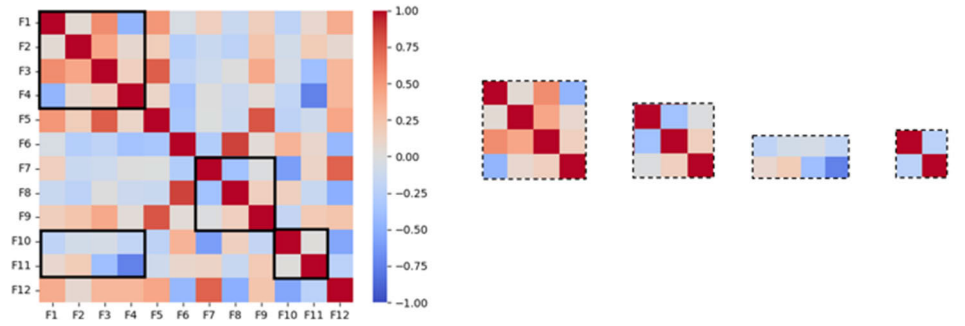
FIGURE 12. Correlation graph using Pearson's correlation for the twelve most correlated features of Breast Cancer dataset.



(a) Feature correlation graph of Breast Cancer dataset where four different separated segments of cells have high correlation



(b) AFWGE's features weights correlation graph for Breast Cancer dataset is segmented following the same segmentation of feature correlation in (a)



(c) CERTIFAI's features changes correlation graph for Breast Cancer dataset is segmented following the same segmentation of feature correlation in (a)

FIGURE 13. Compare (b)AFWGE and (c)CERTIFAI counterfactuals plausibility by comparing which one provides more similar correlation graph to the (a)Breast Cancer dataset features correlation.

The aim of calculating the correlations is to ensure the plausibility of counterfactuals. This means that a model should consider changing more features that are correlated when generating the counterfactuals. For instance, if an applicant is denied a housing loan, a plausible counterfactual may suggest changing multiple associated features to increase the likelihood of loan approval, such as increasing their salary, which may require obtaining higher education, or changing their social status to married, as couples have higher chances of loan approval. However, altering all three correlated features can complicate the generation of counterfactual explanations and affect their sparsity. Hence, it is preferable to change fewer correlated features to achieve counterfactual sparsity while maintaining plausibility. To strike a balance between these requirements, the model should prioritize highly correlated features when generating counterfactuals [31].

5) COMPARISON OF PROCESSING TIME

Finally, we investigated the tradeoff between the quality and the processing time of the counterfactual explanations for the considered algorithms on all datasets. As shown in Fig.13, AFWGE consumes considerably more processing time than CERTIFAI. However, as suggested in [40], the processing time overhead is not significant for all counterfactual explanation applications. In fact, the rapid development of one or more counterfactual explanations is relevant for only certain applications that require an immediate response such as machine teaching, where explanation algorithms need to perform in real-time, and in low-complexity platforms like mobile devices [45].

6) RESULTS SUMMARY

Based on the extensive experimental results on various data sets, the Wilcoxon test indicates that the distances produced by AFWGE are significantly less than those of CERTIFAI. Similarly, the number of feature changes with AFWGE is significantly lower than those of CERTIFAI. Additionally, AFWGE changes features according to weights, demonstrating closer correlations to the dataset features' correlation compared to the number of feature changes correlation exhibited by CERTIFAI. In summary, AFWGE results reported smaller distances and a fewer number of feature changes, indicating better counterfactual explanation effectiveness [40]. Moreover, AFWGE generated counterfactual explanations that fulfill fundamental properties: validity, proximity, sparsity, plausibility, and actionability of explanations. These properties of AFWGE counterfactuals are reviewed and validated in Table. 6.

VI. CONCLUSION

This study introduced an Adaptive Feature Weight Genetic Explanation (AFWGE) method, a novel approach for generating counterfactual explanations in explainable AI. By leveraging a genetic algorithm with embedded feature weights, AFWGE significantly enhances the performance of counterfactual generation, producing more efficient counterfactual explanations. The innovation lies in introducing feature weight adaptation, seamlessly integrated into the optimization process of generating counterfactuals. The empirical results demonstrate the effectiveness of this approach, showcasing superior proximity, sparsity, plausibility, and actionability of the generated counterfactuals. Furthermore, the analysis highlights the role of feature weights as a reliable indicator of feature importance, providing valuable insights for understanding and interpreting the underlying mechanisms of the model. AFWGE contributes to counterfactual explanation generation and offers a robust framework for evaluating the importance of features in machine learning models. This research attempts to bridge the gap in XAI by offering a powerful method for generating interpretable and trustworthy explanations for AI models.

The proposed method currently requires a long processing time; however, we are committed to optimizing its efficiency in the future. This may involve leveraging external libraries to parallelize the algorithm and implementing cache functions for enhanced speed. Additionally, we aim to extend the method's assessment across various AI models to ensure its robust applicability and performance across diverse scenarios.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University for supporting this work.

