Fairness in the Eyes of the Data: Certifying Machine-Learning Models

Shahar Segal[1], Yossi Adi[2], Benny Pinkas[2], Carsten Baum[3], Chaya Ganesh[4], Joseph Keshet[2]
[1]Tel-Aviv University
[2]Bar-Ilan University
[3]Aarhus University
[4]IISc Bangalore

Abstract

We present a framework that allows to certify the fairness degree of a model based on an interactive and privacy-preserving test. The framework verifies any trained model, regardless of its training process and architecture. Thus, it allows us to evaluate any deep learning model on multiple fairness definitions empirically. We tackle two scenarios, where either the test data is privately available only to the tester or is publicly known in advance, even to the model creator. We investigate the soundness of the proposed approach using theoretical analysis and present statistical guarantees for the interactive test. Finally, we provide a cryptographic technique to automate fairness testing and certified inference with only black-box access to the model at hand while hiding the participants' sensitive data.

1 Introduction

Machine learning systems are increasingly being used to inform and influence decisions about people, leading to algorithmic outcomes that have powerful personal and societal consequences. For instance, decisions such as (i) *is an individual likely to commit another crime?* [2]; or (ii) *is an individual likely to default on a loan?* [31] are made using algorithmic predictions. This can be concerning given the many documented cases of models amplifying bias and discrimination from the training data [6, 21, 8, 29]. To address this formally, a line of recent works [10, 14, 18, 20] considers *fairness* in classification by proposing notions of fairness based on similarity measures and formalizing variants of this notion that provide guarantees against discrimination.

One common scenario in which such a discrimination could potentially happen is a setting with a client and a server. The server classifies queries by a client in an automated way using a machine learning model generated by it. On the other hand, the client wants to make sure its queries are treated fairly and its sensitive data is conserved. If the model itself is not a secret, then a client can potentially run tests (such as the ones implied by the references above) on the model to establish its purported fairness without exposing its data. Making a model public, however, is not always in the interest of the server, since it has invested resources such as expertise, data and computation time for the training – and therefore often wants the model to remain proprietary. Moreover, sharing models may in some cases raise security or privacy concerns. It therefore may be deemed appropriate or necessary to outsource any such test to a semi-trusted third party such as a government entity, which would inspect a model and certify its fairness. This raises our first question:

Question 1: Can we design a framework for certifying the fairness of models, giving guarantees to clients while being practically realizable and keeping the model secret from the clients?

Having such a third party relieves the client from testing fairness, but actually just shifts responsibility to someone who might be more qualified to make a judgement about the model. To minimize the

Preprint.

necessary trust between the model owner and the third party, such a test would still be restricted to a black-box scenario. In addition, constructing such a test for establishing fairness guarantees can be difficult on its own. Given that the sources and amount of data is limited, it might be that the third party can only use data in the fairness test that the model owner is familiar with. This might enable the model owner to design an unfair model which successfully passes the examination by the third party. This raises our second question:

Question 2: Can we design a black-box fairness test that gives guarantees even if the test set is (partially) known?

Here, by a black-box test we mean that a test should only query the model M on different inputs but should not make any assumptions about the actual model parameters.

Our contributions In this work, we answer both questions affirmatively. We design an architecture for three (or more) participants which are the model owner (or "server") $\mathcal{S}$, the client $\mathcal{C}$ and a trusted third party $\mathcal{R}$ (also called "regulator"). Our architecture uses techniques from cryptography to construct secure protocols for

1. An interactive test between $\mathcal{S}$ and $\mathcal{R}$ allowing to establish with probability that a model M provided by $\mathcal{S}$ is fair with respect to a set of pre-defined groups. While ensuring that $\mathcal{R}$ does not learn M. This test considers scenarios where $\mathcal{S}$ *is* or *is not* aware of the test data. $\mathcal{R}$ is not involved in the training of M, it only performs certification.
2. An interactive computation between $\mathcal{S}$ and $\mathcal{C}$ which computes a prediction $\hat{y} = M(x)$ from an input x and a model M. The interactive computation neither leaks M to $\mathcal{C}$ nor x to $\mathcal{S}$, and yet makes sure that the model that was used in the prediction has been certified by $\mathcal{R}$.

Our work provides fairness tests necessary for these protocols that have black-box to the model and uses existing highly efficient cryptographic primitives to implement the tests securely. While we motivate the underlying ideas of these tests on an intuitive level and give formal arguments for their soundness, we also provide experimental evidence that the hypotheses that make our tests possible are viable. Since secure and privacy-preserving computation of models, for both training and inference, is a very active research area (e.g. [27, 16, 28, 24]), the performance of the current solutions in this field is continuously improving. As our work assumes the existence of secure protocols for inference, and investigates how to add fairness on top of these in a generic way that is independent of the underlying training algorithm, our approach will benefit in practicality from any independent progress that is made in this direction.

Related work Fairness in algorithms was first investigated by Friedman and Nissenbaum [12]. Since then, further research into data as a source of unfairness in ML decisions has been done e.g. in [17, 7]. Baluta et al. [4] showed how to verify properties of a DNN (fairness among them). In their work they encode the network into *Conjunctive Normal Forms* and then test if it will likely fulfill certain logical constraints. In comparison, our approach is independent of the concrete model parameters and architecture.

Recently, Kilbertus et al. [19] suggested to use cryptographic primatives for *fairness certification*, *fair model training*, and *model decision verification*. However, this work was mainly focused on model training, and did not provide analysis and guarantees for model fairness certification and verification. In contrast, our study focuses on fairness certification of existing models, which we analyze from a theoretical and practical point of view while providing guarantees based on the number of samples available in the test set. We also explore a different scenario where these samples are known to $\mathcal{S}$ during model training, which makes the certification harder.

Several statistical measures of unfairness, and fairness criteria are studied in [11, 32]. These and subsequent works achieve statistical notions of fairness through post-processing the training data, and/or by enforcing constraints at training time. Our work differs from this line of research in that we want to guarantee fairness which is enforced obliviously of the training process. Dwork et al. [10] shows that statistical notions of fairness are inadequate, while [8] established that calibration does not rule out unfair decisions. These results emphasize that fairness is nuanced, complicated, application-specific, and can depend on legal and social contexts. In this work, we answer the orthogonal question of designing a fairness test for a model, *given* an accepted definition of fairness.

2 Preliminaries

Let $\mathcal{X}$ be the set of possible inputs, $\mathcal{G}$ be a finite set of groups that are relevant for fairness (e.g., ethnic groups) and $\mathcal{Y}$ be a finite set of labels. We suppose $\mathcal{X} \times \mathcal{G} \times \mathcal{Y}$ is drawn from a probability space Ω with an unknown distribution $\mathcal{D}$. Let M be a trained model for a classification task of $\mathcal{D}$, we denote $M(x)$ for classification of input $x \in \mathcal{X}$.

The goal of training a model M is usually to achieve low error on unseen data. In addition, when dealing with model fairness, we also take into account a measurement with respect to $\mathcal{G}$. While there are plenty of fairness measurements [30], here we focus on group risk and likelihood based definitions, specifically: *overall risk equality*, *equalized odds* and *demographic parity*. First, we define the conditional *risk* and *likelihood* respectively:

$$\ell_g(M) = \mathop{\mathbb{E}}_{(x,g',y')\sim\mathcal{D}}\left[\mathbb{1}\left\{M(x) \neq y'\right\}|g' = g\right] \tag{Risk}$$

$$\ell_{g,y}(M) = \mathop{\mathbb{E}}_{(x,g',y')\sim\mathcal{D}}\left[\mathbb{1}\left\{M(x) \neq y'\right\}|g' = g, y' = y\right] \tag{Risk with label condition}$$

$$L_{g,y}(M) = \mathop{\mathbb{E}}_{(x,g',y')\sim\mathcal{D}}\left[\mathbb{1}\left\{M(x) = y\right\}|g' = g\right] \tag{Likelihood}$$

where $\mathbb{1}\{\pi\}$ is an indicator function with a predicate π. The empirical conditional risk is defined for a given independent sample set $T = \{(x_1, g_1, y_1), ..., (x_m, g_m y_m)\} \sim \mathcal{D}^m$ as:

$$\bar{\ell}_g(M, T) = \frac{1}{m_g}\sum_{i=1}^{m} \mathbb{1}\{M(x_i) \neq y_i \wedge g_i = g\} \tag{1}$$

where m_g is the number of samples in T from group g.

