Black-Box Fairness Testing with Shadow Models

Weipeng Jiang[1,2], Chao Shen[1,2(✉)], Chenhao Lin[2]
Jingyi Wang[3], Jun Sun[4], and Xuanqi Gao[2]

[1] State Key Laboratory of Communication Content Cognition,
People's Daily Online, Beijing, China 100733
[2] Xi'an Jiaotong University, China
lenijwp@stu.xjtu.edu.cn, chaoshen@xjtu.edu.cn,
linchenhao@xjtu.edu.cn, gxq2000@stu.xjtu.edu.cn
[3] Zhejiang University, China
wangjyee@zju.edu.cn
[4] Singapore Management University, Singapore
junsun@smu.edu.sg

Abstract. Discrimination in decision-making systems is of growing concern as machine learning techniques (especially deep learning) are increasingly applied in systems with societal impact. Multiple recent works have proposed to identify/generate discriminative samples through fairness testing. State-of-the-art fairness testing methods can efficiently generate many discriminative samples, which can be subsequently used to improve the fairness of the model. Unfortunately, the applicability of these approaches is limited in practice as they require the availability of both the model and the training data, i.e., a white-box setting. In a black-box setting (e.g., testing online services), existing approaches are impractical for multiple reasons, e.g., they require huge testing budgets. In this work, we propose a black-box fairness testing approach for neural networks, namely BREAM, which addresses two challenges, i.e., how to generate many discriminative samples without querying many times and how to guide the searching without the original model. Our overall idea is to obtain approximate gradients by training shadow models to effectively guide the discriminative sample generation for black-box DNNs. We also observe the density diversity of the distribution of discrimination, which enables incremental maintenance of shadow models and rational allocation of search resources by dividing multiple subspaces. We evaluated BREAM on three widely adopted datasets for fairness research. The results show that BREAM achieves a 9X higher performance than existing black-box methods, comparable to the state-of-the-art white-box fairness method.

Keywords: Security and privacy of AI · Machine Learning · Fairness.

1 Introduction

Deep neural networks (DNNs) have achieved incredible performance in many applications, such as face recognition [31], self-driving car [6] and vulnerability

detection [21]. Although DNNs have shown great potential, there are also multiple concerns about their dependability and trustworthiness. In particular, fairness property is of rising concern as many DNN applications may have societal impact [20] in domains like justice [4, 17], finance [27] and advertisements [33]. Unfortunately, since societal bias is often deeply rooted in the training data, the resultant DNNs might be discriminative even unintentionally [35].

Intuitively, (individual) discrimination of a DNN means that the model makes different decisions on two samples that differ only by certain features of societal impact (such as `race`, `gender`, and `religion`). These specific attributes are referred to as protected attributes or features [9]. In the machine learning community, multiple lines of work have been proposed aiming to mitigate discrimination of machine learning models either in the data pre-processing stage [14, 16, 42] or the training stage [22, 40] which have been shown to be effective to some extent. However, after a DNN is trained, the fairness requirement of the system should still be properly tested. Even better, the testing results should be able to serve as diagnosis information for mitigating the discrimination in the original model.

There exist multiple fairness testing approaches to identify or generate discriminative samples [3, 15, 18, 37, 41, 43]. For instance, Galhotra *et al.* [18] proposed THEMIS which tests the fairness of a given DNN model by randomly sampling the input domain. Udeshi *et al.* [37] designed AEQUITAS which automatically generates discriminative samples around the training samples. Aggarwal *et al.* [3] developed a method called Symbolic Generation (a.k.a. SG) which searches for more specific discrimination for a particular sample with local explanation. Fan *et al.* [15] propose ExpGA, fusing explanation tools with genetic algorithms to generate discriminative samples for both tabular data and text data. White-box method ADF [43] and EIDIG [41] guide the search of discrimination based on the gradients, i.e., the direction incurring a maximum change in the outputs of the DNN, which are shown to be much more effective and efficient in identifying discrimination than the previous black-box approaches.

However, in reality, many AI applications are provided as resource-constrained black-box APIs, for example, the human resource service provided by HrFlow [2]. It is unlikely that we are allowed to query the target system many times, i.e., such systems are often built to prevent denial-of-service attack [24, 32] or model extraction attack [5, 26, 28, 36, 38, 39]. Or rather, there is a charge for each query, thus fairness testing is subject to budget constraints. The white-box approach is not applicable in this case, and the black-box approach nowadays does not specifically consider the limit of the number of queries. The question is then: *how can we develop an efficient black-box fairness testing approach (without access to the model and the training data) within limited queries?*

In this work, we aim to develop such a **b**lack-box fai**r**ness t**e**sting approach with sh**a**dow **m**odels, namely BREAM, which answers the above question positively. BREAM requires minimal knowledge of the DNN under test. That is, we assume that only the input/output pairs of the DNN can be acquired. Under such a black-box setting, BREAM mainly addresses two important technical challenges. Firstly, how can we effectively guide the search for discriminative

samples? Our remedy is to train shadow models using model extraction techniques [28] and guide the search for discrimination based on the gradients of the shadow models. Secondly, how can we effectively train such shadow models given that we have no access to the training data and are only allowed to query the DNN a small number of times? We propose to first obtain a small number of labeled samples by querying the black-box model and train an initial shadow model. Afterward, we split the whole sample space into multiple sub-spaces and train multiple shadow models for the sub-spaces separately. The motivation is that training a shadow model for a subspace is much easier than training on the entire space, with only a limited number of labeled samples. Note that we additionally assign different weights for the subspace as we observe that discrimination is often unevenly distributed. Once we have the shadow models for different subspaces, we could utilize the gradients of the shadow models to guide the search of discrimination intuitively.

