Attesting Distributional Properties of Training Data for Machine Learning

Vasisht Duddu[1], Anudeep Das[1], Nora Khayata[2], Hossein Yalame[2], Thomas Schneider[2], and N. Asokan[1]

[1] University of Waterloo
[2] Technical University of Darmstadt
{vasisht.duddu, a38das}@uwaterloo.ca, {khayata, yalame, schneider}@encrypto.cs.tu-darmstadt.de, asokan@acm.org

Abstract. The success of machine learning (ML) has been accompanied by increased concerns about its trustworthiness. Several jurisdictions are preparing ML regulatory frameworks. One such concern is ensuring that model training data has desirable *distributional properties* for certain sensitive attributes. For example, draft regulations indicate that model trainers are required to show that training datasets have specific distributional properties, such as reflecting the diversity of the population. We propose the novel notion of *ML property attestation* allowing a prover (e.g., model trainer) to demonstrate relevant properties of an ML model to a verifier (e.g., a customer) while preserving the confidentiality of sensitive data. We focus on the attestation of distributional properties of training data *without revealing the data*. We present an effective hybrid property attestation combining property inference with cryptographic mechanisms.

Keywords: Auditing and Accountability · Machine Learning · Property Inference · Private Computation.

1 Introduction

Machine learning (ML) models are being deployed for a wide variety of critical real-world applications such as criminal justice, healthcare, and finance. This has raised several trustworthiness concerns [41]. There are indications that future regulations will require ML model trainers to account for these concerns [9, 13]. One such concern is to ensure that the training data has desirable *distributional properties* with respect to characteristics such as gender or skin color, e.g., the proportion of training data records with a certain attribute value such as skin-tone=black is consistent with the proportion in the population at large. Forthcoming regulation may require model owners to demonstrate such *distributional equity* in their training data, showing that distributional properties of certain training data attributes fall within ranges specified by regulatory requirements: e.g., the draft *Algorithmic Accountability Act* bill [9] requires operators of automated decision systems to keep track of "the representativeness of the dataset and how this factor was measured including . . . the distribution of the population" (cf. [9, §7.C.(i)]). The European Parliament's proposed AI act [13] stipulates that "datasets . . . shall have the appropriate statistical properties, including, where applicable, as regards the persons or groups of persons on which the high-risk AI system is intended to be used" (cf. [13, Art. 10.3]). This ensures that there are no errors arising from population misalignment, i.e., the model does not accurately represent the target population due to distribution shifts between training data and data seen in the real-world [10].

These regulations do not (yet) spell out technical mechanisms for verifying compliance. In this paper, we introduce the notion of *ML property attestation*, which are technical mechanisms by which a *prover* (e.g., a model trainer) can demonstrate relevant properties about the model to a *verifier* (e.g., regulatory agency or a customer purchasing the trained model). Properties of interest may correspond to either training (relating to the model, its training data, or the training process) or inference (e.g., relating to the inference process, or binding the model to its inputs and/or outputs). We focus on *distributional property attestation*, proving distributional properties of a training dataset to the verifier.

A naïve approach for distributional property attestation is to have the prover reveal the training data to the verifier. But this naïve approach may not be legally or commercially viable, given the sensitivity and/or business value of the training data. We identify four requirements for property attestation: be i) *effective*, ii) *efficient*, iii) *confidentiality-preserving*, iv) *adversarially robust*. Simultaneously meeting all of them is challenging. The natural approaches of using trusted execution environments (TEEs), or cryptographic protocols, like secure two-party computation (2PC) and zero knowledge proofs (ZKPs), either impose deployability hurdles or incur excessive overheads.

An interesting alternative is to adapt *property inference attacks* which infer distributional properties of training datasets [2]. Here, the verifier runs a property inference protocol against the prover's model. Some proposed property inference attacks make strong, unrealistic, assumptions about adversary capabilities, e.g., whitebox model access [50]. We argue that such assumptions are reasonable in our attestation setting where provers and verifiers are *incentivized to collaborate* to complete the attestation. Given the changed adversary model, property inference techniques need to be adapted to ensure adversarial robustness against malicious provers. **Our main contributions are as follows:**

1. the novel notion of *ML property attestation*, and desiderata for effective mechanisms to attest distributional properties of training data (§3), and
2. a *hybrid attestation mechanism*[3], combining a property inference attack technique with 2PC (§4), and extensive empirical evaluation showing its effectiveness (§5 and §6).

[3] Code: https://github.com/ssg-research/distribution-attestation.

2 Background

We first summarize ML notations, distributional properties of training data, property inference attacks, and secure multi-party computation (MPC).
ML Notations. Consider a data distribution $\mathbb{D}$ and a training dataset $\mathcal{D}_{tr} \sim \mathbb{D}$ with $\mathcal{D}_{tr} = \{x_i, y_i\}_i^N$ where the i^{th} tuple consists of a vector of *attributes* x_i and its classification label y_i. An ML classification model is a function $\mathcal{M}^\theta : x \rightarrow y$, parameterized by the model parameters θ, which maps input features x to their corresponding classification label y. During training, θ is iteratively updated by penalizing the model for incorrectly predicting y given $x \in \mathcal{D}_{tr}$. During inference, an input x' to $\mathcal{M}^\theta$ gives the prediction $\mathcal{M}^\theta(x')$. We omit θ in $\mathcal{M}^\theta$.
Distributional Properties of Training Data. We borrow the definition for the distributional property of training data $\mathcal{D}_{tr}$ from Suri and Evans [50]. A distributional property is the ratio of an indicator function, counting different data records applied to a dataset (uniformly sampled from a distribution) with a specific attribute value (e.g., males), and total number of data records or number of records with other attribute value (e.g., females). Examples include the ratio of males to females or whites to non-whites in tabular or image datasets, or the average node degree and clustering coefficient for graph data [50, 51].
Property Inference Attacks. Property inference attacks allow an adversary $\mathcal{A}dv$ to infer such distributional properties about *sensitive* attributes in the data distribution $\mathbb{D}$ (e.g., ratio of males/females) using access to the model $\mathcal{M}$ [17, 51, 50, 55, 58, 43, 56]. The attack assumes that a model trainer and $\mathcal{A}dv$ have access to $\mathbb{D}$ and sampling functions $\mathcal{G}_0$ and $\mathcal{G}_1$ which transform $\mathbb{D}$ to obtain a sub-distribution satisfying a particular property. For instance, $\mathcal{G}_0(\mathbb{D})$ indicates 80% males and 20% females while $\mathcal{G}_1(\mathbb{D})$ indicates 50% males and 50% females. Given models $\mathcal{M}_0$ and $\mathcal{M}_1$ trained on datasets sampled from these sub-distributions $\mathcal{G}_0(\mathbb{D})$ and $\mathcal{G}_1(\mathbb{D})$, $\mathcal{A}dv$ infers whether $\mathcal{M}$ was trained on $\mathcal{G}_0(\mathbb{D})$ or $\mathcal{G}_1(\mathbb{D})$ (i.e., $\mathcal{D}_{tr}$ has 80% males or 50% males).
MPC. This cryptographic protocol allows mutually distrusting parties to jointly compute a function on their private inputs, such that nothing beyond the output is leaked [33]. MPC has been adopted to a wide range of applications, including financial services [1] and privacy-preserving machine learning [31]. We make use of secure two-party computation (2PC), a form of MPC with one dishonest party. Dishonest parties can be either semi-honest (follow the protocol but try to infer the other party's inputs) or malicious (deviate from the protocol, e.g., to break correctness). While maliciously secure MPC protocols are more secure, they come with higher computation and communication costs [57]. For real-world applications, semi-honest security guarantees are often sufficient and give baseline performance numbers [39, 26].

3 Problem Statement

We first present the notion of *ML property attestation* followed by the system and adversary models to attest *distributional properties*. We then identify desiderata for distributional property attestation mechanisms.

Property Attestation. These are technical mechanisms using which a prover $\mathcal{P}$ (e.g., a model trainer) can prove to a verifier $\mathcal{V}$ (e.g., potential customer purchasing the model or regulator) that a certain property about the model holds. For example, distributional property attestation can prove that the proportion of records having a specific value of a given attribute in $\mathcal{P}$'s training dataset $\mathcal{D}_{\mathcal{P}} \sim \mathbb{D}$ meets the value p_{req} expected by $\mathcal{V}$. Both $\mathcal{P}$ and $\mathcal{V}$ know p_{req}. Hereafter, we focus on distributional property attestation.

System and Adversary Models. We assume that the distributional property for the attribute of interest can take a set of n possible values $\overline{\mathbf{p}} = \{p_0, \ldots, p_n\}$ (e.g., proportion of females in the dataset). Following the literature on property inference attacks, we assume that both $\mathcal{P}$ and $\mathcal{V}$ know $\mathbb{D}$ [50, 55, 35, 51]. $\mathcal{D}_{\mathcal{P}}$ is split into $\mathcal{D}_{\mathcal{P}}^{tr}$ and $\mathcal{D}_{\mathcal{P}}^{ver}$: $\mathcal{D}_{\mathcal{P}}^{tr}$ is used to train $\mathcal{M}_p$ with some property, $\mathcal{D}_{\mathcal{P}}^{ver}$, which is not known by $\mathcal{V}$, is used for evaluating attestation to simulate what $\mathcal{V}$ is likely to see in practice. $\mathcal{V}$ has their own dataset $\mathcal{D}_{\mathcal{V}} \sim \mathbb{D}$. $\mathcal{D}_{\mathcal{V}}$ is split into a training dataset ($\mathcal{D}_{\mathcal{V}}^{tr}$) used for building attestation mechanism and test dataset ($\mathcal{D}_{\mathcal{V}}^{test}$) to locally evaluate the mechanism.