We define a metric called the *fairness gap* to be the maximal margin between any two groups (and labels). Formally, we use three well-known measurements:

$$\max_{g_0, g_1 \in \mathcal{G}} \quad |\ell_{g_0}(M) - \ell_{g_1}(M)| \tag{Overall Risk Equality}$$

$$\max_{g_0, g_1 \in \mathcal{G}, y \in \mathcal{Y}} |\ell_{g_0,y}(M) - \ell_{g_1,y}(M)| \tag{Equalized Odds}$$

$$\max_{g_0, g_1 \in \mathcal{G}, y \in \mathcal{Y}} |L_{g_0,y}(M) - L_{g_1,y}(M)| \tag{Demographic Parity}$$

Likewise, the *empirical fairness gap* (EFG) is defined using the empirical approximation of each measurement respectively.

Lastly, we consider a model M as **ϵ-fair** on $(\mathcal{G}, \mathcal{D})$ with respect to a fairness measurement if its fairness gap is smaller than ϵ. We relax the definition by demanding to have ϵ-fairness with confidence $1 - \delta$, which in practice is inherent due to the limited sample set from $\mathcal{D}$. That is, a model M as **ϵ-fair** on $(\mathcal{G}, \mathcal{D})$ under the overall risk equality metric if:

$$\Pr\left[\max_{g_0, g_1 \in \mathcal{G}} |\ell_{g_0}(M) - \ell_{g_1}(M)| > \epsilon\right] \leq \delta \tag{2}$$

Similarly, plugging-in the fairness gap metric for equalized odds and demographic parity yields the corresponding ϵ-fairness definition of each of them.

3 The Framework

In Section 1 we gave an overview about the different participants in our setting. In this section we will make their roles more explicit and describe the security guarantees that are given to each of them as well as the trust relations. Note that we discuss our framework with respect to three participants but it can easily be generalized to any larger number. In particular, it allows for a large number of regulators $\{\mathcal{R}_i\}_{i=1}^k$ that a client $\mathcal{C}$ can choose from or even perform the regulators role by itself.

- The *Server* $\mathcal{S}$ initially generates the model M. Its main objective is to keep M secret.
- The *Client* $\mathcal{C}$ has a private input x and wishes to obtain $\hat{y} \leftarrow M(x)$, where the server provides M (depending on the application, $\mathcal{S}$ might receive $\hat{y}$ as well but not x). The objective of $\mathcal{C}$ is to ensure that M is fair while keeping x private.

- The third participant is the *Regulator* $\mathcal{R}$ who should neither learn M nor x or y. After $\mathcal{S}$ proves the fairness of model M, $\mathcal{R}$ outputs certificate $\texttt{cert}_M$ for the model to attest its validity. $\texttt{cert}_M$ is tied to another certificate $\mathcal{R}$ issued, $\texttt{cert}_{ID}$, which serves as the identity of $\mathcal{R}$. Thus, when $\texttt{cert}_M$ is shown to $\mathcal{C}$ it can verify that indeed $\mathcal{R}$ certified the model using $\texttt{cert}_{ID}$. In addition, $\mathcal{R}$ has access to the sample set T in order to check fairness, which is possibly known to $\mathcal{S}$.

The certificates $\texttt{cert}_{ID}, \texttt{cert}_M$ are implemented using a digital signature scheme and a collision-resistant hash function. Roughly, $\texttt{cert}_{ID}$ is a public verification key that is tied to the identity of $\mathcal{R}$, and $\texttt{cert}_M$ is a signature on a compressed version of the model that is computed using a collision-resistant hash function. Here, a cryptographic signature ensures that only $\mathcal{R}$ could issue $\texttt{cert}_M$, while the hash function forces $\mathcal{S}$ to use the same model with $\mathcal{C}$ that he used when obtaining $\texttt{cert}_M$. We define signature schemes and collision resistant hashing in Appendix A.

At the beginning of the protocol, $\mathcal{R}$ generates its public certificate $\texttt{cert}_{ID}$ and makes it available. $\mathcal{S}$ and $\mathcal{R}$ then interact to generate the model certificate $\texttt{cert}_M$ for model M. In the process $\mathcal{R}$ is allowed to query M an arbitrary number of times to ensure fairness. To perform an inference by $\mathcal{S}$ and $\mathcal{C}$, both first agree on a regulator certificate $\texttt{cert}_{ID}$ that they will use. Then, $\mathcal{C}$ obtains an output $\hat{y}$ based on its input x and on a model M' provided by $\mathcal{S}$. Here, $\mathcal{C}$ only accepts $\hat{y}$ if M' is certified by the regulator behind $\texttt{cert}_{ID}$ for fairness. $\mathcal{C}$ does not learn anything about M', besides $\hat{y}$. Fig. 1 describes the aforementioned process schematically.

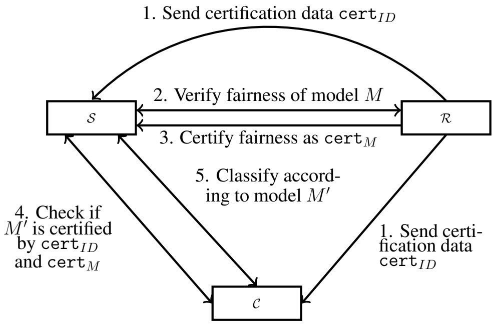
1. Send certification data $\texttt{cert}_{ID}$
2. Verify fairness of model M
$\mathcal{S}$
$\mathcal{R}$
3. Certify fairness as $\texttt{cert}_M$
5. Classify according to model M'
4. Check if M' is certified by $\texttt{cert}_{ID}$ and $\texttt{cert}_M$
1. Send certification data $\texttt{cert}_{ID}$
$\mathcal{C}$

Figure 1: Certification and Verification

Both the inference on M and the verification of $\texttt{cert}_M$ that are necessary in Fig. 1 can be done using secure computation. Secure Computation lets parties perform computations that reveal neither their inputs nor any intermediate values, but only the outputs of the computation. This can be done easily if $\mathcal{S}, \mathcal{R}$ and $\mathcal{C}$ would have access to a "trusted third party" $\mathcal{F}_{\text{SC}}$ which performs the computational task for them. That trusted third party would receive inputs from participants, do the computation, and send the output back to them. In our case, $\mathcal{F}_{\text{SC}}$ would recieve all secret input from the participants and send $\texttt{cert}_M$ to $\mathcal{S}$ after verifying M is fair. $\mathcal{F}_{\text{SC}}$ can also send $M(x)$ to $\mathcal{C}$ if model M' is certified in that manner.

Such a "trusted third party" does not exist, however one can imitate it using secure computation. We provide more information on how to *emulate* $\mathcal{F}_{\text{SC}}$ in Appendix A.3, together with the aforementioned process and its security in Appendix B.

4 Certifying Fairness Interactively

We introduce two interactive tests which allow $\mathcal{R}$ to determine if a model M is ϵ-fair:

1. The model M is queried using a sample set T which is *unknown* to $\mathcal{S}$. We show that fairness guarantees about M can be made given by the empirical fairness gap (EFG) and a lower-bound on the minimal size of the set T with respect to each group $g \in \mathcal{G}$.
2. The model M is queried using a sample set $\tilde{T}$ which is derived from a set T in an augmented and randomized fashion (namely, by making small changes to items from T). The set T as well as the augmentation algorithm are known to $\mathcal{S}$ in advance. We show that this test implies ϵ-fairness of M, given that the augmentation impacts fair and unfair models in a different way.

Both of the aforementioned tests are independent of the representation of M, make no requirement on its training algorithm and only require access to $M(\cdot)$ for different inputs. Looking ahead, this is precisely what allows us to perform those tests interactively between $\mathcal{S}$ and $\mathcal{R}$ without leaking M.

4.1 Certifying Fairness using Private Data

We start by describing the first test when the model M is queried using a sample set T which is *unknown* to $\mathcal{S}$. All definitions below are based on ϵ-fairness with confidence $1 - \delta$ under the overall risk equality fairness metric, however these can be modified to other group-based fairness definitions.

Given a sufficient amount of unknown samples from every group $g \in G$, we approximate the conditional risk of the model M as in (1). In order to achieve a high confidence in the test and a close enough approximation, $\mathcal{R}$ needs to generate enough samples from each group, i.e. a balanced test set T w.r.t $\mathcal{G}$. This is inherent, as we cannot make claims about the behavior of M with respect to g without ever probing M on elements in g.

Denote the *Empirical Fairness Gap* as $EFG = \max_{g_0, g_1 \in \mathcal{G}} |\bar{\ell}_{g_0}(M,T) - \bar{\ell}_{g_1}(M,T)|$. The following theorem states the conditions which guarantee that a model is ϵ-fair with a confidence $1-\delta$.

