Enhanced Fairness Testing via Generating Effective Initial Individual Discriminatory Instances

ZHAO TIAN, College of Intelligence and Computing, Tianjin University, China
MINGHUA MA, Tsinghua University, China
MAX HORT, University College London, UK
FEDERICA SARRO, University College London, UK
HONGYU ZHANG, Chongqing University, China
JUNJIE CHEN, College of Intelligence and Computing, Tianjin University, China

Fairness testing aims at mitigating unintended discrimination in the decision-making process of data-driven AI systems. Individual discrimination may occur when an AI model makes different decisions for two distinct individuals who are distinguishable solely according to protected attributes, such as age and race. Such instances reveal biased AI behaviour, and are called Individual Discriminatory Instances (IDIs).

In this paper, we propose an approach for the selection of the initial seeds to generate IDIs for fairness testing. Previous studies mainly used random initial seeds to this end. However this phase is crucial, as these seeds are the basis of the follow-up IDIs generation. We dubbed our proposed seed selection approach *I&D*. It generates a large number of initial IDIs exhibiting a great diversity, aiming at improving the overall performance of fairness testing.

Our empirical study reveals that *I&D* is able to produce a larger number of IDIs with respect to four state-of-the-art IDI generation approaches, generating 1.86X more IDIs on average. When using the IDIs generated with *I&D* for retraining a machine learning model, the percentage of IDIs in the input space $\mathbb{I}$ is decreased by 24.9% on average, implying that *I&D* is effective for improving the model's fairness.

CCS Concepts: • **Software and its engineering** → **Software verification and validation**; • **Computing methodologies** → **Machine learning**.

Additional Key Words and Phrases: Fairness Testing, Machine Learning Models, Software Fairness

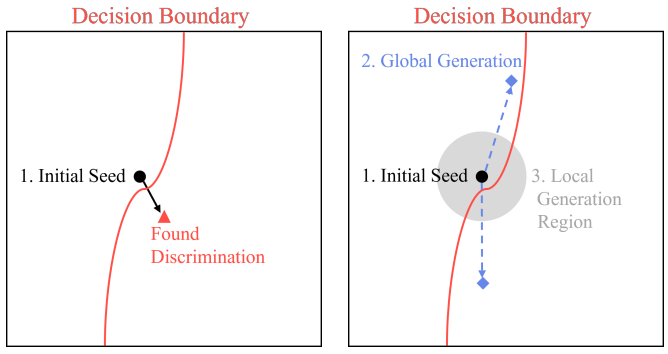
ACM Reference Format:
Zhao Tian, Minghua Ma, Max Hort, Federica Sarro, Hongyu Zhang, and Junjie Chen. 2025. Enhanced Fairness Testing via Generating Effective Initial Individual Discriminatory Instances. *J. ACM* 1, 1, Article 111 (May 2025), 24 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Artificial Intelligence (AI) systems have become increasingly popular over the years, and are now at the core of many data-driven software systems such as loan approval and risk assessments [45]. Although machine learning models have achieved significant performance improvements, their fairness remains a prominent concern that needs to be addressed [7, 12, 34].

Authors' addresses: Zhao Tian, tianzhao@tju.edu.cn, College of Intelligence and Computing, Tianjin University, Tianjin, China; Minghua Ma, minghuama@microsoft.com, Tsinghua University, Beijing, China; Max Hort, max.hort.19@ucl.ac.uk, University College London, London, UK; Federica Sarro, f.sarro@ucl.ac.uk, University College London, London, UK; Hongyu Zhang, hyzhang@cqu.edu.cn, Chongqing University, Chongqing, China; Junjie Chen, junjiechen@tju.edu.cn, College of Intelligence and Computing, Tianjin University, Tianjin, China.

0004-5411/2025/5-ART111 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn



(a) A pair of IDIs (b) Generation

Fig. 1. Sketch map of individual discriminatory instances (IDIs) and their generation process.

Software fairness testing sets out to reveal fairness bugs of the software system (*i.e.*, situations that reveal bias) [14]. Among fairness testing goals, *individual discrimination* has been addressed frequently [5, 16]. Individual discrimination occurs when a machine learning model yields different prediction results for the instances that can only be distinguished by one or more protected attributes, such as age, race, or gender [5, 25]. For example, two individuals with different gender (*e.g.*, female and male), all other attributes being equal, should receive the same response when applying for a bank loan.

To date, fairness testing of machine learning and AI systems has become a topic of interested in Software Engineering research [14, 39]. This includes approaches for generating individual discriminatory instances (IDIs) for fairness testing [14], such as AEQUITAS [44], Symbolic Generation (SG) [3], ADF [50], and EIDIG [49]. All these existing IDI generation approaches follow the same three-phase framework, as depicted in Figure 1. First, they choose *initial seeds* from a given dataset, *i.e.*, the black dot in Figure 1, according to a given strategy.[1] Second, they perform a *global generation* to explore a wider range of IDIs, which is illustrated by the blue squares in Figure 1(b). Once an IDI is found during the *global generation* phase, a *local generation* is performed in the third step. The *local generation* searches for further IDIs in the neighborhood of the collected IDIs from the *global generation* phase, which is depicted in gray in Figure 1(b). Afterwards, the model can be retrained to minimize discrimination using the generated IDIs.

The existing approaches focus on improving the latter two phases to boost the overall performance of fairness testing. For example, AEQUITAS randomly perturbs attribute values in the local generation phase. SG globally creates a decision tree and then performs symbolic generation for the local generation. On neural network models, ADF and EIDIG employ an adversarial sampling technique during both the global and local generation phases. Regarding the first phase (*i.e.*, initial seeds selection), existing approaches adopt random or clustering-based sampling strategies. However, we argue that this phase is very important to the overall performance of fairness testing. Specifically, since this phase is the basis of the follow-up global and local generation phases, the quality of initial seeds could directly affect the performance of IDI generation. On the one hand, as demonstrated in the existing study [49, 50], the number of generated IDIs in the two-generation

[1]Specifically, AEQUITAS selects seeds from the input data using a random sampling mechanism, while other approaches, such as SG, ADF, and EIDIG, cluster the input data using K-means, and then choose initial seeds from each cluster on a round-robin basis.

phases grows linearly with the number of IDIs in the initial seeds. That is, the more initial IDIs provided, the more IDIs may be generated, which is advantageous for fairness testing. On the other hand, initial IDIs with a high diversity could facilitate generating a larger variety of IDIs, which can improve the retraining and debiasing of machine learning models. That is, both the number of IDIs and the diversity of IDIs are pursued in practice, which can be controlled by the quality of the initial IDIs, and thus designing a more effective initial seed selection strategy is desired.

In this work, we propose a novel initial seed selection approach, named *I&D*. Our approach aims to obtain a large initial set of IDIs, and thus improve the overall performance of fairness testing. That is, the contribution of *I&D* is orthogonal to existing fairness testing approaches. *I&D* can be integrated with any existing fairness testing approaches by replacing their initial seed selection method with *I&D*, in order to further improve their performance.

I&D is required to address two major challenges: *1) How do we effectively select seeds to obtain more initial IDIs?* and *2) How do we improve the diversity of initial IDIs?* For the first challenge, we design a novel IDI initialization algorithm that constructs a "chiral" model,[2] which is trained by mutating the protected attributes of the training data. It is more likely to identify initial IDIs if the chiral model predicts differently from the original model, for the same instance. Moreover, the use of a chiral models allows us to obtain IDIs close to the decision boundary of a classification model (see Section 3 for more details).

To overcome the second challenge, we propose to use the SHAP value [29], which is a game-theory-based approach for explaining the prediction output of any machine learning model, to further interpret the difference in prediction behavior between the chiral model and the original model for each initial IDI. Then, *I&D* clusters the initial IDIs based on their SHAP values, and selects diverse initial IDIs from each cluster in a round-robin way for subsequent usage in the global and local generation phases.

To comprehensively evaluate the effectiveness of *I&D*, we undertake a thorough empirical evaluation using three open-source datasets that have been widely used in prior studies [3, 13, 44, 49, 50]. To investigate whether *I&D* boosts the performance of existing IDI generation approaches, such as AEQUITAS, SG, ADF, and EIDIG, we integrate *I&D* with each of them by replacing their initial seed selection strategy with *I&D*, and further investigate the performance on four different machine learning models (*i.e.*, Logistic Regression, Support Vector Machines, Decision Trees, and Multi-layer Perceptron Classifier).

The empirical results reveal that *I&D* can effectively obtain improved initial seeds, and significantly outperform all the compared IDI generation approaches with their original initial seeds. For example, the average number of generated IDIs with *I&D* is 2,342, while the original approaches generated 1,260 IDIs on average. Thereby, *I&D* achieves an improvement of 1.86X. The results show that this is a promising approach for improving IDI initialization. Our results also show that using *I&D* can improve the fairness of these machine learning models considered herein, when such a model is re-trained with the IDIs generated by *I&D*.

Overall, the key contributions of our work are as follows:

- To the best of our knowledge, this is the first study to improve existing fairness testing techniques by providing initial IDIs.
- To produce more, diverse initial IDIs, we design and implement a method called *I&D* through building a chiral model and measuring the SHAP value of each initial IDI for explaining the prediction difference between the chiral and original models.

[2]Chirality is a feature of asymmetry that is important in many research works [10]. An object (model) is chiral if it can be distinguished from its mirror image (trained by mutated data); that is, it cannot be superimposed onto it.

- We conduct an extensive empirical assessment of *I&D* based on three public datasets and four existing IDI generation approaches. The results show that *I&D* can effectively boost the performance of these approaches in terms of both the number of generated IDIs and the reduced discrimination after retraining with the IDIs found.

The replication package for this paper, including all our data, source code, and documentation, is publicly available online at https://anonymous.4open.science/r/fairness-096F/.