BREAM has been implemented as a self-contained toolkit and evaluated with multiple datasets widely adopted by previous studies. Experimental results show that BREAM could generate discriminative samples much more (by an order of magnitude) effectively than existing black-box approaches and is almost (99% on average) as effective as the state-of-the-art white-box approach. Furthermore, BREAM ensures the diversity of identified discrimination by exploring different subspaces which are shown to be more valuable in mitigating the discrimination through retraining the model. In a nutshell, we make the following contributions to this work.

* We propose to effectively address the fairness testing problem of black-box DNN with no access to the training data and limited query budget by adopting shadow model training and guided search with approximate gradients.
* We observe the uneven distribution of discrimination in the input space and propose a smart sampling strategy based on the trained shadow models to identify discrimination while ensuring diversity.
* We evaluate BREAM with multiple benchmark. Our experiments show that BREAM is significantly more effective and efficient than previous black-box fairness testing methods, and even achieves similar effectiveness as the state-of-the-art white-box approach.

2 Background

Discriminative Samples. The individual fairness for DNNs means that a DNN model should output the same label for two individuals that differ only by certain sensitive protected attributes such as gender. We denote by X as a set of containing all possible input samples and A= $\{A_1, A_2, ..., A_n\}$ as all attributes. A DNN model is a function that takes a feature vector $x \in X$ as the input and outputs a label y. We define P $\subseteq$ A as the protected attributes set and NP $\subset$ A as the non-protected attributes set. Besides those, we assume the valuation domain is I_i for each attribute A_i, which means the input domain is $I = I_1 \times I_2 \times ... \times I_n$.

Definition 1 *Let D represent a DNN model. For any $x = (x_1,x_2,...,x_n) \in X$ and $x' = (x'_1,x'_2,...,x'_n) \in X$. The (x, x') is a pair of discriminative samples with respect to D if and only if the following conditions are satisfied:*
(1) $\exists p \in P, x_p \neq x'_p$
(2) $\forall np \in NP, x_{np} = x'_{np}$
(3) $D(x) \neq D(x')$

Jacobian-based Model Extraction and Adversarial Attack. It is a black-box model stealing and adversarial attack method proposed by Nicolas *et al.* in [28]. Firstly, an initial training set should be self-collected because the dataset of the target system is always unavailable. And the architecture of the shadow model is selected by experience and high-level knowledge of the classification task. Then, it repeats the following steps for several rounds:

- *Labeling.* For each sample in the training set, get the label by querying the target model.
- *Training.* Based on the selected architecture and the labeled training set, train a shadow model.
- *Augmentation.* Apply the augmentation technique on the current training set to produce a larger shadow training set. Concretely, for each sample, take a perturbation along the direction of its sign of the Jacobian matrix on the shadow model.

Then adversarial samples are generated based on the shadow model, which can also achieve a successful adversarial attack on the target model with a high probability. This approach provides a solution for performing other operations on black-box models except for adversarial attacks, such as fairness testing. This study indicates that the architecture of the shadow model has a limited impact on the effectiveness of adversarial attacks when the shadow model can behave well on the classification task, and the method can achieve a successful adversarial attack with a few queries.

3 Methodology

BREAM is designed to efficiently generate discriminative samples when only predicted labels are available. BREAM focuses on decision systems where the input is tabular data. As depicted in Figure 1, BREAM requires an API of the target model and the specification of input feature and protected attribute ranges (i.e. which feature is the sensitive/protected attribute users concerned, and possible values). BREAM establishes a database by querying the target model to collect all labeled samples. First, BREAM trains a shadow model by querying the target model and then performs a two-stage generation process utilizing proximity information (i.e. gradient) from the shadow model. As the algorithm is executed, multiple shadow models are trained to perform testing separately, based on the uneven distribution of discriminative samples. Finally, BREAM outputs a list of generated discriminative samples. The BREAM algorithm is described in detail as Algorithm 1.

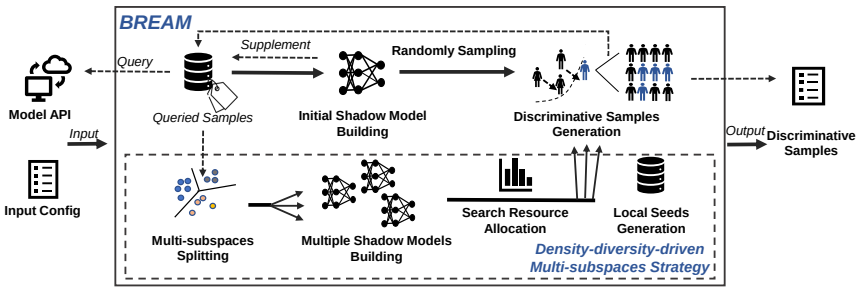


Fig. 1: An Overview of BREAM

3.1 Initial Shadow Model Building

The lack of effective guidance presents a major obstacle to the efficiency of black-box testing, as the decision logic and gradient information of the target model are unknown. To address this problem, we propose a method that constructs a shadow model using a few randomly selected seed samples and Jacobian-based data augmentation (lines 4-10), inspired by the black-box adversarial attack method proposed by Papernot et al. [28]. Specifically, we randomly sample a few seed samples and query the target model to obtain their labels. We then train a shadow DNN model using the self-collected training dataset. We perform Jacobian-based data augmentation on the training dataset and repeat the above process to improve the quality of the shadow model. The structure of the shadow model is empirically determined and is relatively simple. By using this shadow model, we can obtain an approximate estimation of the decision logic and gradient information of the target model.