Theorem 1. *A model M is ϵ-fair with confidence $1-\delta$ if:*

$$EFG < \epsilon \qquad \textit{and} \qquad \min_{g \in \mathcal{G}} m_g \geq \frac{2}{(\epsilon - EFG)^2} \ln \frac{2|\mathcal{G}||\mathcal{Y}|}{\delta} \tag{3}$$

for $T = \{(x_1, g_1, y_1), ..., (x_m, g_m, y_m)\} \sim \mathcal{D}^m$ and where m_g, as in Eq. (1)*, denotes the number of occurrences of g in T.*

Demographic parity and equalized odds can be achieved by using the corresponding EFG definition and minimizing over $\mathcal{G} \times \mathcal{Y}$, counting m_g and $m_{g,y}$ respectively in (3). The full proof of the above theorem can be found in Appendix C.1, and it relies on an Hoeffding-bound argument.

In other terms, we require the EFG of the model to be smaller than ϵ, and the difference between ϵ and the EFG has an impact on the minimum number of samples we require from each group to guarantee ϵ-fairness. We can see a natural trade-off in Theorem 1: A larger sample set is required to verify smaller fairness gaps, as indicated by ϵ. Thus the more data you have, the easier it is to verify that the EFG is close to the actual fairness gap.

4.2 Certifying Fairness using Augmented Data

The disadvantage of the aforementioned test is that all test data T must be hidden so that the model generator $\mathcal{S}$ cannot use it to adapt M accordingly. In other settings, we would like to test for fairness using public data, which can be known to $\mathcal{S}$. This setting is realistic in many scenarios. For example, if labelled data is costly, getting unique labelled data for a test will be difficult for $\mathcal{R}$.

A straightforward argument against this approach is once the data is publicly available, T is not chosen independently of M. Thus, a malicious $\mathcal{S}$ can create an unfair model that memorizes the set T and responds fairly on it, so that it passes the test outlined above. To counter such dishonest training, we need a method to alter the existing samples and force some sort of generalization abilities. We therefore define the notion of an *augmentor*. An augmentor applies random augmentations to the input which alter the sample but still preserves its label and group with high probability, here we use it to generate *new* samples for querying that with high probability were not seen during model training. This is a necessary but not sufficient condition in order to ensure a valid test for M. For example, consider an augmentor that only alters the first few pixels of the image. A model that simply ignores those pixels can still overfit on the rest of the image and pass any test.

Hence, in this work we suggest to use a set of randomized augmentation functions to reduce memorization capabilities of an adversary. For this, the assumption is that ϵ-fair models behave differently from unfair models when queried against the samples augmented by the augmentor. Then, this different behavior can be leveraged to expose the unfair nature of certain models. Our approach follows this assumption to construct a querying test set in the same fashion of the test from Section 4.1.

More specifically, define an algorithm augmentor $\texttt{aug} : \mathcal{X} \times \mathcal{G} \times \{0,1\}^\tau \rightarrow \mathcal{X} \times \mathcal{G}$ that gets as input a random string and a sample and outputs a new augmented sample. The label and group of the new sample should be the same as the original sample with high probability.

We re-define the conditional risk to be on an augmented sample from $\mathcal{D}$. Formally:

$$\ell_{g,\texttt{aug}}(M) = \mathop{\mathbb{E}}_{(x,g',y) \sim \mathcal{D}, r \leftarrow \{0,1\}^\tau} \left[\mathbb{1}\{M(\tilde{x}) \neq y\} \,\middle|\, (\tilde{x}, \tilde{g}) = \texttt{aug}(x, g; r) \wedge g' = \tilde{g} \right].$$

As mentioned before, there is no guarantee that samples augmented by aug yield better results than T itself. We need an additional assumption on the behavior of fair and unfair models when shown augmented samples from aug, thus we call a class of models $\mathcal{M}$ *detectable* if it fulfills that assumption.

Definition 1 (($\epsilon, \alpha, \texttt{aug}$)$-$detectable fairness)**.** Let $\mathcal{A}$ be an arbitrary training algorithm which outputs a model in $\mathcal{M}$. $\mathcal{M}$ has ($\epsilon, \alpha, \texttt{aug}$)-detectable fairness on $\mathcal{D}$ if there exists $m \in \mathbb{N}$ such that for any $T \sim \mathcal{D}^m$ and $M \sim \mathcal{A}(\mathcal{D}, T, \texttt{aug}, \alpha)$, M is ϵ-fair if:

$$\max_{g_0, g_1 \in \mathcal{G}} |\ell_{g_0,\texttt{aug}}(M) - \ell_{g_1,\texttt{aug}}(M)| \leq \alpha \ .$$

Definition 1 allows us to build an interactive test and to empirically find parameters ϵ, α, m and an augmentor for which it appears to be true. It also yields a non-trivial angle both for breaking our overall construction and for improving it. Notice, the above definition does not imply that all ϵ-fair models have this property, and some fair models will not be discovered due to that. In Section 5.2 we demonstrate that some models seems to be detectable. Additionally, we observe that the output of aug is not required to be indistinguishable from a new sample from $\mathcal{D}$. In particular the definition does not rule out that $\mathcal{A}$ is aware of the possible augmentations.

Let $\tilde{T}$ be a sample set T after augmenting each sample. Denote $\bar{\ell}_{g,\texttt{aug}}(M, \tilde{T})$ the empirical conditional risk and $EFG = \max_{g_0, g_1 \in \mathcal{G}} \left|\bar{\ell}_{g_0,\texttt{aug}}(M, \tilde{T}) - \bar{\ell}_{g_1,\texttt{aug}}(M, \tilde{T})\right|$. We state the following,

Theorem 2. *Let $\mathcal{M}$ be a class of models with ($\epsilon, \alpha, \texttt{aug}$)-detectable fairness. Let $T, \tilde{T}, \texttt{aug}$ and $M \in \mathcal{M}$ be as stated above. M is ϵ-fair with confidence $1 - \delta$ if:*

$$EFG < \alpha \qquad \text{and} \qquad \min_{g \in \mathcal{G}} m_g \geq \frac{2}{(\alpha - EFG)^2} \ln \frac{2|\mathcal{G}||\mathcal{Y}|}{\delta} \ .$$

In other words, we can certify ϵ-fairness of a model with high confidence assuming ($\epsilon, \alpha, \texttt{aug}$)-detectable fairness. The proof of this theorem is in Appendix C.2.

5 Experiments

We provide empirical evidence to demonstrate that the assumptions made for the fairness tests in Section 4 are meaningful.

We used six different datasets from various domains: visual (UTKFaces [33], LFW [15], Colored-MNIST [3], and a subset of CelebA[1] [26]), tabular (Adult Income [22]) and spoken (TIMIT [13]). The datasets vary in size and disparity of minority groups and as such some can be used to create fair or unfair models based on their empirical fairness gap (EFG). We demonstrate the variety of our datasets and detail the preprocess in Appendix D.1.

5.1 Private Data Setup

Simulating the necessary setup for Section 4.1, we assume that $\mathcal{R}$ possesses a subset of secret samples to be used to certify a model M for fairness and accuracy. Naturally, we split the data into a training and test subset. Setting ϵ-fair and δ-confidence thresholds, we can certify whether a model is fair using the conditions in Theorem 1. A bottleneck of these conditions is our dependency on the size of the sample set. Datasets with bigger sample set allow us to certify more (fair) models, while we were not able to certify a (fair) model if the sample set was too small, even if it is indeed truly fair under the chosen fairness metric.

We performed our test on the mentioned datasets with $\delta = 0.05$ and varying ϵ of $0.05, 0.075$ and 0.1. For some tasks this gap and confidence level might be intolerable, but for others, such as gender prediction of a face image, which is the task set for UTKFace, LFW and CelebA, it is better than the existing empirical gaps between ethnicity groups of well-known service providers' models [6].

The test results for overall risk equality are shown in Fig. 2. As shown, out of the six datasets only C-MNIST and CelebA produced fair models during our training for $\epsilon = 0.05$, while UTKFace has a fair model for $\epsilon = 0.075$. LFW, Adult Income and TIMIT datasets are all below the threshold of all tests, either due to sample size or a large EFG. Therefore, we focus on the first three datasets as they are the only ones to pass any of our tests.

