Fairness Audit of Machine Learning Models with Confidential Computing

Saerom Park
Sungshin Women's University
Republic of Korea
psr6275@sungshin.ac.kr

Seongmin Kim*
Sungshin Women's University
Republic of Korea
sm.kim@sungshin.ac.kr

Yeon-sup Lim
Sungshin Women's University
Republic of Korea
ylim@sungshin.ac.kr

ABSTRACT

Algorithmic discrimination is one of the significant concerns in applying machine learning models to a real-world system. Many researchers have focused on developing fair machine learning algorithms without discrimination based on legally protected attributes. However, the existing research has barely explored various security issues that can occur while evaluating model fairness and verifying fair models. In this study, we propose a fairness audit framework that assesses the fairness of ML algorithms while addressing potential security issues such as data privacy, model secrecy, and trustworthiness. To this end, our proposed framework utilizes confidential computing and builds a chain of trust through enclave attestation primitives combined with public scrutiny and state-of-the-art software-based security techniques, enabling fair ML models to be securely certified and clients to verify a certified one. Our micro-benchmarks on various ML models and real-world datasets show the feasibility of the fairness certification implemented with Intel SGX in practice. In addition, we analyze the impact of data poisoning, which is an additional threat during data collection for fairness auditing. Based on the analysis, we illustrate the theoretical curves of fairness gap and minimal group size and the empirical results of fairness certification on poisoned datasets.

CCS CONCEPTS

• **Security and privacy → Privacy-preserving protocols**; • **Computing methodologies →** *Supervised learning*.

KEYWORDS

Fairness, Algorithmic audit, Security and privacy, Confidential computing

ACM Reference Format:
Saerom Park, Seongmin Kim, and Yeon-sup Lim. 2022. Fairness Audit of Machine Learning Models with Confidential Computing. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3485447.3512244

1 INTRODUCTION

Nowadays, Machine Learning (ML) models developed with a large amount of data can address various real-world problems such as image classification, text classification, health monitoring, and churn prediction from Web logs [2, 42, 43]. However, ML researchers and practitioners have raised concerns about bias and discrimination in ML-based automated decision-making, where unfairness can come from various sources ranging from inherent bias in training data to amplified bias during training procedures. In EU General Data Protection Regulation (GDPR), Recital 71 refers in particular to fairness-aware processing and data mining technologies [60]. In the ML context, many fair learning methods have been focused on algorithmically mitigating biases while attaining efficient trade-offs between model accuracy and fairness.

Algorithmic fairness is mainly related to legally protected characteristics such as disability, race, and gender, which are sensitive information that can potentially disclose and breach individual privacy [5, 36]. Thus, organizations or corporations implementing algorithmic decision-making systems (i.e., modelers) can not use the protected characteristics to train ML models. Fair learning approaches [36, 41] have been proposed to work with such a restriction, i.e., training fair models without the protected characteristics. However, it is still required to have the protected characteristics for assessing the fairness of the trained model.

To resolve the lack of available sensitive information during fairness audit on ML software, Veale and Binns [60] postulate that a third party (i.e., regulator) possesses test data with the protected characteristics for the fairness evaluation. In this approach, there should be a strong trust relationship between the regulator and the modeler: the regulator has to trust the ML prediction results from the modeler, or the modeler should allow access to its algorithm and code if the regulator performs a direct audit on the algorithm. The problem is that full access to ML models can invade the modeler's intellectual property. Therefore, companies might have their own internal auditing teams or turn to the private auditing firm [45]. Several studies [31, 36] proposed a public fairness audit utilizing multi-party computation (MPC) to protect model confidentiality and data privacy. Although those approaches enable a public fairness audit under the assumption of semi-honest security, they require significant computational overhead.

In this study, we propose a public fair audit framework that assesses the fairness of ML models and attests to the fairness-aware ML models. Our framework provides a technical solution to enable secure fairness audit by leveraging confidential computing based on hardware enclave. We address critical design challenges in facilitating fair ML audit under a practical threat model. Our key contributions are as follows:

*corresponding author.

WWW '22, April 25–29, 2022, WWW, Lyon
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512244

- We provide a generic fairness audit framework that is not dependent on any particular application or concept of fairness so that the framework can be applied to various algorithmic fairness problems.
- We explore potential threats or attacks in the fairness certification/verification process from the perspective of all participants, considering various adversaries such as a malicious modeler, a curious or disguised regulator or data owner, or an outside invader, described in §3.2.
- We design our framework to address these security issues by leveraging hardware enclaves in confidential computing, enhancing security mechanisms based on enclave remote attestation, and constructing the chain of trust from certification to verification through public scrutiny and enclave sandboxing, explained in §4.
- We show that our framework yields a small computational overhead in real-world datasets through experiments using an SGX-based implementation of fairness certification.

The remainder of this paper is organized as follows. §2 provides background on fairness notions in ML and confidential computing. §3 brings up possible threats in realizing a secure fairness audit framework from the perspective of all participants while §4 describes our fairness audit framework based on confidential computing to address the potential threats. We demonstrated the effectiveness of the fairness certification on real-world datasets in §5. §6 and §7 discusses related work and concludes this study.

2 BACKGROUND

2.1 Fairness notions in ML

Various fairness notions have been proposed in the previous studies, such as statistical disparity, equalized odds, and individual fairness [25, 63]. In this study, we propose a fairness audit framework that can be applied to statistical fairness notions that require sensitive group information because the main privacy challenge of fairness auditing comes from restrictions on the collection of sensitive variables. The popular statistical fairness notions include disparate treatment, disparate impact, disparate mistreatment, and equalized odds. Disparate treatment means that the decision-making system gives different outputs even if non-sensitive attributes have the same or similar values, but sensitive attribute values are different. On the other hand, disparate impact exists when the decision-making system provides predictions based on implicit correlations between the outputs and sensitive attributes. Disparate mistreatment considers that misclassification rates are different depending on sensitive attributes. Equalized odds imply that the instances from different sensitive groups have the same or similar predictions if they come from the same label group.

Suppose that we have a binary classifier $f(\boldsymbol{x})$, non-sensitive attributes $\boldsymbol{x} \in \mathbb{R}^p$, sensitive attribute $z \in \mathcal{Z}$, class label $y \in \{-1, 1\}$ and the estimated output $\hat{y} = \text{sign}(f(\boldsymbol{x}))$. Then, the above notions can be represented as:

- Disparate treatment:$P(\hat{y}|\boldsymbol{x}, z) \neq P(\hat{y}|\boldsymbol{x}), \forall z$
- Disparate impact (DI):$P(\hat{y} = 1|z) \neq P(\hat{y} = 1), \forall z$
- Disparate mistreatment
  - Overall misclassification rate (OMR): $P(\hat{y} \neq y|z) \neq P(\hat{y} \neq y), \forall z$
  - False positive rate (FPR): $P(\hat{y} \neq y|y = -1, z) \neq P(\hat{y} \neq y|y = -1), \forall z$
  - False negative rate (FNR): $P(\hat{y} \neq y|y = 1, z) \neq P(\hat{y} \neq y|y = 1), \forall z$
- Equalized odds: $P(\hat{y} = 1|y, z) = P(\hat{y} = 1|y), \forall y, z$

In this study, we implement all of the above notions for fairness auditing. In a regression problem, for a predictor $g(x) : \mathbb{R}^p \to \mathbb{R}^q$, the following notions are relevant Agarwal et al. [3]:

- Statistical parity: $P(g(x) \geq a|z) \neq P(g(x) \geq a), \forall z, a$
- Bounded group loss: $\mathbb{E}[\ell(y, g(x))|z] \leq \zeta, \forall z$

It is difficult to obligate ML providers to meet the fairness notions through laws and regulations across the use cases because the fairness notions associated with ML models are highly dependent on the application domains. Therefore, we propose a certification-based framework that encourages ML providers to construct fair models that are attractive to their clients.

2.2 Confidential Computing

Confidential computing is a new paradigm in cloud computing to keep privacy-sensitive data more safe and secure [1]. By leveraging commoditized trusted execution environment (TEE) technology provided by CPU vendors [8, 26], it enables service providers to achieve isolated execution of their services within a hardware-protected memory region—*an enclave*. With a processor-specific key, the CPU package cryptographically protects an execution of enclave code from underlying software components running in the host system, including OS and hypervisor. This introduces a new opportunity to service providers who are reluctant to migrate privacy-sensitive services to the untrusted cloud [10]. In fact, the execution model of confidential computing perfectly fits well with privacy-preserving ML services running on the public cloud infrastructure; researchers have proposed TEE-based ML/DL prediction and training systems [21, 27, 32, 39, 46, 58, 66].