2 BACKGROUND

In this section, we first introduce the concept and notation of individual discrimination. Then, we review the existing IDI generation approaches. Finally, we describe the SHAP value, which is a commonly adopted model explanation approach.

2.1 Individual Discrimination

AI systems utilise various types of machine learning models, including decision trees [28], regression analysis [46], and neural networks [37]. Following prior studies on fairness testing [3, 44, 49, 50], we focus on the binary classification problem, which is important in AI systems [28]. To demonstrate the generality of our approach, we do not focus on a specific type of machine learning model in our study. We denote the machine learning model $\mathcal{M} : X \rightarrow Y$, and it generates a predicted class label $y \in Y$ with the highest probability for a given instance $x \in X$. $A = A_1, A_2, ..., A_n$ represents a set of attributes (features) in X. Assuming that each attribute A_i ($i \in [1, n]$) has a domain value space of $\mathbb{I}_i$, then the total input domain of x is equal to all possible combinations of attribute value spaces, *i.e.*, $\mathbb{I} = \mathbb{I}_1 \times \mathbb{I}_2 \times \ldots \times \mathbb{I}_n$.

Finding and generating individual discriminatory instances (IDIs) for a given machine learning model is the first step towards reducing discrimination and achieving individual fairness [3, 44, 49, 50]. Discrimination is frequently described in terms of a group of protected attributes, such as age, race, and gender. Individual discrimination occurs when a machine learning model makes different decisions for two identical individuals apart from protected attributes. Note that the list of protected attributes is often application-specific and unrelated to the prediction goal, which is provided in advance [50]. Deleting the protected attributes from the training data would not eliminate the bias, since individual discrimination may remain due to various correlations between protected and non-protected qualities [13].

Formally, the IDI x of a machine learning model can be defined as follows:

$$\begin{cases} \exists p \in P, x_p \neq x'_p \\ \forall q \in NP, x_q = x'_q \\ f(x) \neq f(x') \end{cases} \tag{1}$$

where x' exists in $\mathbb{I}$, $P \subset A$ is a set of protected attributes like race and gender. $NP \subset A$ is the set of non-protected attributes, $P \cup NP = A$, and $P \cap NP = \varnothing$.

We use the Census Income dataset as a running example.[3] Section 4 contains more information about this dataset. From this dataset, we consider the following two instances x and x':

$$x : [3, 5, 3, 0, 2, 8, 3, 0, \mathbf{1}, 2, 0, 40, 0, 0], \ f(x) = 0$$
$$x' : [3, 5, 3, 0, 2, 8, 3, 0, \mathbf{0}, 2, 0, 40, 0, 0], \ f(x') = 1$$

In the list, the attributes of an instance are represented as integers that are the model's input. Gender, which is displayed in bold, is assumed to be the protected attribute here. x denotes a male

[3] https://archive.ics.uci.edu/ml/datasets/adult

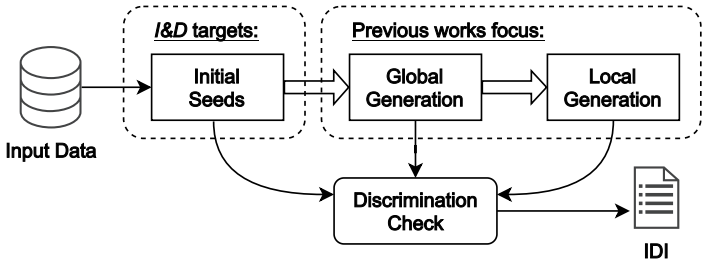


Fig. 2. A typical framework for IDI generation.

and x' denotes a female. Except for gender, we can see that x and x' have identical attribute values. Since the model $\mathcal{M}$ has different prediction outcomes $f(x)$ and $f(x')$, we say that x and x' are a pair of IDIs for the model.

In summary, machine learning models could make biased decisions for IDIs, which is destructive to fairness. It is important to effectively generate IDIs and improve the fairness of the machine learning model through testing and retraining.

Over the years, several IDI generators have been proposed [3, 44, 49, 50]. In Figure 2, we summarize a typical framework for IDI generation.

Initial seeds. At first, *initial seeds* are selected from the dataset, which are used in the subsequent steps for IDI generation.

THEMIS [20] has been the first work to address individual discrimination testing. It chooses initial seeds at random and then tests if they are IDIs without using a global or local generation phase. AEQUITAS [44] uses the same random initialization procedure. SG [3], ADF [50], and EIDIG [49] use K-means to cluster the data. Afterwards, they select initial seeds from each cluster in a round-robin fashion.The purpose of clustering is to improve the diversity of initial seeds, however, the initial seeds are still randomly sampled, resulting in only few initial IDIs to be chosen. Aggarwal et al. [3] investigated the importance of the seed data used by their fairness testing technique SG. Their results showed that seed data based on the training data set allows for a higher number of IDIs generated than when using random seeds. To the best of our knowledge, no existing work tackles the problem of improving the quality of IDIs during the initial seeds phase. We argue that this phase serves as the foundation for the subsequent global and local generation and is therefore worth to investigate.

Global generation. Secondly, the algorithm performs a *global generation* phase to extend from initial seeds in order to cover the input domain $\mathbb{I}$ across a broad range.

AEQUITAS uniformly samples instances and applies a discrimination check to identify IDIs among them. SG creates a decision tree to approximate the machine learning model under test. Symbolic execution is used to explore the input domain. ADF adopts gradients to maximize the difference between the deep neural networks outputs of two similar instances. EIDIG increases the efficiency of ADF through a memorization technique.

Local generation. Following the global generation phase, the IDIs found are utilised for *local generation*, which explores their neighborhood for additional IDIs.

AEQUITAS assumes that IDIs are close to one another in a local domain. The intuition is to add small modifications to the current IDIs in order to find more IDIs locally. SG examines locality to evaluate if a small change in the input can influence the model's judgment. Depending on the minimal gradient absolute value on each attribute, ADF looks for more discriminatory occurrences

among the IDIs' neighbors. EIDIG reduces the number of gradient computations performed in ADF by exploiting prior knowledge of gradients.

2.2 SHAP Value

To gain a better understanding of machine learning models, SHapley Additive exPlanations (SHAP) [29] is often adopted. SHAP value is a popular black-box model-agnostic interpretable approach, which utilizes Shapley Values, a game theory-based method, to approximate and explain the relationship between the input instance and the output prediction. Moreover, SHAP values are calculated as a consistent measure of feature importance, which is also time-saving in terms of computation [19]. Specifically, given an input instance x, the model generates a prediction value $f(x)$, and a SHAP value is assigned to each feature of the instance. Formally, x_i denotes the feature i of the instance x, and x_{base} denotes the base value that is the mean of the target class for all instances. The output of the model prediction result $f(x)$ can be formulated as follows:

$$f(x) = x_{base} + \sum_{i=1}^{n} \mathcal{S}(x_i) \tag{2}$$

where $\mathcal{S}(x_i)$ is the SHAP value of the feature x_i. For example, we compute the SHAP value[4] of the previous examples x and x' (shown in Section 2.1) based on the decision tree model:

$$\mathcal{S}(x) : [-0.038, -0.014, 0.012, -0.191, -0.208, -0.260, 0.006,$$
$$-0.001, \textbf{-0.047}, 0.021, 0.005, -0.034, -0.008]$$
$$\mathcal{S}(x') : [0.105, 0.070, 0.107, -0.059, -0.191, -0.122, 0.001,$$
$$-0.001, \textbf{0.181}, 0.036, 0.016, 0.103, -0.003]$$

Note that $x_{base} = 0.757$, indicating the average confidence score of $y = 0$ predicted labels in the total instances. In this example, the sum of $\mathcal{S}(x)$ and x_{base} is 0 ($f(x)$), whereas the sum of $\mathcal{S}(x')$ and x_{base} is 1 ($f(x')$). From the SHAP values, we can determine that *gender* is the greatest positive feature (0.181) in $\mathcal{S}(x')$. This implies that it has the strongest positive relationship with the classification result of all attributes in the model. Meanwhile, we also observe that *gender* shows a negative relationship (-0.047) with the classification result in $\mathcal{S}(x)$.

In this work, the SHAP value is utilized to explain the machine learning model at the instance level. Other metrics measuring the feature importance (introduced in Section 7) can also be substituted for SHAP value to explain the machine learning models.

3 THE I&D APPROACH

In this section, we introduce our proposal, dubbed *I&D*, for initial IDI generation. Figure 3 gives an overview of our approach. First, *I&D* designs a novel **IDI initialization** component, which leverages a chiral model to effectively obtain initial IDIs close to the decision boundary of a classification model (Section 3.1). Then, *I&D* exploits a **diversity improvement** component, which combines the SHAP value and clustering algorithm to select more diverse IDIs from those obtained by the previous IDI initialization component (Section 3.2). Finally, *I&D* is integrated with the existing state-of-the-art approaches, such as AEQUITAS, SG, ADF, and EIDIG, by simply replacing their initial seed generator (based for example on random sampling or clustering) with the IDIs generated by *I&D* (*i.e.*, instead of using their own initial seeds, we feed each of these state-of-the-art approaches with the initial seeds generated by *I&D*).