3.2 Discriminative Samples Generation

A series of white-box methods [41, 43] have amply demonstrated the surprising effectiveness of using gradients to guide the discovery of discriminative examples. Based on the shadow model, we can obtain approximate gradient information of the target black box model. Thus, we design a shadow-model-driven approach to search and generate discriminative examples. As shown in Figure 2, the generation consists of two stages: the global stage and the local stage. It is worth emphasizing that we detect discrimination by accessing the target model to obtain the predicted label (as marked in `red`), and at all other times accessing the shadow model to obtain approximate gradient (as marked in `blue`). The purpose of global generation is to gradually move the seed samples closer to the decision boundary, where there is a greater likelihood of prediction difference due to weak perturbations of protected attributes. First, a sample will be checked for discrimination, i.e., by enumerating its protected attributes to check if there are inconsistent predictions. If discriminative, it is sent directly to the local stage; otherwise, it will be calculated the gradient. When computing the gradient, a

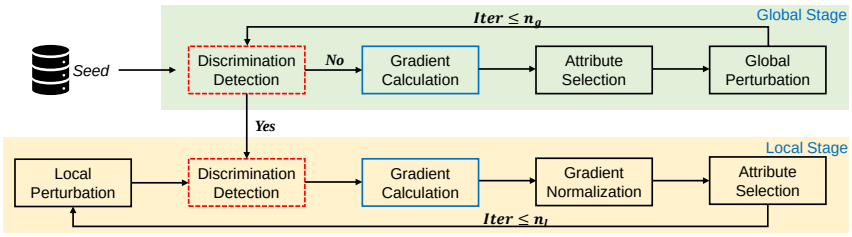


Fig. 2: Pipeline of Two-stage Generation

Table 1: Density Distribution of Discriminative Samples.

Dataset	Prot.Attr.	Density			
		Clu.1	Clu.2	Clu.3	Clu.4
Census	*gender*	0.045	0.023	0.005	0.009
Census	*race*	0.094	0.039	0.007	0.003
Census	*age*	0.149	0.081	0.014	0.008
Bank	*age*	0.219	0.091	0.078	0.037
Credit	*age*	0.240	0.196	0.122	0.075
Credit	*gender*	0.082	0.051	0.039	0.024

pair of (x, x') that differ only in protected properties will be computed simultaneously. Further, based on this pair of gradients, a small perturbation is applied in the direction of the same sign to obtain a new sample. The process is repeated until discrimination is found or the number of iterations is capped. Global generation searches across the entire sample space and aims to increase diversity, while local generation aims to further exploit globally found discriminative samples. In the local stage, the gradient is used to select those non-protected attributes that have less impact on the decision to be perturbed. The idea behind this is that a pair of discriminative examples remain unchanged in their predictions after being perturbed, i.e., they remain discriminative. To ensure diversity, we do not choose the attribute with the smallest gradient, but calculate a probability distribution for the choices. The probability distribution is specifically obtained by normalizing the inverse of the gradient.

3.3 Density-diversity-driven Multi-subspaces Strategy

Density Diversity of Discriminative Samples. To enhance the performance of the discriminative samples generation, we aimed to explore the distribution of discriminative samples by investigating the likelihood of their existence in different locations in the entire sample space. Imbalance or bias in the training data can lead to individual fairness deficiencies in models, and we assumpt that the proportion of discriminative samples may vary across different subspaces. To validate our assumption, we conduct an empirical study using three fairness-related datasets and several protected attributes (details provided in §4). We

Algorithm 1: BREAM

Input: f,$conf$
Parameter: n_{ini}, n_{aug}, n_g, n_l, γ, n_f
Output: A list of samples.

Let $Iters \leftarrow zero$
while $QueryTimes <= Limits$ **do**
  if $Iters$ is 0 **then**
    Let $D \leftarrow n_{ini}$ randomly samples
    Let $L \leftarrow \emptyset$
    for i in [0,1,...,n_{aug}] **do**
      $LetL \leftarrow f(D)$
      Let $M \leftarrow$ Train a shadow model on D and L
      Perform Jacobian-based dataset augmentation on D
    end for
    Let $Base \leftarrow$ Number of present labeled data by querying f
    while New labeled data $< Base$ **do**
      Generation(M,Random Seeds)
    end while
  else
    Let $Cnum \leftarrow 2 * Iters$
    $D_0, D_1, ..., D_{Cnum-1} \leftarrow$ Perform K-Means on D into $Cnum$ Clusters
    for Each cluster D_i **do**
      Let $M_i \leftarrow$ Train a shadow model on D_i
      Let $Den_i \leftarrow$ Proportion of discriminative samples in D_i
      Let $Seed_i \leftarrow$ Generate local seeds for D_i
    end for
    Let $N \leftarrow$ Normalsize Den and perform an integerization
    Let $Base \leftarrow$ Number of present labeled data by querying f
    while New labeled data $< Base$ **do**
      for i in [0,1,...,$Cnum - 1$] **do**
        Generation(M_i,$Seed_i$,N_i)
      end for
    end while
  end if
  $Iters \leftarrow Iters + 1$
end while
return discriminative samples

performed large-scale random sampling and applied K-Means clustering [25] to divide the samples into four clusters, each representing a local subspace. We then calculate the proportion of discriminative samples in each subspace to measure density. Our results, presented in Table 1, show that the density of discriminative samples varies significantly across different subspaces. We refer to this uneven distribution as density diversity.