The goal of $\mathcal{P}$, who has trained a model $\mathcal{M}_p$ on $\mathcal{D}_{\mathcal{P}}^{tr}$, is to succeed in property attestation to comply with regulation. $\mathcal{V}$'s goal is to ensure that attestation succeeds if $\mathcal{D}_{\mathcal{P}}^{tr}$ meets p_{req} even if $\mathcal{P}$ tries to fool the attestation process. We assume that $\mathcal{P}$ has given $\mathcal{V}$ whitebox access to $\mathcal{M}_p$. This is reasonable since $\mathcal{P}$ is incentivized to co-operate with $\mathcal{V}$ to complete the attestation successfully. However, $\mathcal{P}$ does not want to disclose $\mathcal{D}_{\mathcal{P}}^{tr}$ to $\mathcal{V}$ for confidentiality/privacy.

Requirements. A property attestation mechanism must be:

- **R1 Confidentiality-preserving:** $\mathcal{V}$ learns no additional information about $\mathcal{D}_{\mathcal{P}}^{tr}$;
- **R2 Effective:** correctly identify if $\mathcal{D}_{\mathcal{P}}^{tr}$ meets p_{req}, with acceptably low false accepts (FA)/rejects (FR)
- **R3 Adversarially robust:** meet **R2**, with respect to FA, even if $\mathcal{P}$ misbehaves
- **R4 Efficient:** impose an acceptable computation and communication overhead.

4 Distributional Property Attestation Mechanisms

Property attestation by simply revealing $\mathcal{D}_{\mathcal{P}}^{tr}$ to $\mathcal{V}$ violates **R1** and is susceptible to manipulations by a malicious $\mathcal{P}$ ($\mathcal{P}_{mal}$). We discuss three different property attestation mechanisms satisfying **R1** by design and examine **R2**-**R4** for each mechanism: inference-based attestation, cryptographic attestation using MPC, and a hybrid attestation combining the benefits of both.

Inference-based Attestation. Recall that property inference attacks infer statistical properties of training data given access to the victim's model. Hence, these attacks can be adapted for property attestation. Unlike the attack where whitebox model access to $\mathcal{A}dv$ is a strong assumption, $\mathcal{P}$ and $\mathcal{V}$ have an incentive to collaborate to complete the attestation successfully, making whitebox access reasonable. However, directly applying property inference attacks is not possible as there are differences between the two settings (inference attack vs. attestation) in terms of their:

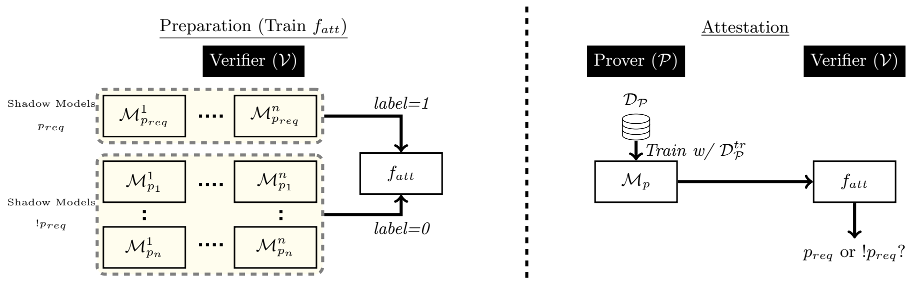


Fig. 1: **Inference-based Attestation**: During preparation, $\mathcal{V}$ trains f_{att} using the first layer parameters of models trained on the training data $\mathcal{D}^{tr}_\mathcal{P}$ with p_{req} ($\{\mathcal{M}^i_{p_{req}}\}^{\mathcal{N}_m}_{i=1}$) and $!p_{req}$ ($\{\mathcal{M}^i_{!p_{req}}\}^{\mathcal{N}_m}_{i=1}$). During attestation, $\mathcal{V}$ uses first layer parameters of $\mathcal{M}_p$ to attest if it was indeed trained on $\mathcal{D}^{tr}_\mathcal{P}$ with p_{req} or not.

– **objective**: the attack distinguishes between two property values while attestation requires differentiating p_{req} from all others ($!p_{req}$).
– **requirement**: attestation has the additional requirement of robustness **R3**, i.e., resist $\mathcal{P}_{mal}$'s attempts to fool $\mathcal{V}$.

We show how property inference attacks can be adapted to attestation and describe the inference-based attestation below.

Method. Given access to $\mathcal{M}_p$, $\mathcal{V}$ uses an attestation classifier (f_{att}) to attest if $\mathcal{D}^{tr}_\mathcal{P}$ satisfies p_{req} using the first layer parameters of $\mathcal{M}_p$ as input to f_{att}. The first layer parameters are more effective to capture distributional properties for successful property inference than subsequent layers [50]. To train f_{att}, $\mathcal{V}$ uses $\mathcal{D}^{tr}_\mathcal{V}$ and generates multiple sub-distributions $\{\mathcal{G}_0(\mathbb{D}), \ldots, \mathcal{G}_n(\mathbb{D})\}$ corresponding to property values in $\overline{\mathbf{p}}$ and samples datasets $\{\mathcal{D}_0, \ldots, \mathcal{D}_n\}$. In practice, this is done by sampling datasets multiple times with different properties from $\mathcal{D}_\mathcal{V}$. For each dataset and property value, $\mathcal{V}$ trains $\mathcal{N}_m$ "shadow models" $\{\{\mathcal{M}^i_0\}^{\mathcal{N}_m}_{i=1}, \ldots, \{\mathcal{M}^i_n\}^{\mathcal{N}_m}_{i=1}\}$. These mimic the models that $\mathcal{V}$ could encounter during attestation.

$\mathcal{V}$ trains f_{att} using the first layer parameters of models trained on $\mathcal{D}^{tr}_\mathcal{P}$ with p_{req} ($\{\mathcal{M}^i_{p_{req}}\}^{\mathcal{N}_m}_{i=1}$) and $!p_{req}$ ($\{\mathcal{M}^i_{!p_{req}}\}^{\mathcal{N}_m}_{i=1}$). $\mathcal{V}$ uses $\mathcal{D}^{test}_\mathcal{V}$ for evaluating f_{att}. Attestation effectiveness is evaluated using $\mathcal{D}^{ver}_\mathcal{P}$. We present a visualization of inference-based attestation in Figure 1.

Cryptographic Attestation. Property attestation can be securely achieved using cryptographic protocols (e.g., MPC, ZKPs) by proving that (a) $\mathcal{D}^{tr}_\mathcal{P}$ meets p_{req} (`DistCheck`), and (b) $\mathcal{M}_p$ was trained on $\mathcal{D}^{tr}_\mathcal{P}$ to ensure that a misbehaving $\mathcal{P}$ does not change $\mathcal{D}^{tr}_\mathcal{P}$ after (a) (Figure 2). We use 2PC due to their practicality (see §8 for discussion on alternative approaches).

Assumptions. $\mathcal{P}$ may deceive $\mathcal{V}$ about p_{req}, acting maliciously. However, $\mathcal{V}$, interested in purchasing $\mathcal{M}_p$, has no incentive to cheat, but may seek additional details about $\mathcal{D}^{tr}_\mathcal{P}$. Thus, we assume $\mathcal{V}$ behaves semi-honestly.

Setup. To account for $\mathcal{P}_{mal}$, we could use malicious two-party protocols directly between $\mathcal{P}$ and $\mathcal{V}$, which is prohibitively expensive. Instead, $\mathcal{P}$ and $\mathcal{V}$ rely on secure outsourced computation to independent and non-colluding servers $\mathcal{S}_1$ and

$\mathcal{S}_2$ as done in prior work [45] and in practical deployments [15]. $\mathcal{S}_1$ and $\mathcal{S}_2$ can be instantiated by different companies, which according to data protection laws must protect user data, thus cannot share their data with each other. Outsourcing also allows to flexibly instantiate the cryptographic protocol, i.e., our construction generalizes to 2PC/MPC or ZKP protocols.

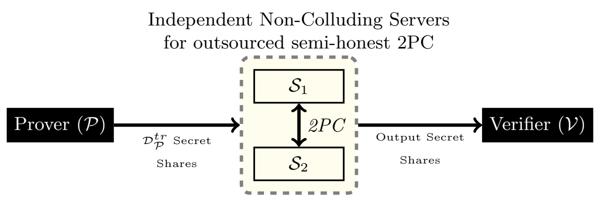


Fig. 2: **Cryptographic Attestation**: $\mathcal{P}$ sends the secret shares of the training data $\mathcal{D}^{tr}_{\mathcal{P}}$ to $\mathcal{S}_1$ and $\mathcal{S}_2$. The servers securely compute "DistCheck" for $\mathcal{D}^{tr}_{\mathcal{P}}$ and train $\mathcal{M}_{2pc}$ on $\mathcal{D}^{tr}_{\mathcal{P}}$ with their secret shares using 2PC. The output shares are then sent to $\mathcal{V}$ for reconstructs the outputs.