[1] We annotated 8,500 celebrities out of 10,177 in the dataset for ethnicity using Amazon Mechanical Turk. Three turkers annotated three images of each of the 8,500 celebrities, resulting in 177,683 images classified as either Asian, African, Caucasian or Other. The annotations can be downloaded from `https://github.com/will/be/published/`.

Table 1: Fairness test in the private and public settings on the 3 image datasets. ”Regular” refers to the model trained fairly, while ”Bias” refers to the biased sampled training.

		Private Setting				**Public Setting**			
Dataset	Model	Accuracy		Risk Equality EFG		Accuracy		Risk Equality EFG	
		Regular	Bias	Regular	Bias	Regular	Bias	Regular	Bias
UTKFace	ResNet18	89.76	88.56	0.012	0.093	96.11	91.44	0.027	0.139
C-MNIST	LeNet	98.11	74.01	0.001	0.450	89.17	67.97	0.007	0.340
CelebA	ResNet18	97.63	96.95	0.007	0.034	96.60	97.02	0.010	0.033

To further evaluate the setup, we trained the same models with a tainted batch sampler. The sampler showed less samples from the smallest minority group-label pair (g_1, y_1) in each batch in order to generate a synthetic sample disparity. We denoted these models as *Bias* in Fig. 2 and Table 1. The taint resulted in an almost as accurate model with a much larger EFG, suggesting they are less fair. For equalized odds, only CelebA had enough samples to certify a model for $\epsilon = 0.05$. It requires at least twice as many samples (since we count $m_{g,y}$ instead of m_g). We find it interesting as it implies the amount of data should be a consideration even for which definition of fairness is practical to choose. We detail the results for equalized odds and demographic parity metrics in Appendix D.2. Results for bias C-MNIST does not appear in Fig 2 since its performance is significantly worse.

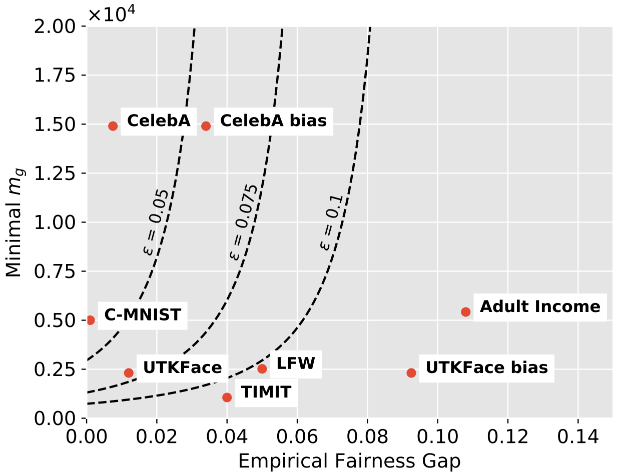

Figure 2: Private fairness test borders by EFG and the minimal m_g. Left to the dashed border is the area where a model would pass the test for that ϵ with $\delta = 0.05$. Dots indicate the overall risk equality results for each dataset.

5.2 Augmented Public Data Setup

In this setup, all the data used in the test is known to all participants. With that in mind, we show a potential augmentor for datasets of images and demonstrate empirically that unfair models cannot pass our test as both accurate and fair. Even though the training set is the same as the test set, the key difference is that $\mathcal{S}$ fixes M before we apply our augmentor to the dataset with new randomness. This generates diverse enough samples, for which models that are fair and generalize well on augmented samples pass the test, while models which are either unfair or bad at generalization fail the test. Our augmentations include rotation, cropping, blanked pixels [34] and added Gaussian noise. Each augmentation was set to be invoked at a certain probability threshold which was chosen randomly. The augmented images should keep the same label and group as the original image to the human eye. By doing so, we hope to generate varied data that cannot be easily be reversed or overfitted on.

We tested our three image datasets using the overall risk equality metric, the results are in Table 1. We used the same method to generate fair and unfair models as in Section 5.1 with the following difference: during training we invoked the augmentor per sample to generate a new augmented sample each time. When we trained the models on the original dataset, the models were not able to generalize on the augmented data.

The results show that there exists a margin in EFG between the fair and unfair models on UTKFace, C-MNIST and CelebA, while the margin is different between datasets, potentially due to their varying size and different complexities of the tasks. This suggests the existence of some α per dataset, based on Definition 1, but we were not able to pinpoint the exact α. We conducted further attempts to characterize α in Appendix D.5.

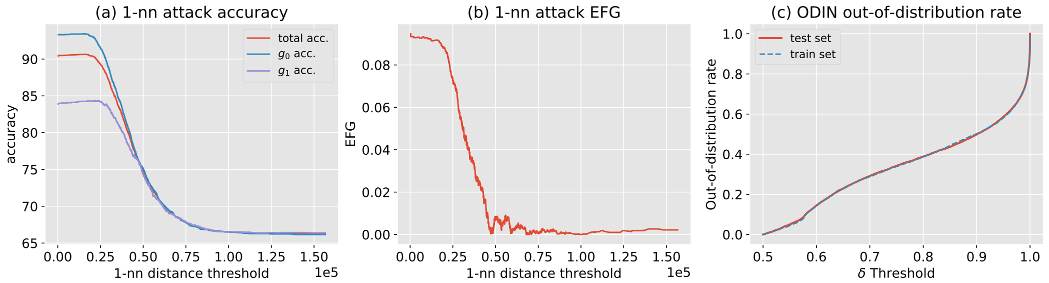

Figure 3: (a)-(b) 1-NN attack accuracy and EFG; (c) ODIN out-of-distribution rate.

Attacks against public fairness tests As we assume that our test works without knowledge of the concrete model, our scheme might be susceptible to an indirect attack on our augmentor. For example, if the model could distinguish between the public data available and a new sample, it could try and behave fairly during the test, but unfairly when an actual new sample is shown. To mimic such an attack, we tested on UTKFace whether it is easy to fool our test using a simple k-nearest neighbour algorithm (kNN) or an out-of-distribution detection technique (ODIN [25]) on top of a fair classifier to identify the augmented samples. For a fixed threshold distance from our augmented dataset, we would switch to the unfair model and otherwise output the class of identified by the kNN. For the ODIN attack we would create a threshold for detect out of distribution samples to switch to the unfair classifier. Ideally these attacks uses the kNN or fair classifier to pass the test as fair when needed, while future new samples (being "far enough" in threshold terms) invoke the unfair model as predictor. We gave the kNN augmented samples of the test as referenced neighbors and plotted the accuracy and EFG by the threshold distance for k=1 in Fig. 3a-b (larger k had worse results). In order to have a similar EFG the fair model have, this approach leads to a drop in accuracy from 91.44% to 81.9%. In the other attack, we tuned ODIN's hyperparameters, taking the values which had at least 95% success rate identifying test samples and had the best results at detecting new samples as out-of-distribution. Further details on tuning are listed in Appendix D.4. We plotted the train and test detection rate as out-of-distribution by threshold in Fig. 3c. As can be seen, the sets are detected at similar rates, and are nearly indistinguishable. This resulted in a similar fair or unfair behavior depending on the chosen threshold. These experiments suggest that these types of attacks are not a good approach to attack our proposed test, as this hybrid models cannot pass as both fair and accurate enough for practical applications.

6 Discussion & Future Work

We present an interactive test to certify fairness of any machine learning model using cryptographic methods such as secure computation. The interactive test ensures $\mathcal{R}$ does not learn M, x does not leak to $\mathcal{S}$, and M does not leak to $\mathcal{C}$, yet it verifies the model was used during inference has been certified by $\mathcal{R}$. We experimented with two scenarios where the test data is either public or private. We provide analysis and guarantees for the test data, as well as rigorously define the relation between the empirical fairness gap to the sample set sizes.

We believe creating regulatory entities, such as our abstract entity $\mathcal{R}$, can be a step towards standardizing fairness. Once these roles are set in place, our framework can guarantee users are treated fairly in a secure manner. Moreover, from our guarantees and experiments we noticed not all fairness definitions are created equally, some are harder to verify and require a much larger volumes of data, i.e. equalized odds requires at least twice as many samples as overall risk equality. This makes room for consideration on what practical definition should we aim for with respect to limited resources or what compromise needs to be made in terms of fairness gap and certainty (ϵ and δ).

For future work we would like to further explore the public data scenario. Specifically, to characterize the detectable fairness hyper-parameter α and its relation to other parameters like the sample set size T, the amount of randomness used per augmentation, etc. Additionally, we would like to explore whether these parameters can be estimated *in advance*, without having to conduct experiments on a dataset. Our results in Section 5 suggest that this is a challenge on its own. Moreover, as we are dealing with large models we also require to hash the model inside secure computation. This step has substantial cost (see Appendix B.4), and it is an open question if it could be made more efficient in practice using different ideas than ours. Lastly, the proposed method is focused on group-based fairness definitions, exploring other fairness definitions is also an interesting research direction.

