Thinking Out of the (Black)-Box

Tools for machine learning audits
in the presence of *deceptive* model providers.

Doctoral Thesis

—

Augustin Godinot

Abstract

Machine learning-based prediction services are now widely deployed across industries by companies, governments, and individuals. Yet, these services often rely on a complex AI supply chain, whose components (training data, models, infrastructure), while critical to their performance, are partially or completely hidden to the final users. Thus, to an external user or regulator, these prediction services appear as black-boxes, complicating their evaluation and opening avenues for manipulations. In the presence of deceptive model providers, this thesis aims to understand the fundamental limits to black-box auditing and designing protocols to provide guarantees beyond the black-box interaction model. This manuscript presents three contributions towards that goal. First, I present a formalization of this quest for the minimal assumption beyond the black-box as a prior construction problem and provide a new audit method leveraging the labeled data available to the auditor. Then, I study the benefits of requesting the hypothesis class used by the platform to inform the audit. Finally, in an attempt to cheaply detect post-audit attacks, I introduce a new model fingerprint baseline and theoretical analysis to detect model change.

Table of Contents

List of Figures

List of Theorems

Glossary

AI. Artificial Intelligence 23, 24, 29, 30, 35, 100, 101

API. Application Programming Interface 29, 33, 35, 38, 40

CSAM. Child Sexual Abuse Material 26

DMA. Digital Markets Act 30

DSA. Digital Services Act 30, 99

EU. European Union 30

FPR. False Positive Rate 31, 91, 102

ML. Machine Learning v, ix, 23, 24, 25, 26, 29, 30, 32, 33, 34, 35, 36, 37, 44, 45, 99, 101, 102

NLP. Natural Language Processing 35

OECD. Organisation for Economic Co-operation and Development 35

PII. Personally Identifiable Information 26

ROC. Receiver-Operator Curve 91

SGD. Stochastic Gradient Descent 28

TPR. True Positive Rate 31, 91

Résumé en Français

Les algorithmes d'apprentissage automatique ont trouvé de nombreuses applications dans nos vies. Des algorithmes de traitement d'image permettant à nos smartphones de prendre des photos parfaites malgré des capteurs minuscules, à l'accélération des prévisions météorologiques, en passant par l'assistance à la rédaction, la prédiction de fraude, ou la détection de pathologies, l'apprentissage automatique est utilisé autant par les entreprises, les gouvernements que par le grand public. L'observation de son déploiement massif, particulièrement au travers du paradigme du « Machine Learning as a Service » (MLaaS), met en lumière un paradoxe structurel : alors que ces systèmes régissent des décisions à fort enjeu, ils demeurent, pour la majorité des acteurs, des « boîtes noires » hermétiques. Pour l'utilisateur final ou le régulateur, l'interaction se réduit à une interface d'entrée-sortie, occultant totalement les données d'entraînement, l'architecture du modèle, ses paramètres internes et ses objectifs.

Dans ce contexte d'opacité, l'audit des systèmes d'apprentissage automatique émerge comme un instrument indispensable de gouvernance. Il se définit comme une procédure d'évaluation méthodique et indépendante visant à vérifier la conformité d'un système d'apprentissage automatique avec un ensemble d'exigences prédéfinies, qu'elles soient techniques, éthiques ou réglementaires. Le point de départ de ces travaux de thèse est l'adoption d'une approche d'audit qui prend en compte le comportement potentiellement antagoniste du fournisseur de services, rompant ainsi avec l'hypothèse de coopération qui sous-tend implicitement la majorité des protocoles d'audit actuels. En effet, dans un environnement économique concurrentiel, soumis à des cadres réglementaires stricts tels que l'AI Act européen, un fournisseur dispose d'incitations rationnelles à adopter des comportements stratégiques, voire trompeurs.