Method. We consider secret sharing over a ring with Q elements where a secret input x is split into two shares, x_1 and x_2 such that $x = x_1 + x_2 \mod Q$ [31]. Each share x_1 and x_2 looks random, i.e., given only one share, one cannot learn any information about the secret. For (a), `DistCheck` computes the distributional property directly over $\mathcal{D}^{tr}_{\mathcal{P}}$ and comparing with p_{req}. Here, $\mathcal{P}$ sends secret shares of $\mathcal{D}^{tr}_{\mathcal{P}}$ to $\mathcal{S}_1$ and $\mathcal{S}_2$ who jointly perform `DistCheck` by running secure accumulation and comparison using 2PC. For (b), $\mathcal{P}$ sends secret shares of the initial model weights to obtain $\mathcal{M}_p$ to $\mathcal{S}_1$ and $\mathcal{S}_2$. Together with the previously obtained shares of $\mathcal{D}^{tr}_{\mathcal{P}}$, $\mathcal{S}_1$ and $\mathcal{S}_2$ jointly run secure training. $\mathcal{S}_1$ and $\mathcal{S}_2$ send the resulting secret shares of `DistCheck` and the final model parameters of the trained model $\mathcal{M}_{2pc}$ to $\mathcal{V}$ who adds the received shares to get the results of `DistCheck` and the trained model weights. The *correctness* property of 2PC convinces $\mathcal{V}$ that both `DistCheck` and training are run correctly on $\mathcal{D}^{tr}_{\mathcal{P}}$. We implement secure computation between $\mathcal{S}_1$ and $\mathcal{S}_2$ using CrypTen [31], a framework for efficient privacy-preserving ML that supports one semi-honest corruption for two parties (see Appendix A for more details, security and correctness of the protocols).

Hybrid Property Attestation. Cryptographic attestation is costly, while inference-based attestation can have unacceptably high false acceptance or false rejected rates (FAR or FRR respectively) (Table 1). Relying solely on either is inadequate. Therefore, we propose a hybrid attestation scheme that first uses inference-based attestation with cryptographic attestation as a fallback. Depending on the application, $\mathcal{V}$ can fix an acceptably low FAR or FRR:

– **Fixed FAR.** For accepted provers ($\mathcal{P}$s), no further action is needed. If the inference-based attestation fails (FR), $\mathcal{P}$s can request re-evaluation with cryptographic attestation.

– **Fixed FRR.** If inference-based attestation is rejected, there is no provision for re-appeal since FRR is low. For accepted $\mathcal{P}$s, $\mathcal{V}$ may do a random "spot-check" using cryptographic attestation.

Assumptions. We assume FAR and FRR are fixed at 5%. Additionally, $\mathcal{V}$ uses $\mathcal{M}_p$ for inference-based attestation and $\mathcal{M}_{2pc}$ is obtained by 2PC training. We assume that $\mathcal{P}$ shares the hyperparameters for training $\mathcal{M}_p$ to obtain $\mathcal{M}_{2pc}$ to be perfectly equivalent. This can be done by a fidelity check, i.e., sending arbitrary inputs and ensuring outputs from $\mathcal{M}_p$ and $\mathcal{M}_{2pc}$ are equal.

Method. While hybrid attestation is straight-forward for fixed FAR, we describe the methodology for random spot-checks of accepted $\mathcal{P}$s for fixed FRR. Let z be the total FA on $\mathcal{D}_{\mathcal{V}}^{test}$ from the inference-based attestation and $\mathcal{N}_a$ denote the number of accepted $\mathcal{P}$s. Knowing z, $\mathcal{V}$ can randomly sample $\mathcal{N}_{spchk}$ spot-checks where $z \leq \mathcal{N}_{spchk} \leq \mathcal{N}_a$. $\mathcal{V}$ then uses cryptographic attestation to eliminate any FA in the sampled set thus reducing the overall FAR. To compute the new FAR, we first compute the probability of finding t FAs from the sample of $\mathcal{N}_{spchk}$ $\mathcal{P}$s. We model the probability distribution over FA as hypergeometric distribution which computes the likelihood of selecting t FAs in a sample of $\mathcal{N}_{spchk}$ from a population of z falsely accepted $\mathcal{P}$s without replacement: $\mathbb{P}(T = t) = \binom{z}{t}\binom{\mathcal{N}_a - z}{\mathcal{N}_{spchk} - t}/\binom{\mathcal{N}_a}{\mathcal{N}_{spchk}}$, where $t \in [0, z]$. We compute the effective #FAs as $\#FA_{new} = \#FA_{old} - t'$ where $t' = argmax_{t \in [0,z]}\mathbb{P}(T = t)$. $\mathcal{N}_{spchk}$ will determine the FAR and cost incurred.

5 Experimental Setup

We describe the different datasets used and corresponding model architectures, followed by the metrics used for evaluation.

Datasets and Model Architecture. We use the datasets, properties, and model architectures same as in prior work on property inference attacks [50].

BONEAGE is an image dataset which contains X-Ray images of hands, with the task being predicting the patient's age in months. The dataset is converted to a binary classification task for classifying the age of the patient. We focus on the ratios of the females the property of interest. We consider the following permissible ratios ($\overline{\mathbf{p}}$): ["0.2" - "0.8"]. Here, the sensitive attribute is implicit as part of the metadata.

BONEAGE model is a pre-trained DenseNet [21] model for feature extraction of the images, followed by a three-layer network of size [128, 64, 1] for classification with ReLU activations. We train the model for 100 epochs with a batch size of 8192, learning rate of 0.001, and weight decay of 1e-4.

ARXIV is a directed graph dataset representing citations between computer science ArXiv papers. The classification task is to predict the subject area for the papers. The property considered for attestation is the mean node-degree of the graph dataset. We use the following permissible ratios ($\overline{\mathbf{p}}$): ["9" - "17"]. The graph dataset is sampled to satisfy a specific mean-node degree which is implicitly included in the dataset.

ARXIV model is a four-layer graph convolutional network which maps the input graph data to low dimensional embedding for node classification tasks. The graph convolution layer sizes are [256, 256, 256, and 40] with ReLU activation. We use dropout with 0.5 drop probability. We train the model for 100 epochs with a learning rate of 0.01, and weight decay of 5e-4.

CENSUS is a tabular dataset which consists of several categorical and numerical attributes like age, race, education level. The classification task is to predict whether an individual's annual income exceeds 50K. However, we have two variants of this dataset based on the property: (a) CENSUS-R which considers the distribution of whites and (b) CENSUS-S which considers the distribution of females in the dataset. For both, we consider the following permissible ratios ($\overline{\mathbf{p}}$): ["0.0" - "1.0"]. Both CENSUS-S and CENSUS-R explicitly include the sensitive attributes in the dataset.

CENSUS model is a three layer deep neural network with the hidden layer dimensions: [32, 16, 8] with ReLU activation function. We train the model for 100 epochs with a learning rate of 0.01, batch size of 64, and no weight decay.

Our f_{att} is based on permutation invariant networks, based on DeepSets architecture [53], used in the property inference literature [50, 17]. The classifier generates a representation of the input model parameters independent of the ordering of the neurons in a layer. Suri and Evans [50] suggest that the first layer captures distributional properties better than subsequent layers. Hence, we use first layer's model parameters as input to f_{att} which outputs the training data's property.

Metrics. We describe different metrics to measure the effectiveness of inference based attestation. A model trained on a dataset with p_{req} is considered as the positive class. Accuracy indicates the success of $\mathcal{P}$'s model and $\mathcal{V}$'s shadow models on the task. True Acceptance Rate (TAR) measures the fraction of models where $\mathcal{V}$ correctly attests that $\mathcal{M}_p$ was indeed trained from a dataset with p_{req}. True Rejection Rate (TRR) measures the fraction of models where $\mathcal{V}$ correctly rejects attestation of $\mathcal{M}_p$ w.r.t. p_{req}. FAR and FRR measure the extent of $\mathcal{V}$ incorrectly accepting or rejecting attestation respectively. Equal Error Rate (EER) indicates the value at which the FAR and FRR are equal. TAR and TRR should ideally be 1.00 while FAR, FRR and EER be 0.00.

For cryptographic attestation, we indicate computation cost (ω_{crpt}^{comp}) as the execution time for `DistCheck` and secure model training; and communication cost (ω_{crpt}^{comm}) as the amount of data transferred during attestation. For hybrid attestation with fixed FAR where $\mathcal{P}$ relies on cryptographic fallback, the expected cost is $\mathbb{P}_{inf} \times \omega_{inf} + \mathbb{P}_{crpt} \times \omega_{crpt}$ where $\mathbb{P}_{inf} = 1$. As $\omega_{inf} \ll \omega_{crpt}$, the cost reduces to $\mathbb{P}_{crpt} \times \omega_{crpt}$. Similarly, for fixed FRR, $\mathcal{V}$ conducts spot-checks with a probability of $\mathbb{P}_{spchk}$. The expected cost in this case is $\mathbb{P}_{spchk} \times \omega_{crpt}$. Both $\mathbb{P}_{crpt}$ and $\mathbb{P}_{spchk}$ are computed on $\mathcal{D}_{\mathcal{V}}^{test}$.

6 Experimental Evaluation

For different requirements: effectiveness, adversarial robustness and efficiency, we first evaluate the inference-based and cryptographic attestation and identify their limitations, then evaluate the hybrid attestation.