Broader Impact

Our framework touches two important and sensitive subjects: fairness and privacy. As such, it has tremendous value from an ethical perspective, but also should be treated with careful consideration as a formal notion of fairness does not always align with what we think is fair and just. We believe our work is a step in the larger context of looking at Machine Learning problems through a cryptographic lens. This opens up the prospect of benefiting from what cryptography has to offer – balancing integrity with privacy.

The benefit of our framework is that it is a method to certify and evaluate fairness under a formal definition thereof, while it also conserves the privacy of clients and the intellectual property of model providers. Clients and providers benefit from it as it bridges the potential mistrust between them. It can also standardize fairness in the form of trusted regulators which are acknowledged as trustworthy in evaluating models. We find this to be an aspect that hopefully has a positive impact on society. However, we need to be careful by what we consider as fair. There is no ”one size fits all” fairness definition which complicates the validity of a certification in social terms. In particular, we focused on specific notions of fairness in our work, as these lead to an implementable result. The question of which fairness notion is the most applicable in a certain setting is independent of it and beyond the scope of this work. Our work leaves the definition of fairness to policy makers, regulators and other experts, and gives a way to take an accepted definition of fairness and certify a model for it. It should also be noted though that we guarantee fairness only up to certain probability, and this might give people a false sense of fairness of a model even when there's a chance it is not.

What is interesting about our work is that issues like fairness and transparency are considered to be at odds with another desirable feature – privacy. But our work brings these together. Moreover, cryptographic models for various primitives and protocols ranging from encryption to secure computation are designed to work in worst-case adversarial environments. This seems to be necessary for successful deployment of machine learning in certain applications and we hope that bringing this mindset to the machine learning field might be beneficial.

References

[1] Victor Arribas Abril, Pieter Maene, Nele Mertens, and Nigel Smart. 'bristol fashion' mpc circuits. `https://homes.esat.kuleuven.be/~nsmart/MPC/`. Last accessed on 11/16/2019.

[2] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: theres software used across the country to predict future criminals. and its biased against blacks. propublica 2016, 2016.

[3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[4] Teodora Baluta, Shiqi Shen, Shweta Shinde, Kuldeep S Meel, and Prateek Saxena. Quantitative verification of neural networks and its security applications. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1249–1264, 2019.

[5] Assi Barak, Daniel Escudero, Anders Dalskov, and Marcel Keller. Secure evaluation of quantized neural networks. Cryptology ePrint Archive, Report 2019/131, 2019. `https://eprint.iacr.org/2019/131`.

[6] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91, 2018.

[7] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[8] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 797–806. ACM, 2017.

[9] Ivan Damgård, Daniel Escudero, Tore Frederiksen, Marcel Keller, Peter Scholl, and Nikolaj Volgushev. New primitives for actively-secure mpc over rings with applications to private machine learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1102–1120. IEEE, 2019.

[10] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM, 2012.

[11] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2015.

[12] Batya Friedman and Helen Nissenbaum. Bias in computer systems. *ACM Transactions on Information Systems (TOIS)*, 14(3):330–347, 1996.

[13] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. Darpa timit acoustic-phonetic continuous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n*, 93, 1993.

[14] Úrsula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1944–1953, 2018.

[15] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[16] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. {GAZELLE}: A low latency framework for secure neural network inference. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1651–1669, 2018.

[17] Matthew Kay, Cynthia Matuszek, and Sean A Munson. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3819–3828. ACM, 2015.

[18] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, pages 2569–2577, 2018.

[19] Niki Kilbertus, Adrià Gascón, Matt J. Kusner, Michael Veale, Krishna P. Gummadi, and Adrian Weller. Blind justice: Fairness with encrypted sensitive attributes. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2635–2644. PMLR, 2018.

[20] Michael P Kim, Aleksandra Korolova, Guy N Rothblum, and Gal Yona. Preference-informed fairness. *arXiv preprint arXiv:1904.01793*, 2019.

[21] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[22] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, page to appear, 1996.

[23] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th International Conference on Computer Vision*, pages 365–372, Sep. 2009.

[24] Nishant Kumar, Mayank Rathee, Nishanth Chandran, Divya Gupta, Aseem Rastogi, and Rahul Sharma. Cryptflow: Secure tensorflow inference. *arXiv preprint arXiv:1909.07814*, 2019.

[25] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[27] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 19–38. IEEE, 2017.

[28] M Sadegh Riazi, Christian Weinert, Oleksandr Tkachenko, Ebrahim M Songhori, Thomas Schneider, and Farinaz Koushanfar. Chameleon: A hybrid secure computation framework for machine learning applications. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 707–721, 2018.

[29] Rachael Tatman and Conner Kasten. Effects of talker dialect, gender & race on accuracy of bing speech and youtube automatic captions. In *INTERSPEECH*, pages 934–938, 2017.

[30] Sahil Verma and Julia Rubin. Fairness definitions explained. In *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, pages 1–7. IEEE, 2018.

[31] Kaveh Waddell. How algorithms can bring down minorities credit scores. *The Atlantic*, 2, 2016.

[32] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

[33] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[34] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017.

A Cryptographic Primitives

We now describe the cryptographic primitives that are necessary to implement our framework from Section 3 in more detail: *Signatures*, *Collision-Resistant Hash Functions* and *Secure Computation*.

A.1 Signatures

Cryptographic signatures can be thought of as a computational analogue to hand-written signatures. We give a schematic explanation of signature schemes in Figure 4. Here, a pair of a public verification key vk and a secret signing key sk are generated together by the key generation algorithm `KeyGen`. sk will be used by the signing algorithm `Sign` to create a signature σ on a message m, while the verification algorithm `Verify` decides if a pair (m, σ) is valid according to the verification key vk or not. Secure signature schemes guarantee *unforgeability*, which means that given vk and arbitrarily many signature pairs $\{(m_i, \sigma_i)\}_{i \in [\ell]}$, it is hard to generate a valid signature σ on a message m, where $m \neq m_i$ for all $i \in [\ell]$.

A.2 Collision-Resistant Hashing

We will use a *Collision-Resistant Hash Function* $H_k : \{0,1\}^n \times \{0,1\}^{2n} \to \{0,1\}^n$, which is an efficiently computable function such that it is hard for any polynomial-time algorithm (in n) that is given a random k to come up with x_1, x_2 such that $H_k(x_1) = H_k(x_2)$. In practice, one uses e.g. SHA-3 to implement H_k for a k that is fixed in advance. Since the input length of SHA-3 is fixed, in order to hash longer messages, one can apply H_k recursively using a Merkle Tree (see Figure 4). For such a Merkle Tree it can be proven that if H_k is collision-resistant then $\widehat{H}_k$ is too.

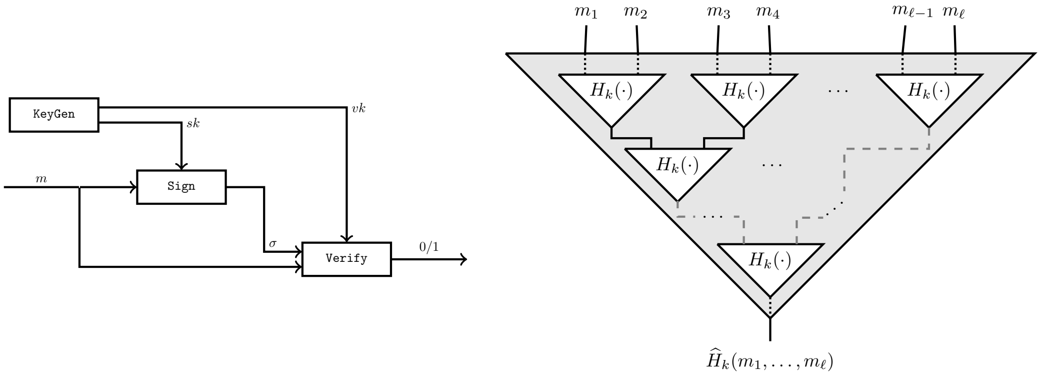

Figure 4: Signatures (left), Merkle Trees (right)

A.3 Secure Computation

We further let parties perform computations on shared data such that the computation does not reveal their inputs, for purposes as mentioned in Section 3.