Ce manuscrit adresse donc ce scénario du pire cas : comment garantir la fiabilité, l'équité et la performance d'un algorithme lorsque son concepteur tente activement de dissimuler ses défaillances ? Bien-entendu, si l'auditeur ne dispose d'aucune autre information ni aucun autre accès au système qu'au travers de requête-réponse, il n'a aucun moyen de détecter quelconque manipulation de la part du fournisseur de services. Pour aborder ce problème, tout au long de cette thèse nous nous efforcerons de répondre à la question

Afin de dépasser le modèle de l'audit en boîte noire, de quelle information supplémentaire l'auditeur a-t'il besoin pour obtenir des garanties de validité de ses mesures?

Chapitre 1: Introduction

Ce chapitre introductif pose le contexte général de la thèse : l'omniprésence des services de prédiction basés sur l'apprentissage automatique dans des secteurs critiques et la complexité croissante de leur chaîne d'approvisionnement. Nous effectuons un bref rappel des notations et pratiques régissant la création de systèmes d'apprentissage automatique. Ensuite, ce chapitre détaille les différentes étapes d'un audit: la définition des objectifs, les mesures utilisées et les différents niveaux d'accès au système. Enfin, nous présentons les motivations qu'un fournisseur aurait à manipuler un audit et le modèle de sécurité correspondant: un fournisseur qui manipule les réponses du modèle sur l'ensemble d'audit pour paraître équitable et performant.

Chapitre 2: À la recherche d'un a priori

Dans ce deuxième chapitre, nous abordons la formalisation de l'audit comme un problème de construction d'une *connaissance a priori* pour l'auditeur. Face à l'impossibilité théorique de garantir la conformité d'un modèle boîte noire sans hypothèses supplémentaires, nous introduisons le cadre de l'*audit avec a priori*. Nous illustrons notre proposition à travers l'utilisation de données étiquetées dont disposerait l'auditeur comme source de cet a priori pour contraindre l'espace des modèles possibles et détecter les manipulations.

Nous développons une analyse théorique démontrant que la connaissance de la distribution des données permet de borner la capacité de nuisance du fournisseur. Nous introduisons le concept d'« inéquité dissimulable », qui quantifie l'ampleur de l'inéquité de traitement qu'un fournisseur peut cacher tout en satisfaisant les contraintes de l'audit.

Sur le plan empirique, nous évaluons cette méthode sur des jeux de données réels (comme CelebA et ACSEmployment). Nos expériences montrent comment l'inéquité dissimulable évolue en fonction du budget de requêtes de l'auditeur et des différentes stratégies de manipulation du fournisseur. Nous mettons en évidence que si l'accès à un jeu de données étiqueté renforce l'audit, il existe toujours une marge de manœuvre pour le fournisseur, qui diminue à mesure que l'auditeur acquiert plus de données.

Chapitre 3: Connaissance de la classe d'hypothèses

Ce chapitre examine une hypothèse alternative pour renforcer l'audit : la connaissance par l'auditeur de la classe d'hypothèses (la famille de modèles et l'algorithme d'entraînement) utilisée par le fournisseur. Nous cherchons à déterminer si cette information permet de concevoir des audits plus robustes contre les manipulations, notamment via des stratégies d'audit actif où l'auditeur choisit ses requêtes en fonction des réponses précédentes de la plateforme.

Nous démontrons que pour de nombreuses classes de modèles, aucune stratégie d'audit actif ne surpasse l'échantillonnage aléatoire simple. Nous établissons un lien formel entre la manipulabilité d'une classe de modèles et sa capacité (mesurée par la complexité de Rademacher). Plus un modèle est complexe et capable d'apprendre des motifs variés, plus il est facile pour le fournisseur de dissimuler son comportement réel sans être détecté par l'audit.

L'analyse est illustrée par l'étude des modèles « dictionnaires » et étendue aux modèles classiques (arbres de décision, modèles linéaires, réseaux de neurones). Les expériences confirment que pour les modèles modernes à grande capacité, malgré la connaissance de la classe d'hypothèses par l'auditeur, le fournisseur peut aisément tromper l'audit. Le coût en termes de performances pour le fournisseur de tromper l'audit reste négligeable, ce qui souligne les limites de cette approche pour la régulation des systèmes d'IA complexes.