6.1 Inference-based Attestation

Effectiveness (R2). We first evaluate the effectiveness of f_{att} in distinguishing between models trained on $\mathcal{D}_{\mathcal{P}}^{tr}$ with p_{req} and $!p_{req}$ using AUC score under FAR-TAR curves (see Appendix A of our full paper [12]). We find that for some p_{req} values, f_{att} is less effective. Hence, we relax the attestation requirement to exactly match p_{req} by increasing the window size to ±1 (i.e., classify between $\{p_{req}$-1,p_{req},p_{req} +1$\}$ and $!\{p_{req}$-1,p_{req},p_{req} +1$\}$) [4]. Based on the results, we identify the best window sizes on $\mathcal{D}_{\mathcal{V}}^{test}$: ±1 for all p_{req} for BONEAGE and ARXIV; 0 for the edge p_{req} values (i.e., "0.00" and "1.00") and ±1 for all middle p_{req} values for CENSUS-R and CENSUS-S. $\mathcal{V}$ can make these decisions on $\mathcal{D}_{\mathcal{V}}^{test}$ before finalizing f_{att}.

Assuming that $\mathcal{V}$ fixes (a) FAR, or (b) FRR at 5%, we present the corresponding TRR and TAR in Table 1 on $\mathcal{D}_{\mathcal{P}}^{ver}$. At either end of the spectrum of p_{req} values, attestation is effective (high TRR/TAR). However, we observe a high FAR and FRR for the middle p_{req} values indicating that attestation is less effective. Furthermore, we provide EER values that, for specific p_{req} values, demonstrate lower rates than both FAR and FRR. This implies the existence of a more optimal threshold than the currently used 5%. In summary, inference-based attestation is *ineffective for certain p_{req} values* and cannot be used on its own.

Robustness (R3). $\mathcal{P}_{mal}$ can fool $\mathcal{V}$ by modifying $\mathcal{M}_p$'s first layer parameters ($\mathcal{M}_p^1$) to trigger FA. $\mathcal{P}_{mal}$ adds adversarial noise δ to the first layer parameters: $\mathcal{M}_p^1+\delta$ where $\delta = argmax_{||\delta||_p<\epsilon}\ L(\ f_{att}\ (\mathcal{M}_p^1+\delta\ ),p_{req}\ )$, L is the f_{att}'s loss and $||\cdot||_p$ is the l_p norm to ensure δ to minimize accuracy degradation. Since, $\mathcal{P}_{mal}$ does not have access to f_{att}, they train a "substitute model" on $\mathcal{D}_{\mathcal{P}}^{tr}$ which mimics f_{att}. For worst-case analysis under the attack, we assume that the substitute model's architecture is the same as f_{att}. δ is then computed with respect to this substitute model and the FA are expected to transfer to f_{att}. To restore any $\mathcal{M}_p$'s accuracy loss, $\mathcal{P}_{mal}$ can freeze $\mathcal{M}_p^1$ and fine-tune the remaining layers. We empirically evaluate this and confirm that accuracy of model after fine-tuning is close to the original accuracy while still being able to fool the attestation (results in Appendix C of our full paper [12]). We use Autoattack [11] with $\epsilon = 8/255$, and L_∞ norm for the distance function. As $\mathcal{P}_{mal}$ has access to the models shared with $\mathcal{V}$, we evaluate on $\mathcal{D}_{\mathcal{P}}^{ver}$.

Attack Success. $\mathcal{P}_{mal}$ wins if f_{att} incorrectly classifies perturbed models as having been trained with p_{req}. We measure the attack success using FAR. Note that the FAR here is restricted to $\mathcal{D}_{\mathcal{P}}^{ver}$ containing models with adversarial noise. Under "w/o Defence" in Table 2, the high FAR values indicate that the attack is indeed successful (f_{att} is not robust).

[4] We continue to use p_{req} and $!p_{req}$ to refer to these windows.

Table 1: TAR and TRR with 5% thresholds for FAR and FRR respectively along with EER across different p_{req} windows on $\mathcal{D}_{\mathcal{P}}^{ver}$. The p_{req} value within the window is indicated in **bold**. Edge p_{req} values have higher effectiveness than middle p_{req} values due to higher distinguishability in first layer parameters [50].

ARXIV

p_{req} **Range**	**TAR**	**TRR**	**EER**
{**9**, 10}	1.00	0.99	0.02
{9, **10**, 11}	1.00	1.00	0.01
{10, **11**, 12}	0.24	0.83	0.16
{11, **12**, 13}	0.61	0.68	0.19
{12, **13**, 14}	0.78	0.85	0.10
{13, **14**, 15}	0.92	0.93	0.07
{14, **15**, 16}	0.87	0.90	0.08
{15, **16**, 17}	1.00	1.00	0.00
{16, **17**}	1.00	1.00	0.00

BONEAGE

p_{req} **Range**	**TAR**	**TRR**	**EER**
{**0.20**, 0.30}	0.96	0.96	0.03
{0.20, **0.30**, 0.40}	0.99	1.00	0.02
{0.30, **0.40**, 0.50}	0.87	0.88	0.09
{0.40, **0.50**, 0.60}	0.53	0.65	0.21
{0.50, **0.60**, 0.70}	0.39	0.72	0.25
{0.60, **0.70**, 0.80}	0.98	0.98	0.03
{0.70, **0.80**}	0.95	0.95	0.05

CENSUS-S

p_{req} **Range**	**TAR**	**TRR**	**EER**
{**0.00**}	1.00	1.00	0.00
{0.00, **0.10**, 0.20}	0.49	0.49	0.19
{0.10, **0.20**, 0.30}	0.70	0.72	0.14
{0.20, **0.30**, 0.40}	0.23	0.56	0.25
{0.30, **0.40**, 0.50}	0.12	0.30	0.37
{0.40, **0.50**, 0.60}	0.13	0.23	0.41
{0.50, **0.60**, 0.70}	0.15	0.22	0.34
{0.60, **0.70**, 0.80}	0.12	0.26	0.35
{0.70, **0.80**, 0.90}	0.59	0.58	0.19
{0.80, **0.90**, 1.00}	0.60	0.59	0.19
{**1.00**}	1.00	1.00	0.00

CENSUS-R

p_{req} **Range**	**TAR**	**TRR**	**EER**
{**0.00**}	1.00	1.00	0.00
{0.00, **0.10**, 0.20}	0.21	0.64	0.19
{0.10, **0.20**, 0.30}	0.75	0.89	0.10
{0.20, **0.30**, 0.40}	0.22	0.59	0.23
{0.30, **0.40**, 0.50}	0.14	0.16	0.39
{0.40, **0.50**, 0.60}	0.10	0.15	0.42
{0.50, **0.60**, 0.70}	0.13	0.26	0.39
{0.60, **0.70**, 0.80}	0.05	0.36	0.32
{0.70, **0.80**, 0.90}	0.65	0.77	0.13
{0.80, **0.90**, 1.00}	0.35	0.41	0.26
{**1.00**}	1.00	1.00	0.00

Improving Robustness. We propose adversarial training of f_{att} where $\mathcal{V}$ includes models with adversarial noise to train f_{att}. Our goal is to reduce FAR on perturbed models while retaining utility on clean $\mathcal{D}_{\mathcal{P}}^{ver}$.

We present the results of adversarial training under "w/ Defence" in Table 2. First, the FAR values in "w/ Defence" are lower (than in "w/o Defence"). Hence, adversarial training of f_{att} successfully mitigates the perturbation attack, thus making inference-based attestation adversarially robust, satisfying **R3**. Second, the difference in utility on clean $\mathcal{D}_{\mathcal{P}}^{ver}$ (measured using AUC score under FAR-TAR curves indicated by $\mathcal{U}$) between "w/ Defence" and "w/o Defence" is small. We also evaluate the effectiveness using TRR, TAR, and EER on clean $\mathcal{D}_{\mathcal{P}}^{ver}$ which are available in Appendix B of our full paper [12]. We use robust f_{att} in the rest of the paper. Similar to f_{att}, robust f_{att} is still ineffective for some p_{req}.

Efficiency (R4). $\mathcal{V}$ trains multiple shadow models and f_{att}. We measure the total training time to train 10 attestation classifiers and 1000 shadow models on a single NVIDIA A100 GPU. Training f_{att} took a total of 200 mins for BONEAGE; 12 mins for ARXIV, 6 mins for CENSUS-S and CENSUS-R. Training 1000 shadow models took a total of 173 mins for BONEAGE, 123 mins for ARXIV and 50 mins for CENSUS-S and CENSUS-R.

BONEAGE, being an image dataset trained on a large neural network, takes the maximum time for both f_{att} and shadow models. On the other hand, CENSUS has a small number of tabular data records and with a small MLP

Table 2: **Robustness against first layer parameter perturbations with and without a defense.** Utility ($\mathcal{U}$) is calculated using AUC on FAR-TAR for a clean dataset. FAR $\rightarrow$ lack of robustness. Green $\rightarrow \mathcal{U}$ decreases or FAR $< 5\%$, red $\rightarrow \mathcal{U}$ increases or FAR $\geq 5\%$.