Multiple Shadow Models Building. With the generation process described above, the number of labeled samples increases gradually, enabling us to con-

struct an improved shadow model. To achieve this, we propose dividing the input space into multiple subspaces to train separate shadow models rather than retraining the initial shadow model. This approach allows us to estimate the potential density of discriminative samples in each subspace and adjust the frequency of generating samples in each subspace accordingly, allocating more resources to high-density areas. Additionally, since the majority of labeled samples are generated during individual discriminative sample generation, the resulting dataset may be unbalanced, which can negatively impact the performance of a single shadow model trained on the dataset. By focusing on local subspaces, we can mitigate this issue. It is also worth noting that our approach generates discriminative samples that are likely to be near decision boundaries, which has been shown to improve the accuracy of trained shadow models in previous research [5, 39]. Based on the above idea, we make an update iteration whenever the number of labeled samples doubles. In each iteration, we first cluster the labeled dataset into multiple clusters using the K-Means method. We utilize currently known samples directly without additional sampling to minimize unnecessary queries. The number of clusters increases with each iteration (line 16). For each cluster, we train a shadow model on its samples (line 19). Although multiple subspaces are divided, the total number of samples is also increasing, so it is expected that the training samples for each model will not be too thin and thus will not cause very serious overfitting.

Search Resource Allocation. Next, we allocate more resources to subspaces with a higher density of discriminative samples. We use the proportion of discriminative samples in each cluster (line 20) as an estimate of the potential density in the subspace it covers. To allocate more resources to subspaces with a high potential density, we normalize the density and assign search weights to different spaces (line 23). To address zero-density situations, we apply Laplace smoothing [7] during normalization. The normalized results are converted to integers by multiplying by a small factor, which can be used directly as the number of iterative rounds of the generation process. We achieve a complete allocation of search resources through iterative loops (lines 25-29).

Local Seeds Generation for Each Subspace. Another challenge in the discriminative sample generation process based on the shadow model is to generate suitable seeds for local shadow models. Random sampling is not feasible for local models as the seeds must be within the approximate coverage of the corresponding local shadow model. Moreover, directly using samples from the clusters is not ideal either, as it can lead to overfitting and a lack of diversity. To address this, we propose a local seed generation strategy. Specifically, for each cluster, we apply a moderate perturbation to each sample and add them to the new seed set with probability γ, generating new samples within the effective functional area of each shadow model. Additionally, for the old samples, we retain them with probability γ^2, allowing unexplored samples to be considered.

Failure-rate-triggered Early Terminating. To address the density diversity and improve the local phase in the generation process, we introduce a control parameter, n_f, as the threshold of failure times. If the local generation fails frequently and the number of failures reaches a threshold value, we terminate the local iteration early to reduce unnecessary queries to the target model. This design is effective because frequent failures suggest a low density of discriminative samples in the local subspace.

4 Experiments

4.1 Dataset and Experimental Settings

We choose two popular black-box methods AEQUITAS [37] and SG [3], and one of the state-of-the-art white-box methods, ADF [43], for baseline comparison. We re-implement existing methods based on the source code used by ADF from Github [44], and make the following two improvements to achieve a fair comparison: firstly, we record the query history to avoid duplicate queries to the target model for each method; secondly, we change the global generation and local generation in AEQUITAS and ADF to alternate execution in the same way as BREAM, to facilitate the control of the same number of queries. Here we use random samples as seeds for all black-box methods for a fair comparison. While for the white-box method ADF, we use original training data as seeds because we suppose it as a reference upper bound. THEMIS is not used for comparisons, since it is shown to be less effective [18]. We do not evaluate ExpGA [15], because it is similar to SG in that it exploits local interpretability (which we will discuss later because it is a very resource-unfriendly way to query limitation). Table 2 shows the value of the main parameters set of BREAM in our experiments. Some not mentioned parameters during generation phases are the same as the ADF. For all baseline methods, we adopt the default parameters or the best strategy used in their original papers, (except for n_g and n_l in ADF and AEQUITAS, which are consistent with BREAM). Notice that SG does not take into account the limited number of queries, so the local explanation phase may cost a huge number of queries. The default number of locally sampling in SG is 2000. In order to trade off the cost of queries and the accuracy of local explanation, we choose a relatively small number as twice the input dimension, if it is further reduced, we believe it is insufficient to support building decision trees.

Following baseline works, we choose the same three open-source datasets from [12] to evaluate our approach. The details of the three adopted datasets:

* *Census* [10]: This dataset is used to predict whether the income of an adult is above $50,000. It contains over 32,000 pieces of data with 13 attributes. We focus on its three protected attributes *age*, *gender*, and *race*.
* *Credit* [13]: A small dataset with 600 data classifies people described by 20 attributes as good or bad credit risks. The protected attributes we are concerned with include *age* and *gender*.

Table 2: Configuration of experiments.