Chapitre 4: Surveillance des changements de modèle

Face aux limites de l'audit ponctuel identifiées dans les chapitres précédents, nous nous tournons dans ce chapitre vers la surveillance continue via l'empreinte de modèle (fingerprinting). L'objectif est de détecter si le modèle déployé a été modifié après l'audit par le fournisseur, par exemple pour réduire les coûts d'exploitation ou favoriser un groupe d'utilisateurs.

Nous proposons le cadre d'analyse QuRD (Query, Representation, Detection) pour déconstruire et évaluer les méthodes de fingerprinting existantes. Cette analyse révèle que la littérature actuelle souffre de faiblesses méthodologiques : les benchmarks utilisés sont souvent triviaux et les méthodes complexes basées sur des exemples adversariaux s'avèrent fragiles.

Nous introduisons une méthode de référence simple mais efficace, nommée AKH, qui performe aussi bien, voire mieux, que les techniques de l'état de l'art sur la plupart des tâches de détection de vol ou de modification de modèle. Les résultats expérimentaux sur des benchmarks comme ModelReuse et SACBench montrent que des stratégies simples d'échantillonnage sont souvent suffisantes pour détecter les changements. Nous concluons que la complexité des méthodes actuelles

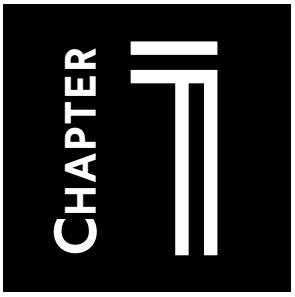
n'est pas justifiée et recommandons l'usage de bases de référence robustes pour les futurs travaux de surveillance.

Chapitre 5 : Conclusion

Le dernier chapitre synthétise les apports de la thèse. Nous avons montré que l'audit boîte noire pur est intrinsèquement limité face à des adversaires capables de manipulations. Les tentatives pour renforcer l'audit par la connaissance de la classe d'hypothèses se heurtent à la grande capacité des modèles modernes, qui permet de cacher des comportements déviants avec un coût minime pour le fournisseur.

Nous discutons des implications de ces résultats pour la gouvernance de l'IA. Puisque l'audit technique seul ne suffit pas à garantir la fiabilité, nous suggérons de déplacer la charge de la preuve vers les fournisseurs (par exemple via des certificats cryptographiques, bien que coûteux) et de réintégrer les utilisateurs dans la boucle de gouvernance. Enfin, nous élargissons la perspective aux dynamiques de pouvoir inhérentes aux systèmes algorithmiques, soulignant que les solutions techniques doivent s'accompagner d'une réflexion politique et sociale sur le contrôle des infrastructures numériques.

Introduction



> Snowball conjured up pictures of fantastic machines which would do their work for them while they grazed at their ease in the field or improved their minds with reading and conversation.
>
> — Animal Farm, Orwell

From the image processing algorithm enabling our smartphones to take pixel-perfect pictures despite tiny sensors [5] to speeding-up weather predictions [6, 7], and providing writing assistance [8], Machine Learning (ML) systems have found ways in our pockets and in our lives. Because it allows to analyze and use the huge amount of data harvested along the development of the internet, Artificial Intelligence (AI) has enabled technologies that would not have been feasible otherwise. Among them, we could cite speech recognition, translation, information retrieval, object recognition, recommender systems, large scale biometric identification or targeted online advertising.

Despite the growing social and individual impact of the decisions made by these ML systems, the data and proprietary algorithms that run them are obscured by their model providers, presented to the public as a **black-box** that requires no user understanding. This lack of transparency creates an **accountability gap**. Without insight into internal workings, decision logic, and training data, it is virtually impossible for users or regulators to **verify claims** of performance, detect harm, or assign liability for negative outcomes. Thus, the responsibility to scrutinize these ML systems, hold their model providers accountable and raise public awareness on how to use them has fallen on scientists, journalists and regulators. In this setting, the present manuscript studies audits as a crucial tool to hold ML systems providers accountable to their users and society.