ARXIV

p_{req} **Range**	w/o Defence $\mathcal{U}$	w/o Defence FAR	w/ Defence $\mathcal{U}$	w/ Defence FAR
{**9**, 10}	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.01	0.00 ± 0.00
{9, **10**, 11}	1.00 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00
{10, **11**, 12}	0.92 ± 0.00	1.00 ± 0.00	0.88 ± 0.00	0.00 ± 0.00
{11, **12**, 13}	0.96 ± 0.00	1.00 ± 0.00	0.96 ± 0.00	0.00 ± 0.00
{12, **13**, 14}	0.93 ± 0.00	1.00 ± 0.00	0.96 ± 0.01	0.00 ± 0.00
{13, **14**, 15}	0.99 ± 0.00	1.00 ± 0.00	0.96 ± 0.01	0.00 ± 0.00
{14, **15**, 16}	0.99 ± 0.00	1.00 ± 0.00	1.00 ± 0.01	0.00 ± 0.00
{15, **16**, 17}	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00
{16, **17**}	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.01	0.00 ± 0.00

BONEAGE

p_{req} **Range**	w/o Defence $\mathcal{U}$	w/o Defence FAR	w/ Defence $\mathcal{U}$	w/ Defence FAR
{**0.20**, 0.30}	0.99 ± 0.00	1.00 ± 0.00	0.99 ± 0.00	0.00 ± 0.00
{0.20, **0.30**, 0.40}	0.99 ± 0.00	1.00 ± 0.00	0.99 ± 0.00	0.00 ± 0.00
{0.30, **0.40**, 0.50}	0.92 ± 0.00	1.00 ± 0.00	0.92 ± 0.00	0.00 ± 0.00
{0.40, **0.50**, 0.60}	0.86 ± 0.00	1.00 ± 0.00	0.84 ± 0.00	0.00 ± 0.01
{0.50, **0.60**, 0.70}	0.87 ± 0.00	0.28 ± 0.00	0.85 ± 0.00	0.25 ± 0.00
{0.60, **0.70**, 0.80}	0.99 ± 0.00	1.00 ± 0.00	0.99 ± 0.00	0.02 ± 0.00
{0.70, **0.80**}	0.95 ± 0.00	0.04 ± 0.00	0.95 ± 0.00	0.00 ± 0.00

CENSUS-S

p_{req} **Range**	w/o Defence $\mathcal{U}$	w/o Defence FAR	w/ Defence $\mathcal{U}$	w/ Defence FAR
{**0.00**}	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00
{0.00, **0.10**, 0.20}	0.83 ± 0.01	0.26 ± 0.05	0.82 ± 0.01	0.01 ± 0.01
{0.10, **0.20**, 0.30}	0.92 ± 0.00	0.12 ± 0.02	0.92 ± 0.01	0.04 ± 0.01
{0.20, **0.30**, 0.40}	0.78 ± 0.01	0.11 ± 0.02	0.79 ± 0.00	0.10 ± 0.02
{0.30, **0.40**, 0.50}	0.66 ± 0.00	0.34 ± 0.10	0.66 ± 0.01	0.29 ± 0.03
{0.40, **0.50**, 0.60}	0.67 ± 0.01	0.39 ± 0.02	0.67 ± 0.00	0.22 ± 0.03
{0.50, **0.60**, 0.70}	0.62 ± 0.01	0.19 ± 0.02	0.62 ± 0.01	0.14 ± 0.01
{0.60, **0.70**, 0.80}	0.68 ± 0.00	0.48 ± 0.01	0.68 ± 0.01	0.35 ± 0.04
{0.70, **0.80**, 0.90}	0.89 ± 0.00	0.32 ± 0.02	0.89 ± 0.00	0.32 ± 0.05
{0.80, **0.90**, 1.00}	0.89 ± 0.01	0.65 ± 0.03	0.89 ± 0.00	0.22 ± 0.03
{**1.00**}	1.00 ± 0.00	0.02 ± 0.00	1.00 ± 0.00	0.00 ± 0.00

CENSUS-R

p_{req} **Range**	w/o Defence $\mathcal{U}$	w/o Defence FAR	w/ Defence $\mathcal{U}$	w/ Defence FAR
{**0.00**}	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00
{0.00, **0.10**, 0.20}	0.75 ± 0.01	0.82 ± 0.02	0.77 ± 0.01	0.35 ± 0.03
{0.10, **0.20**, 0.30}	0.95 ± 0.00	0.60 ± 0.03	0.95 ± 0.00	0.08 ± 0.01
{0.20, **0.30**, 0.40}	0.71 ± 0.01	0.83 ± 0.02	0.71 ± 0.01	0.13 ± 0.05
{0.30, **0.40**, 0.50}	0.75 ± 0.00	0.82 ± 0.07	0.74 ± 0.00	0.05 ± 0.01
{0.40, **0.50**, 0.60}	0.64 ± 0.00	0.79 ± 0.05	0.63 ± 0.00	0.11 ± 0.03
{0.50, **0.60**, 0.70}	0.60 ± 0.01	0.31 ± 0.02	0.60 ± 0.00	0.31 ± 0.02
{0.60, **0.70**, 0.80}	0.83 ± 0.00	0.30 ± 0.02	0.82 ± 0.01	0.26 ± 0.01
{0.70, **0.80**, 0.90}	0.96 ± 0.00	0.10 ± 0.01	0.96 ± 0.01	0.19 ± 0.00
{0.80, **0.90**, 1.00}	0.70 ± 0.01	0.46 ± 0.03	0.75 ± 0.01	0.34 ± 0.02
{**1.00**}	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00

classifier, takes the least training time. Note that this is a one-time cost which can be parallelized among multiple GPUs. Hence, $\mathcal{V}$'s cost for inference-based

attestation is reasonable. f_{att} can then be used for multiple attestation for the same property and has to be trained on new property only if p_{req} changes.
Summary. Inference-based attestation satisfies **R3** robustness and **R4** efficiency but has a *poor effectiveness.*

6.2 Cryptographic Attestation

Effectiveness (R2). Cryptographic attestation operates over $\mathcal{D}_{\mathcal{P}}^{tr}$ confidentially to correctly check whether the distributional properties match p_{req}. Hence, we have *zero FAR and FRR.*
Robustness (R3). Using outsourcing, $\mathcal{P}$'s inputs are secret-shared between $\mathcal{S}_1$ and $\mathcal{S}_2$ who learn nothing (non-colluding assumption) and have no incentive to cheat (semi-honest assumption). Furthermore, $\mathcal{P}$ *only* performs the input sharing of the training data $\mathcal{D}_{\mathcal{P}}^{tr}$ and initial model weights, thus $\mathcal{P}$ cannot cheat during proof generation. Hence, this attestation is robust.
Efficiency (R4). We use protocols for semi-honest parties. We present the computation and communication cost for a single cryptographic attestation in Table 3. We indicate the costs for BONEAGE and CENSUS but omit the evaluation on ARXIV as there are no PyTorch frameworks for secure GNN training which is required for the CrypTen library. We observe that the cost for `DistCheck` is low, but the cost for secure ML training is high. Hence, cryptographic attestation is difficult to be used in practice for multiple $\mathcal{P}$s.

Table 3: Computation (ω_{crpt}^{comp}) and communication costs (ω_{crpt}^{comm}) of cryptographic attestation for a single $\mathcal{P}$ averaged over 20 runs.

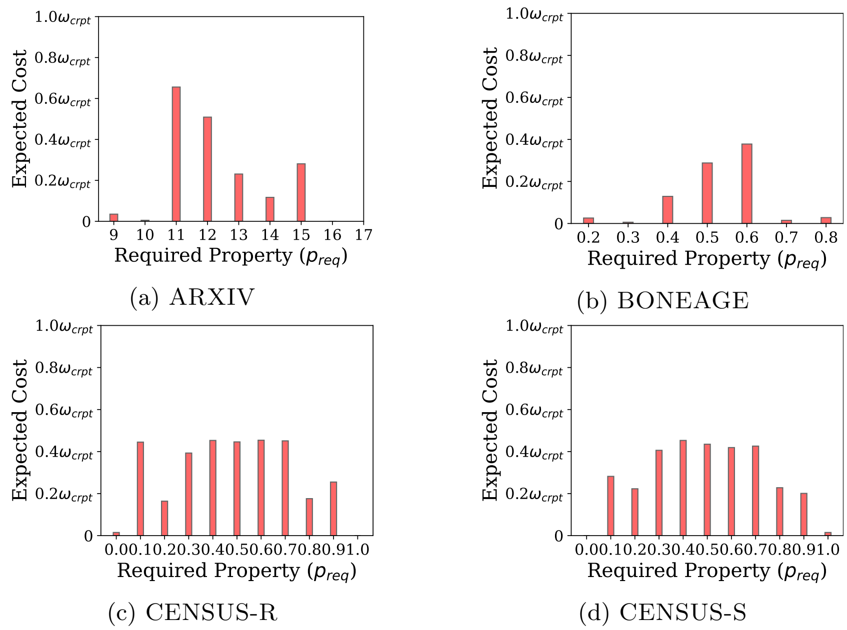
Datasets	ω_{crpt}^{comp} (s)		ω_{crpt}^{comm} (GB)	
	DistCheck	Training	DistCheck	Training
BONEAGE	1.30 ± 0.05	1367.31 ± 27.95	0.01	228.54
CENSUS-R	1.54 ± 0.15	1081.00 ± 17.00	0.01	874.06
CENSUS-S	1.68 ± 0.15	2109.78 ± 65.20	0.01	1438.38

Summary. Cryptographic attestation satisfies **R2** effectiveness and **R3** robustness, but *lacks efficiency* which limits its scalability to multiple $\mathcal{P}$s.

6.3 Hybrid Attestation

We present the effectiveness of hybrid attestation with a fixed FAR (or FRR), its impact on the respective FRR (or FAR) and the expected cost incurred.