Parameter	Value	Description
n_ini	500	number of initial samples
n_aug	2	iteration of data augmentation
n_g	10	max.iteration of global generation
n_l	200	max.iteration of local generation
γ	0.6	probability of save seed samples
n_f	160	threshold of failure times

Table 3: Target DNN models.

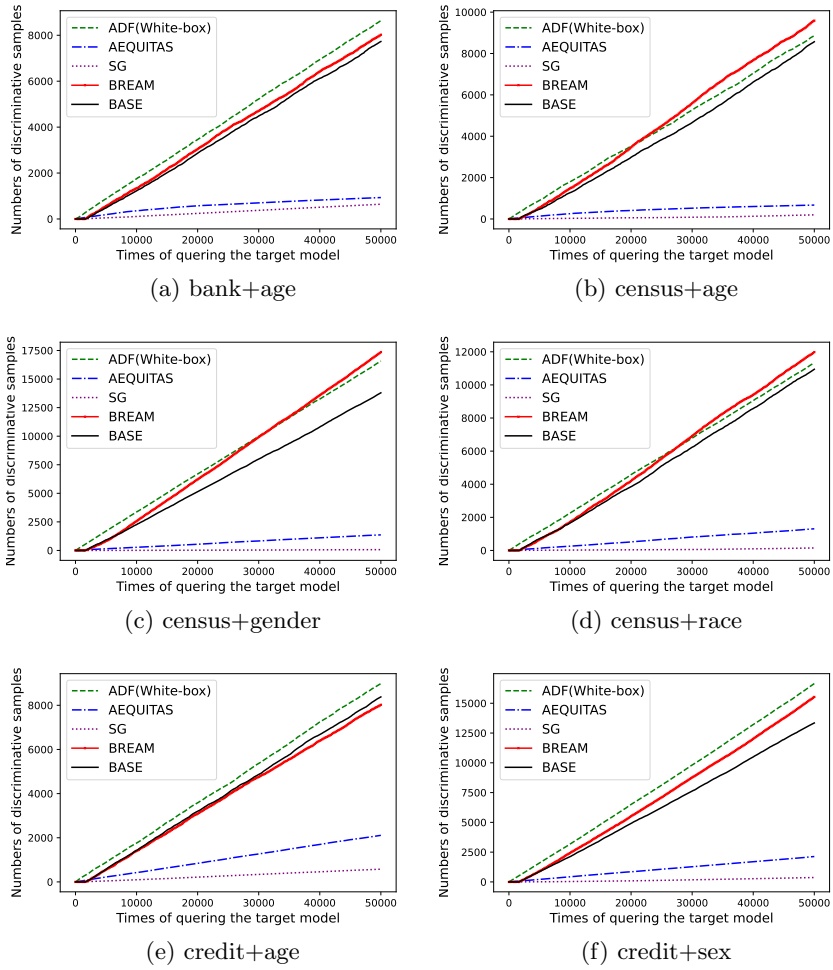
Dataset	Pieces of Data	DNN model	Accuracy
Census	32561	Six-layer FC	88.3%
Credit	600	Six-layer FC	99.3%
Bank	45211	Six-layer FC	93.9%

* *Bank* [11]: The dataset contains over 45,000 samples with 16 attributes. It is collected by a Portuguese banking institution and used to train models predicting whether customers will subscribe to a term deposit. The only protected attribute is *age*.

We apply the same data pre-processing and selection of target models for a fair comparison. Table 3 shows details of target DNN models in our experiments. Note the accuracy is evaluated over the data set. Based on the above models, we conduct a series of experiments. We filter out duplicate samples and record the discriminative samples. Previous SG generated 500,000 samples for quantitative experiments. Recall that our study focuses on scenarios with a limited number of queries to the target model, so we reduce the threshold of queries as 50,000 and argue that this is sufficient to fully demonstrate the performance of those approaches. To reduce random effects, all our experimental results are the average of five runs. We implement our approach based on Keras [8]. We conduct our experiments on a Server with one Intel Xeon E5-2620 2.10GHz CPU and Ubuntu 16.04 operating system.

4.2 Effectiveness and Efficiency

We systematically measure the number of discriminative samples (i.e. `NDS`) generated as the number of queries rises for different methods. Note the structure of shadow models taken in BREAM is as M1 in Table 4. Results are shown in Figure 3. It can be observed that the BREAM achieves a significant improvement over AEQUITAS and SG. Besides, all experimental results show that the efficiency of BREAM is close to that of the white-box method ADF, and even better in some experiments, as Figure 3b, 3d, 3c. To test the effectiveness of our proposed density-diversity-driven multi-subspaces strategy, we set a



(a) bank+age (b) census+age

(c) census+gender (d) census+race

(e) credit+age (f) credit+sex

Fig. 3: Comparison with existing methods on the efficiency of fairness testing. The horizontal axis represents the number of times to query the target model, and the vertical axis represents the number of discriminative samples generated. Note the BASE is used as a reference in the ablation experiment to verify the effectiveness of the multi-subspaces strategy.

Table 4: Structures of different shadow models.

Names	Hidden Layers	Neurons Configuration
M1	2	(64, 16)
M2	3	(64, 32, 8)
M3	3	(64, 16, 4)

comparison method that retraining the initial model at the same moment with BREAM and keeps other parts unchanged. It is named as `BASE` in Figure 3. It can be found that BREAM make an incremental performance on than BASE in most experiments. The achievement of surpassing ADF also comes precisely from the enhancements brought by this multi-subspaces strategy. Of course, the counterexample shown in Figure 3e can not be ignored, and we believe this may be due to the relatively small number of unfair instances in sum, leading to a premature overdraft of search on some subspaces. It can be observed that the experiments shown in Figure 3e, 3a have fewer discriminative samples for all approaches than others, and BREAM performs relatively the worst.