A **Machine Learning audit** is an independent, systematic verification of claimed properties of a ML system. ML audits have been used to verify claims on

performance metrics and disparities [9], privacy leakage [10] and safety measures [11]. However, effective auditing comes with its own challenges. It requires the auditor to collect data that is both representative of the system's actual use case and relevant to the tested property. Moreover, auditors have limited access to the studied ML systems, often restricting them to input-output queries, interacting with the system as a black-box. Finally, the complex nature and massive scale of modern ML models makes it extremely challenging to understand the causes of discrepancies or undesired outcomes in predictions, even when observational data is available.

As if the auditing task was not complicated enough, the existing ML auditing literature has overlooked a critical factor: **model providers have incentives to evade, cheat or manipulate ML audits**. Because of the large data and compute requirements [12], emerging data licensing practices [13], and strategic partnerships [14, 15], companies providing ML systems are incentivized to act as monopolistic profit machines [16]. As a result, thanks to its monopoly position, the AI system provider can present whatever output they want to their users without risking churn. It thus becomes rational to alter what is presented to the auditor to pass the audit without having to modify the system exposed to the users. In the end, the provider only has to **detect the audit** (e.g. by detecting the use of public evaluation datasets, detecting specific IP addresses or looking for specific query patterns) and **manipulate what is exposed to the auditor** (e.g. alter the data, predictions or internal documents sent to the auditor) to pass the audit without having to modify the system exposed to the users.

Even forcing the provider to grant the auditor full access to the ML system would not completely alleviate manipulations: how would the auditor know if they are not in a sandbox? Therefore, in this manuscript, I seek a sensible trade-off: continue to treat the provided APIs and user interfaces of the AI system as black-boxes but look for **additional information** to help interpret and detect manipulations of the audit results. As obtaining this additional information can be costly for the auditor, one of the challenges will be to understand *how much* is needed.

In light of how easy it is to manipulate ML audits and how important they are to hold ML providers accountable, the following question will guide us along the manuscript.

> To go beyond the black-box model, what *minimal* additional *information* should the auditor request to achieve meaningful audit validity *guarantees*?

The purpose of this chapter is first to establish some background and notations I will use to describe ML systems (Section 1.1) and their evaluation via ML

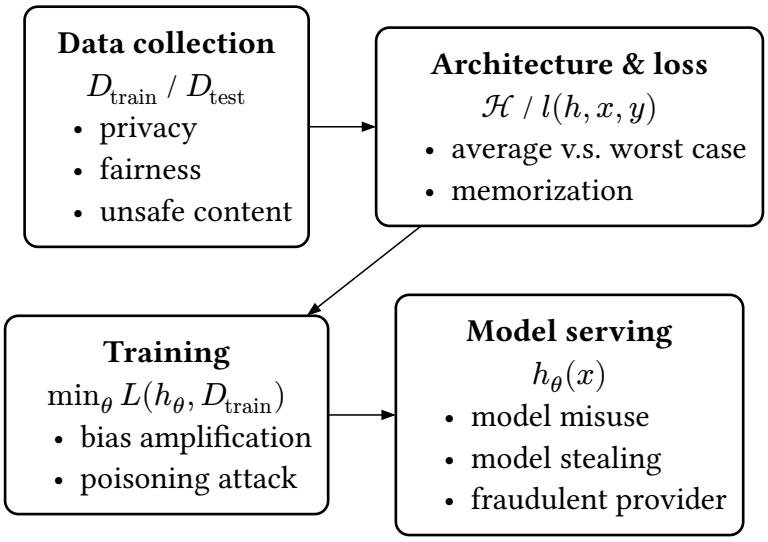


Figure 1.1: The Machine Learning pipeline and examples of challenges associated with each step.

audits (Section 1.2). Then, I will detail how the interactions between the auditor and the model provider can be weaponized by the model provider to game the audit (Section 1.3). Finally, I will succinctly present the contributions of this thesis (Section 1.4).

1. A primer on Machine Learning systems

The term *Machine Learning* was coined by A. Samuel in 1959 to describe computers playing each other to learn checkers [17]. The reason we turn to Machine Learning is to transform data into *predictions*, e.g. about the weather, the next word I will type on my keyboard or the presence of a cat in an image. There are a lot of moving parts in modern ML systems: prediction often go through a series of filters and business rules before finally reaching the user, the training data is continuously collected, the model providing the predictions is regularly retrained and the quality of predictions is monitored. For simplicity, I will use the following convention: the *predictor* will refer to the end to end ML system, whereas the *model* will designate the single, instantiated and trained ML model. For example, in a product that detects hate speech, there might be a model to embed the text into vectors, a classifier trained separately and a layer to abstain when the confidence is too low. The predictor is the entire pipeline, while embedding model, the classifier and the uncertainty layer are all different models.