Secure Computation can be imagined as the existence of a "trusted third party" $\mathcal{F}_{\texttt{SC}}$ which performs a computational task for certain parties. $\mathcal{F}_{\texttt{SC}}$ would receive the inputs from both participants, do the computation, and send the output to the participants. The task of this party is outlined in Figure 5: As is common in the secure computation literature, this description assumes that the computation is done by a circuit K. Participant $\mathcal{P}_1$ provides to the trusted party its input x_1, while participant $\mathcal{P}_2$ provides its input x_2. The trusted party computes $K(x_1, x_2)$ and sends its outputs to the respective participants. By this definition this "idealized box" $\mathcal{F}_{\texttt{SC}}$ achieves the desired privacy objective.

Such a trusted third party $\mathcal{F}_{\texttt{SC}}$ as described in Figure 5 does not necessarily exist in the real world, but *it can be emulated* using cryptographic tools as a protocol consisting of two (or more) entities sending messages to each other over a network. Guarantees in these protocols can be given if at least one of the participants is acting honestly throughout the process.

Two parties $\mathcal{P}_1, \mathcal{P}_2$ can talk to this trusted third party.

Input: Upon message $(\texttt{Input} - \mathcal{P}_1, x_1)$ from $\mathcal{P}_1$ and $(\texttt{Input} - \mathcal{P}_2, x_2)$ from $\mathcal{P}_2$ store x_1, x_2 locally.

Compute: Upon input $(\texttt{Compute}, K)$ from $\mathcal{P}_1$ and $\mathcal{P}_2$ and if x_1, x_2 have been stored:

1. Check if x_1, x_2 have suitable size for the circuit K. If not, output $(\texttt{Abort})$.
2. If x_1, x_2 have suitable size then compute $(y_1, y_2) = K(x_1, x_2)$ and store y_1, y_2 locally.

Output: Upon input $(\texttt{Output})$ from $\mathcal{P}_1$ and $\mathcal{P}_2$ and if y_1, y_2 have been computed, send y_1 to $\mathcal{P}_1$ and y_2 to $\mathcal{P}_2$.

Figure 5: A Trusted Third Party $\mathcal{F}_{\texttt{SC}}$ for Secure Computation.

The two most popular approaches for implementing Figure 5 are based on cryptographic paradigms called *Fully Homomorphic Encryption* (FHE) and *Secure Multiparty Computation* (MPC). For comparison, current FHE schemes are constrained by their demand for computational power and they at best can evaluate a few hundred AND-gates of the circuit K per second. MPC on the other hand, which has a higher demand in terms of communication, can achieve a much better throughput. In particular, there exist MPC schemes that are tailored at efficiently implementing the function $M(\cdot)$, such as e.g. [5, 9].

B Implementing the Framework

In this section we describe how to implement the framework from Section 3 using the tests from Section 4. While the implementation is described at a high level, it is easy to instantiate each of the components based on existing cryptographic tools and the experimental results from Section 5.

B.1 Creating the Test

Consider a design of an interactive test based on the set $T = \{(x_1, g_1, y_1), ..., (x_m, g_m, y_m)$ and parameters δ, ϵ as follows:

1. The regulator $\mathcal{R}$ computes the minimal m_g fulfilling Equation (3) by assuming $EFG = 0$. If T does not contain enough samples from each group, then $\mathcal{R}$ aborts. If $\mathcal{R}$ does not abort, it tells $\mathcal{S}$ the total number of inputs m that will be checked.
2. $\mathcal{R}$ and $\mathcal{S}$ run a secure computation of a functionality $\mathcal{F}_{\texttt{Check}}$ which is described below. $\mathcal{S}$ inputs M into $\mathcal{F}_{\texttt{Check}}$ while $\mathcal{R}$ inputs $(\{(x_i, g_i, y_i)\}_{i \in [m]})$. The functionality $\mathcal{F}_{\texttt{Check}}$ consists of the following steps:
   (a) Compute $\hat{y}_i \leftarrow M(x_i)$ for all $i \in [m]$.
   (b) For all $i \in [m]$, compute a bit b_i as 1 if $\hat{y}_i = y_i$ and 0 otherwise.
   (c) Based on the b_i values, compute for each group g the empirical risk $\bar{\ell}_g(M, T)$ based on Eq. 1.
   (d) Based on the result of the previous step, evaluate Eq. (3) of Theorem 1 (checking ϵ-fairness). Output 1 if the statement holds and 0 otherwise.

Based on the statement of Theorem 1 it follows that $\mathcal{F}_{\texttt{Check}}$ will output 1 if and only if the model M provided by $\mathcal{S}$ is ϵ-fair with confidence $1 - \delta$.

The secure computation of $\mathcal{F}_{\texttt{Check}}$ implements the functionality as a Binary circuit K that is evaluated on secret inputs. We examine the size of this circuit in Section B.4.

A test using augmented data If $\mathcal{R}$ instead wishes to use public and augmented data as for Theorem 2 then this will only work assuming that M is (ϵ, α)-detectable as defined in Definition 1. In such a setting $\mathcal{R}$ would now create a test set T' from T locally using an augmentor $\texttt{aug}$ and then follow the exact same path as for the public data (albeit with different constants).

B.2 Algorithms

We now describe how to use the circuit K from Section B.1 to implement the framework. The overall approach is as follows: Initially, $\mathcal{R}$ generates a signature key pair and distributes the verification key to all other participants. Then $\mathcal{R}$ and $\mathcal{S}$ run a secure computation which runs the test of Section B.1 and computes a Merkle tree hash $\hat{H}_k(M)$ of the model M. If the test finds that the model is fair, then $\mathcal{R}$ signs ($\hat{H}_k(M), \epsilon, \delta$, fairness definition string) and sends the signature to $\mathcal{S}$. By signing ϵ, δ and a fairness definition string we allow multiple fairness definitions and hyperparameters to be certified.

Later, whenever $\mathcal{S}$ and $\mathcal{C}$ run a certified inference for ϵ, δ and a fairness definition, then in addition to running a secure computation of $M(x)$, the functionality will also recompute the hash $\hat{H}_k(\cdot)$ of the model provided by $\mathcal{S}$ and output it to $\mathcal{C}$, while $\mathcal{S}$ sends the signature on the model to $\mathcal{C}$. $\mathcal{C}$ can then locally check if $\mathcal{R}$ originally issued the signature on the hash for those hyperparameters and fairness definition, given the public verification key of $\mathcal{R}$. The overall protocols are outlined in Figure 6.

We will have three participants $\mathcal{S}, \mathcal{C}, \mathcal{R}$ as outlined before. Let ($\texttt{KeyGen}, \texttt{Sign}, \texttt{Verify}$) be a signature scheme and $H_k(\cdot)$ be a collision-resistant hash function whose key k is a common input to all parties. Moreover, let $\mathcal{F}_{\text{SC}}$ be a functionality for secure computation as outlined in Figure 5. $\mathcal{S}$ has a model M as input, $\mathcal{R}$ has a fairness validation set T as well as parameters ϵ, δ and $fair$ the fairness definition string.

Setup: This reflects Step 1 of Figure 1.

1. $\mathcal{R}$ uses $\texttt{KeyGen}$ to generate a key pair (sk, vk). $\mathcal{R}$ keeps sk private and sends vk to $\mathcal{C}, \mathcal{S}$ as $\texttt{cert}_{ID}$.

Certification: This reflects Steps $2, 3$ of Figure 1.

1. $\mathcal{S}, \mathcal{R}$ input the same values into $\mathcal{F}_{\texttt{Check}}$ as they do in Section B.1.
2. Let K be the circuit as outlined in Section B.1. Create a circuit $K_{\texttt{cert}}$ that performs the following:
   First run K on the respective inputs as before, computing $\mathcal{F}_{\texttt{Check}}$. Denote the output bit of this circuit as b. Then compute the Merkle tree output $h \leftarrow \hat{H}_k(M)$ based on the hash function H_k. Finally, output (b, h) to $\mathcal{R}$.
3. Both parties run a secure computation of $K_{\texttt{cert}}$ using $\mathcal{F}_{\text{SC}}$.
4. If the output bit b is 1, then $\mathcal{R}$ computes $\sigma(h) \leftarrow \texttt{Sign}_{sk}(h, (\epsilon, \delta, fair))$ and sends it as $\texttt{cert}_M$ to $\mathcal{S}$.

Inference: This reflects Steps $4, 5$ of Figure 1.