REFERENCES

[1] Z. Dikopoulou, S. Moustakidis, and P. Karlsson, "GLIME: A new graphical methodology for interpretable model-agnostic explanations," 2021, *arXiv:2107.09927*.
[2] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti, "Local rule-based explanations of black box decision systems," 2018, *arXiv:1805.10820*.
[3] D. Wang, Q. Yang, A. Abdul, and B. Y. Lim, "Designing theory-driven user-centric explainable AI," in *Proc. CHI Conf. Human Factors Comput. Syst.* Glasgow, U.K.: ACM, May 2019, pp. 1–15, doi: 10.1145/3290605.3300831.
[4] S. Sharma, J. Henderson, and J. Ghosh, "CERTIFAI: A common framework to provide explanations and analyse the fairness and robustness of black-box models," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.* New York, NY, USA: ACM, Feb. 2020, pp. 166–172, doi: 10.1145/3375627.3375812.
[5] Dr. M. Turek. *Explainable Artificial Intelligence (XAI)*. Defense Advanced Research Projects Agency. Accessed: Mar. 14, 2022. [Online]. Available: https://www.darpa.mil/program/explainable-artificial-intelligence
[6] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, "Evaluating the robustness of neural networks: An extreme value theory approach," 2018, *arXiv:1801.10578*.
[7] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," 2016, *arXiv:1608.04644*.
[8] M. Hosny and S. Al-Malak, "An adaptive genetic algorithm approach for optimizing feature weights in multimodal clustering," in *Advances in Intelligent Systems and Computing*, vol. 1229, K. Arai, S. Kapoor, and R. Bhatia, Eds. Cham, Switzerland: Springer, 2020, pp. 181–197, doi: 10.1007/978-3-030-52246-9_13.
[9] B. Ustun and C. Rudin, "Supersparse linear integer models for optimized medical scoring systems," *Mach. Learn.*, vol. 102, no. 3, pp. 349–391, Mar. 2016, doi: 10.1007/s10994-015-5528-6.
[10] J. Jung, C. Concannon, R. Shroff, S. Goel, and D. G. Goldstein, "Simple rules to guide expert classifications," *J. Roy. Stat. Soc. A, Statist. Soc.*, vol. 183, no. 3, pp. 771–800, Jun. 2020, doi: 10.1111/rssa.12576.

[11] B. Mittelstadt, C. Russell, and S. Wachter, "Explaining explanations in AI," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2019, pp. 279–288, doi: 10.1145/3287560.3287574.

[12] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, Jul. 2019, doi: 10.3390/electronics8080832.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," 2016, *arXiv:1602.04938*.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, vol. 32, no. 1, p. 9.

[15] S. Wachter et al., "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harvard J. Law Technol.*, vol. 31, no. 2, pp. 841–887, 2018.

[16] O. Radley-Gardner, H. Beale, and R. Zimmermann, *Fundamental Texts on European Private Law*. Portland, OR, USA: Hart, 2016, doi: 10.5040/9781782258674.

[17] A. Weller, "Challenges for transparency," in *Proc. Workshop Human Interpretability Mach. Learn. ICML*, Sydney, NSW, Australia, 2017, pp. 1–8.

[18] R. D. Bock and M. Lieberman, "Fitting a response model for *n* dichotomously scored items," *Psychometrika*, vol. 35, pp. 179–197, Jun. 1970.

[19] J. W. Smith, J. Everhart, W. Dicksont, W. Knowler, and R. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," in *Proc. Annu. Symp. Comput. Appl. Med. Care*, Nov. 1988, pp. 261–265.

[20] A. Stivala, G. Robins, Y. Kashima, and M. Kirley, "Diversity and community can coexist," *Amer. J. Community Psychol.*, vol. 57, nos. 1–2, pp. 243–254, Mar. 2016, doi: 10.1002/ajcp.12021.

[21] B. Ustun, A. Spangher, and Y. Liu, "Actionable recourse in linear classification," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2019, pp. 10–19, doi: 10.1145/3287560.3287566.

[22] I.-C. Yeh and C.-H. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 2473–2480, Mar. 2009, doi: 10.1016/j.eswa.2007.12.020.

[23] C. Russell, "Efficient search for diverse coherent explanations," 2019, *arXiv:1901.04909*.

[24] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 582–597, doi: 10.1109/SP.2016.41.

[25] M. Schleich, Z. Geng, Y. Zhang, and D. Suciu, "GeCo: Quality counterfactual explanations in real time," 2021, *arXiv:2101.01292*.

[26] A.-H. Karimi, G. Barthe, B. Balle, and I. Valera, "Model-agnostic counterfactual explanations for consequential decisions," in *Proc. 23rd Int. Conf. Artif. Intell. Statist. (AISTATS)*, Palermo, Italy, 2020, pp. 12–27.

[27] D. Mahajan, C. Tan, and A. Sharma, "Preserving causal constraints in counterfactual explanations for machine learning classifiers," in *Proc. CausalML Workshop NeurIPS*, 2019.