[4]https://github.com/slundberg/shap

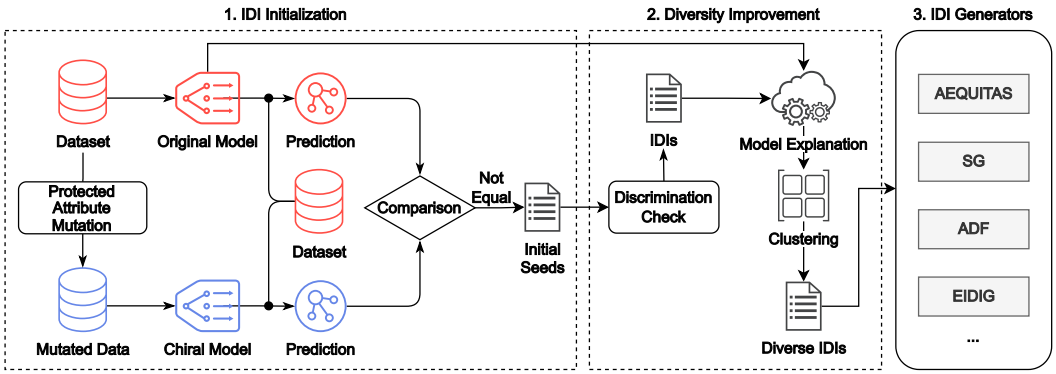


Fig. 3. Overview of our proposed approach *I&D*.

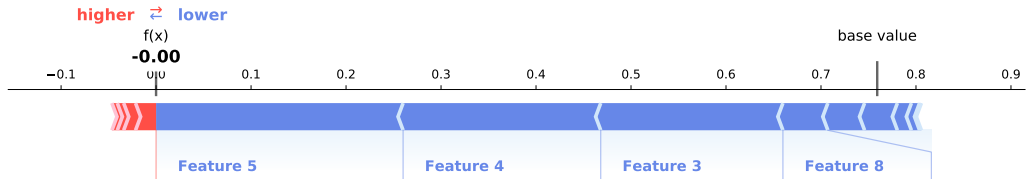
3.1 IDI Initialization

To generate initial seeds, we design a novel IDI initialization component based on a "chiral" model. Chirality is a feature of asymmetry that is important in many research areas [10], such as chemistry, mathematics, and biology. An object is chiral if it can be distinguished from its mirror image; that is, it cannot be superimposed onto it. Here we borrow this concept to mutate the protected attributes of a dataset and train a new model, *i.e.*, the chiral model, which should yield similar results as the original model. Ordinarily, one would assume for both models (original and chiral), to make the same predictions. However, due to non-deterministic training procedures of machine learning models, we are able to detect inconsistent/different predictions among the two models, which highlight that some of the instances are more difficult to predict and might be prune to cause inconsistent predictions.

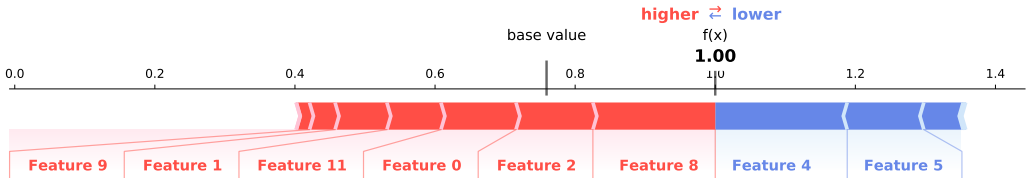
As shown in Figure 3, the IDI initialization component first builds a prediction model, named *original model*. Then, it mutates the protected attributes in the dataset to obtain *mutated data*. This is a simple step because we only need to alter the values of protected attributes and leave the rest of the attributes and labels unchanged. For binary protected attributes, one can simply flip the value from 0 to 1 and from 1 to 0. For other protected attributes, we modify their value from their input domain at random. Note that we choose random values rather than a permutation of all values in their input domain, because the mutated data should have the same size as the original dataset. Furthermore, because some attributes, such as age, have a broad variety of input domain, supplying every possible value for a protected attribute may lead to a combinatorial explosion.

After mutating the protected attribute, we utilize the mutated data to train a *chiral model* with the same structure and hyper-parameters as the *original model*. These two models are then used to predict the label of every instance in the dataset. If the prediction outputs are different, the instance is subject to a discrimination check method. The discrimination check method is conducted in accordance with the definition of an IDI. An instance is considered individually discriminatory if, when the values of its non-protected attributes remain unchanged but the values of its protected attributes are altered across all possible combinations, the predicted labels differ. Since *I&D* is not based on a specific machine learning model, *I&D* treats the machine learning model as a black box. Thereby, we do not only ensure that *I&D* finds error-prone, but also discriminative instances for the use as initial seeds. In our experiments, we also consider a set of all IDIs as initial seeds, without using chiral models to detect error-prune instances (see Section 4.3 for more details).

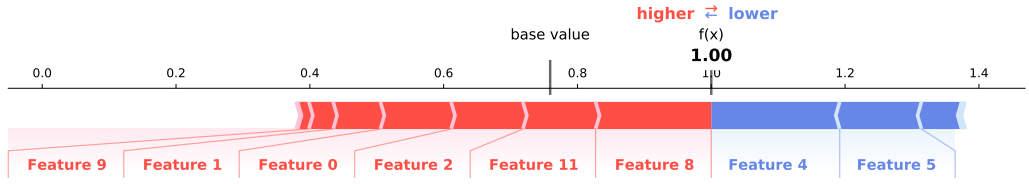
To gain insights into why the chiral model is applicable, we adopt the SHAP value to explain the predictions made by black box models. The SHAP value can intuitively demonstrate the contribution



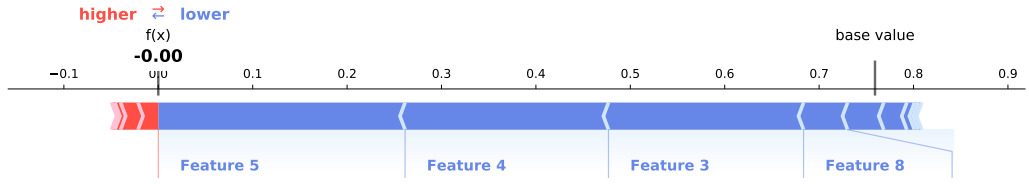
(a) x on original model



(b) x' on original model



(c) x on chiral model



(d) x' on chiral model

Fig. 4. In the original model, the SHAP value between x and x' is different. The SHAP value of x on the original model (a) is comparable to that of x' on the chiral model (d). (b) and (c) is similar, likewise. It's no surprise that the chiral model on the IDI looks like a mirror image of the original model.

of each feature (of input instances) to the final prediction result. Specifically, we compare the SHAP values for the same instance predicted by both the original model and the chiral model. Figure 4 illustrates the force plots of the SHAP values for a pair of IDIs (denoted as x and x'). In the axis above, the bold number (*e.g.*, **-0.00** in Figure 4(a)) is the predicted confidence of models. For our studied binary classification problem, higher predicted confidence leads the model to predict 1 and lower confidence leads the model to predict 0. Besides, the *base value* is the mean of the target class for all instances. The confidence values are sorted from left to right, with the lowest value being at the left-most position. Below the axis, the SHAP values of features are displayed, with red bars indicating positive contributions (that push the model confidence higher) and blue bars indicating negative ones (that push the model confidence lower). Features that have more of an impact on the predicted confidence are located closer to the dividing boundary between red and blue, and the size of that impact is represented by the size of the bar. As described in Section 2.2,

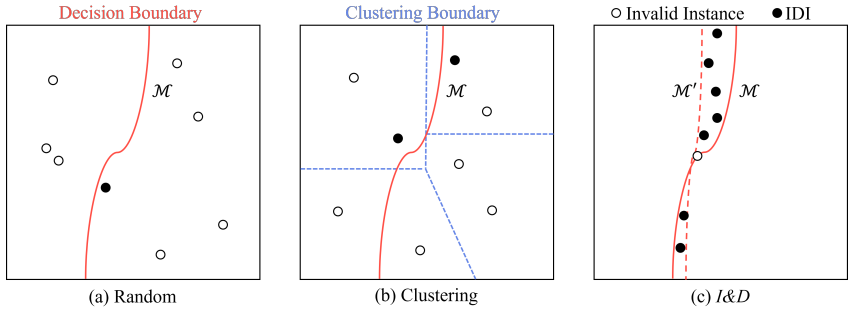


Fig. 5. Comparison of different initialization methods. (a) samples the entire input space for 8 instances; (b) divides the input space in four clusters and receives 2 instances from each; (c) illustrates *I&D*, which samples instances close to the decision boundary of the original model (solid) and the chiral model (dashed).

the actual prediction output is denoted by $f(x)$, which is the sum of all features' SHAP values plus the base value. From Figure 4(a) and Figure 4(b), we can observe that, given the pair of IDIs x and x', the original model has the different prediction outputs. When comparing the chiral model to the original model, the SHAP values of IDIs in the chiral model are almost identical to that of the original model. For example, Figure 4(b) is comparable to Figure 4(c). As a result, the chiral model's predicted output differs from the original output. Note that the SHAP value for x on the chiral model is not the same as x' on the original model. The rank of features in Figure 4(b) and Figure 4(c) differs. It is important to note that this mirrored relationship of SHAP values between the chiral and original models is consistent across the dataset and not confined to instances within specific decision boundaries. This consistency across the dataset suggests that the chiral model maintains the predictive dynamics of the original model while reflecting a chiral symmetry in feature contributions.

Qualitative evaluation. In the following, we provide the reader with a theoretical explanation of how *I&D* augments and improves existing IDI generation approaches. This is further supported by the empirical evaluation presented in Section 5.

Existing state-of-the-art or widely-used individual fairness testing approaches (e.g., AEQUITAS, ADF, SG, and EIDIG) typically rely on either simple random or clustering-based strategies for initial seed selection, without explicitly exploring how to enhance the quality of IDIs at the initialization stage. Specifically, the random strategy selects seeds indiscriminately from the original dataset, lacking focus on potential IDIs. While the clustering-based method improves the diversity of initial seeds, it still depends on random sampling within clusters, offering limited improvement in selecting rare IDIs. Given the extreme sparsity of IDIs in the original dataset, both strategies are inadequate for effectively retrieving them. Moreover, neither approach considers the quality of selected IDIs, that is, whether the selected seeds contribute meaningfully to improving model robustness.