Recent hardware-based TEEs support enclave attestation that enables proving the integrity and genuinity of enclaves running on the remote platform (e.g., cloud) [4]. For example, in the case of Intel SGX, a verifier can validate a report created by a target enclave by asking Intel Attestation Service (IAS) that the report is signed with a valid attestation key. Then, the verifier can figure out whether the enclave is loaded on the real SGX hardware or not (e.g., emulation). In addition, the attestation procedure contains cryptographic verification on the enclave's hash measurement to check its integrity. Note that it is possible to establish a secure channel by combining TLS handshake[23, 55] or Diffie-Hellman key exchange (DHKE) [4] with enclave attestation.

3 PROBLEM STATEMENT

In this paper, we aim to develop a framework for auditing the fairness of an ML model and certifying/verifying a fair model to enable clients to use a certified ML model for inferences while preserving data privacy and model confidentiality.

3.1 Deployment Scenario

We consider a scenario in which four participants work on certification, verification, and use of an ML model: 1) multiple data owners who provide data for model fairness test, 2) a regulator that performs fairness audits on an ML model, 3) a modeler (ML service provider) that trains a model and provides its inference API, and 4) clients who request ML inferences to the modeler.

In this scenario, the modeler wants to demonstrate the fairness of their ML model and obtain a certificate of fairness. Then, the regulator tests and certifies/verifies an ML model from the modeler so that clients can safely use the modeler's fair model for inferences. The regulator's model fairness test is based on data given from multiple data owners. To ensure no interest between a regulator and ML modeler, we assume that the regulator is not involved in a model training. As a result, we can summarize the requirements and capabilities of each participant as follows:

- **Modeler** provides inferences API based on its trained ML model to a regulator and clients without exposing the secrets of the model, such as code, weights, and parameters.
- **Regulator** needs validation datasets that contain sensitive information. Given a dataset, a regulator can compute fairness metrics like those described in §2.1 to evaluate the fairness of ML models. Based on the values of the obtained metrics, it determines whether a model is fair according to its own policy and issues the certificate of fairness.
- **Data owners (Owners)** provide datasets that a regulator leverages to evaluating the fairness of ML models. Throughout the entire certification procedure, including data transfer and fairness evaluation, they want to protect the privacy of the entire data as well as sensitive information to stakeholders (e.g., face, textual, and speech data), such as other participants and a regulator.
- **Clients** want to use a trustworthy ML inference API that guarantees fairness while preserving the privacy of their request data.

3.2 Potential Threats and Challenges

Given the requirements and capabilities in §3.1, we consider potential threats and attacks from the perspective of each participant. We assume that the attacker could be a malicious modeler, a curious or disguised regulator or data owner, a cloud service provider (if required), or an outside invader.

Modeler-side (T1) When a public regulator verifies the modeler's ML API, there exists a risk that a mistrustful regulator asks the details about its ML model. If a malicious attacker imposters such a regulator, the attacker can exploit the information to extract the modeler's secrets. To avoid the threat, the corporations entrust the fairness evaluation of their ML software to private auditing firms that might similarly request access to the codes of their ML software but are trusted. However, the clients (ML API users) might doubt the authenticity of the verification results by private auditors.

Regulator-side (T2,T3) A malicious modeler might ask a regulator to certify an unfair model as a fair one. To this end, the modeler submits a fair model, which is not an exact one to be evaluated, during the regulator's test, and then deploys the unfair model after obtaining certification (**T2**). Also, a regulator worries about the reliability of the collected data from multiple data owners (e.g., data poisoning on sensitive group information) (**T3**).

Data-owners-side (T4) Data owners can lose data privacy and confidentiality by the impersonate attack that disguises a regulator or eavesdropping on the communication between data owners and a regulator. Although prior studies proposed a cryptographically protected fair certification/verification framework using MPC [31, 54], they had high communication costs between two parties and protected only sensitive group information (i.e., demographics) because of computational complexity.

Clients-side (T5,T6) The ML service users can be deceived by certification forgery (**T5**). For example, a malicious modeler might manipulate the certificate of ML fairness for itself without going through the regulator or provide an unfair ML API only at runtime. Moreover, clients worry that their data will be stolen by modelers or other outside attackers in unintended ways while requesting the inference results to the ML API (**T6**).

Several fair ML studies have been interested in some threats above [31, 41, 54, 60] and fairness evaluation or audit [11, 12, 30, 52, 53]. However, we have found that existing workflows of fairness assessment do not lend themselves to address the threats (**T1-T6**). We aim to propose a secure fairness audit framework to deal with the aforementioned security issues. The following section will describe the system components that constitute our framework and how each component can be helpful to mitigate **T1-T6**.

The considered scope of threat model. Although we explore various potential threats in the fairness audit process in this study, we do not consider how to build fair ML models or how to cope with possible attacks in the inference phase, such as model extraction, inversion, and evasion attacks. [19, 20, 38, 57, 59]. In our system architecture, we assume that an enclave and the CPU package are the only trustworthy components; thus, as in the traditional threat model using hardware enclaves [10, 26], it is possible that an attacker controls the entire software stack, including the operating system and hypervisor. Also, we consider the case where only CPU processes workloads for fairness certification to securely protect Fair ML models and their prediction routines within an enclave; we do not deal with untrusted computing units such as GPU. Note that we do not consider side-channel attacks on TEE because enclave hardening is beyond the scope of this study.

4 PROPOSED FRAMEWORK

In this section, we present our fairness audit framework that aims to counter the threats (**T1-T6**) presented in §3.2 by leveraging hardware enclaves in confidential computing. Regulator, Modeler, Owners and Clients denote regulator, modeler, data owners and clients in §3.1, respectively.

4.1 System Overview

Our fairness audit framework consists of sub-modules for the participants (Regulator, Modeler, Owners and Clients) described in §3.1. While most fairness audit processes concentrate on the evaluation of ML fairness, our framework excogitates a secure fair ML audit, from assessing the fairness of ML APIs (certification) to ensuring the use of certified fair ML APIs (verification).

Figure 1 illustrates the overview of the interaction between the modules in the certification phase in our proposed framework. Regulator consists of fairness auditing and data aggregation enclaves:

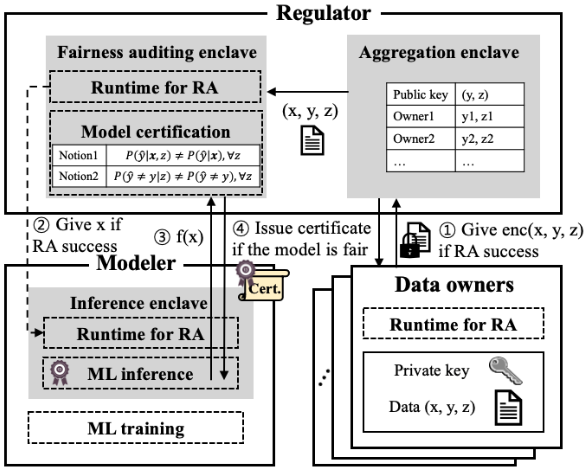


Figure 1: Model certification overview. Note that "RA" denotes remote attestation and "ML" denotes machine learning.

1) The fairness auditing enclave certifies and verifies an ML API trained by Modeler. Modeler requests its model certification to the fairness auditing enclave. The fairness auditing enclave checks the fairness of the model based on inference results on test data, and it issues a certificate if the model is fair. 2) The data aggregation enclave manages and collects the fairness test data sent from Owners. We explain the detailed design of certification phase in §4.2 and discuss potential issues in the the data aggregation enclave in §4.4. After Regulator successfully certifies the fair ML APIs, our framework performs the verification phase related to running the ML inference API in practice (See Figure 2). If Clients query whether a model is fair to Modeler, Modeler's inference enclave replies with the certificate issued by the fairness auditing enclave in Regulator. Then, Clients validate the model's fairness through the certificate so that it obtains trustworthy inferences from the certified API. We will describe the sub-process for the verification phase in §4.3.

Under our system design, an adversary aims to break the model secrecy (**T1**), disturb the calculation of fairness notions (**T2**, **T3**), uncover sensitive variables during fairness certification by eavesdropping on user input (**T4**), vitiate fairness certificates (**T5**), or swindle a client's data during inference phase (**T6**). We propose design components (**D1**-**D5**) and elaborate them to mitigate the aforementioned potential threats with a theoretical analysis (**A1**).