The practice of predictive model building can be stylized in four steps: 1) data collection, 2) architecture and loss choice, 3) training and evaluation and finally 4) model serving. The purpose of this section is threefold: to define notations used throughout the manuscript, to provide a general background on the craft of train-

ing ML models, and to highlight well-known pitfalls and systemic vulnerabilities that can arise at each of the four stages.

1.1. Data collection

The first step to create a predictor is to collect data related to the problem at hand. This data could be images, texts, physical measurement, anything that can ultimately be represented as numbers or vectors. Each data point x is paired with a *target* prediction y. The space of all possible points is called the input space $\mathcal{X}$, the space of all targets, the output space $\mathcal{Y}$.

In supervised learning, the target is a label or value, also called *ground truth*, observed when collecting the data point. It describes the observed outcome for the piece of data if refers to, e.g. the animal pictured in the image, or the temperature associated to the measured sensor voltage. Otherwise, the target could be the data point itself (e.g. in generative modeling), a transformed version of the data point (e.g. in machine translation) or even a textual description (e.g. in image captioning). Once collected, the data is split into a *train* dataset $D_{\text{train}} \subset \mathcal{X} \times \mathcal{Y}$ that will be used to build the predictive model and a *test* dataset $D_{\text{test}} \subset \mathcal{X} \times \mathcal{Y}$ that will be used to evaluate it.

How the data is collected, its quality, and diversity directly affects the downstream performance of the predictor [18] but not only.

Privacy The collection of Personally Identifiable Information (PII) in the training data exposes the *privacy* of people, even if the data is not shared outside of the ML system. In fact, the predictions can be used to reconstruct private information in the training data [19], guess if some person was in the training data [20] or even discover the name of a person from a portrait image [21].

Fairness Biases in the data can create large *performance disparities* in the trained predictor. For example, computer vision models are very good at learning shortcuts [22]: if all images of cows are with a blue sky background then any cow image in the U.K. will not be correctly classified.

Unsafe content With the imperative to build ever larger datasets, from all corners of the Internet, filtering harmful content such as Child Sexual Abuse Material (CSAM) [23] becomes crucial but remains difficult. Worse, against common intuition, just scaling the dataset size can actively increase racial and gender biases of downstream predictions [24].

1.2. Model architecture and loss

Once the model provider collected the data, they have to choose how to generate predictions and what a good prediction is. How to generate predictions is specified

by a model architecture or hypothesis class. A model architecture is a parametrized class of functions

$$\mathcal{H} = \{h_\theta : \mathcal{X} \to \mathcal{Y} \mid \theta \in \Theta\}, \tag{1.1}$$

in which functions, parametrized by a set of parameters $\theta \in \Theta$, take a data point as input and return a prediction as output. The quality of a prediction is specified by the model provider via a *loss* function

$$l : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+. \tag{1.2}$$

The loss takes a prediction function $h_\theta \in \mathcal{H}$, a data point $x \in \mathcal{X}$ and a target prediction $y \in \mathcal{Y}$ and returns a positive value $l(h, x, y)$ that is low if the prediction $h_\theta(x)$ is close to the target y and high otherwise. Finally, to estimate the performance of its prediction function h_θ, the losses of individual data points are aggregated into a single value called the *empirical risk* $L(h_\theta, D_{\text{train}})$.

Definition 1.1 (Empirical risk) *Given a predictor $h : \mathcal{X} \to \mathcal{Y}$, a dataset $D \subset \mathcal{X} \times \mathcal{Y}$ and a loss function l, the empirical risk is defined as*

$$L(h_\theta, D_{\text{train}}) = \frac{1}{|D_{\text{train}}|} \sum_{(x,y) \in D_{\text{train}}} l(h_\theta, x, y).$$

The choice of loss function defines how the optimal model behaves, and, as with the data, the choice of loss and architecture have an impact beyond the performance of the resulting predictions.