Fixed FAR Analysis Recall that on fixing FAR, rejected $\mathcal{P}$s can request re-evaluation using cryptographic attestation as a fallback.
Effectiveness (R2). Hybrid attestation will *not change the 5% fixed FAR value.* We compute the effective FRR on using cryptographic attestation as a fallback. In practice, only $\mathcal{P}$s with FR have the incentive to request re-evaluation using



(a) ARXIV (b) BONEAGE

(c) CENSUS-R (d) CENSUS-S

Fig. 3: **Fixed FAR@5%: Expected cost on $\mathcal{D}^{ver}_{\mathcal{P}}$.** ω_{crpt} is a placeholder for ω^{comp}_{crpt} and ω^{comm}_{crpt}. Expected cost is less than cryptographic attestation (=ω_{crpt}).

cryptographic attestation. If such $\mathcal{P}$s undergo re-evaluation, *FRR is zero* as the cryptographic attestation will rectify any erroneous decision.

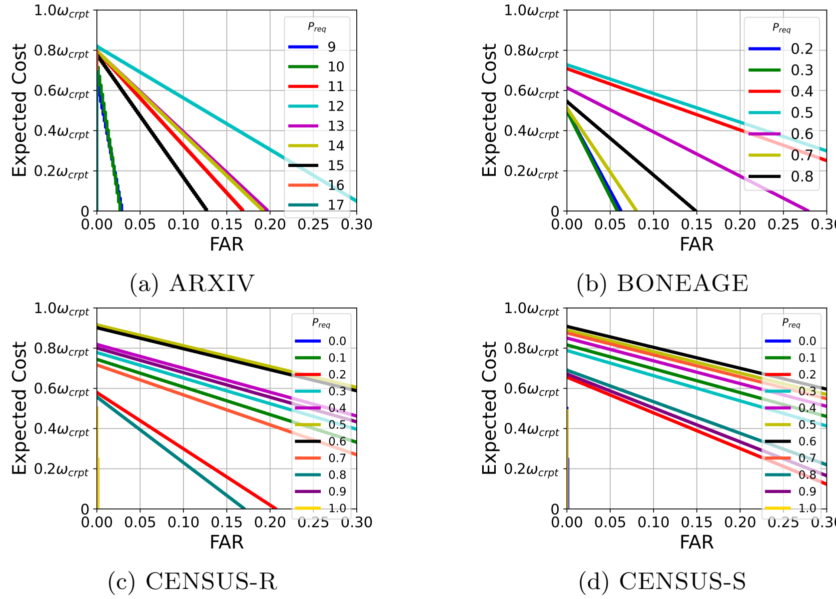
Expected Cost (R4). We evaluate the expected cost on $\mathcal{D}^{ver}_{\mathcal{P}}$. This gives the actual estimate of the cost incurred during attestation. We assume that $\mathcal{P}$s are rational, so only $\mathcal{P}$s with FR will request a re-evaluation using cryptographic attestation. Here, to compute the expected cost, $\mathbb{P}_{crpt} = \frac{\mathcal{N}_{rej}}{\mathcal{N}}$, where $\mathcal{N}_{rej}$ is the total number of rejected $\mathcal{P}$s.

We present the expected cost in Figure 3 where the values of ω_{crpt}, a placeholder for ω^{comp}_{crpt} or ω^{comm}_{crpt}, are from Table 3. Compared to cryptographic attestation with an expected cost of ω_{crpt}, hybrid attestation has a lower expected cost across different datasets and p_{req}. Additionally, since $\mathbb{P}_{crpt}$ depends on $\mathcal{N}_{rej}$ computed from inference-based attestation, the edge p_{req} values, where inference-based attestation is effective, have lower expected cost than middle p_{req} values.

Fixing FRR Here, recall that $\mathcal{V}$ conducts random spot-checks to reduce FAR.

Effectiveness (R2). $\mathcal{V}$'s choice of $\mathcal{N}_{spchk}$ determines FAR. No spot-checks corresponds to same FAR as inference-based attestation while spot-checks for all accepted $\mathcal{P}$s indicates zero FAR.

Efficiency (R4). The expected cost incurred per $\mathcal{P}$ increases with $\mathcal{N}_{spchk}$. No spot-checks corresponds to no expected cryptographic cost while spot-checks for all accepted $\mathcal{P}$s incurs a high expected cost. Hence, $\mathcal{V}$ decides $\mathcal{N}_{spchk}$ based on their application's requirement.



(a) ARXIV

(b) BONEAGE

(c) CENSUS-R

(d) CENSUS-S

Fig. 4: **Fixed FRR@5%: Trade-off between FAR and expected cost on varying $\mathcal{N}_{spchk}$ on $\mathcal{D}^{test}_{\mathcal{V}}$.** Expected cost is less than cryptographic attestation ($=\omega_{crpt}$) and effectiveness is better than inference-based attestation. ω_{crpt} is a placeholder for both ω^{comp}_{crpt} and ω^{comm}_{crpt}.

We present this trade-off between FAR and expected cost using cryptographic attestation by varying $\mathcal{N}_{spchk}$ on $\mathcal{D}^{test}_{\mathcal{V}}$ in Figure 4. We use $\mathbb{P}_{spchk} = \frac{\mathcal{N}_{spchk}}{\mathcal{N}}$ for the expected cost. Increasing $\mathcal{N}_{spchk}$, increases the expected cost while FAR decreases. Using Figure 4, $\mathcal{V}$ can determine $\mathcal{N}_{spchk}$. Once $\mathcal{V}$ decides on a suitable $\mathcal{N}_{spchk}$ using $\mathcal{D}^{test}_{\mathcal{V}}$, the actual cost and FAR value can be read from a plot (similar to Figure 4) for $\mathcal{D}^{ver}_{\mathcal{P}}$ corresponding to the chosen $\mathcal{N}_{spchk}$ (see Appendix E of our full paper [12]).

Carefully choosing $\mathcal{N}_{spchk}$ leads to a notable reduction in FAR compared with $\mathcal{N}_{spchk} = 0$ (x-axis). Additionally, we have lower expected cost compared to conducting spot-checks for all accepted $\mathcal{P}$s and purely cryptographic attestation (y-axis where $\mathcal{N}_{spchk} = 0$).

Summary. Hybrid attestation is more effective than inference-based attestation and incurs a lower expected cost than cryptographic attestation.

7 Related Work

Property Attestation in trusted computing [46, 32] allows attesting if $\mathcal{P}$'s system satisfies the desired (security) requirements without revealing its specific software or hardware configuration. We are the first to introduce such a notion in ML while presenting mechanisms for distributional property attestation.

Property Testing compares the closeness of two distributions using mean and standard deviation [3]. In contrast, we need $\mathcal{V}$ to test if $\mathcal{D}_{\mathcal{P}}^{tr}$ corresponds to the distribution expected by $\mathcal{V}$ *without* having access to $\mathcal{D}_{\mathcal{P}}^{tr}$. One can conceivably implement property testing using 2PC, which will be similar to our cryptographic attestation protocol. Chang et al. [5] combine MPC and ZKP with property testing to check for data quality. However, they consider a different setting with multiple parties and their evaluation does not account for ML.

Auditing ML Models has been explored by adapting membership inference attacks to check for compliance with "Right to Erasure" [36, 49, 34, 22]. Juarez et al. [24] use property inference to check for a specific case of distribution shift from balanced data ($p_{req} = 0.5$). Our scheme is broader by allowing attestations for arbitrary properties as required by $\mathcal{V}$. Further, their scheme is insufficient for attestation as it lacks effectiveness (**R2**) and robustness (**R3**). We address these concerns in our work. Additionally, cryptographic primitives can help audit models for fairness w.r.t. output predictions [30, 42, 47] which is different from property attestation considered in this work. "Proof-of-Learning" (also known as proof-of-training) proves that a model was trained on a specific dataset using ML [23]. However, such ML based schemes can be evaded [54, 16]. Garg et al. [18] propose proof-of-training by combining ZKP with MPC-in-the-head in a concurrent and independent work. They mention the possibility of attesting properties, but do not implement it. Moreover, their approach is limited to logistic regression (e.g., [14]). Our hybrid approach using MPC is currently the best available approach that scales to larger models.

Property Inference Attacks have been explored in different domains: image, graphs and tabular data, threat models and classification tasks [55, 58, 50, 17, 2, 35, 51, 7, 6, 58]. Defending against them is an open problem [27, 8, 20, 51].

Privacy-Preserving ML is an active research field, with much focus placed on cryptographic methods for privacy-preserving supervised and deep learning inference and training [38, 44, 29, 37]. See the survey [40] for an overview.

8 Discussions

Outsourcing as a Trade-off between Security and Efficiency. We cannot run cryptographic attestation using semi-honest 2PC protocols directly between a malicious prover $\mathcal{P}_{mal}$ and $\mathcal{V}$, because $\mathcal{P}_{mal}$ can easily change the outcome of "DistCheck" for p_{req} by flipping its share of the output bit.