4.2 Fair Model Certification

In the proposed framework, the fairness auditing and data aggregation enclaves take charge of fair model certification. In the certification phase, we need to consider modeler-side (**T1**), regulator-side (**T2,T3**) and data-owners-side (**T4**) threats. To cope with these threats, we introduce **D1-D2** based on confidential computing with executing multiple enclaves in a public cloud infrastructure.

D1. Secure transmission of sensitive information (T4) Our fairness audit begins with a data aggregation enclave collecting test data from Owners. The communication between data aggregation and owner enclaves is encrypted to prevent data disclosure during the collection process: encrypted data are sent by Owners and decrypted in a hardware enclave by establishing a secure channel. Extending enclave attestation to utilize cryptographic protocols (e.g., TLS or DHKE) can prevent attackers from impersonating Regulator or eavesdropping Owners' data. Note that our privacy protection is not necessarily limited to the sensitive group information z unlike [31, 54]. Also, Clients' data can be similarly protected in verification. Regulator-side threat (**T3**) on the aggregation enclave will be explored in the analysis **A1** in §4.4.

D2. Enclave remote attestation of fairness auditing enclave and ML inference enclave (T1, T2) To request a model certification, Modeler first initiates enclave attestation procedure with the fairness auditing enclave to confirm its model to be certified. Once the attestation completes, the fairness auditing enclave collects the inference results for the fairness test data using the inference API provided by Modeler. The remote attestation verifies the confidentiality of the enclave publisher to prevent an adversary from impersonating participants and verifies the integrity of codes and data to ensure that a malicious modeler cannot compromise the integrity of the inference enclave. In our proposed framework, both Modeler and Regulator mutually perform enclave attestation to authenticate each others' identities, for example, to figure out the disguised regulator. Note that Modeler only exposes the cryptographic hash of the inference enclave and the inference results of the ML model. Therefore, Modeler can protect model confidentiality while requesting the certification procedure, i.e., it does not disclose the code and parameters of the model to any participant. However, it is still difficult for Clients to be aware of whether the attested enclaves only behave in determined ways. Thus, we will propose **D5** to address this issue in §4.3.

Computation of ML model fairness for certification. Using the collected inference results on the test data, the fairness auditing enclave computes the fairness metrics and determines whether the model is fair. Thus, we want to describe the formulation of ML fairness for certification. Suppose that $\mathcal{R}_z(f)$ represents metrics of a group z for fairness of function f such as misclassification, false positive rate, and false negative rate. Without loss of generality, we can consider the misclassification rate (risk) $\mathcal{R}_z(f) = \mathbb{E}_{x,y,z'}[\mathbb{I}[f(x) \neq y|z' = z]]$, where $\mathbb{I}$ is an indicator function. Our framework estimates this group risk with the collected data $\mathcal{D} = \{(x_1, y_1, z_1), \ldots, (x_N, y_N, z_N)\} = \bigcup_{z' \in \mathcal{Z}} \mathcal{D}_{z'}$ where $\mathcal{D}_{z'} = \{(x, y, z) \in \mathcal{D} : z = z'\}$. The estimated group risk is denoted by $R(\mathcal{D}_z) = \frac{1}{m_z} \sum_{i \in \mathcal{D}_z} \mathbb{I}[f(x_i) \neq y_i, z_i = z]$, where $|\mathcal{D}_z| = m_z$. We can define the fairness gap of a classifier f $\max_{z_0, z_1 \in \mathcal{Z}} |\mathcal{R}_{z_0}(f) - \mathcal{R}_{z_1}(f)|$ as in [54]. Based on this, we can define the classifier f is (ϵ, δ)-fair if:

$$Pr\left[\max_{z_0, z_1 \in \mathcal{Z}} |\mathcal{R}_{z_0}(f) - \mathcal{R}_{z_1}(f)| > \epsilon\right] \leq \delta. \tag{1}$$

We can also compute the empirical fairness gap from the given dataset as follows:

$$G(R, \mathcal{D}) = \max_{z_0, z_1 \in \mathcal{Z}} |R(f, \mathcal{D}_{z_0}) - R(f, \mathcal{D}_{z_1})| \tag{2}$$

For simplicity, $G(\mathcal{D})$ denote $G(R, \mathcal{D})$. In the fairness certification phase, we can ensure the fairness of ML model by using the empirical fairness gap (2) as described in the claim of [54]. This empirical gap can be computed for other fairness notions in §2.1. Note that the code for calculating fairness metrics can be shared and agreed upon in public because the code itself does not reveal any confidential information such as data or models. Based on the results of this certification, the fairness auditing enclave issues a certificate of fairness to the target ML inference API. In §4.3, we present how Clients use the ML inference API signed by Regulator.

The most appropriate fairness notion differs according to the real-world problem for which ML models are applied. Our framework adaptively supports other fairness notions since the fairness auditing enclaves can compute them without complex design changes in fairness check algorithms, different from other MPC-based privacy-preserving certification studies [31, 54].

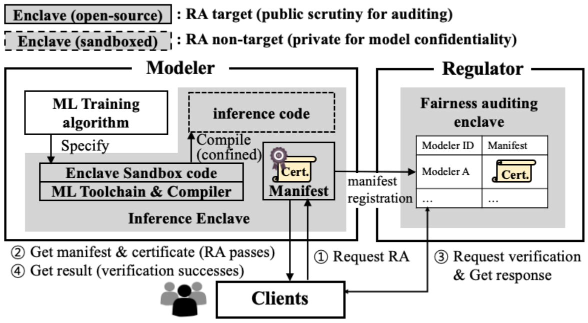


Figure 2: Model verification overview.

4.3 Fair Model Verification

To guarantee the end-to-end trustworthiness from Regulator to Clients, there needs to be a client-side verification mechanism against other participating enclaves regarding a certified model to which Regulator issues a certificate. In our framework, Clients verify the integrity of Modeler’s inference and Regulator’s fairness auditing enclaves through the enclave attestation primitives. However, the following issues still need to be addressed: 1) Clients cannot recognize whether a target enclave to be attested is genuine unless the code of enclave is publicly available and well scrutinized [27, 65]. 2) there is no safeguard for clients to detect misbehavior of Modeler and Regulator; for example, a malicious modeler deploys an unfair model, and a malicious regulator issues a forged certificate. Our framework addresses these issues by introducing **D3-D5**.

D3. Public scrutiny of fairness auditing enclave (T5) Public scrutiny of the fairness auditing enclave helps Clients to validate the integrity of their target regulator’s enclave: they combine the openness with the attestation in **D2** to validate the authenticity of fairness audit conducted by Regulator. If Regulator attempts an unauthorized change on the fairness audit procedure, Clients promptly recognize the integrity of the certification becomes broken by comparing it with the public scrutinized code. Note that the fairness certificates are implemented based on the standard cryptographic libraries (e.g., OpenSSL), so enabling public scrutiny does not reveal any Regulator’s intellectual property or secret.

However, public scrutiny is infeasible in the case of the verification of Modeler’s inference enclave due to privacy concerns. As stated in the threat **T1** of §3.2, Modeler does not want to expose a trained model (e.g., model parameters and the corresponding training algorithm) since the loss of confidentiality of the proprietary model leads to irreparable damage to the disclosing party’s business or an unintended exposure of their vulnerability. Also, enclave attestation cannot verify the deployed certificate residing on Modeler because the procedure does the integrity of the initial code and data of the target enclave, not runtime data retrieved from external entities (e.g., certificates issued from fairness auditing enclave). Therefore, an additional approach is required to enable Clients to audit the issued certificate while preserving the model confidentiality. The model verification procedure has to address two security concerns as discussed in §3.2: 1) How to verify the issued certificate to the ML inference enclave (**T5**). 2) How to verify whether Modeler utilizes the certified model in actual service without disclosing the model confidentiality and Clients’ data (**T1,T2,T6**).

D4. Verifying fairness certificate manifest (T5) After Regulator populates the certificate to Modeler’s inference enclave, Clients verify whether the inference enclave passes the fairness test and Regulator has indeed utilized it for the certification. However, the issued certificate is dynamically loaded (or received from the network) after bootstrapping an enclave, which means that its integrity cannot be checked through remote attestation. To address this, we take a similar approach that measures the integrity of dynamically-linked libraries by pre-registering the specification manifest within an enclave [14]. Our verification procedure imposes an inference enclave to include a hard-coded credential as a manifest (e.g., self-signed certificate by Regulator) before registering the cryptographic hash of an inference enclave for remote attestation.