As shown in Figure 5, we illustrate a sketch map of the three different IDI initialization approaches, namely Random initialization (shown in Figure 5(a)), Clustering-based initialization (shown in Figure 5(b)), and our proposed *I&D* initialization (shown in Figure 5(c)). Each box in this figure represents the data space in two dimensions. The solid red line denotes the decision boundary of the original model, while the dashed red line in Figure 5(c) represents the decision boundary of the chiral model. For the initial seeds, we chose eight instances, marked by circles. The black circles are IDIs after the discrimination check, while the white circles are data points without discrimination. Because IDIs have such a small proportion in the dataset, the random selection

approach is obviously difficult to capture [2]. The clustering-based method divides the data space into numerous groups and obtains seed instances in a round-robin fashion from each cluster. The purpose of clustering is to increase the diversity of initial IDIs. Because these cases represent distinct clusters of the dataset, these approaches may gain more IDIs than random sampling.

Unlike existing approaches, *I&D* takes into account both the dataset and the model. We develop a chiral model, indicated as $\mathcal{M}'$ in this figure, that has a different decision boundary than the original model. As a result, we can explicitly catch initial IDIs by distinguishing predicted outcomes from the two models. The IDIs found by *I&D* are thereby bound within the gap between the original model and the chiral model's decision boundary. Instances located near the decision boundary, where small changes in positive instances can turn them into negative instances, require special attention. IDIs from this region are more likely to reduce bias in the model as these instances are more prone to being misclassified. With retraining based on IDIs, the boundary can be made more robust to handle these instances better.

3.2 Diversity Improvement

A desired property of the initial seeds generation is to obtain a diverse set of IDIs. In this context, we use the diversity of IDIs to capture the diversity in predictions made by a machine learning model according to attribute feature importance (*i.e.*, we aim to find IDIs that occur for different reasons). In particular, we use SHAP values (Section 2.2) to determine the feature importance for predictions made for discriminative instances.

SHAP values are calculated as a consistent measure of feature importance, which is also time-saving in terms of computation [19]. The model generates a prediction value for each instance, and a SHAP value is assigned to each feature of the instance.

In accordance with existing approaches for generating diverse IDIs, we also follow a clustering-based procedure. However, these approaches group the original dataset directly and this practice might be ineffective because it does not take the prediction model into account. In fact, different models may have different decision-making strategies on the dataset, and the clustering boundary may not precisely overlap with the decision boundary, as shown in Figure 5 (b). Therefore, the diversity of IDIs may not be reflected effectively without considering the original model. On the other hand, it is challenging to directly assess the diversity of IDIs of machine learning models, since these models are often difficult to interpret, besides the testing of such models is desired to be model agnostic, considering them as a black box [3].

Therefore, we design a novel diversity improvement component, which combines the SHAP value and clustering algorithms. Specifically, we first use the SHAP value to explain the IDIs found in our designed IDI initialization component, which outputs the SHAP value score of each IDI. Then, based on their SHAP values, we use DBSCAN [38] to cluster these IDIs. DBSCAN is a powerful data density-based clustering algorithm that clusters IDIs without the need for specifying the target number of clusters [30]. IDIs in each cluster are then chosen in a round-robin process to obtain the higher-diversity IDIs until a specified number of initial seeds is reached. This upper limit (*i.e.*, search budget) on the number of initial seeds is a parameter adopted by previous studies [2, 49, 50]. If the budget set ends up to be larger than the number of IDIs actually generated, one can fill it up by using random sampling.[5]

[5]It is worth noting that our experiments (Section 5) show that this had not been needed since *I&D* was always able to identify more initial IDIs than those required by the search budget. In RQ3 (Section 5.3) we further evaluate the usefulness of this component for fairness testing and improvement.

Table 1. Datasets used in this study.

Dataset	# Rows	# Attributes	Protected Attribute
Census	48,842	12	Age, Race, Gender
German	1,000	24	Age, Gender
Bank	45,211	16	Age

We then integrate *I&D* with each existing IDI generation approach (namely AEQUITAS, SG, ADF, and EIDIG) by simply replacing their initial seeds with the IDIs generated by *I&D*, which makes our approach very straightforward to integrate with existing IDI generator.

4 EMPIRICAL STUDY DESIGN

In this section, we describe the design of the empirical study we carried out to assess the effectiveness of our proposal for initial seed generation, dubbed *I&D*.

Specifically, this study aims to address the following research questions (RQs):

- **RQ1:** How well does *I&D* perform when integrated with existing IDI generation methods?
- **RQ2:** How effective is *I&D* for testing different type of machine learning models?
- **RQ3:** How do the hyper-parameters of *I&D* influence its performance?

4.1 Datasets

To evaluate the use *I&D*, we employ three commonly used datasets in the software fairness literature [12, 13, 24, 49, 50]. We investigate the following three datasets (with additional information provided in Table 1):

- **Adult Census Income (Census)**.[6] Barry Becker retrieved this dataset from the 1994 Census database. Its objective is to forecast whether a person earns more than $50,000 per year based on their personal data.
- **German Credit (German)**.[7] This dataset classifies people described by a set of attributes as good or bad credit risks.
- **Bank Marketing (Bank)**.[8] This dataset comes from a Portuguese bank and is used to estimate if a customer would sign up for a term deposit based on their information.

We pre-process the datasets following the previous work ADF [50] by binning the numerical attributes. Also, we remove the feature that directly indicates the prediction label.

4.2 Machine Learning Models

Because ADF and EIDIG are only capable of generating IDIs for neural networks, we use the typical fully-connected neural network following the existing studies as the test model in RQ1 (Section 5.1), and RQ3 (Section 5.3). We choose hyper-parameters following their respective settings [49, 50]. The number of neural network layers is set to 6, the size of neural network hidden states is set to 30, 20, 15, 10, 5, and 1, respectively. During the training phase, we adopt the binary cross-entropy loss function, Nadam [15] as the optimizer, 30 training epochs, 128 batch size, and 0.01 learning rate.

[6]https://archive.ics.uci.edu/ml/datasets/adult
[7]https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)
[8]https://archive.ics.uci.edu/ml/datasets/bank+marketing

For RQ2 (Section 5.2), we build four widely-used machine learning models, *i.e.*, Logistic Regression (LR), Support Vector Machine Classifier (SVC), Decision Tree Classifier (DTC), and Multi-layer Perceptron Classifier (MLP) following AEQUITAS [2] based on Scikit-learn [32]. The hyper-parameters are set according to their respective models' default settings, which we believe have little impact on the results of our experiments because they can attain reasonably high performance (the average F1-score is about 89%).

4.3 Baseline

To show the improvement of *I&D* on existing fairness testing approaches, and the advantage of using a chiral model for selecting instances that are relevant for discrimination checks, we compare *I&D* against a baseline which performs a simpler strategy for selecting initial seeds. Instead of using the chiral model to filter for instances for which we apply a discrimination check in the datasets, we propose a baseline that considers all IDIs in the datasets for initial seeds. In accordance with the procedure of *I&D*, we first train the *original model*. Afterwards, we apply a discrimination check for each instance in the dataset, which inverts the protected attribute and compares predictions made by the *original model*. When the number of data instances is unrestricted, the resulting set of IDIs used as initial seeds in existing fairness testing approaches (AEQUITAS, SG, ADF and EIDIG) is a superset of the initial seeds selected by *I&D*. However, when the number of data instances is limited, the baseline can only select instances randomly, whereas *I&D* first filters instances using the chiral model. With this baseline approach, we aim to investigate whether the chiral model can provide IDIs of higher quality in contrast to simply selecting the IDIs from the data instances. In the following, we refer to this baseline as "revert". Although our study primarily focuses on initial seed selection for individual fairness testing, we also compare our *I&D* with a state-of-the-art seed selection technique from fuzzing, namely DiPri [35], to provide a more comprehensive evaluation. DiPri is a distance-based seed selection approach designed to enhance fuzzing effectiveness (e.g., improving code coverage). DiPri assigns a priority score to each seed based on both its distance to other seeds and its test coverage, and ranks seeds accordingly to form an optimal initial seed set. To adapt DiPri to the individual fairness testing, we remove its test coverage-based component.

4.4 Implementation

We implemented *I&D* in Python based on TensorFlow [1] and Scikit-learn [32]. The default ϵ (distance threshold) and the minimum sample size of DBSCAN is set to 0.09 and 10, respectively [38]. Moreover, we used the public implementations of four IDI generation approaches, *i.e.*, AEQUITAS [44], SG [3], ADF [50] and EIDIG [49].

We conducted all the experiments on a Linux server with Intel(R) Xeon(R) E5-2640 v4 @ 2.40GHz CPU, 128GB memory, and Ubuntu 18.04 as the operating system.

4.5 Fairness Testing Metrics

The approach proposed in this work aims to improve existing solutions for individual fairness. To assess the potential improvement of our approach to find initial IDIs, we combine *I&D* with four state-of-the-art IDI generation approaches, *i.e.*, AEQUITAS, SG, ADF, and EIDIG, respectively. We compare these approaches before and after the integration with *I&D*. In this study, we consider both effectiveness and usefulness as measures of performance. The count of IDIs has been widely-used in previous work [3, 44, 49, 50] to evaluate the performance of IDI generation for fairness testing, and thus we adopted the same metric. Although the count of IDIs is a single measure, it is used in two different manners.

The More the Better for Generation (↑). Our goal for IDI generation is to effectively generate a large number of IDIs given the computational constraints ("the more IDIs the better"). We set the limit of initial seeds, and the global and local generation limit both to 100. As a consequence, 100, 100 is the maximum number of the two-phase searched instances (100 global and 100*100 local instances). We also evaluate the influence of these limits in RQ3 (Section 5.3).