In order to represent this result more quantitatively, we count the number of discriminative samples generated by 50,000 times queries to quantify our results in Table 5. It manifests that the BREAM achieves an efficiency not inferior to the ADF on three benchmarks. And this is achieved with the premise that there are some extra queries used on initial sampling in the BREAM. This means that, in practice, we can perform efficient fairness testing with a few queries for specific black-box DNN systems.

We average the improvements in the efficiency of BREAM over 6 benchmarks as an overall measure. BREAM identifies 9.3 times of discriminative samples as much as AEQUITAS on average, which is the existing state-of-the-art black-box method under this limit scenario. Also, its effectiveness arrives 99% of ADF. BREAM meets our expectations to approach the level of the state-of-the-art white-box method. Additionally, it can be observed that SG behaves poorly under the limited times to query the target model. The result is in line with our expectations because it needs to sample a lot and build a local decision tree before generating each time. Unlike this, BREAM only costs a small number of queries at the very beginning. The results indicate that focusing on the limited black-box task is meaningful, considering that current black-box studies do tend to overlook a few limits.

4.3 Threats from Structures of Shadow Models

Although BREAM shows remarkable performance, there may be a measure of the impact of the structures and parameters of shadow models on effectiveness. In our approach, the structures and parameters are determined artificially by experience and prior knowledge. In addition, the structures and parameters of shadow models can be relatively simple as we think, since they only need to be fitted over a small subspace. We believe that their effects are limited if the shadow

Table 5: Comparison on numbers of discriminative samples generated with 50,000 queries to the target model. W refers to the white-box approach.

Dataset	Prot.Attr.	NDS			
		AEQ.	SG	ADF(W)	BREAM
Bank	*age*	932	642	**8637**	8022
Census	*age*	672	199	8860	**9589**
Census	*gender*	1362	70	16576	**17357**
Census	*race*	1307	148	11345	**11981**
Credit	*age*	1609	577	**8984**	8024
Credit	*sex*	1304	369	**16653**	15531

Table 6: Comparison of numbers of discriminative samples generated by BREAM with different shadow models.

Dataset	Prot.Attr.	Fluc.	NDS		
			M1	M2	M3
Bank	*age*	6.4%	8022	7726	8238
Census	*age*	9.0%	9589	8755	9397
Census	*gender*	2.1%	17357	17743	17569
Census	*race*	4.5%	11981	11455	11900
Credit	*age*	2.9%	8024	8117	8258
Credit	*sex*	2.8%	15531	15518	15097

model works, because behaviors of DNN models with different architectures are proven to have transferability [28, 34].

In order to estimate this potential threat to the effectiveness of BREAM from an empirical perspective, We conduct some experiments. Here we use three different structures of shadow models to run the BREAM separately. Details about these models can be found in Table 4. The neuron configuration shows the number of neurons per layer of the neural network. M1 is which one we used in the above comparison experiment. Table 6 shows the behavior of these models. The `Fluc.` represents the fluctuation of effectiveness. We formalize this fluctuation by the ratio of the extreme difference to the mean. It can be observed that the fluctuation is capped at 10%. On average, the metric value is only 4.6%.

4.4 Time Performance

We measure the time needed to generate a single new discriminative sample for all methods. In detail, we count the time and number of discriminative samples and calculate the average time required to discover a new discriminative sample. The results are shown in Table 7. It can be seen that ADF has the best efficiency in finding discriminative samples in terms of time complexity, BREAM is on the next. Although BREAM takes some time to train shadow models, it

Dataset	Prot.Attr.	Time(ms)			
		AEQ.	SG	ADF(W)	BREAM
Bank	*age*	591	319	33	58
Census	*age*	528	734	34	47
Census	*gender*	244	2343	48	102
Census	*race*	323	1155	44	99
Credit	*age*	763	449	59	98
Credit	*sex*	485	745	82	125

Table 7: Time(ms) for BREAM to generate a new discriminative sample. W refers to the white-box approach.

finds far more discriminative samples than AEQUITAS and SG, which leads to better efficiency in time complexity. Quantitatively, in terms of time complexity, BREAM spends 78% less time than AEQUITAS and 87% less time than SG. Besides, it costs 79% more time than ADF.

5 Related Work

Fair Machine Learning Classifiers In the field of machine learning, designing and training fair classifiers that avoid discrimination [14, 16, 19, 23] has become important, since fairness is a wide demand for people in the real world. These previous studies focus on achieving fairness from theoretical aspects by preprocessing training data and modifying existing classifiers. Our study is aimed to discover the discrimination in the DNN-model-based classifiers and help improve fairness.