Figure 2 illustrates the overall procedure. First, Clients initiate remote attestation, and if it succeeds, Clients retrieve the manifest with the fairness certificate from Modeler. Then, Clients ask Regulator to verify the signed certificate extracted from the manifest, which should be statically registered within the Regulator’s inference enclave after the model certification. To distinguish manifest, the fairness auditing enclave manages a Modeler-certificate mapping table. The registered manifest enforces Modeler not to tamper fairness certificate or launch a different enclave since such an attempt results in verification failure or attestation failure, respectively. Therefore, Clients make sure that they use a certified fair model by verifying the integrity of Modeler’s inference enclave, which embeds the signing certificate chain. This chain of trust starting from the Regulator’s certificate ensures the fairness of model. Note that a certificate issued by Regulator is available in public, thus, each data owner can similarly validate the model’s fairness after the deployment of the certified inference enclave.

D5. In-enclave compilation with ML toolchain and sandboxing (T1,T2,T6) Combining enclave attestation with state-of-the-art enclave sandboxing techniques [27, 28] mediates the second issue. Based on a compiler-based instrumentation [61], malicious attempts to reveal Clients’ data in the inference enclave (**T6**) is confined as the enclave sandboxing based on software fault isolation (SFI) guarantees that Modeler’s enclave cannot leak training data in an unintended purpose. Meanwhile, public scrutiny is also required to audit the inference procedure, similar to verifying the correctness of fairness auditing enclave. For this, our framework takes attestation strategy [27, 65] that separates the ML inference codebase into two parts as Figure 2 shows: public in-enclave part to be attested and private part to achieve model confidentiality. The codes related to ML toolchain and enclave sandboxing are opened to clients, while an enclave utilizes model parameters and a training algorithm to be involved in runtime compilation generated by the ML toolchain. Modeler's sensitive data becomes outside the scope of attestation as an enclave dynamically loads and consumes it to generate an inference code. Then, Clients attest the integrity of ML toolchain and sandbox, both of which are not specific to the proprietary model and publicly auditable whether it deliberately leaks sensitive information. Such design choice explicitly spells out what Clients have to trust, while Modeler keeps their secret private.

4.4 Data Aggregation

Our framework evaluates model fairness based on test data collected from Owners through the aggregation enclave that protects data privacy using **D1**. After constructing the test data $\mathcal{D} = \{(x_1, y_1, z_1), \dots, (x_N, y_N, z_N)\}$, the aggregation enclave sends $\mathcal{D}$ to the fairness auditing enclave. Although the main functionality of the aggregation enclave is to collect test datasets, additional functionalities can be easily implemented in our framework to handle more concerns such as bias and poisoning of the test data.

First, the aggregation enclave can refine the test data to mitigate the inherent bias of data. Note that collected datasets in the real world might be biased; for example, many publicly available face datasets are strongly biased in terms of race (toward Caucasian faces) [9, 29]. Kärkkäinen and Joo [29] show that the trained model with the biased dataset yields poor performance with new balanced datasets. Similarly, such a bias can affect the results of fairness evaluation. However, it is difficult for the existing studies on privacy-preserving fairness certification [31, 36] to mitigate the bias in the collected test data since their cryptographic tools do not allow data inspection based on sensitive group information. In contrast, the aggregation enclave can refine the test data while protecting the privacy of data by using **D1**.

Next, the aggregation enclave can check whether the amount of collected test data is enough to correctly evaluate fairness even when data poisoning exists in the data. It is straightforward that as the aggregation enclave collects more test data, the evaluation becomes robust to data poisoning. The following **A1** provides our theoretical analysis that determines the minimum amount of test data for correct fairness certification in the presence of data poisoning. Referring to this analysis, the aggregation enclave can estimate the required amount of data according to fairness requirements and examine whether the collected data is sufficient.

A1. Consideration of partially poisoned data (T3) Assume that some Owners try to distract our fairness certification by manipulating the data they provide (data poisoning). Unlike the conventional definition of data poisoning, whose target is a set of elements in a data instance, we define data poisoning as flips of only sensitive attributes in a data instance. The following proposition, which extends the claim based on (ϵ, δ)-fairness in [54] , states the condition on the corruption ratio α to certify the classifier f in terms of fairness gap for the risk $\mathcal{R}_z$, where the flipping probability from z to z' is $P(z'|z) \leq \alpha \frac{1}{|\mathcal{Z}|-1}$ if $\forall z' \neq z$ and $P(z'|z) \geq 1 - \alpha$, otherwise.

Proposition 1. *A classifer f becomes (ϵ, δ)-fair if the followings are satisfied for any $0 < \beta \leq \frac{\min_{z'} m_{z'}}{\max_{z'} m_{z'}}$:*

(a) $0 \leq G(\hat{\mathcal{D}}) < \epsilon - 2\gamma$, *where* $\gamma = \frac{2\alpha}{\beta(1-\alpha)+\alpha}$.

(b) $\min_{z'} m_{z'} \geq \frac{2}{(\epsilon - G(\hat{\mathcal{D}}) - 2\gamma)^2} \ln \frac{2|\mathcal{Z}|}{\delta}$.

Proof. The proof is deferred to the Appendix A. □

Proposition 1 demonstrates that the empirical fairness gap of poisoned dataset $G(R, \hat{\mathcal{D}})$ can be used to certify the (ϵ, δ)-fairness of the classifier f when m_z satisfy the conditions. We can extend it to other fairness notions by replacing the group risk $\mathcal{R}_z(f)$ under the condition $p(z'|z, y) = p(z'|z)$ and $m_{z,y} = |\mathcal{D}_{z,y}|$ for the subgroups $\mathcal{D}_{z,y} = \{(x', y', z') \in \mathcal{D} : y' = y, z' = z\}$, as in [54].

5 EVALUATION

In this section, we demonstrate the efficacy and effectiveness of the fairness certification through experiments using a real implementation in Intel SGX. Also, we validate the assumptions in Proposition 1 by inspecting a theoretical curve, and we show the empirical effect of the corrupted group on fairness certification. We use five real-world datasets which are popularly used in fair ML studies: Adult, Bank, COMPAS, German and LSAC (The detailed explanation of these datasets are given in Appendix B.1).

Implementation Setup: We implement our fair auditing enclave by using Intel SGX [26], a representative commoditized TEE technology for x86 architecture. Among SGX-based enclave solutions, we utilize sgx-lkl Open Enclave Edition [50] that supports various programming language runtimes, including Python [1]. We evaluate model certification time and inference time on Quad-core Intel i7-10700K (3.80 GHz CPU,8 physical cores) with Ubuntu 18.04 and Linux 5.4.0 version. Note that we run our experiments on Docker container since containers are a common option for deploying and managing confidential cloud services [37].

ML baselines: In this experiment, we use fairness-aware and fairness-unaware methods as ML baselines. We train logistic regression (LR), support vector machine (SVM), and neural network (NN) as fairness-unaware methods and fair logistic regression (FLR) [63] and fair neural network (FNN) based on adversarial training [40] as fairness-aware methods [2]. We expound the setting of these ML models in Appendix B.

SGX overhead on fairness certification. The fairness audit enclave calculates fairness metrics based on the prediction results from the inference enclave. Thus, we measure computation times of fairness metrics and inference times. First, we use FLR model to measure the certification time in all five datasets. Table 1 shows the evaluation results of fairness certification in the fairness auditing enclave, where the numbers are the averages from 20 runs. The values of accuracy and fairness metrics is given Table 3 in Appendix

[1] SGX-LKL-OE is available at https://github.com/lsds/sgx-lkl (2021/11/17)

[2] We use scikit-learn [48] for LR and SVM and Pytorch [47] for NN. The implementation of fairness-aware methods is based on the python codes (FLR: https://github.com/mbilalzafar/fair-classification, FNN: https://github.com/equialgo/fairness-in-ml)

Table 1: The computation time of fair logistic regression (FLR) model certification with or without SGX, where DI, OMR, FPR and FNR denote fairness notions defined in §2.1