1. $\mathcal{S}$ sends σ to $\mathcal{C}$. $\mathcal{C}$ also knows the signature verification key vk.
2. $\mathcal{S}$ and $\mathcal{C}$ run a secure computation, where $\mathcal{S}$ inputs $\tilde{M}$ and $\mathcal{C}$ inputs its input x. For this secure computation they construct a circuit $K_{\texttt{inf}}$ as follows:
   (a) Compute $\hat{y} \leftarrow \tilde{M}(x)$.
   (b) Compute $\tilde{h} \leftarrow \hat{H}_k(\tilde{M})$.
3. $\mathcal{C}, \mathcal{S}$ run a secure computation of $K_{\texttt{inf}}$ using $\mathcal{F}_{\text{SC}}$. $\mathcal{C}$ obtains as output $(\hat{y}, \tilde{h})$ while $\mathcal{S}$ does not obtain anything.
4. $\mathcal{C}$ computes $b \leftarrow \texttt{Verify}_{vk}(\tilde{\sigma}, (\tilde{h}, \epsilon, \delta, fair))$. If this is true then $\mathcal{C}$ accepts $\hat{y}$. Otherwise it rejects it.

Figure 6: Protocol $\pi_{\texttt{Framework}}$ for Certified Inference

B.3 Security

We now give a sketch of the argument about the security of $\pi_{\texttt{Framework}}$ with respect to Section 3. This must naturally stay on a high level, since we did not make the security properties of the framework formal.

First, we note that $\pi_{\texttt{Framework}}$ leaks to $\mathcal{R}$ and $\mathcal{C}$ the Merkle-tree hash h of the model. But since it can be assumed that M has high entropy and the implementation of H_k is a cryptographic hash function, the leakage of h should be tolerable.[2] That being said, we base our security argument on statements

[2] It is possible in principle to reduce this leakage by computing $\mathcal{R}$'s signature of h, and the signature verification by $\mathcal{C}$, in a secure computation, but this will considerably increase the overhead. In the other direction, if we are willing to leak some more information then the circuit K can be modified to output to $\mathcal{R}$ whether M successfully classified each input x_i and let $\mathcal{R}$ compute the ϵ-fairness of the model locally. This will simplify the secure computation at the cost of leaking more data to $\mathcal{R}$.

about the security of the building blocks that are used, which can be instantiated using well-known cryptographic constructions:

- *The functionality $\mathcal{F}_{\texttt{SC}}$ can be implemented using a secure protocol.* As mentioned in Section 2 this can be done using secure two-party or multi-party computation (MPC).
- *There exist secure signature and hashing schemes.*

Given these primitives, we can assume that the certification and inference steps of Figure 6 are as secure as if they were computed by a trusted party: Assume that in the inference step, the signature $\tilde{\sigma}$ and output $\tilde{h}$ of $\mathcal{F}_{\texttt{SC}}$ are validated. This can only happen due to 3 cases: (i) $\tilde{\sigma}$ was generated for $\tilde{M}$ by $\mathcal{R}$ (which is the desired course of events); (ii) $\tilde{\sigma}$ was issued by $\mathcal{R}$ but for a different $\tilde{M}'$; or (iii) $\tilde{\sigma}$ was never issued by $\mathcal{R}$. In the last case, $\mathcal{S}$ must have broken the security of the signature scheme. In the second case, $\mathcal{S}$ must have broken the collision-resistance of H_k. Therefore, either $\mathcal{S}$ managed to break the signature scheme or the hash function, or $\mathcal{R}$ signed $\tilde{M}$. $\mathcal{R}$ computes this signature if and only if the model passed the test of Section B.1. Based on the statement of Theorem 1 it follows that this test passes if and only if the model $\tilde{M}$ provided by $\mathcal{S}$ is ϵ-fair with confidence $1-\delta$.

B.4 Efficiency

We now estimate the efficiency of implementing our framework using $\pi_{\texttt{Framework}}$. We first claim that it only makes sense to run our framework in settings where the ML inference is done using a secure computation: If the inference is not computed using a secure computation, then one option is for the client to learn the model and run by itself a check for fairness, or send the model to another party and ask it to do this check. Another option is that the client simply hands over its input to the model owner, but this would require prohibitively expensive zero-knowledge proofs, to be computed at the owner side, to attest to fairness of the output without revealing anything about the model.

Therefore, given that inference is done via secure computation, the parties must incur the cost of running a secure computation of the inference, and the efficiency of the framework should be measured by the additional overhead that is added on top of the secure inference.

The main computational tasks that are run by $\pi_{\texttt{Framework}}$ are as follows:

- The **Certification** phase runs m instances of a secure computation of inference and in addition computes a hash of the model and checks the accuracy of the output.
- The **Inference** phase runs a single secure computation of the inference and in addition computes a hash of the model.

The **Certification** phase is a one-time event, and therefore its overhead is less critical. Theorem 1 shows that the number of samples m_g per group should be $m = \frac{2}{(EFG-\epsilon)^2} \ln \frac{2|G|}{\delta^2}$. Setting for example $EFG = 0.05, \epsilon = 0.1, \delta = 0.2$ and considering $|G| = 100$ groups, we get that $m_g \approx 6800$, which does not seem to be too far off from existing training set sizes.

In more detail, we describe here the cost of implementing the different steps of the circuit K which computes the certification, as described in Section B.1: Step (a) needs to implement the inference m times. This is by far the largest component of the circuit. Step (b) computes m comparisons, which are easy. Step (c) computes $\bar{\ell}_g(M,T)$ for each group g, based on Eq. 1. This computation must sum the b values for each group g. To make this step efficient, the circuit must hard-wire the connections for these summations, and the locations of the inputs from each g can be known. (There is no need to hide these locations from $\mathcal{S}$.) Eq. 1 also computes a division by m_g, but there is no need to compute the division and the circuit forwards $m_g \cdot \bar{\ell}_g(M,T)$ to the next step. Step (d) tests Eq. 3 for each pair of g_0, g_1, namely computes $\bar{\ell}_{g_0}(M,T) - \bar{\ell}_{g_1}(M,T)$. Since the input to this step is $m_{g_i} \cdot \bar{\ell}_{g_i}(M,T)$ then the test in this equation should be changed appropriately (which is straightforward, especially if $m_{g_0} = m_{g_1}$).

As for the cost of computing $M(\cdot)$, current secure computation implementations for this task only hide the weights of a DNN but reveal the actual network structure and activation functions. We assume that our secure computation will also only hide the weights as this seems to be a standard assumption. Therefore, we ask what is the additional cost of hashing this data over the default cost of using the weights in the computation of the model.

There is a lot of current work on lightweight hashing schemes for usage in zero-knowledge proofs, and it is reasonable to expect that a lot of improvements in this area will be made in the near future. As a baseline, we consider the Keccak-F function, which is the basis of the SHA3 standard. That function takes a 1600 bit input and can be implemented by a Boolean circuit of 38,400 AND gates (see [1]), i.e. 24 AND gates per input bit. If we use a Merkle tree then the total number of hashes is twice the number of input blocks[3]. Therefore, the total cost is about 48 AND gates per input bit.

Now, with regards to the secure evaluation of the model (not considering special MPC implementations for secure inference[4]), let us consider a setting where the weights have 32 bit fixed-point values. The cost per each weight (when used in DNN inference) must be at least that of multiplying the weight with either an input or output of a hidden layer and adding all these products together (neglecting the cost of the activation function). Multiplying the weight with a 32 bit value costs 185 ANDs per input bit, while adding up the result would only require 6 ANDs per input bit (see [1]), and we therefore take the assumption that the total cost of the secure computation is 191 AND gates per bit of the weights (neglecting the activation function). Therefore the fairness verification increases the cost of inference in this model by only about 25%. While using optimized implementations for inference will make the additional overhead from hashing larger, we can in practice lower the cost of hashing drastically by exploiting special properties of $\mathcal{F}_{\texttt{SC}}$ which allow the use of homomorphic commitments. We leave such specialized hashing techniques as interesting future work.

C Theorems Proofs

C.1 Theorem 1

Using Hoeffding's concentration bound we get that:

$$\Pr\left[\left|\bar{\ell}_g(M,T) - \ell_g(M)\right| > \frac{\epsilon - EFG}{2}\right] \leq 2e^{-m_g \frac{(\epsilon - EFG)^2}{2}} = \frac{\delta}{|\mathcal{G}||\mathcal{Y}|}$$

By a union bound it follows that $|\bar{\ell}_g(M,T) - \ell_g(M)| \leq (\epsilon - EFG)/2$ for all $g \in \mathcal{G}$ with probability $1-\delta$. Given that this event holds then, by applying the triangle inequality twice, for any $g_0, g_1 \in \mathcal{G}$ we have:

$$|\ell_{g_0}(M) - \ell_{g_1}(M)| \leq |\ell_{g_0}(M) - \bar{\ell}_{g_0}(M,T)| + EFG + |\bar{\ell}_{g_1}(M,T) - \ell_{g_1}(M)| \leq \epsilon$$

Hence $\max_{g_0,g_1 \in \mathcal{G}} |\ell_{g_0}(M) - \ell_{g_1}(M)| \leq \epsilon$ with confidence $1-\delta$. Similar arguments show the same is true for equalized odds and demographic parity.