The Fewer the Better After Retraining (↓). Following previous work [44, 49, 50], for each IDI generation approach, we also use its generated IDIs to retrain the original model and then measure the fairness of the retrained model. We use 5% of the IDIs to augment the training dataset for retraining [49]. To determine the fairness of a model before and after retraining, we estimate the percentage of IDIs in the input space $\mathbb{I}$, which captures all possible combination of attribute values from $A = A_1, A_2, ..., A_n$. The lower the percentage of IDIs the better. Thus, fairness improvements are indicated by a decrease in the percentage of IDIs in the input space. In accordance with Zhang et al. [49], we sample 10,000 instances from $\mathbb{I}$, uniformly at random, to determine the percentage of IDIs. We repeat this procedure 5 times and report averaged results.

We follow the typical practice to evaluate the performance of these machine learning classification models based on the F1-score. Each dataset was split, with 60% of it serving as the training set and 40% as the test set. During the retraining phase, we add 5% of the generated IDIs into the training data and keep the parameters of the original model unchanged. The number of epochs is set to 10, because the loss is steady after retraining for 10 epochs. We experimented with each approach ten times and averaged the results to limit the impact of randomness. We observe that the F1-score of these models on the test set remain constant before and after retraining, indicating that these models are not overfitting on the IDIs. Following ADF and EIDIG, we use the notion of majority voting to determine the label of created IDIs based on the decisions of several models, *i.e.*, LR, SVC, DTC, and MLP.

Other metrics [11, 41] exist, but they are not well-suited for evaluating the IDI task. For instance, generalized entropy indices, such as the Theil index [41], though categorized as individual fairness metrics, focus on subjective perceptions of fairness. The Theil index attempts to capture how each individual perceives the fairness of their outcome by defining a benefit function based on the discrepancy between what the individual deserves and what the algorithm provides. Besides, metrics like GEI Theil index [22] are rarely used in practice and present challenges, such as the need to configure benefit values, which vary depending on the context and are often difficult to determine. Therefore, our study primarily focuses on standard metrics that are widely recognized and commonly used in fairness testing. These metrics are well-established in the literature, providing reliable, interpretable, and easy-to-implement indicators of model fairness.

5 EMPIRICAL STUDY RESULTS

In this section, we describe the results of the empirical study as designed in Section 4 to answer our three RQs.

5.1 RQ1: Performance of *I&D*

Overall Effectiveness of *I&D*. The performance comparison between existing IDI generation approaches and their corresponding variants with *I&D* is shown in Table 2. Note that "origin" refers to the original IDI generation approach (*i.e.*, AEQUITAS, SG, ADF, and EIDIG), "revert" denotes the combination of the original IDI generation approach with the revert baseline (as described in Section 4.3), "DiPri" denotes the combination of the original IDI generation approach with the DiPri baseline (as described in Section 4.3), and "*I&D*" represents the integration of the original IDI generation approach with our proposed *I&D*. In particular, *I&D* contains two main components:

IDI initialization component and diversity improvement component. To further investigate the contribution of each component, we constructed a variant of *I&D* that removes the diversity improvement component (*i.e.*, "*w I*"). Besides, Each column in this table represents the number of IDIs produced for each dataset and protected attribute.

According to these results, we find that all four IDI generation approaches with *I&D* achieve superior effectiveness compared to all three compared baselines (*i.e.*, origin, revert, and DiPri) for all datasets and protected attributes under study. The average number of generated IDIs across all original approaches is 1, 260, 1, 246 for original approaches combined with the revert baseline, and 1, 334 for original approaches combined with the DiPri baseline. In contrast, original approaches combined with *I&D* generate an average of 2, 342 IDIs. Specifically, the original approaches combined with *I&D* achieve average improvements of 1.86X, 1.88X, and 1.76X over the original approaches alone, as well as those combined with the revert and DiPri baselines, respectively. We also observe that among these studied IDI generation approaches, EIDIG with *I&D* can generate the highest number of IDIs (3, 420 on average). Besides, the number of IDIs generated by AEQUITAS with *I&D* is increased by 5.69X and 12.35X compared to AEQUITAS and AEQUITAS with revert, marking the largest improvement among all the four IDI generation approaches. One possible reason for the revert baseline's lower effectiveness is the lower quality of the selected IDIs, which limits the ability of AEQUITAS's global and local search algorithms to find additional IDIs in their neighborhood. These findings significantly demonstrate the effectiveness of our *I&D*.

We further investigate the contribution of both main components (*i.e.*, IDI initialization component and diversity improvement component) in *I&D*. We observe that "*w I*" also demonstrates superior effectiveness compared to all three baselines (*i.e.*, random, revert, and DiPri) across all datasets and protected attributes. Specifically, the original approaches generate an average of 1, 260 IDIs, while their combinations with the revert and DiPri baselines generate averages of 1, 246 and 1, 334 IDIs, respectively. In contrast, "*w I*" generates an average of 2, 076 IDIs, which is still lower than the 2, 342 generated by *I&D*. As a result, "*w I*" outperforms all three baselines (*i.e.*, original, revert, and DiPri) but is less effective than *I&D*. Overall, the results demonstrate that the both components are important for *I&D* to improve the effectiveness of IDI generation approaches.

Comparison of IDI Initialization Component. We further compare the IDI initialization component of *I&D* with the original IDI initialization methods used in IDI generation approaches: Random Sampling (as employed in AEQUITAS) and Clustering-based Sampling (utilized in SG, ADF, and EIDIG), as well as the IDI initialization components of revert and DiPri baselines. We set the limit of instances for each protected attribute to 1, 000 for the Adult and Bank datasets, and to 500 for the German dataset, as the dataset only contains 1, 000 instances. In particular, the revert baseline also employs a random sampling strategy to select 1, 000 instances prior to the comprehensive discrimination check, whereas our *I&D* utilizes the chiral model to select 1, 000 (or 500) instances. After obtaining 1, 000 (or 500) instances using each IDI initialization method, we apply a discrimination check to evaluate the IDI initialization rate (*i.e.*, the percentage of valid IDIs in these instances). The average IDI initialization rate is then calculated by repeating each experiment ten times. By comparing these IDI initialization methods, we aim to demonstrate that *I&D* can effectively identify individual discrimination before the discrimination check.

Table 3 presents a comparison of the IDI initialization rates among four IDI initialization methods: Random Sampling (denoted as *Random*), Clustering-based Sampling (denoted as *Clustering*), the IDI initialization component of the revert baseline (denoted as *revert*), the IDI initialization component of the DiPri baseline (denoted as *DiPri*), and the IDI initialization component of *I&D* (denoted as "*w I*"). First, the evaluation results show that "*w I*" achieves superior effectiveness compared to all the baselines in terms of IDI initialization rate for all datasets and protected attributes studied herein.

Table 2. RQ1-Effectiveness: Total number of generated IDIs by each of the existing approach with and without *I&D* (↑).

Approach		Census			German		Bank	Average
		Age	Race	Gender	Age	Gender	Age	
AEQ.	origin	359	271	56	1039	261	388	396
	revert	345	42	80	468	54	106	183
	DiPri	614	501	404	456	102	439	419
	w I	2982	861	1093	2951	1323	756	1661
	I&D	**3634**	**970**	**1613**	**3314**	**1530**	**2457**	**2253**
SG	origin	186	96	46	324	190	57	150
	revert	206	79	99	290	192	21	148
	DiPri	213	82	87	319	198	62	160
	w I	234	115	118	341	223	94	188
	I&D	**254**	**138**	**123**	**387**	**243**	**108**	**209**
ADF	origin	2640	883	693	3400	795	3321	1955
	revert	3216	660	656	3596	922	3525	2096
	DiPri	3379	973	801	3726	879	3682	2240
	w I	3941	**1537**	**1956**	4998	2744	3906	3180
	I&D	**4363**	1493	1933	**5150**	**3344**	**4636**	**3487**
EIDIG	origin	3788	1295	1103	3854	1082	4116	2540
	revert	3789	1202	1021	4458	1015	3855	2557
	DiPri	3593	1237	1390	3933	969	3972	2516
	w I	3974	1308	**1988**	4868	3185	4319	3274
	I&D	**4637**	**1408**	1869	**5030**	**3204**	**4371**	**3420**

* AEQ. is short for AEQUITAS.

In particular, the average IDI initialization rate of "*w I*" is 65.6%, while the other three approaches (*i.e.*, *Random*, *Clustering*, *Revert*, and *DiPri*) are just 8.1%, 9.3%, 9.2%, and 10.7%, respectively. In addition, the IDI initialization rate for "*w I*" ranges from 41.5% to 78.3% across all datasets and protected attributes, suggesting that *I&D* has consistent high initialization effectiveness. This is because *Clustering* and DiPri accounts for the characteristics of the dataset, whereas *Random* and *Revert* do not. In contrast, "*w I*" considers the characteristics of both the model and the dataset. In fact, we train a chiral model by mutating the protected attributes of the dataset. By comparing the original and chiral models, we can explicitly extract individual discrimination before the discrimination check. Therefore, existing IDI initialization methods that do not account for model characteristics do not perform well for IDI initialization. The IDI initialization component of our *I&D* is the first attempt to overcome this problem, and our findings indicate that it is a promising approach.

The last row in Table 3 shows the running time of *Random*, *Clustering*, *Revert*, *DiPri*, and "*w I*". We observe that *Random* and *Revert* require the least amount of time (0.001 seconds), *Clustering* requires 0.002 seconds, *DiPri* requires 0.859 seconds, whereas our "*w I*" takes 2.692 seconds on average. This can be explained by the training of the chiral model, which is as expensive as training the original model. In our case, additional time consumption is within the range of a few seconds,

Table 3. RQ1-Initialization: The IDI initialization rate (↑) and running time (↓) comparison based on 1000 selected instances.