Dataset	Certification time (msec)				Certification time w/ SGX (msec) (Overhead (%))			
	DI	OMR	FPR	FNR	DI	OMR	FPR	FNR
Adult	108	108	109	109	119 (9.5%)	116 (7.8%)	118 (7.9%)	117 (7.6%)
Bank	101	100	101	101	109 (8.1%)	107 (6.4%)	109 (7.3%)	109 (7.5%)
COMPAS	16.1	16.0	16.1	16.2	20.8 (29%)	17.4 (8.9%)	17.4 (8.1%)	17.2 (6.4%)
German	2.45	2.42	2.44	2.45	3.09 (26%)	3.07 (26%)	3.01 (23%)	3.04 (24%)
LSAC	59.4	59.1	59.5	59.4	67.5 (14%)	62.6 (6.0%)	63.8 (7.3%)	63.8 (7.3%)

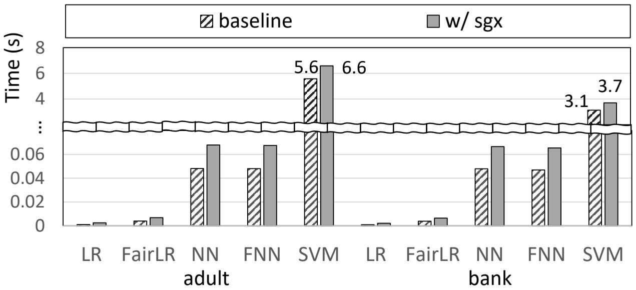


Figure 3: Consumed inference time

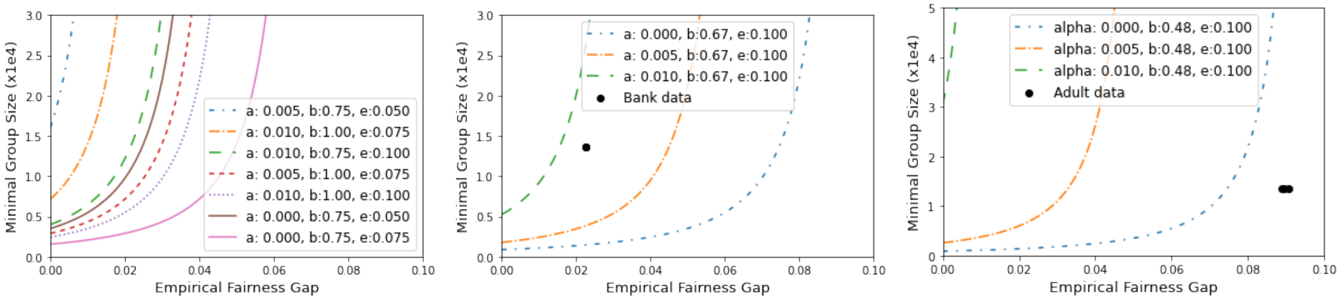
B.3. For large dataset such as Adult and Bank, the computational overhead is relatively small (6.4-9.5%) while German dataset has the highest overhead (23-25%). This is because the portion of the CPU time consumed by SGX-related operations (e.g., paging and context switching [33]) dominates the CPU time for fairness certification as the size of the dataset becomes smaller. To demonstrate our SGX-based fairness certification has a small overhead in practice, we compare our results with respect to the existing MPC-based fairness certification. Due to the limitation on reproducing the MPC-based alternatives, we indirectly refer to the results given in [31], where the MPC certification for disparate impact takes 250ms for German and 802ms for Adult, respectively. Considering that the MPC approach additionally incurs significant communication overhead, we believe our proposed framework is more practical than MPC-based certification.

In addition, we evaluate the inference times of LR, SVM, NN, FLR and FNN two largest datasets, Adult and Bank, as described in Table 2. Figure 3 shows the total inference time from loading the model parameters to predicting the inference results. When measuring the performance, we exclude the time for loading test data. The difference between LR and FLR comes from whether the intercept term is added in the data matrix (FLR) or not (LR). Note that SGX provides near-native processor speed for compute-intensive workloads [56]. As the model complexity of FLR and LR is relatively low, the overhead of utilizing SGX becomes more dominant than compute-intensive inference calculation, which leads to 1.66x-2.43x slowdown compared to the baseline. Because we use the same network structure for the classifier in NN and FNN, the inference time is the same (1.4x and 1.39 slowdown for Adult and Bank, respectively). The SVM model needs matrix-matrix multiplication between the matrix of support vectors and input matrix of test data, where the sizes of input matrices are $13,567 \times 50$ and $13,564 \times 45$, and the number of support vectors for Adult and Bank is 11,570 and 7,242, respectively. As a result, SVM has the longest inference time but moderate computational overhead (1.18x-1.19x slowdown).

Theoretical curve with the corrupted group variable. In §4.2, we discussed the model fairness certification when considering a corruption attack of sensitive group variables. Figure 4 illustrates the theoretical curves for the fairness gap of mis-classification rate based on proposition 1. The curve is affected by ϵ, δ in (ϵ, δ)-fairness, the corruption ratio α and the group ratio β. Figure 4 (a) shows the theoretical curve for various parameter combinations of α, β, and ϵ denoted by a,b and e, respectively, setting $\delta = 0.05$ for all cases. As a result, the fairness certification requires more test data as the corruption or imbalance of sensitive group variables deteriorates. Figure 4 (b) and (c) demonstrate the test results for the fairness gap of mis-classification rate. To evaluate the certification on real datasets, we trained fair logistic regression (FLR) models for Bank and Adult datasets. The imbalance ratio β of Bank and Adult datasets is 0.67 and 0.48, respectively. We conducted the test for $\alpha = [0.0, 0.005, 0.001]$ and $\epsilon = 0.1$. We found that Bank was able to certify a fair model if the corruption ratio was small $\alpha = 0.005$ whereas Adult was not able to certify a fair model even if there was no corruption of sensitive group. Thus, we ascertain that it is significant to retain enough test samples to certify the fairness of ML models, especially if assuming the existence of corruption.

6 RELATED WORK

There have been a number of studies that discussed and addressed algorithmic discrimination issues. They aimed to preprocess data for fair-learning [13, 18], manipulate model predictions without discrimination [25, 49], mitigate bias in a trained model [63, 64], and verify/certify the fairness of machine-learning models [30, 53]. Based on these studies, there are toolkits implemented such as AIF360 [11], FairLearn[12], Aequitas [53] and GerryFair [30]. Although these toolkits support state-of-arts fairness algorithms and metrics, they do not care about various security issues in implementing practical services in some aspects. First, these toolkits assume that all data and models are processed by a trusted party; one should inspect and utilize the entire data to verify and certify a machine-learning model with guaranteed fairness. These data can include protected variables such as race and gender that users



(a) Theoretical Curve

(b) Bank dataset

(c) Adult dataset

Figure 4: The illustration of certification condition in Proposition 1 for corruption ratio α, group ratio β and certification level ϵ: (a) theoretical curve for different parameters, (b)-(c) certification curve and empirical result of Bank dataset and Adult datasets

do not want to disclose, which can raise a significant privacy concern. Second, the existing toolkits are not designed for a distributed environment such as a cloud system, which has become one of the popular platforms to provide machine-learning-based services. Even though each functionality of the toolkits can be used for one service component of the distributed system, they do not properly consider security and privacy issues during required data exchange for verifying and certifying models in terms of fairness.

An interesting line of work introduced trusted third parties to help privacy-preserving FAML. Veale and Binns [60] discussed a multi-party data governance model that a trusted third party is enlisted to collect data on the protected attributes of training data. They posit semi-honest modelers because the third party needs to request the inference of the ML model to the modeler. In addition, the trusted third party that has sensitive information can be a target of attack. To resolve these problems, cryptographic tools such as multi-party computation and homomorphic encryption can be applied. Even though such approaches can provide data privacy, they do not sufficiently satisfy requirements for practical services. For example, Kilbertus et al. [31] and Segal et al. [54] presented privacy-preserving fair certification and verification of ML algorithms that protect sensitive variables and model secrecy by using MPC. However, they postulate that both parties are semi-honest (passive security) and online during computations with high communication costs. In this study, we extend, explore and address potential threats by designing a secure protocol and introducing confidential computing in fair ML auditing.

Utilizing trusted processors to protect a training set and DL/ML model becomes a major research direction in cloud computing. The pioneered studies have focused on securely performing cloud-assisted Deep Neural Network (DNN) inference [22, 24, 32, 35, 39, 51, 58], which improves the computational efficiency but suffers from data privacy. By leveraging commodity CPU feature that supports hardware-protected isolated execution [26], the proposed systems protect data owner's privacy throughout the overall DL inference procedure, while guaranteeing the protection against information exposure problem and model stealing [22, 24, 27, 58]; performing system-level optimization to reduce the inference latency [32, 39, 58]; improving scalability [67]; offering compatibility by importing Tensorflow into hardware-based memory region without modification [35, 51]. Starting from DNN inference, researchers have leveraged TEE technology to achieve privacy-preserving federated learning[44, 65, 66], enabling secure but collaborative data aggregation between multiple data owners. Also, several studies enhance the security of training and inference procedures running on the trusted processor by leveraging data obliviousness to defend against data-dependent access pattern leakage for machine learning [46], DNN inference [21], and XGBoost [37]. However, none of the previous studies has considered how to securely audit the fairness of the training model and certify the result.