[28] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viegas, and J. Wilson, "The what-if tool: Interactive probing of machine learning models," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 56–65, 2019.

[29] C. Guo, J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger, "Simple black-box adversarial attacks," 2019, *arXiv:1905.07121*.

[30] N. Wiratunga, A. Wijekoon, I. Nkisi-Orji, K. Martin, C. Palihawadana, and D. Corsar, "DisCERN: Discovering counterfactual explanations using relevance features from neighbourhoods," 2021, *arXiv:2109.05800*.

[31] X. Dastile, T. Celik, and H. Vandierendonck, "Model-agnostic counterfactual explanations in credit scoring," *IEEE Access*, vol. 10, pp. 69543–69554, 2022, doi: 10.1109/ACCESS.2022.3177783.

[32] N. Panagant, S. Bureerat, and K. Tai, "A novel self-adaptive hybrid multi-objective meta-heuristic for reliability design of trusses with simultaneous topology, shape and sizing optimisation design variables," *Structural Multidisciplinary Optim.*, vol. 60, no. 5, pp. 1937–1955, Nov. 2019, doi: 10.1007/s00158-019-02302-x.

[33] F. Ming, W. Gong, L. Wang, and L. Gao, "Balancing convergence and diversity in objective and decision spaces for multimodal multi-objective optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 2, pp. 474–486, Apr. 2023, doi: 10.1109/TETCI.2022.3221940.

[34] A. S. Eesa and W. K. Arabo, "A normalization methods for backpropagation: A comparative study," *Sci. J. Univ. Zakho*, vol. 5, no. 4, p. 319, Dec. 2017, doi: 10.25271/2017.5.4.381.

[35] B. Becker and R. Kohavi, "Adult," UCI Mach. Learn. Tech. Rep., 1996, doi: 10.24432/C5XW20.

[36] W. Wolberg, O. Mangasarian, N. Street, and W. Street, "Breast cancer Wisconsin (Diagnostic)," UCI Mach. Learn. Tech. Rep., 1995, doi: 10.24432/C5DW2B.

[37] (1988). *Pima Indians Diabetes Database*. [Online]. Available: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[38] R. A. Fisher, "Iris," UCI Mach. Learn. Tech. Rep., 1988, doi: 10.24432/C56C76.

[39] R. Guidotti, "Counterfactual explanations and how to find them: Literature review and benchmarking," *Data Mining Knowl. Discovery*, 2022, pp. 1–55, doi: 10.1007/s10618-022-00831-6.

[40] G. Navas-Palencia, "Optimal counterfactual explanations for scorecard modelling," 2021, *arXiv:2104.08619*.

[41] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, and F. Turini, "Factual and counterfactual explanations for black box decision making," *IEEE Intell. Syst.*, vol. 34, no. 6, pp. 14–23, Nov. 2019, doi: 10.1109/MIS.2019.2957223.

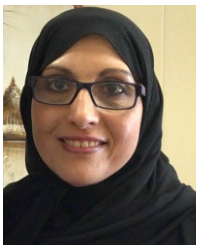
[42] N. Chawla and W. Wang, "Generalized inverse classification," in *Proc. SIAM Int. Conf. Data Mining*, Jun. 2017, pp. 785–794, doi: 10.1137/1.9781611974973.

[43] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. San Francisco, CA, USA: ACM, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[44] F. Dominici, J. J. Faraway, M. Tanner, and J. Zidek, *Texts in Statistical Science Series*.

[45] Y. Zhao, "Fast real-time counterfactual explanations," 2020, *arXiv:2007.05684*.

EBTISAM ALJALAUD received the Bachelor of Science degree in computer science from Imam Mohammad Ibn Saud Islamic University (IMSIU), and the Master of Science degree in computer science from King Saud University (KSU), Riyadh, Saudi Arabia, where she is currently pursuing the Ph.D. degree in computer science. Her research interests include artificial intelligence and algorithms.



MANAR HOSNY received the B.Sc. and M.Sc. degrees in computer science from The American University in Cairo, and the Ph.D. degree from Cardiff School of Computer Science and Informatics, Cardiff University, U.K., in 2010. Her extensive teaching experience encompasses various computer science subjects at both the graduate and undergraduate level. She is currently an Associate Professor and the former Vice Chair of the Computer Science Department, College of Computer and Information Sciences, King Saud University (KSU), Riyadh, Saudi Arabia. Her research interests include heuristic and metaheuristic algorithms for solving combinatorial optimization problems, especially focusing on routing and scheduling problems. Additionally, she has expertise in HCI, data mining, social network analysis, and explainable AI. Furthermore, she has contributed significantly to the field, with over 60 publications in reputable conferences and journals. Her research articles have been recognized and accepted by the academic community, demonstrating the quality and impact of her work. She also serves as an Associate Editor for the *Journal of King Saud University-Computer and Information Sciences*.