Fairness Testing Systematic testing and validation of the fairness of machine learning models from the software engineering aspect are still in its infancy. This is also what we want to discuss in this paper. Galhotra *et al.* proposed THEMIS [18] which firstly defines the software fairness and discrimination and fairness measurement metrics, then gives a causality-based algorithm to evaluate the fairness of models by randomly sampling and calculating the frequency of discriminative samples. THEMIS is generally inefficient since it is based on random sampling without any guidance. Then Udeshi *et al.* proposed AEQUITAS [37], which designs a two-phase generation method. It first runs the global generation by random sampling to find some discriminative samples and then starts the local generation to perturb faintly to obtain more discriminatory instances with a greedy strategy, which are motivated by the robustness of models. In [3], Galhotra *et al.* developed a black-box method called Symbolic Generation (a.k.a SG), which firstly generates a decision tree by local explanation tools such as LIME [29] to approximate the DNN model decision then performs symbolic execution with the decision tree to generate test samples, and it repeats the above process to find more discrimination. Recently, Zhang *et al.* proposed a white-box fairness testing method ADF [43] based on adversarial sampling, which also contains a global generation phase and a local generation phase. It samples from

training data and generates the discriminative sample by making a given input progressively closer to the decision boundary, and then perturbs these identified instances by the gradient guidance. Zhang *et al.* further optimize this gradient-based search approach by introducing momentum. Above works is mainly for tabular-data system, but of course there are some other systems. For example, CHECKLIST proposes many templates and gender entity words, thus testing the discrimination in the NLP system [30]. Detailed empirical comparison between our work and this previous approach has been shown above in Section 4.
Model Extraction Attacks. The shadow-model-based strategy of BREAM is inspired by the model extraction attacks. The Jacobian-based shadow DNN training [28] used in our work and adversarial-samples-based method [5,39] mentioned above all belong to shadow-model-based model extraction attacks, which aim to simulate functions and decision boundaries of the target model. The attacker does not know the exact structure of the target model, so enormous queries are often required, which brings a challenge. We actually face similar problems in our work and we have come up with some new solutions. The details can be found in Section 3. This shadow-model-based method is relatively practical, and there are also some other attack techniques. Equation-solving attack [36,38] is designed for the traditional machine learning methods, which solved the parameters of models, under the premise of knowing algorithms and structures of models. Meta-model-based extraction attack [26] tries to infer the properties of the target model, such as the number of network layers, type of activation functions, etc., by training an additional meta-model. The meta-model takes the results of querying the target model as input and outputs the properties.

6 Discussion and Future Work

Recalling BREAM presented in this paper, we propose a black-box guidance strategy based on global interpretability, in contrast to the previous methods that primarily rely on local interpretability. While interpretability offers advantages, it also consumes the query budget. To address this, BREAM, aims to efficiently leverage a general guidance approach from a global perspective, thereby avoiding the repetitive construction of local interpretability and minimizing resource consumption. Specifically, we achieve global interpretability through model extraction, which is commonly considered query-intensive. But we believe that this is not a concern for BREAM. On the one hand, the model of a decision system based on tabular data is relatively simple, and on the other hand, the investment in building the initial show model will pay off consistently. The experimental results have validated the effectiveness of our strategy.

However, there is still a lot of room for improvement in BREAM. Firstly, the current version of BREAM only supports tabular data. Secondly, if the target system deploys a defense mechanism against model extraction, the effectiveness of BREAM may be affected. Later, we will explore how BREAM can be extended to more scenarios and its performance against model extraction defense mechanisms.

7 Conclusion

In this paper, we propose BREAM, an automated black-box DNN fairness testing method that is able to discover discriminative samples efficiently for a target model. Unlike existed works, we focus on the query-limited and label-only black-box setting, which is close to the real world and maximizes transferability. BREAM only requires permission to access specific inputs and predicted labels for a black-box model. BREAM trains shadow models to perceive information about the target model, then benefits from the approximate gradients obtained by shadow models to guide the discriminative samples generation process. Experimental results show that BREAM significantly outperforms current black-box methods and achieves the performance level of the current state-of-the-art white-box method in efficiency. The code of BREAM is available at [1].

Acknowledgement

This research is supported by National Key Research and Development Program of China (2020YFB1406900), National Natural Science Foundation of China (U21B2018, 62161160337, U20B2049, U1736205, 61802166, 62102359, 62006181, 62132011, U20A20177, 62206217), State Key Laboratory of Communication Content Cognition (Grant No. A02103) and Shaanxi Province Key Industry Innovation Program (2023-ZDLGY-38, 2021ZDLGY01-02). Chao Shen is the corresponding author.

References

1. Code of black-box fairness testing with shadow models. https://github.com/lenijwp/Black-box-Discrimination-Finder
2. Embedding api-build faster great ai algorithms for hr. https://hrflow.ai/embedding//
3. Aggarwal, A., Lohia, P., Nagar, S., Dey, K., Saha, D.: Black box fairness testing of machine learning models. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. pp. 625–635 (2019)
4. Angwin, J., Larson, J., Mattu, S., Kirchner, L.: Machine bias. ProPublica, May **23**, 2016 (2016)
5. Batina, L., Bhasin, S., Jap, D., Picek, S.: CSI NN: Reverse engineering of neural network architectures through electromagnetic side channel. In: 28th USENIX Security Symposium (USENIX Security 19). pp. 515–532. USENIX Association, Santa Clara, CA (Aug 2019), https://www.usenix.org/conference/usenixsecurity19/presentation/batina
6. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)
7. Cherian, V., Bindu, M.: Heart disease prediction using naive bayes algorithm and laplace smoothing technique. Int. J. Comput. Sci. Trends Technol **5**(2), 68–73 (2017)