7 DISCUSSION AND CONCLUSION

In this work, we propose a novel fair auditing framework incorporating confidential computing technology to address security issues such as privacy, confidentiality, and trustworthiness. Our framework is flexible for various fairness metrics and extendable for various ML models while still preventing security breaches, which helps integrate the fairness assessment into Web applications that use ML models. By building the chain of trust through confidential computing, our framework provides secure fairness certification and verification of trained ML models. Therefore, our framework is one of promising solutions for algorithmic fairness that does not delegate to ML service providers or rely on legislature to drive regulation.

Although we design our framework to consider various aspects of threats and address them in fairness auditing, our framework has several limitations: First, we do not provide an adaptive defense method against data poisoning in the test dataset. We only theoretically analyze its impact and show that sufficient data collection can mitigate it. Second, we do not consider possible attacks during the inference phase. In future work, we will enhance our framework to address such attacks.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable feedback. This work was partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) No. 2021R1G1A1006326 and funded by the Ministry of Education(2018R1D1A1A02085851).

REFERENCES

[1] [n. d.]. Confidential Computing Consortium. Available at https://confidentialcomputing.io/ (2022/02/23).

[2] Carlos Affonso, André Luis Debiaso Rossi, Fábio Henrique Antunes Vieira, André Carlos Ponce de Leon Ferreira, et al. 2017. Deep learning for biological image classification. *Expert Systems with Applications* 85 (2017), 114–122.

[3] Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. 2019. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*. PMLR, 120–129.

[4] Ittai Anati, Shay Gueron, Simon Johnson, and Vincent Scarlata. 2013. Innovative technology for CPU based attestation and sealing. In *Proceedings of the 2nd international workshop on hardware and architectural support for security and privacy*, Vol. 13. Citeseer, 7.

[5] McKane Andrus, Elena Spitzer, Jeffrey Brown, and Alice Xiang. 2020. "What We Can't Measure, We Can't Understand": Challenges to Demographic Data Procurement in the Pursuit of Fairness. *arXiv preprint arXiv:2011.02282* (2020).

[6] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine bias. ProPublica. *See https://www. propublica. org/article/machine-bias-risk-assessments-in-criminal-sentencing* (2016).

[7] Peter Arcidiacono, Shakeeb Khan, and Jacob L Vigdor. 2011. Representation versus assimilation: How do preferences in college admissions affect social interactions? *Journal of Public Economics* 95, 1-2 (2011), 1–15.

[8] ARM. 2009. ARM Security Technology – Building a Secure System using TrustZone Technology. http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf.

[9] Solon Barocas and Andrew D Selbst. 2016. Big data's disparate impact. *Calif. L. Rev.* 104 (2016), 671.

[10] Andrew Baumann, Marcus Peinado, and Galen Hunt. 2015. Shielding applications from an untrusted cloud with haven. *ACM Transactions on Computer Systems (TOCS)* 33, 3 (2015), 1–26.

[11] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. 2018. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943* (2018).

[12] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. 2020. Fairlearn: A toolkit for assessing and improving fairness in AI. *Microsoft, Tech. Rep. MSR-TR-2020-32* (2020).

[13] Flavio P Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. 2017. Optimized pre-processing for discrimination prevention. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 3995–4004.

[14] Chia che Tsai, Donald E. Porter, and Mona Vij. 2017. Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX. In *2017 USENIX Annual Technical Conference (USENIX ATC 17)*. USENIX Association, Santa Clara, CA, 645–658. https://www.usenix.org/conference/atc17/technical-sessions/presentation/tsai

[15] Alexandra Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data* 5, 2 (2017), 153–163.

[16] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*. 797–806.

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

[18] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 259–268.

[19] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 1322–1333.

[20] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[21] Karan Grover, Shruti Tople, Shweta Shinde, Ranjita Bhagwan, and Ramachandran Ramjee. 2018. Privado: Practical and secure DNN inference with enclaves. *arXiv preprint arXiv:1810.00602* (2018).

[22] Zhongshu Gu, Heqing Huang, Jialong Zhang, Dong Su, Ankita Lamba, Dimitrios Pendarakis, and Ian Molloy. 2018. Securing input data of deep learning inference systems via partitioned enclave execution. *arXiv preprint arXiv:1807.00969* (2018).

[23] Juhyeng Han, Seongmin Kim, Daeyang Cho, Byungkwon Choi, Jaehyeong Ha, and Dongsu Han. 2020. A secure middlebox framework for enabling visibility over multiple encryption protocols. *IEEE/ACM Transactions on Networking* 28, 6 (2020), 2727–2740.

[24] Lucjan Hanzlik, Yang Zhang, Kathrin Grosse, Ahmed Salem, Maximilian Augustin, Michael Backes, and Mario Fritz. 2021. Mlcapsule: Guarded offline deployment of machine learning as a service. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3300–3309.

[25] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *NIPS*.

[26] Matthew Hoekstra, Reshma Lal, Pradeep Pappachan, Vinay Phegade, and Juan Del Cuvillo. 2013. Using innovative instructions to create trustworthy software solutions. *HASP@ ISCA* 11, 10.1145, 2487726–2488370.

[27] Tyler Hunt, Congzheng Song, Reza Shokri, Vitaly Shmatikov, and Emmett Witchel. 2018. Chiron: Privacy-preserving machine learning as a service. *arXiv preprint arXiv:1803.05961* (2018).

[28] Tyler Hunt, Zhiting Zhu, Yuanzhong Xu, Simon Peter, and Emmett Witchel. 2016. Ryoan: A Distributed Sandbox for Untrusted Computation on Secret Data. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. USENIX Association, Savannah, GA, 533–549. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/hunt

[29] Kimmo Kärkkäinen and Jungseock Joo. 2019. Fairface: Face attribute dataset for balanced race, gender, and age. *arXiv preprint arXiv:1908.04913* (2019).

[30] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*. PMLR, 2564–2572.

[31] Niki Kilbertus, Adrià Gascón, Matt Kusner, Michael Veale, Krishna Gummadi, and Adrian Weller. 2018. Blind justice: Fairness with encrypted sensitive attributes. In *International Conference on Machine Learning*. PMLR, 2630–2639.

[32] Kyungtae Kim, Chung Hwan Kim, Junghwan" John" Rhee, Xiao Yu, Haifeng Chen, Dave Tian, and Byoungyoung Lee. 2020. Vessels: efficient and scalable deep learning prediction on trusted processors. In *Proceedings of the 11th ACM Symposium on Cloud Computing*. 462–476.

[33] Seongmin Kim, Juhyeng Han, Jaehyeong Ha, Taesoo Kim, and Dongsu Han. 2018. Sgx-tor: A secure and practical tor anonymity network with sgx enclaves. *IEEE/ACM Transactions on Networking* 26, 5 (2018), 2174–2187.

[34] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2016. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807* (2016).

[35] Roland Kunkel, Do Le Quoc, Franz Gregor, Sergei Arnautov, Pramod Bhatotia, and Christof Fetzer. 2019. Tensorscone: A secure tensorflow framework using intel sgx. *arXiv preprint arXiv:1902.04413* (2019).

[36] Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H Chi. 2020. Fairness without demographics through adversarially reweighted learning. *arXiv preprint arXiv:2006.13114* (2020).

[37] Andrew Law, Chester Leung, Rishabh Poddar, Raluca Ada Popa, Chenyu Shi, Octavian Sima, Chaofan Yu, Xingmeng Zhang, and Wenting Zheng. 2020. Secure collaborative training and inference for xgboost. In *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*. 21–26.

[38] Sungyoon Lee, Jaewook Lee, and Saerom Park. 2020. Lipschitz-certifiable training with a tight outer bound. *Advances in Neural Information Processing Systems* 33 (2020).

[39] Taegyeong Lee, Zhiqi Lin, Saumay Pushp, Caihua Li, Yunxin Liu, Youngki Lee, Fengyuan Xu, Chenren Xu, Lintao Zhang, and Junehwa Song. 2019. Occlumency: Privacy-preserving remote deep-learning inference using sgx. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–17.