Proof. Let $[\cdot]^1$ denote shares held by $\mathcal{P}_{mal}$ and $[\cdot]^2$ shares held by $\mathcal{V}$. Let $\mathcal{D}_{\mathcal{P}mal}$ denote the dataset that only $\mathcal{P}_{mal}$ holds and wants to use for cryptographic attestation. Furthermore, let v denote the true result of the verification and *out* the output of the verification protocol. Since p_{req} is known to $\mathcal{P}_{mal}$, it knows whether or not $\mathcal{D}_{\mathcal{P}mal}$ fulfills the requirement and can fool the $\mathcal{V}$ by flipping the outcome, if the requirement is not met. If $\mathcal{P}_{mal}$ wants to flip the outcome of the verification, $\mathcal{P}_{mal}$ sets $[out]^1 = 1 \oplus [v]^1$, s.t. verification yields $out = [out]^1 \oplus [out]^2 = 1 \oplus [v]^1 \oplus [v]^2 = 1 \oplus v$, where $1 \oplus v = true$ iff the true outcome $v = false$. $\square$ Alternatively, robustness against $\mathcal{P}_{mal}$ can be achieved with malicious protocols [57, 28], however, at a high cost. Then, how can we account for $\mathcal{P}_{mal}$ without using maliciously secure 2PC protocols? For this, we use secure outsourcing by introducing additional non-colluding semi-honest servers $\mathcal{S}_1$ and $\mathcal{S}_2$ that carry out the cryptographic protocol on behalf of $\mathcal{P}$ and $\mathcal{V}$ [4].

Alternative Protocols and their Limitations. Instead of outsourcing, we can replace MPC with other cryptographic protocols like ZKP. Non-Interactive ZKPs can be reused, thus the cost amortizes over multiple parties. However, they incur a high cost making them impractical as do other cryptographic mechanisms [40, 18]. On the other hand, our MPC based approach can scale to neural networks. Further, TEEs offer an alternative approach, but may pose a deployment hurdle by requiring all $\mathcal{P}$s and $\mathcal{V}$ to have a TEE. Hence, designing more efficient protocols for property attestation is left as future work.

Relation with Fairness. Fairness involves a subsequent evaluation of model predictions to gauge the consistency of metrics like the false positive rate among various subgroups. The selection of an appropriate reference dataset holds significant importance, making it unclear whether the model is actually fairness [48]. Biased datasets tend to yield more inequitable models compared to unbiased counterparts [25]. Hence, distribution equity is a prerequisite for fairness.

Whitebox Access and Inference Attacks by $\mathcal{V}$. Our setting is attestation for regulatory compliance, i.e., both $\mathcal{P}$ and $\mathcal{V}$ *co-operate* as both want attestation to succeed. If $\mathcal{V}$ is a potential buyer, whitebox access to the model is natural. If $\mathcal{V}$ is a regulator, whitebox access is still reasonable because $\mathcal{V}$ is "honest-but-curious", i.e., $\mathcal{V}$ may misuse any available information, but will not deviate from the specified protocol. Hence, $\mathcal{V}$ *must not be given the training dataset* ($\mathcal{D}^{tr}_{\mathcal{P}}$), but can be trusted not to mount other inference attacks. Also, *whitebox access is not needed* for cryptographic attestation: $\mathcal{V}$ never sees $\mathcal{D}^{tr}_{\mathcal{P}}$ in the clear because the computation is over encrypted (secret-shared) data. Further, we can also add DP to minimize privacy risks without losing attestation accuracy as distribution inference is *more successful* with DP *assuming* $\mathcal{V}$ knows the DP hyperparameters, which is reasonable in the attestation setting [51]. Thus, we expect inference-based attestation will be *more effective* with DP.

Acknowledgements

This work is supported in part by Intel (Private AI consortium) and the Government of Ontario. Additionally, this project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 850990 PSOTI). It was co-funded by the Deutsche Forschungsgemeinschaft (DFG) within SFB 1119 CROSSING/236615297 and GRK 2050 Privacy & Trust/251805230.

References

1. Atapoor, S., Smart, N.P., Alaoui, Y.T.: Private Liquidity Matching Using MPC. In: CT-RSA (2022)

2. Ateniese, G., Mancini, L.V., Spognardi, A., Villani, A., Vitali, D., Felici, G.: Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. Int. J. Secur. Netw. (2015)
3. Canonne, C.L.: A survey on distribution testing: Your data is big. but is it blue? Theory of Computing (2020)
4. Carter, H., Mood, B., Traynor, P., Butler, K.R.B.: Outsourcing secure two-party computation as a black box. In: Secur. Commun. Networks (2015)
5. Chang, I., Sotiraki, K., Chen, W., Kantarcioglu, M., Popa, R.: HOLMES: Efficient distribution testing for secure collaborative learning. In: USENIX Security (2023)
6. Chase, M., Ghosh, E., Mahloujifar, S.: Property inference from poisoning. arXiv:2101.11073 (2021)
7. Chaudhari, H., Abascal, J., Oprea, A., Jagielski, M., Tramèr, F., Ullman, J.: SNAP: Efficient extraction of private properties with poisoning. In: S&P (2023)
8. Chen, M., Ohrimenko, O.: Protecting global properties of datasets with distribution privacy mechanisms. arXiv:2207.08367 (2022)
9. Congress, U.: H.r.6580 - algorithmic accountability act of 2022 (2022), https://www.congress.gov/bill/117th-congress/house-bill/6580/text
10. Coston, A., Kawakami, A., Zhu, H., Holstein, K., Heidari, H.: A validity perspective on evaluating the justified use of data-driven decision-making algorithms. In: SaTML (2023)
11. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: ICML (2020)
12. Duddu, V., Das, A., Khayata, N., Yalame, H., Schneider, T., Asokan, N.: Attesting distributional properties of training data for machine learning (full version). arXiv:2308.09552 (2023)
13. EC, E.C.: Regulation of the european parliament and of the council laying down harmonized rules on artificial intelligence (artificial intelligence act) (2021)
14. Eisenhofer, T., Riepel, D., Chandrasekaran, V., Ghosh, E., Ohrimenko, O., Papernot, N.: Verifiable and provably secure machine unlearning. arXiv:2210.09126 (2022)
15. Englehardt, S.: Next steps in privacy-preserving telemetry with Prio. https://blog.mozilla.org/security/2019/06/06/next-steps-in-privacy-preserving-telemetry-with-prio/ (2019)
16. Fang, C., Jia, H., Thudi, A., Yaghini, M., Choquette-Choo, C.A., Dullerud, N., Chandrasekaran, V., Papernot, N.: Proof-of-learning is currently more broken than you think. arXiv:2208.03567 (2023)
17. Ganju, K., Wang, Q., Yang, W., Gunter, C.A., Borisov, N.: Property inference attacks on fully connected neural networks using permutation invariant representations. In: CCS (2018)
18. Garg, S., Goel, A., Jha, S., Mahloujifar, S., Mahmoody, M., Policharla, G.V., Wang, M.: Experimenting with zero-knowledge proofs of training. In: CCS (2023)
19. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: STOC (1987)
20. Hartmann, V., Meynent, L., Peyrard, M., Dimitriadis, D., Tople, S., West, R.: Distribution inference risks: Identifying and mitigating sources of leakage. arXiv:2209.08541 (2022)
21. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Computer Vision and Pattern Recognition (2017)
22. Huang, Y., Li, X., Li, K.: Ema: Auditing data removal from trained models. In: Medical Image Computing and Computer Assisted Intervention (2021)

23. Jia, H., Yaghini, M., Choquette-Choo, C.A., Dullerud, N., Thudi, A., Chandrasekaran, V., Papernot, N.: Proof-of-learning: Definitions and practice. In: S&P (2021)
24. Juarez, M., Yeom, S., Fredrikson, M.: Black-box audits for group distribution shifts. arXiv:2209.03620 (2022)
25. Kamiran, F., Calders, T.: Data pre-processing techniques for classification without discrimination. Knowledge and Information Systems (2011)
26. Kaviani, D., Popa, R.A.: MPC Deployments. https://mpc.cs.berkeley.edu (2023)
27. Kawamoto, Y., Murakami, T.: Local obfuscation mechanisms for hiding probability distributions. In: ESORICS (2019)
28. Keller, M.: MP-SPDZ: A versatile framework for multi-party computation. CCS (2020)
29. Keller, M., Sun, K.: Secure Quantized Training for Deep Learning. In: ICML (2022)
30. Kilbertus, N., Gascon, A., Kusner, M., Veale, M., Gummadi, K., Weller, A.: Blind justice: Fairness with encrypted sensitive attributes. In: ICML (2018)
31. Knott, B., et al.: CrypTen: Secure multi-party computation meets machine learning. In: NeurIPS (2021)
32. Kostiainen, K., Asokan, N., Ekberg, J.: Practical property-based attestation on mobile devices. In: TRUST (2011)
33. Lindell, Y.: Secure multiparty computation. CACM (2020)
34. Liu, X., Tsaftaris, S.A.: Have you forgotten? a method to assess if machine learning models have forgotten data. arXiv:2004.10129 (2020)
35. Melis, L., Song, C., Cristofaro, E.D., Shmatikov, V.: Exploiting unintended feature leakage in collaborative learning. In: S&P (2019)
36. Miao, Y., Xue, M., Chen, C., Pan, L., Zhang, J., Zhao, B.Z.H., Kaafar, D., Xiang, Y.: The audio auditor: User-level membership inference in internet of things voice services. In: PETS (2021)
37. Mohassel, P., Rindal, P.: ABY^3: A Mixed Protocol Framework for Machine Learning. In: CCS (2018)
38. Mohassel, P., Zhang, Y.: SecureML: A system for scalable privacy-preserving machine learning. In: S&P (2017)
39. MPC-Alliance: MPC Alliance. https://www.mpcalliance.org (2023)
40. Ng, L.L., Chow, S.M.: SoK: Cryptographic neural-network computation. In: S&P (2023)
41. Papernot, N., McDaniel, P., Sinha, A., Wellman, M.P.: SoK: Security and privacy in machine learning. In: EuroS&P (2018)
42. Park, S., Kim, S., Lim, Y.s.: Fairness audit of machine learning models with confidential computing. In: WWW (2022)
43. Pasquini, D., Ateniese, G., Bernaschi, M.: Unleashing the tiger: Inference attacks on split learning. In: CCS (2021)
44. Patra, A., Schneider, T., Suresh, A., Yalame, H.: ABY2.0: Improved Mixed-Protocol Secure Two-Party Computation. In: USENIX Security (2021)
45. Riazi, M.S., Weinert, C., Tkachenko, O., Songhori, E.M., Schneider, T., Koushanfar, F.: Chameleon: A hybrid secure computation framework for machine learning applications. In: ASIACCS (2018)
46. Sadeghi, A.R., Stüble, C.: Property-based attestation for computing platforms: Caring about properties, not mechanisms. In: Workshop on New Security Paradigms (2004)
47. Segal, S., Adi, Y., Pinkas, B., Baum, C., Ganesh, C., Keshet, J.: Fairness in the eyes of the data: Certifying machine-learning models. In: AIES (2021)