Dataset	Protected Attribute	Random	Clustering	Revert	DiPri	w I
Census	Age	13.8%	16.4%	14.0%	16.5%	**78.0%**
	Race	2.9%	4.3%	4.0%	3.9%	**70.3%**
	Gender	3.3%	3.8%	6.6%	3.4%	**78.3%**
German	Age	18.8%	19.4%	19.1%	25.8%	**61.6%**
	Gender	8.2%	8.8%	8.2%	9.6%	**63.8%**
Bank	Age	1.8%	3.1%	3.0%	4.8%	**41.5%**
Average		8.1%	9.3%	9.2%	10.7%	**65.6%**
Running Time (s) (↓)		**0.001**	0.002	**0.001**	0.859	2.692

but is also acceptable for larger models, as the training needs only be performed once, which can be done offline.

Usefulness of *I&D*. To show the usefulness of *I&D*, we leverage the generated IDIs to retrain the original model to mitigate the bias and investigate whether retraining leads to a fairness improvement. Specifically, we use 5% of the IDIs generated by each approach to augment the training dataset for model retraining. To measure the fairness of a machine learning model, we sample the entire input space $\mathbb{I}$, uniformly at random, to determine the percentage of IDIs when making predictions with the model (see Section 4.5 for more details). This is performed before and after retraining with the IDIs found before (see Table 2).

Table 4 illustrates these results. The "*Before*" row indicates the IDI percentage for each dataset and protected attribute before retraining, proceeding rows show the IDI percentage after retrained with the corresponding approach (*i.e.*, AEQUITAS, SG, ADF, and EIDIG) and initial seed strategy (*i.e.*, the original approach, the revert baseline, the DiPri baseline, and our *I&D*). The average F1-score on the test set remains similar over all approaches, ranging from 87.6% to 89.2%, which indicates that the retrained models are not overfitting on the generated IDIs. In particular, the average percentage of IDIs remained is 8.3% for the original approaches, 8.6% for approaches with revert, and 8.2% for approaches with DiPri, while IDI percentage of approaches with *I&D* is 6.3%, representing reductions of 24.9%, 27.3%, and 23.2%, respectively. Therefore, *I&D*, on average, is better than the baseline approaches.

Furthermore, we investigate the contribution of each component in *I&D*. For the first IDI initialization component component, we compare "*w I*" with the original, revert, and DiPri baselines. To compare these approaches, we investigate the number of remaining IDIs after retraining the neural network model by them. More specifically, "*w I*" has 1.1%, 1.4%, and 1.0% fewer remaining IDIs on average across all datasets than the original, revert, and DiPri baselines, respectively. This result indicates that the IDI initialization component in *I&D* can improve the model's fairness due to the higher-quality IDIs obtained by the chiral models. For the second diversity improvement component, we further compare *I&D* with "*w I*". From these results, we observe that *I&D* has fewer remaining IDIs than "*w I*" in almost all cases (with only four exceptions due to the potential randomness of experiments). For the exceptional cases, "*w I*" just has slightly 0.4% fewer remaining IDIs than *I&D* on average. In particular, on all other cases, *I&D* has 1.2% fewer remaining IDIs than

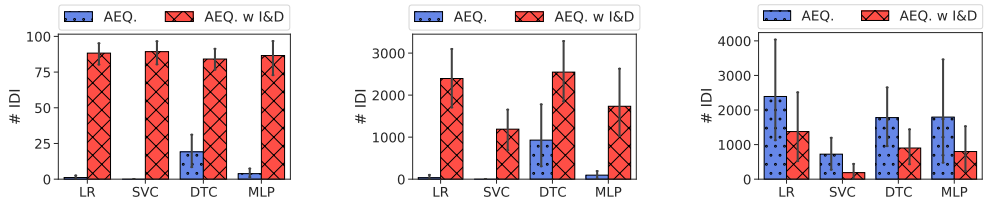
Table 4. RQ1-Usefulness: Percentage of discriminative instances in the input space $\mathbb{I}$, before and after retraining (↓). The best configuration for each approach and dataset is highlighted in bold. Variance over 5 runs is reported (±).

Approach		Census			German		Bank	F1-score
		Age	Race	Gender	Age	Gender	Age	
Before		14.6±0.3	13.2±0.2	6.2±0.1	24.9±0.3	8.3±0.2	11.8±0.3	88.3
AEQ.	origin	15.6±0.2	10.9±0.2	8.2±0.1	6.4±0.1	4.3±0.1	6.1±0.2	87.6
	revert	11.4±0.3	12.6±0.2	5.5±0.2	8.0±0.2	6.6±0.2	5.0±0.1	87.8
	DiPri	11.6±0.2	9.9±0.2	5.9±0.2	6.4±0.1	5.6±0.3	5.1±0.2	87.8
	w I	10.9±0.2	8.6±0.2	3.1±0.2	6.8±0.2	3.5±0.2	4.6±0.2	87.8
	I&D	**9.6±0.1**	**8.0±0.3**	**2.8±0.2**	**4.2±0.1**	**3.1±0.2**	**4.5±0.1**	88.0
SG	origin	10.1±0.2	10.5±0.1	5.8±0.2	11.7±0.2	9.2±0.3	12.0±0.2	88.0
	revert	14.2±0.2	10.3±0.1	7.9±0.2	16.1±0.2	8.2±0.3	11.2±0.3	88.3
	DiPri	13.7±0.2	11.7±0.1	6.6±0.1	14.3±0.3	8.3±0.1	11.4±0.2	88.0
	w I	12.8±0.3	11.3±0.1	4.2±0.2	11.8±0.2	6.0±0.2	11.0±0.1	87.9
	I&D	**8.0±0.2**	**9.3±0.1**	**4.1±0.2**	**9.1±0.2**	**5.5±0.1**	**10.5±0.3**	88.1
ADF	origin	10.5±0.2	8.8±0.3	4.2±0.1	6.7±0.3	4.4±0.2	6.8±0.1	89.2
	revert	11.1±0.2	7.5±0.2	5.3±0.1	4.8±0.2	5.9±0.3	7.0±0.3	89.0
	DiPri	11.3±0.1	8.1±0.2	3.8±0.1	4.6±0.2	5.7±0.2	6.7±0.1	89.0
	w I	8.9±0.2	**7.2±0.2**	**2.9±0.1**	**4.3±0.2**	6.7±0.1	6.5±0.2	89.1
	I&D	**7.7±0.2**	7.9±0.3	3.2±0.1	4.4±0.1	**4.1±0.2**	**5.6±0.2**	89.0
EIDIG	origin	13.1±0.2	9.5±0.2	5.3±0.1	6.7±0.2	**4.4±0.1**	8.7±0.3	88.9
	revert	12.7±0.2	10.2±0.1	5.0±0.1	6.4±0.2	6.3±0.1	7.2±0.2	88.6
	DiPri	12.8±0.2	9.7±0.1	4.7±0.2	7.1±0.3	5.7±0.2	6.9±0.3	88.9
	w I	12.7±0.2	8.6±0.3	**3.6±0.1**	5.7±0.2	4.9±0.2	6.7±0.2	89.0
	I&D	**11.4±0.1**	**8.3±0.2**	4.1±0.2	**4.5±0.1**	5.1±0.1	**5.1±0.1**	89.1

* AEQ. is short for AEQUITAS.

"*w I*" on average across all the datasets and IDI generation approaches. This result indicates that *I&D* can improve the model's fairness due to a diverse set of IDIs. Overall, the results demonstrate that the both components are important for *I&D* to improve model's fairness.

Answer to RQ1: The use of *I&D* improves the average number of IDIs by 1.86X for AEQUITAS, SG, ADF, and EIDIG, as opposed to not using *I&D*. Furthermore, after retraining with the IDIs generated by *I&D*, the percentage of IDIs in the input space $\mathbb{I}$ is decreased by 24.9% on average, implying that *I&D* is effective for improving the model's fairness. Besides, both the IDI initialization and diversity improvement components make contributions to the overall effectiveness of *I&D*, demonstrating the necessity of each of them.



(a) Global Generation (↑) (b) Local Generation (↑) (c) After Retraining (↓)

Fig. 6. RQ2: Comparison of number of IDIs generated (a, b) and remaining after retraining (c) using different models. Results are averaged over the three datasets and protected attributes. AEQ. is short for AEQUITAS.

5.2 RQ2: Testing Different Models

To further investigate the effectiveness of *I&D* with different machine learning models, we choose representative AEQUITAS with *I&D* for experiments because SG is not efficient [50] and the ADF and EIDIG approaches are both aimed only at neural network models.

The results of AEQUITAS with or without using *I&D* for different machine learning models are shown in Figure 6. Specifically, Figure 6(a) and Figure 6(b) illustrate the global generation and local generation comparison, respectively. From these figures, we can observe that AEQUITAS with *I&D* significantly outperforms the original AEQUITAS for all four models (*i.e.*, LR, SVC, DTC, and MLP). The average number of IDIs generated by AEQUITAS with *I&D* during the global generation phase is 87, compared to just 6 with the original AEQUITAS, representing a 14.50X increase. Note that the initial seeds are used directly in the global generation phase of AEQUITAS, meaning that the IDI initialization component of *I&D* is more effective than the random sampling. In terms of the local generation phase, the average number of IDIs generated by AEQUITAS with *I&D* is 1, 967, whereas the original AEQUITAS is 266, an improvement of 7.39X. In particular, the LR model has the highest number of IDIs improvement (87 in the global phase and 2, 359 in the local phase on average) among all models, while the SVC model has the smallest number of IDIs improvement (89 in the global phase and 1, 190 in the local phase on average). The simplest of these four machine learning models is LR, which takes into account the linear relationship between features and the prediction objective.

While *I&D* is able to increase the number of IDIs generated by AEQUITAS in the global and local generation phases, by 14.50X and 7.39X respectively, we note that the increase in the global generation phase is almost twice as high as in the local phase. A potential reason for this can be seen in the creation of duplicated IDIs without *I&D*, as the local search explores the neighborhood of fewer seeds.