[40] Gilles Louppe, Michael Kagan, and Kyle Cranmer. 2017. Learning to Pivot with Adversarial Networks. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/48ab2f9b45957ab574cf005eb8a76760-Paper.pdf

[41] Natalia L Martinez, Martin A Bertran, Afroditi Papadaki, Miguel Rodrigues, and Guillermo Sapiro. 2021. Blind Pareto Fairness and Subgroup Robustness. In *International Conference on Machine Learning*. PMLR, 7492–7501.

[42] Maw Maw, Su-Cheng Haw, and Chin-Kuan Ho. 2021. Utilizing data sampling techniques on algorithmic fairness for customer churn prediction with data imbalance problems. *F1000Research* 10, 988 (2021), 988.

[43] Marcin Michał Mirończuk and Jarosław Protasiewicz. 2018. A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications* 106 (2018), 36–54.

[44] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. 2021. PPFL: privacy-preserving federated learning with trusted execution environments. *arXiv preprint arXiv:2104.14380* (2021).

[45] Alfred Ng. [n. d.]. Can Auditing Eliminate Bias from Algorithms? Available at https://themarkup.org/ask-the-markup/2021/02/23/can-auditing-eliminate-bias-from-algorithms (2021/02/23).

[46] Olga Ohrimenko, Felix Schuster, Cédric Fournet, Aastha Mehta, Sebastian Nowozin, Kapil Vaswani, and Manuel Costa. 2016. Oblivious multi-party machine learning on trusted processors. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*. 619–636.

[47] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith

Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

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

[49] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. 2017. On fairness and calibration. *arXiv preprint arXiv:1709.02012* (2017).

[50] Christian Priebe, Divya Muthukumaran, Joshua Lind, Huanzhou Zhu, Shujie Cui, Vasily A Sartakov, and Peter Pietzuch. 2019. SGX-LKL: Securing the host OS interface for trusted execution. *arXiv preprint arXiv:1908.11143* (2019).

[51] Do Le Quoc, Franz Gregor, Sergei Arnautov, Roland Kunkel, Pramod Bhatotia, and Christof Fetzer. 2020. secureTF: a secure TensorFlow framework. In *Proceedings of the 21st International Middleware Conference*. 44–59.

[52] Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. 2020. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 469–481.

[53] Pedro Saleiro, Benedict Kuester, Loren Hinkson, Jesse London, Abby Stevens, Ari Anisfeld, Kit T Rodolfa, and Rayid Ghani. 2018. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577* (2018).

[54] Shahar Segal, Yossi Adi, Benny Pinkas, Carsten Baum, Chaya Ganesh, and Joseph Keshet. 2021. Fairness in the eyes of the data: Certifying machine-learning models. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 926–935.

[55] Hossein Shafagh, Anwar Hithnawi, Andreas Dröscher, Simon Duquennoy, and Wen Hu. 2015. Talos: Encrypted query processing for the internet of things. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*. 197–210.

[56] Fahad Shaon, Murat Kantarcioglu, Zhiqiang Lin, and Latifur Khan. 2017. Sgx-bigmatrix: A practical encrypted data analytic framework with trusted processors. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1211–1228.

[57] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 3–18.

[58] Florian Tramer and Dan Boneh. 2018. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. *arXiv preprint arXiv:1806.03287* (2018).

[59] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*. 601–618.

[60] Michael Veale and Reuben Binns. 2017. Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data. *Big Data & Society* 4, 2 (2017), 2053951717743530.

[61] Bennet Yee, David Sehr, Gregory Dardyk, J Bradley Chen, Robert Muth, Tavis Ormandy, Shiki Okasaka, Neha Narula, and Nicholas Fullagar. 2009. Native client: A sandbox for portable, untrusted x86 native code. In *2009 30th IEEE Symposium on Security and Privacy*. IEEE, 79–93.

[62] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2017. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*. 1171–1180.

[63] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. 2019. Fairness constraints: A flexible approach for fair classification. *The Journal of Machine Learning Research* 20, 1 (2019), 2737–2778.

[64] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 335–340.

[65] Chengliang Zhang, Junzhe Xia, Baichen Yang, Huancheng Puyang, Wei Wang, Ruichuan Chen, Istemi Ekin Akkus, Paarijaat Aditya, and Feng Yan. 2021. Citadel: Protecting Data Privacy and Model Confidentiality for Collaborative Learning with SGX. *arXiv preprint arXiv:2105.01281* (2021).

[66] Xiaoli Zhang, Fengting Li, Zeyu Zhang, Qi Li, Cong Wang, and Jianping Wu. 2020. Enabling execution assurance of federated learning at untrusted participants. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 1877–1886.

[67] Jianping Zhu, Rui Hou, XiaoFeng Wang, Wenhao Wang, Jiangfeng Cao, Boyan Zhao, Zhongpu Wang, Yuhui Zhang, Jiameng Ying, Lixin Zhang, et al. 2020. Enabling rack-scale confidential computing using heterogeneous trusted execution environment. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1450–1465.

A PROOF SKETCH OF PROPOSITION 1

Suppose the poisoned dataset $\hat{\mathcal{D}} = \bigcup_{z \in \mathcal{Z}} \hat{\mathcal{D}}_z$ where each group $\hat{\mathcal{D}}_z$ has the flipped sensitive variable with the ratio at most α and $|\hat{\mathcal{D}}_z| = \hat{m}_z$. For simplicity, $R(f, \mathcal{D})$ is denoted by $R(\mathcal{D})$.

$$
\begin{aligned}
|R(\mathcal{D}_z) - R(\hat{\mathcal{D}}_z)| &= \Bigg| \frac{1}{m_z} \left\{ \sum_{i=1}^{m_z} \mathbb{I}[f(x_i) \neq y_i] \right\} \\
&\quad - \frac{1}{\hat{m}_z} \left\{ \sum_{i=1}^{m_z^\alpha} \mathbb{I}[f(x_i) \neq y_i] + \sum_{i=m_z^\alpha+1}^{\hat{m}_z} \mathbb{I}[f(\hat{x}_i) \neq \hat{y}_i] \right\} \Bigg| \\
&\leq \left| \left(\frac{1}{m_z} - \frac{1}{\hat{m}_z} \right) \sum_{i=1}^{m_z^\alpha} \mathbb{I}[f(x_i) \neq y_i] \right| + \left| \frac{1}{m_z} \sum_{i=m_z^\alpha+1}^{m_z} \mathbb{I}[f(x_i) \neq y_i] \right| \\
&\quad + \left| \frac{1}{\hat{m}_z} \sum_{m_z^\alpha+1}^{\hat{m}_z} \mathbb{I}[f(\hat{x}_i) \neq \hat{y}_i] \right| \\
&\leq \left| \left(\frac{1}{m_z} - \frac{1}{\hat{m}_z} \right) m_z^\alpha \right| + \left(1 - \frac{m_z^\alpha}{m_z} \right) + \left(1 - \frac{m_z^\alpha}{\hat{m}_z} \right) \quad (**)
\end{aligned}
$$

where $m_z^\alpha = \lceil m_z(1-\alpha) \rceil$.

We can simplify the equation (**) depending on the cases. The cases can be as follows:

(i) $2\left(1 - \frac{m_z^\alpha}{\hat{m}_z}\right) \leq 2\left(1 - (1-\alpha)\frac{m_z}{\hat{m}_z}\right)$ if $m_z \leq \hat{m}_z$.

(ii) $2\left(1 - \frac{m_z^\alpha}{m_z}\right) \leq 2\left(1 - (1-\alpha)\right) = 2\alpha$ if $m_z > \hat{m}_z$.