48. Shamsabadi, A.S., Wyllie, S.C., Franzese, N., Dullerud, N., Gambs, S., Papernot, N., Wang, X., Weller, A.: Confidential-PROFITT: Confidential PROof of fair training of trees. In: International Conference on Learning Representations (2023)
49. Song, C., Shmatikov, V.: Auditing data provenance in text-generation models. In: KDD (2019)
50. Suri, A., Evans, D.: Formalizing and estimating distribution inference risks. PETS (2022)
51. Suri, A., Lu, Y., Chen, Y., Evans, D.: Dissecting distribution inference. In: SaTML (2023)
52. Weng, C., Yang, K., Xie, X., Katz, J., Wang, X.: Mystique: Efficient conversions for zero-knowledge proofs with applications to machine learning. In: USENIX Security (2021)
53. Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R.R., Smola, A.J.: Deep sets. In: NeurIPS (2017)
54. Zhang, R., Liu, J., Ding, Y., Wang, Z., Wu, Q., Ren, K.: "adversarial examples" for proof-of-learning. In: S&P (2022)
55. Zhang, W., Tople, S., Ohrimenko, O.: Leakage of dataset properties in Multi-Party machine learning. In: USENIX Security (2021)
56. Zhang, Z., Chen, M., Backes, M., Shen, Y., Zhang, Y.: Inference attacks against graph neural networks. In: USENIX Security (2022)
57. Zheng, W., Deng, R., Chen, W., Popa, R.A., Panda, A., Stoica, I.: Cerebro: A platform for multi-party cryptographic collaborative learning. In: USENIX Security (2021)
58. Zhou, J., Chen, Y., Shen, C., Zhang, Y.: Property inference attacks against GANs. arXiv:2111.07608 (2021)

Appendix

A Details for Cryptographic Attestation

Protocol Instantiation. Given the proof objectives, our goal now is to find a concrete cryptographic protocol variant to instantiate property attestation. To this end, we need the following: (1) primitives for ML training, (2) security against $\mathcal{P}_{mal}$, and (3) an efficient protocol instantiation that allows us to use the cryptographic property attestation in a real setting. We rule out TEEs because of their susceptibility to side-channel attacks, hence violating (2). Because of their impracticality to deploy in real-world sized models, thus violating (3), we also rule out Homomorphic Encryption (HE) [40]. As of now, there are efficient ZKPs for verifiable inference [52], but not for backpropagation during ML training, violating (1) and ruling out ZKPs for our instantiation. As state-of-the-art works in PPML based on MPC such as CrypTen [31] satisfy all three required properties for our cryptographic property attestation, we choose MPC as instantiation.
Property Attestation as a Cryptographic Protocol. Assuming a malicious $\mathcal{P}$ and semi-honest $\mathcal{V}$, we construct a cryptographic protocol based on MPC in the outsourcing setting with two non-colluding semi-honest servers $\mathcal{S}_1$ and $\mathcal{S}_2$. The protocol consists of the following steps:
1. Initiate input-sharing phase between $\mathcal{P}$ and $\mathcal{S}_1$, $\mathcal{S}_2$.

2. $\mathcal{S}_1$ and $\mathcal{S}_2$ run `DistCheck` on their input shares of $\mathcal{D}^{tr}_{\mathcal{P}}$.
3. $\mathcal{S}_1$ and $\mathcal{S}_2$ securely train on their input shares, which yields $\mathcal{M}_{2pc}$
4. $\mathcal{S}_1$ and $\mathcal{S}_2$ send output shares of `DistCheck` and $\mathcal{M}_{2pc}$ to $\mathcal{V}$ for reconstruction of plaintext outputs.
5. $\mathcal{V}$ checks if `DistCheck` succeeded using the output shares.

(1) Input-sharing Phase. $\mathcal{P}$ computes additive secret-shares of the training dataset ($\mathcal{D}^{tr}_{\mathcal{P}}$). Hence, the prover computes $[\mathcal{D}^{tr}_{\mathcal{P}}]^1$, $[\mathcal{D}^{tr}_{\mathcal{P}}]^2$ such that $[\mathcal{D}^{tr}_{\mathcal{P}}]^1 + [\mathcal{D}^{tr}_{\mathcal{P}}]^2 = \mathcal{D}^{tr}_{\mathcal{P}}$ and sends $[\mathcal{D}^{tr}_{\mathcal{P}}]^1$ to $\mathcal{S}_1$ and $[\mathcal{D}^{tr}_{\mathcal{P}}]^2$ to $\mathcal{S}_2$.
(2) Secure Computation of `DistCheck`. Given the input shares of $\mathcal{D}^{tr}_{\mathcal{P}}$, $\mathcal{S}_1$ and $\mathcal{S}_2$ compute `DistCheck` by computing the distributional property of $\mathcal{D}^{tr}_{\mathcal{P}}$ and comparing against p_{req}.
(3) Secure Training of $\mathcal{M}_{2pc}$. Given the input shares of both $\mathcal{D}^{tr}_{\mathcal{P}}$ and $\mathcal{M}_p$, both servers jointly run the protocols for secure training as described in [31]. CrypTen has efficient secure protocols for both the forward pass and back propagation. We refer to [31] for the protocol details. We emphasize that $\mathcal{S}_1$ and $\mathcal{S}_2$ use the previously obtained shares of $\mathcal{D}^{tr}_{\mathcal{P}}$ from the input-sharing phase, because they were used for `DistCheck`. This leaves no room for $\mathcal{P}$ to cheat by choosing different shares of another dataset $\mathcal{D}' \neq \mathcal{D}^{tr}_{\mathcal{P}}$ for training.
(4) Verify `DistCheck`. $\mathcal{S}_1$ and $\mathcal{S}_2$ send the output shares $[v]^1$ and $[v]^2$ of `DistCheck` to $\mathcal{V}$ who now locally reconstructs the output $v = [v]^1 + [v]^2$. Now, $v = 1$ iff `DistCheck` was successful. Then, $\mathcal{V}$ reconstructs $\mathcal{M}_{2pc}$ by locally adding the output shares from $\mathcal{S}_1$ and $\mathcal{S}_2$, i.e., $\mathcal{M}_{2pc} = [\mathcal{M}_{2pc}]^1 + [\mathcal{M}_{2pc}]^2$.
Security and Correctness of Cryptographic Attestation. Since we implement cryptographic attestation using CrypTen, we refer to [31] for the detailed security proofs. Assuming CrypTen's protocols satisfy security and correctness, we discuss the security (i.e., preserving the privacy of $\mathcal{P}$'s dataset $\mathcal{D}^{tr}_{\mathcal{P}}$) and correctness for cryptographic attestation.

The privacy of $\mathcal{D}^{tr}_{\mathcal{P}}$ naturally follows from the security guarantees of linear secret-sharing [19, 31]. For correctness, we identify two cases:

- **when $\mathcal{P}$ does not cheat** and correctly creates an input sharing for $\mathcal{D}^{tr}_{\mathcal{P}}$, then, correctness follows from the underlying secret-sharing scheme.
- **when $\mathcal{P}$ cheats**, then $\mathcal{P}$ can
  - simply abort instead of providing a valid sharing to escape attestation, hence the whole attestation fails.
  - create incorrect shares $[\mathcal{D}']^1$ and $[\mathcal{D}']^2$ where $[\mathcal{D}']^1 + [\mathcal{D}']^2 = \mathcal{D}' \neq \mathcal{D}_{\mathcal{P}}$. However, since the two shares indeed compute $\mathcal{D}'$, the input-sharing is done correctly, just for a different input value. MPC does not secure against choosing the "wrong" input value. However, this is not a problem, because if $\mathcal{D}'$ satisfies p_{req}, we still obtain a valid model with respect to the distributional property.

After the input-sharing phase, $\mathcal{P}$ does not participate in the protocol. Hence, there is no further cheating as $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{V}$ are semi-honest. Secure computation of `DistCheck` only consists of secure additions and comparison, hence the correctness and privacy of `DistCheck` as well as secure training of $\mathcal{M}_{2pc}$ directly follows from the security guarantees of CrypTen [31].