The number of remaining IDIs after model retraining is shown in Figure 6(c). Note that the retraining procedure is similar to that described in Section 5.1. According to this figure, the number of retraining IDIs has decreased dramatically from AEQUITAS with *I&D* to the original AEQUITAS, regardless of models. Specifically, after retraining using IDIs generated by the original AEQUITAS, the IDIs remained are 1, 673, whereas with *I&D* is just 816, a 51.2% decrease. The LR model has the greatest number of IDIs drop (1, 014 on average) among all models, whereas the SVC model has the least number of IDIs decline (534 on average).

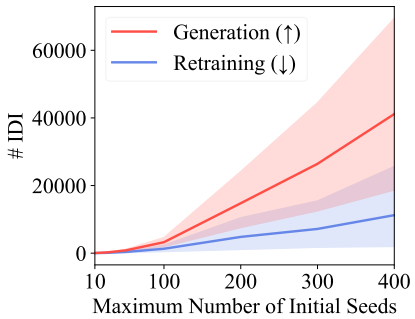


Fig. 7. RQ3: The average number of generated IDIs (red) and the average number of remaining IDIs after retraining with the same IDIs (blue) by AEQUITAS with *I&D* under different values for the maximum number of initial seeds.

Answer to RQ2: Regardless of the machine learning model, *I&D* can significantly enhance the IDI generation approach to generate more IDIs, with an improvement of 14.50X on average. Moreover, *I&D* can also help models further reduce the number of IDIs after retraining, with a 51.2% reduction on average.

5.3 RQ3: Hyper-parameter Sensitivity

The only hyper-parameter of *I&D* is the maximum number of initial seeds. Intuitively, the number of generated IDIs depends on the maximum number of seeds for the generating phases, which include initial seeds, global and local generation limits [44, 49]. In line with previous work [44], we set the number of initial seeds and global generation limits to be the same. This is because the global generation phase utilizes these initial seeds as input (see Figure 2). If the maximum number of initial seeds is less than the global generation limit, random sampling would be needed to fill the gap, which would not accurately evaluate our *I&D*. Conversely, if the maximum number of initial seeds exceeds the global generation limit, only a subset would be used, also leading to an inaccurate evaluation of hyper-parameter sensitivity. Therefore, we set identical limits for initial seeds, global and local generation (consistent with RQ1). To answer RQ3, we examine the average number of generated IDIs and the average number of remaining IDIs after retraining with the same IDIs, as the maximum number of initial seeds ranges from 10 to 400 (see Figure 7). The average number of IDIs generated by AEQUITAS with *I&D* is shown by the red line with error bound. The blue line depicts the average number of remaining IDIs after retraining with the same IDIs. Note that since both model retraining and post-retraining testing use the same generated IDIs, the maximum number of remaining IDIs cannot exceed the number of generated IDIs. Consequently, the blue line will always be below the red line. The number of generated IDIs increases dramatically, because the search space expands exponentially as the limit rises (from 1010 to 160400). The number of remaining IDIs increases slower than the number of generated IDIs because retraining the model with more IDIs can reduce bias even further. We do not set a higher limit because the generating time of the AEQUITAS also rises exponentially as the limit increases. When the limit is 400, it will take more than 14 hours to execute IDI generation on a single protected attribute. *I&D*, on the other hand, is efficient in handling higher limits. Given the limit is 1000, it only takes 2.692 seconds, as shown in Table 3. In the future, we will further explore the effect of varying one parameter

while keeping others constant (*e.g.*, the number of initial seeds, global generation limit, and local generation limit).

Answer to RQ3: The maximum number of initial seeds, as well as global and local generation limits, influence the final number of generated IDIs. As the limits increase, the number of generated IDIs and generation time increase dramatically, whereas the number of remained IDIs after retraining increase slower.

6 THREATS TO VALIDITY

We evaluated *I&D* with three datasets. They are the most common public benchmarks used in the fairness testing literature. However, further datasets could be considered for future work to strengthen our results. Furthermore, the generated IDIs lack ground-truth labels and rely on voting from multi-model prediction outputs [26]. When retraining the model with more produced labels than its initial dataset size, the model may not be helpful. Moreover, *I&D* is open-sourced and dataset-independent. If more datasets become accessible in the future, it will be simple to expand our analysis.

The machine learning models used to conduct our experiments are also used in prior studies, such as AEQUITAS [44] and ADF [50]. Because the datasets are simple, with a maximum of 16 features, basic models like fully-connected deep neural networks can handle them. Our approach, on the other hand, is broad and does not rely on any specific models. Since the main idea of *I&D* is to train a chiral model and compute SHAP values to cluster, which is straightforward to implement even for more sophisticated models.

7 RELATED WORK

7.1 Fairness in Software Engineering

Fairness is a critical non-functional testing property of data-driven applications and machine learning software [48]. As such, it has received an increasing attention from both the software engineering [9, 13, 23, 47] and machine learning research communities [6, 43]. Among others, Brun *et al.* [9] named this "software fairness" and called for software engineers to combat such discrimination and build fair software. Since then, fairness concerns have been addressed in different stages of the software development process [40]. In addition to software testing [14], fairness with regards requirements [18] and the design of fair algorithms [13] have been investigated. Fair design approaches have been introduced in various stages of the development process, including pre-processing, in-processing, and post-processing [8, 12, 31, 47]. Fairkit [22, 27], AI Fairness 360 [43], and Fairea [24] aim to mitigate bias. Our study complements these methods for detecting and mitigating bias.

7.2 Individual Discriminatory Testing

Various types of approaches have been proposed for fairness testing of machine learning models in the past few years. THEMIS [5, 20] first defined software fairness testing in terms of individual discrimination. However, THEMIS is inefficient because it relies on random sampling without generating IDIs. AEQUITAS [44], SG [3], ADF [50] and EIDIG [49] are a series of IDI generation frameworks. A detailed description of these approaches is presented in Section 2.1.

To allow for an interpretability for discrimination found in deep neural networks, Zheng et al. [51] proposed the fairness testing framework NeuronFair. NeuronFair is able to detect biased neurons in the different layers of deep networks by measuring the differences in their activation

for instances with different protected attribute values. A high difference in neuron activation for such instances indicates that a neuron is biased.

Another possibility for generating IDIs, is the use of search-based methods, such as genetic algorithms [17, 33]. For example, Fan et al. [17] first used model explanation techniques (see Section 7.3) to find initial seeds, which are then used as a starting point for a genetic algorithm to find IDIs. Perera et al. [33] used genetic algorithms for fairness testing when dealing with a regression based problem (*i.e.*, predicting waiting time for emergency departments). Given an initial population of randomly generated instances, the genetic algorithm was designed to find instances which maximizes the difference in predictions for instances that only differ in protected attributes.

They concentrate on generating IDIs effectively and efficiently in the global and local generation phases but overlook the importance of the seed selection phase for improving fairness testing results. To fill this gap, in this work we have proposed a novel way to generate initial seeds, which is able to further improve IDIs effectiveness and diversity. Our work is orthogonal to previous work, as it can be applied to existing IDIs generator by simply replacing their own initial seed generation strategy.

7.3 Model Explanation

In addition to SHAP [29] there exist other model explanation techniques. For example, Local Interpretable Model-agnostic Explanation (LIME) illustrates the model with a decision tree-like structure. Decision rules [4] are approaches that are easily understood by humans. However, they are only useful when they have human-reasonable size. In this work, we require an explanation approach at instance level to calculate feature importance [21, 52]. The Shapley value [36] is the foundation for various approaches that credit a machine learning model's prediction on an instance to its underlying features. The Shapley value can be calculated using a variety of algorithms [42], from which we choose the SHAP value [29], since it is capable of efficiently explaining a wide range of models.

8 CONCLUSION

Fairness testing can be used to detect individual discriminating instances (IDIs) and asses an AI system's fairness. In this paper, we have proposed a novel initialization approach for fairness testing, *I&D*, to aid in the initial phase of IDI generation. *I&D* compares the prediction output between the original model and a chiral model and uses the SHAP value to improve the diversity of IDIs. The usefulness of *I&D* was demonstrated through an empirical study on three widely-used datasets for fairness testing research. The average number of IDIs generated by using our *I&D* approach achieves improvements of 1.86X and exceeds that of the existing approaches. Furthermore, we discover that by utilizing the generated IDIs to retrain the model and test IDIs again, the remaining IDIs are reduced by 24.9%, thus outperforming other approaches. We also show how the fairness of widely-used models like Logistic Regression, Support Vector Machines, Decision Trees, and Neural Networks can be improved by using *I&D*. The contributions of the key components are also supported by our experiments. Overall, the results show that the initial seed phase is an important step in the fairness testing procedure, for increasing the number of generated IDIs and proceeding fairness improvements, and should receive more attention.

In future work, we aim to investigate additional methods for selecting initial seeds for comparisons with the chiral model. While the focus of *I&D* is individual fairness, in particular counterfactual fairness (*i.e.*, treating individuals with different protected attributes equally) [11], one could investigate the impact of fairness testing and retraining on other fairness definitions (*e.g.*, group fairness metrics and non-classification tasks). Moreover, *I&D* could be applied to datasets of different domains (*e.g.*, textual [17] and visual [51]).

ACKNOWLEDGMENTS

Max Hort and Federica Sarro are supported by the ERC Advanced fellowship grant no. 741278 (EPIC). Hongyu Zhang is supported by the Australian Research Council (ARC) Discovery Projects (DP200102940, DP220103044)

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX symposium on operating systems design and implementation (OSDI)*. ACM, 265–283.

[2] Aniya Agarwal, Pranay Lohia, Seema Nagar, Kuntal Dey, and Diptikalyan Saha. 2018. Automated test generation to detect individual discrimination in AI models. *CoRR, arXiv preprint arXiv:1809.03260* (2018).