8. Chollet, F., et al.: Keras. https://github.com/keras-team/keras (2015)
9. Corbett-Davies, S., Goel, S.: The measure and mismeasure of fairness: A critical review of fair machine learning. arXiv preprint arXiv:1808.00023 (2018)
10. Dua, D., Graff, C.: UCI adult data set. https://archive.ics.uci.edu/ml/datasets/adult (2017)
11. Dua, D., Graff, C.: UCI bank marketing data set. https://archive.ics.uci.edu/ml/datasets/bank+marketing (2017)
12. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
13. Dua, D., Graff, C.: UCI statlog (german credit data) data set. https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data) (2017)
14. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: Proceedings of the 3rd innovations in theoretical computer science conference. pp. 214–226 (2012)
15. Fan, M., Wei, W., Jin, W., Yang, Z., Liu, T.: Explanation-guided fairness testing through genetic algorithm. In: Proceedings of the 44th International Conference on Software Engineering. pp. 871–882 (2022)
16. Feldman, M., Friedler, S.A., Moeller, J., Scheidegger, C., Venkatasubramanian, S.: Certifying and removing disparate impact. In: proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. pp. 259–268 (2015)
17. Ferral, K.: Wisconsin supreme court allows state to continue using computer program to assist in sentencing. the capital times, july 13, 2016
18. Galhotra, S., Brun, Y., Meliou, A.: Fairness testing: testing software for discrimination. In: Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering. pp. 498–510 (2017)
19. Goh, G., Cotter, A., Gupta, M., Friedlander, M.P.: Satisfying real-world goals with dataset constraints. In: Advances in Neural Information Processing Systems. pp. 2415–2423 (2016)
20. HLEG, A.: High-level expert group on artificial intelligence. Ethics Guidelines for Trustworthy AI (2019)
21. Huang, G., Li, Y., Wang, Q., Ren, J., Cheng, Y., Zhao, X.: Automatic classification method for software vulnerability based on deep neural network. IEEE Access **7**, 28291–28298 (2019)
22. Kamiran, F., Calders, T., Pechenizkiy, M.: Discrimination aware decision tree learning. In: 2010 IEEE International Conference on Data Mining. pp. 869–874. IEEE (2010)
23. Kamishima, T., Akaho, S., Asoh, H., Sakuma, J.: Fairness-aware classifier with prejudice remover regularizer. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 35–50. Springer (2012)
24. Liang, L., Zheng, K., Sheng, Q., Huang, X.: A denial of service attack method for an iot system. In: 2016 8th international conference on Information Technology in Medicine and Education (ITME). pp. 360–364. IEEE (2016)
25. Lloyd, S.P.: Least squares quantization in pcm. IEEE Trans **28**(2), 129–137 (1982)
26. Oh, S.J., Augustin, M., Schiele, B., Fritz, M.: Towards reverse-engineering black-box neural networks (2018)
27. Olson, P.: The algorithm that beats your bank manager. cnn money, march 15, 2011
28. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia conference on computer and communications security. pp. 506–519 (2017)

29. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144 (2016)
30. Ribeiro, M.T., Wu, T., Guestrin, C., Singh, S.: Beyond accuracy: Behavioral testing of nlp models with checklist. arXiv preprint arXiv:2005.04118 (2020)
31. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
32. Schuba, C.L., Krsul, I.V., Kuhn, M.G., Spafford, E.H., Sundaram, A., Zamboni, D.: Analysis of a denial of service attack on tcp. In: Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No. 97CB36097). pp. 208–223. IEEE (1997)
33. Sweeney, L.: Discrimination in online ad delivery. Queue **11**(3), 10–29 (2013)
34. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
35. Tramer, F., Atlidakis, V., Geambasu, R., Hsu, D., Hubaux, J.P., Humbert, M., Juels, A., Lin, H.: Fairtest: Discovering unwarranted associations in data-driven applications. In: 2017 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 401–416. IEEE (2017)
36. Tramèr, F., Zhang, F., Juels, A., Reiter, M.K., Ristenpart, T.: Stealing machine learning models via prediction apis. In: 25th {USENIX} Security Symposium ({USENIX} Security 16). pp. 601–618 (2016)
37. Udeshi, S., Arora, P., Chattopadhyay, S.: Automated directed fairness testing. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering. pp. 98–108 (2018)
38. Wang, B., Gong, N.Z.: Stealing hyperparameters in machine learning. In: 2018 IEEE Symposium on Security and Privacy (SP). pp. 36–52. IEEE (2018)
39. Yu, H., Yang, K., Zhang, T., Tsai, Y.Y., Ho, T.Y., Jin, Y.: Cloudleak: Large-scale deep learning models stealing through adversarial examples. In: Proceedings of Network and Distributed Systems Security Symposium (NDSS) (2020)
40. Zhang, B.H., Lemoine, B., Mitchell, M.: Mitigating unwanted biases with adversarial learning. In: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. pp. 335–340 (2018)
41. Zhang, L., Zhang, Y., Zhang, M.: Efficient white-box fairness testing through gradient search. In: Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis. pp. 103–114 (2021)
42. Zhang, L., Wu, Y., Wu, X.: Achieving non-discrimination in data release. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1335–1344 (2017)
43. Zhang, P., Wang, J., Sun, J., Dong, G., Wang, X., Dai, T., Wang, X., Dong, J.S.: White-box fairness testing through adversarial sampling. 42nd International Conference on Software Engineering (2020)
44. Zhang, P., Wang, J., Sun, J., Dong, G., Wang, X., Dai, T., Wang, X., Dong, J.S.: White-box fairness testing through adversarial sampling. https://github.com/pxzhang94/ADF (2020)