C.2 Theorem 2

Similar to the proof in C.1, we get that:

$$\Pr\left[\left|\bar{\ell}_{g,\texttt{aug}}(M,\tilde{T}) - \ell_{g,\texttt{aug}}(M)\right| > \frac{\alpha - EFG}{2}\right] \leq \frac{\delta}{|\mathcal{G}||\mathcal{Y}|}$$

which implies that $|\ell_{g_0,\texttt{aug}}(M) - \ell_{g_1,\texttt{aug}}(M)| \leq \alpha$ with confidence $1-\delta$ for all groups. Since $\mathcal{M}$ is (ϵ, α)-detectable fairness, then M is ϵ-fair with the same confidence.

D Experiments Details

D.1 Datasets

- **UTKFace** [33] is a dataset of face images with attribute annotation for age, ethnicity (called *race* and annotated as *black, asian, white* or *other*), and gender (*male* or *female*). We

[3] We can improve on that by having the circuit output to $\mathcal{C}$ the results of the first layer of the Merkle tree, and have $\mathcal{C}$ locally compute the rest of the tree. For this to work, we will on the other hand have to add random values to each input block to avoid lookup table-based attacks on preimages of H_k.

[4] This analysis neglects recent works such as e.g. [5] that apply to special types of networks only. We believe that the accuracy of the networks such as MobileNets that are used in [5] is too low to be of use for fairness testing.

focused on gender prediction as a task across two ethnicity groups *black* and *white* and discarded all other samples, so we were left with 14,604 samples, which we split equally to train and test sets. The dataset consists of 70% *white* and 30% *black*, 53% *male* and 47% *female*.

- **MNIST** is originally a dataset of hand-written digits from 0 to 9. The dataset is used to predict the digit in the image without additional annotations, hence there are 10 classes across the dataset without any allocation of groups. We changed the task to a binary classification and synthetically generated two fairness groups of digits based on MNIST data. Therefore, we assign the label 0 to the digits 0-4 and the label 1 to the digits 5-9. We randomly colored half of the dataset's digits in red as was done in [3], resulting in 50% red digits and 50% white digits – these were the fairness groups. We called this dataset *C(olored)-MNIST*.
- **LFW** [15] is a dataset of face images with attributes annotation [23]. Using the "Black" attribute we divided the data into two groups, while using "Male" as a binary label.
- **CelebA** [26] is a face recognition dataset consisting of more than 10,000 different celebrities with gender labelling. We annotated 8,500 celebrities out of 10,177 in the dataset for ethnicity using Amazon Mechanical Turk. Three turks annotated three images of each of the 8,500 celebrities, resulting in 177,683 images classified as either Asian, African, Caucasian or Other[5]. During our experiments we merged all but the Caucasian group to produce a large dataset to showcase our setup, having over 30,000 samples for the minority group.
- **Adult Income** [22] is a tabular features dataset with a label for low/high income. We used the gender feature as group affiliation and income for labels. During the preprocessing all numeric features were normalized, while categorical features were transformed into one-hot vectors in order to be used later by DNN models.
- **TIMIT** [13] is a voice recognition dataset with dialects and gender annotation. We used the different dialects as groups and gender of speaker as label. To have more samples per dialect, we merged dialects which have much in common and are considered similar, namely we merge New England with New York City and Northern with North Midland, while discarding the rest.

Dataset	Size	g_0, y_0	g_0, y_1	g_1, y_0	g_1, y_1
UTKFace	14,604	37.5%	31.51%	15.87%	15.12%
LFW	13,144	74.22%	21.52%	3.24%	1.02%
CelebA	177,683	33.97%	49.26%	8.67%	8.11%
C-MNIST	70,000	25%	25%	25%	25%
Adult Income	48,842	46.54%	20.3%	29.52%	3.62%

Table 2: Data distribution across groups and labels for each of the datasets.

D.2 Private Data Setup Full Results

Full results for overall risk equality, equalized odds and demographic parity can be found in Table 3.

D.3 Public Data Setup Full Results

We've experimented with cutting the UTKFace dataset in half, to see how it affects the margin and α. We also have results for the LFW dataset. Since we could not generate a fair model in LFW, we have no reference or evidence of margin, but empirically the EFG seems high suggesting the augmentation would work on it as well. Results are in Table 4.

D.4 ODIN Tuning

We tuned ODIN's 3 hyperparameters - T temperature, ϵ perturbation and δ threshold. We chose T from among $\{1, 10, 100, 1000\}$, ϵ from 30 evenly spaced numbers between 0 and 0.01 and took

[5] The annotations can be downloaded from `https://github/will/be/published/`

Dataset	Model	Accuracy	Risk Equality		Equalized Odds		Demographic Parity	
			EFG	ϵ-Test	EFG	ϵ-Test	EFG	ϵ-Test
UTKFace	ResNet18	89.76	0.012	Failed†	0.067	Failed	0.007	Failed†
UTKFace	Bias-ResNet18	88.56	0.093	Failed	0.115	Failed	0.088	Failed
CelebA	ResNet18	97.63	0.007	Passed	0.017	Passed	0.083	Failed
CelebA	Bias-ResNet18	96.95	0.034	Failed	0.045	Failed	0.039	Failed
C-MNIST	LeNet	98.11	0.001	Passed	0.022	Failed†	0.001	Passed
C-MNIST	Bias-LeNet	74.01	0.450	Failed	0.485	Failed	0.464	Failed
LFW	ResNet18	91.06	0.049	Failed†	0.398	Failed	0.065	Failed
Adult Income	MLP	84.17	0.108	Failed	0.377	Failed	0.182	Failed
TIMIT	LeNet	89.07	0.040	Failed†	0.117	Failed	0.082	Failed

Table 3: Fairness test using private data with $\epsilon = 0.05$ and $\delta = 0.05$. "Failed†" refers to insufficient sample size to certify the fairness of the model.

Dataset	Model	Accuracy		Risk Equality EFG	
		Fair	Bias	Fair	Bias
UTKFace - Half Size	ResNet18	92.13	87.03	3.56	12.91
UTKFace	ResNet18	96.11	91.44	2.72	13.88
C-MNIST	LeNet	89.17	67.97	0.65	34.04
LFW	ResNet18	91.98	-	7.45	-
CelebA	ResNet18	96.60	97.02	1.01	3.28

Table 4: Fairness test using public data with an augmentor on the 4 image datasets. "Half Size" refers to the dataset with half of the samples removed.

those which yielded the best results for any $\delta \in [0, 1]$. We note that the hyperparameter tuning had little effect, as most of the values chosen performed very similarly.

D.5 Testing with unknown margin

In certain scenarios it might be hard to determine α necessary for the fairness test in advance. For example, UTKFace in the augmented public data setup has a different fairness gap than the one seen when private data is used, and the fairness gap is influenced by the sample set size. To further investigate the nature of our augmentation under the assumption that the gap is unknown, we tested the models under an increasingly larger degree of augmentation (frequency that each augmentation is invoked) for each sample. The changes in accuracy and fairness gap are presented in Table 5. The accuracy decreases as we increase the augmentation degree, which fits the idea that it might be hard to generalize on the augmented data. Hence, more augmentation yields less accuracy. The fairness gap, on the other hand, had inconclusive results: increasing the augmentation degree had little or no effect on CelebA dataset, while it greatly varied on UTKFace between fair and unfair models.

Augmentation	Dataset	Fair Model		Unfair Model	
Degree		Acc.	Risk-Eq EFG	Acc.	Risk-Eq EFG
25%	UTKFace	97.08	3.56	91.74	18.51
50%	UTKFace	94.58	2.03	91.14	15.4
75%	UTKFace	91.74	3.15	88.16	13.36
25%	CelebA	97.24	1.03	96.99	4.82
50%	CelebA	95.84	0.84	96.08	4.33
75%	CelebA	93.84	0.79	94.67	4.42

Table 5: Accuracy and overall risk equality EFG for different degrees of augmentation