[3] Aniya Aggarwal, Pranay Lohia, Seema Nagar, Kuntal Dey, and Diptikalyan Saha. 2019. Black box fairness testing of machine learning models. In *Proceedings of the 27th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. ACM, 625–635.

[4] Rakesh Agrawal, Ramakrishnan Srikant, et al. 1994. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, Vol. 1215. Citeseer, 487–499.

[5] Rico Angell, Brittany Johnson, Yuriy Brun, and Alexandra Meliou. 2018. Themis: Automatically testing software for discrimination. In *Proceedings of the 26th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. ACM, 871–875.

[6] R. Berk, H. Heidari, S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth. 2017. A Convex Framework for Fair Regression. *FAT-ML Workshop* (2017).

[7] Sumon Biswas and Hridesh Rajan. 2020. Do the machine learning models on a crowd sourced platform exhibit bias? an empirical study on model fairness. In *Proceedings of the 28th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. ACM, 642–653.

[8] Sumon Biswas and Hridesh Rajan. 2021. Fair Preprocessing: Towards Understanding Compositional Fairness of Data Transformers in Machine Learning Pipeline. In *Proceedings of the 29th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. ACM, 981–993.

[9] Yuriy Brun and Alexandra Meliou. 2018. Software fairness. In *Proceedings of the 26th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. ACM, 754–759.

[10] Robert S Cahn, Christopher Ingold, and Vladimir Prelog. 1966. Specification of molecular chirality. *Angewandte Chemie International Edition in English* 5, 4 (1966), 385–415.

[11] Simon Caton and Christian Haas. 2024. Fairness in machine learning: A survey. *Comput. Surveys* 56, 7 (2024), 1–38.

[12] Joymallya Chakraborty, Suvodeep Majumder, and Tim Menzies. 2021. Bias in Machine Learning Software: Why? How? What to do?. In *Proceedings of the 29th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. ACM, 429–440.

[13] Joymallya Chakraborty, Suvodeep Majumder, Zhe Yu, and Tim Menzies. 2020. Fairway: A way to build fair ml software. In *Proceedings of the 28th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. ACM, 654–665.

[14] Zhenpeng Chen, Jie M Zhang, Max Hort, Federica Sarro, and Mark Harman. 2022. Fairness Testing: A Comprehensive Survey and Analysis of Trends. *arXiv preprint arXiv:2207.10223* (2022).

[15] Timothy Dozat. 2016. Incorporating nesterov momentum into adam. (2016).

[16] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, 214–226.

[17] Ming Fan, Wenying Wei, Wuxia Jin, Zijiang Yang, and Ting Liu. 2022. Explanation-Guided Fairness Testing through Genetic Algorithm. In *Proceedings of the 44th International Conference on Software Engineering*.

[18] Anthony Finkelstein, Mark Harman, S Afshin Mansouri, Jian Ren, and Yuanyuan Zhang. 2009. A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making. *Requirements engineering* 14, 4 (2009), 231–245.

[19] Alexandre Fréchette, Lars Kotthoff, Tomasz Michalak, Talal Rahwan, Holger Hoos, and Kevin Leyton-Brown. 2016. Using the shapley value to analyze algorithm portfolios. In *Proceedings of the 30th Conference on Artificial Intelligence (AAAI)*, Vol. 30. ACM.

[20] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. 2017. Fairness testing: testing software for discrimination. In *Proceedings of the 25th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. ACM, 498–510.

[21] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *Computing Surveys (CSUR)* 51, 5 (2018), 1–42.

[22] Max Hort, Zhenpeng Chen, Jie M Zhang, Federica Sarro, and Mark Harman. 2022. Bias Mitigation for Machine Learning Classifiers: A Comprehensive Survey. *arXiv preprint arXiv:2207.07068* (2022).

[23] Max Hort and Federica Sarro. 2021. Did You Do Your Homework? Raising Awareness on Software Fairness and Discrimination. In *Proceedings of the 36th International Conference on Automated Software Engineering (ASE)*. IEEE.

[24] Max Hort, Jie M Zhang, Federica Sarro, and Mark Harman. 2021. Fairea: a model behaviour mutation approach to benchmarking bias mitigation methods. In *Proceedings of the 29th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. ACM, 994–1006.

[25] Nasif Imtiaz, Justin Middleton, Joymallya Chakraborty, Neill Robson, Gina Bai, and Emerson Murphy-Hill. 2019. Investigating the effects of gender bias on GitHub. In *Proceedings of the 41st International Conference on Software Engineering (ICSE)*. IEEE, 700–711.

[26] Heinrich Jiang and Ofir Nachum. 2020. Identifying and correcting label bias in machine learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 702–712.

[27] Brittany Johnson, Jesse Bartola, Rico Angell, Katherine Keith, Sam Witty, Stephen J Giguere, and Yuriy Brun. 2020. Fairkit, Fairkit, on the Wall, Who's the Fairest of Them All? Supporting Data Scientists in Training Fair Models. *arXiv preprint arXiv:2012.09951* (2020).

[28] Wei-Yin Loh. 2011. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1, 1 (2011), 14–23.

[29] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*. ACM, 4768–4777.

[30] Minghua Ma, Shenglin Zhang, Junjie Chen, Jim Xu, Haozhe Li, Yongliang Lin, Xiaohui Nie, Bo Zhou, Yong Wang, and Dan Pei. 2021. Jump-Starting Multivariate Time Series Anomaly Detection for Online Service Systems. In *Proceedings of the USENIX Annual Technical Conference (ATC)*. USENIX, 103–114.

[31] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *Computing Surveys (CSUR)* 54, 6 (2021), 1–35.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[33] Anjana Perera, Aldeida Aleti, Chakkrit Tantithamthavorn, Jirayus Jiarpakdee, Burak Turhan, Lisa Kuhn, and Katie Walker. 2022. Search-based fairness testing for regression-based machine learning systems. *Empirical Software Engineering* 27, 3 (2022), 1–36.

[34] Hung Viet Pham, Shangshu Qian, Jiannan Wang, Thibaud Lutellier, Jonathan Rosenthal, Lin Tan, Yaoliang Yu, and Nachiappan Nagappan. 2020. Problems and opportunities in training deep learning software systems: an analysis of variance. In *Proceedings of the 35th International Conference on Automated Software Engineering (ASE)*. ACM, 771–783.

[35] Ruixiang Qian, Quanjun Zhang, Chunrong Fang, Ding Yang, Shun Li, Binyu Li, and Zhenyu Chen. 2024. Dipri: Distance-based seed prioritization for greybox fuzzing. *ACM Transactions on Software Engineering and Methodology* 34, 1 (2024), 1–39.

[36] Alvin E Roth. 1988. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press.

[37] Warren S Sarle. 1994. *Neural networks and statistical models*. Citeseer.

[38] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. 2017. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *Transactions on Database Systems (TODS)* 42, 3 (2017), 1–21.

[39] Ezekiel Soremekun, Mike Papadakis, Maxime Cordy, and Yves Le Traon. 2022. Software Fairness: An Analysis and Survey. https://doi.org/10.48550/ARXIV.2205.08809

[40] Ezekiel Soremekun, Mike Papadakis, Maxime Cordy, and Yves Le Traon. 2022. Software Fairness: An Analysis and Survey. *arXiv preprint arXiv:2205.08809* (2022).

[41] Till Speicher, Hoda Heidari, Nina Grgic-Hlaca, Krishna P Gummadi, Adish Singla, Adrian Weller, and Muhammad Bilal Zafar. 2018. A unified approach to quantifying algorithmic unfairness: Measuring individual &group unfairness via inequality indices. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2239–2248.

[42] Mukund Sundararajan and Amir Najmi. 2020. The many Shapley values for model explanation. In *Proceedings of International Conference on Machine Learning (ICML)*. PMLR, 9269–9278.

[43] Trusted-AI. 2021. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. (2021). https://github.com/Trusted-AI/AIF360.

[44] Sakshi Udeshi, Pryanshu Arora, and Sudipta Chattopadhyay. 2018. Automated directed fairness testing. In *Proceedings of the 33rd International Conference on Automated Software Engineering (ASE)*. ACM.

[45] Zan Wang, Ming Yan, Junjie Chen, Shuang Liu, and Dongdi Zhang. 2020. Deep learning library testing via effective model generation. In *Proceedings of the 28th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. ACM, 788–799.

[46] Sanford Weisberg. 2005. *Applied linear regression*. Vol. 528. John Wiley & Sons.
[47] Jie M Zhang and Mark Harman. 2021. "Ignorance and Prejudice" in Software Fairness. In *Proceedings of the 43rd International Conference on Software Engineering (ICSE)*. IEEE, 1436–1447.
[48] Jie M Zhang, Mark Harman, Lei Ma, and Yang Liu. 2020. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering* (2020).
[49] Lingfeng Zhang, Yueling Zhang, and Min Zhang. 2021. Efficient white-box fairness testing through gradient search. In *Proceedings of the 30th International Symposium on Software Testing and Analysis (ISSTA)*. ACM, 103–114.
[50] Peixin Zhang, Jingyi Wang, Jun Sun, Guoliang Dong, Xinyu Wang, Xingen Wang, Jin Song Dong, and Ting Dai. 2020. White-box fairness testing through adversarial sampling. In *Proceedings of the 42nd International Conference on Software Engineering (ICSE)*. ACM, 949–960.
[51] Haibin Zheng, Zhiqing Chen, Tianyu Du, Xuhong Zhang, Yao Cheng, Shouling Ji, Jingyi Wang, Yue Yu, and Jinyin Chen. 2022. Neuronfair: Interpretable white-box fairness testing through biased neuron identification. In *Proceedings of the 44th International Conference on Software Engineering*. 1519–1531.
[52] Alexander Zien, Nicole Krämer, Sören Sonnenburg, and Gunnar Rätsch. 2009. The feature importance ranking measure. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 694–709.