In the case (i), suppose that $m_0 = \min_{z'} m_{z'}$ and $m_1 = \max_{z'} m_{z'}$. Then,

$$
\frac{m_z}{\hat{m}_z} \geq \frac{m_0}{\hat{m}_0} \geq \frac{m_0}{(1-\alpha)m_0 + \alpha m_1} \geq \frac{\beta}{\beta(1-\alpha) + \alpha}. \tag{3}
$$

because $\min_z \frac{m_z}{\hat{m}_z} = \frac{m_0}{\hat{m}_0}$, considering the corruption method. By applying (3) to the case (i), we can obtain:

$$
|R(\mathcal{D}_z) - R(\hat{\mathcal{D}}_z)| \leq \frac{2\alpha}{\beta(1-\alpha) + \alpha} := \gamma, \tag{4}
$$

where $\gamma > 2\alpha$. Using this relationship, we can obtain

$$
\begin{aligned}
|R(\mathcal{D}_z) - \mathcal{R}_z(f)| &\geq |R(\hat{\mathcal{D}}_z) - \mathcal{R}_z(f)| - |R(\hat{\mathcal{D}}_z) - R(\mathcal{D}_z)| \\
&\geq |R(\hat{\mathcal{D}}_z) - \mathcal{R}_z(f)| - \gamma. \qquad (5)
\end{aligned}
$$

Using Hoeffding's inequality and (5) for any $z \in \mathcal{Z}$,

$$
\begin{aligned}
&Pr\left[|R(\hat{\mathcal{D}}_z) - \mathcal{R}_z(f)| > \frac{\epsilon - G(\hat{\mathcal{D}})}{2} \right] \\
&= Pr\left[|R(\mathcal{D}_z) - \mathcal{R}_z(f)| > \frac{\epsilon - 2\gamma - G(\hat{\mathcal{D}})}{2} \right] \\
&\leq 2\exp\left(-m_z \frac{(\epsilon - 2\gamma - G(\hat{\mathcal{D}}))^2}{2} \right) \leq \frac{\delta}{|\mathcal{Z}|}, \qquad (6)
\end{aligned}
$$

where $\frac{\epsilon - G(\hat{\mathcal{D}})}{2} > \gamma > 0$ by the condition (a). Then,

$$
Pr\left[\exists z \in \mathcal{Z} : |R(\hat{\mathcal{D}}_z)(h) - \mathcal{R}_z(f)| > \frac{\epsilon - 2\gamma - G(\hat{\mathcal{D}})}{2} \right] \tag{7}
$$

$$
\leq \sum_z Pr\left[|R(\hat{\mathcal{D}}_z)(h) - \mathcal{R}_z(f)| > \frac{\epsilon - 2\gamma - G(\hat{\mathcal{D}})}{2} \right] \leq \delta \tag{8}
$$

Given that $|\mathcal{R}_z(f) - R(\mathcal{D}_z)| \leq \frac{\epsilon - 2\gamma - G(\hat{\mathcal{D}})}{2}$ for $z = z_0, z_1$,

$$
\begin{aligned}
|\mathcal{R}_{z_0}(f) - \mathcal{R}_{z_1}(f)| &\leq |\mathcal{R}_{z_0}(f) - R(\mathcal{D}_z)| \\
&\quad + G(\hat{\mathcal{D}}) + 2\gamma + |R(\hat{\mathcal{D}}_{z_1}) - \mathcal{R}_{z_1}(f)| \leq \epsilon.
\end{aligned}
$$

Thus, $\max_{z_0, z_1 \in \mathcal{Z}} |\mathcal{R}_{z_0}(f) - \mathcal{R}_{z_1}(f)| \leq \epsilon$ with confidence $1 - \delta$.

B EXPERIMENTAL DETAILS

B.1 Data Description

In §5, we used five real-world datasets: Adult, Bank, COMPAS, German and LSAC. Table 2 contains information about the datasets where each entire dataset was split into training and test data by 7 : 3. The Adult income dataset (Adult) [17] was originated from the 1994 US Census database, and its class label is whether the annual income is above 50K/year or not [3]. We considered gender (male as $z = 1$ and female as $z = 0$) as a sensitive variable and did not use 'race' and 'fnlwgt' as input varables [62]. Also, we remove records that has missing values. The Bank marketing dataset (Bank) [17] has a class label which indicates whether the client subscribed a term deposit [4]. We considered marital status as a binary sensitive variable. The dataset was preprocessed to have 45 input variables, by one-hot encoding all the categorical variables. The COMPAS Recidivism Racial Bias dataset (COMPAS) [6] has a class label which indicates whether a criminal defendant committed crimes after two years [5]. We considered race as a binary sensitive variable, and the dataset was preprocessed to have 5 input variables as in [63]. The German credit dataset (German) [17] has a class label to predict each applicant's credit risk [6]. We considered gender as a binary sensitive variable, and the dataset was preprocessed to have 23 input variables by one-hot encoding all the nominal variables. The Law School Admission Council (LSAC) was collected from all of the public law schools in the United States in July 2007 [7] [7]. The class label of LSAC is to predict the acceptance and rejection of

Table 2: Description of real datasets

Dataset	# of variables	# of samples	
		Training	Test
Adult	50	31,655	13,567
Bank	45	31,647	13,564
COMPAS	5	5,049	2,165
German	23	700	300
LSAC	9	18,585	7,966

[3] http://archive.ics.uci.edu/ml/datasets/Adult

[4] https://archive.ics.uci.edu/ml/datasets/bank+marketing

[5] https://github.com/propublica/compas-analysis

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

[7] http://www.seaphe.org/databases.php

Table 3: The empirical fairness gap of real datasets, where DI, OMR, FPR and FNR are defined in §2.1. The bold numbers indicate the best performance for each dataset and each metric.

Dataset	Methods	Acc(%)	Empirical Fairness Gap(%)			
			DI	**OMR**	**FPR**	**FNR**
Adult	LR	84.56	17.44	12.31	7.73	5.18
	SVM	84.90	16.98	12.43	6.52	**4.86**
	NN	**85.07**	18.13	11.26	7.27	9.23
	FLR	83.20	4.31	10.94	**1.59**	24.08
	FNN	75.32	**2.82**	**2.76**	10.07	24.08
Bank	LR	**90.26**	2.25	2.79	0.93	2.73
	SVM	89.33	0.80	4.17	0.72	4.27
	NN	89.87	2.04	2.75	0.63	**0.49**
	FLR	83.74	1.19	2.87	1.43	0.57
	FNN	89.04	**0.11**	**2.65**	**0.26**	1.15
COMPAS	LR	66.97	20.78	4.73	11.32	25.69
	SVM	66.84	18.81	3.07	9.80	23.28
	NN	**67.67**	19.01	4.54	9.42	23.80
	FLR	56.58	**0.99**	**0.05**	**3.69**	**4.19**
	FNN	57.69	17.54	3.48	16.61	15.59
German	LR	**79.33**	**0.00**	11.67	16.78	0.95
	SVM	78.00	0.42	7.92	9.21	3.89
	NN	79.00	3.33	**5.00**	5.24	3.25
	FLR	78.00	5.83	14.17	25.29	3.89
	FNN	69.33	**0.00**	15.83	**0.00**	**0.00**
LSAC	LR	82.48	1.53	1.08	4.35	0.35
	SVM	80.02	**0.00**	2.13	**0.00**	**0.00**
	NN	**83.02**	1.39	1.77	2.14	0.68
	FLR	81.89	0.58	**0.66**	3.36	0.51
	FNN	81.53	1.46	1.15	4.12	0.37

law applicants. We used 10 input variables, considered gender as a sensitive variable and did not use 'race' as an input variables.

B.2 ML Model Details

To evaluate our fairness certification framework, we use LR, SVM, NN, FLR, and FNN. We need to set some hyperparameters for ML methods except for LR. For SVM model, we used the linear kernel. For FLR, we minimize the loss function subject to the fairness constraint based on the covariance metric for DI [63] while setting the threshold of constraint to 0.0 (perfect fairness). For both NN and FNN, we use 4-layer NNs as classifiers, where each hidden layer consists of 32 hidden nodes with the ReLU activation, and dropout in the training data, and the last layer has a single node with the sigmoid activation. Also, we use the binary cross entropy loss function. FNN consists of two NNs: a classifier network and an adversary network that predicts the sensitive attribute values from the predicted output of the classifier. We use the 4-layer NN that has 3 hidden layers with 32 hidden nodes and ReLU activation and output layer with a single output and sigmoid activation. After pre-training classifier and adversary network for 5 epochs, the adversary network is trained for 50 epochs while training the classifier on a single batch for each epoch as in [40]. The balance parameter between the classifier's loss and the adversary network's loss is set 100.0.

B.3 Additional Evaluation Results

Table 3 shows the evaluation results of LR, SVM, NN, FLR and FNN on four real datasets. The empirical fairness gap is calculated the difference of performance metrics between the worst group and the best group. We have found that the best model can vary depending on what kind of fairness metrics we need to consider. Some researchers have studied the trade-off between fairness and accuracy [15, 16, 34] and the possibility of inconsistent relationships between multiple fairness criterion [34, 49]. Therefore, multidisciplinary studies are needed to achieve social consensus on how to construct fairness certification standards in various practical applications of ML models.