Average case Choosing to minimize the *average* loss, while mathematically convenient, can be harmful when there are identifiable groups in the data (e.g. genders, races, income, location ...). In fact, even if a model has a low empirical risk, there might exist individuals or groups on which the model perform much worse than the average [25].

Memorization Modern chat-bots are prone to memorization. For example, prompting LIama 3.1 70B with the beginning of a sentence of Harry Potter will make it regurgitate large parts of the book [26]. It turns out that the number of parameters of the chosen architecture plays a crucial role in how much data it will *memorize* [27]. This has implications on *intellectual property*: model providers argue that collecting data to create a model is *fair use*, i.e. the model is a substantial transformation of the data. However, if the model regurgitates the data easily, it might be considered as a copy [28].

1.3. Model training

Having collected the necessary training data and chosen a suitable architecture and loss function, the model provider now has to find a good choice of parame-

ters $\theta \in \Theta$ according to its loss function. Choosing "good" parameters values θ^* amounts to minimizing the empirical risk on the train data.

$$\theta^* \in \arg\min_{\theta} L(h_\theta, D_{\text{train}}) \tag{1.3}$$

Since deriving the exact analytical solution for θ^* is infeasible in general, the parameters are computed using variants of gradient descent. First introduced by A.-L. Cauchy to solve systems of equations [29], gradient descent iteratively refines an initial guess θ_0 towards the optimal solution θ^*. In the convex setting, i.e. when the empirical risk is convex in the parameter θ, theory tells us that we are guaranteed to reach the global optimum using gradient descent. See [30] an in-depth overview of convergence results.

However, the massive size of the train set D_{train} makes computing the risk gradient $\nabla_\theta L$ prohibitively costly in terms of memory. Modern optimization methods thus rely on improvements of the Stochastic Gradient Descent (SGD) [31] algorithm. To avoid computing $\nabla_\theta L$, the idea is to sample a *batch* $B \subset D_{\text{train}}$ of b points and compute the gradient $\nabla_\theta L(h_\theta, B)$. The choice of optimization method is not benign. Not only it requires tuning important hyperparameters (e.g. learning rate or the batch size b), it also carries fairness and security risks.

Bias amplification Models reproduce the bias in the training data, they also exacerbate it. It has been shown that the imbalance, a.k.a. the difference in the number of training sample from two groups, slows the training dynamics of the minority group [32]. As a result, classical training techniques such as gradient normalization and early stopping favor even more the majority class, almost suppressing the impact of the minority on the final model [33].

Poisoning attacks One way to guarantee an unbiased SGD estimate of the full gradient is to sample the batches B uniformly from D_{train}. However, just by reordering data batches, an adversary can insert any behavior in the model (i.e. poison it) without altering the data itself or the rest of the training procedure [34].

1.4. Model serving

Once all the data, architecture and training techniques have been chosen, the challenge is to reduce as much as possible the cost of serving the predictor. With models sizes continuously increasing, optimization of the serving phase, via combinations of model sparsity [35] or quantization [36] has become more common.

All of these operations need to be tuned correctly as they can greatly affect the performance of the resulting predictor. In addition, even if the model was perfectly trained and thoroughly evaluated, the data it was based on might become outdated due to *distribution shifts* [37]. It requires continuously collecting new data and re-training or adapting the model. Finally, at this model serving stage anybody

can finally access the model or its predictions, which involves an additional set of challenges.

Model misuse The content produced by generative models have reached a disturbing level of fidelity with original content. Without citing the litany of potential misuses of this fidelity, one example is the generation of non-consensual intimate images [38], sometimes, by teenagers [39]. Preventing such misuse is not only technically difficult, it also requires a discussion on what to censor and who takes the decision. A discussion that large model providers have for now tried to elude.

Model stealing Often, model providers do not return only raw predictions. When serving the model via an Application Programming Interface (API), they also include other useful information such as the model confidence or even the prediction scores (the score the predictor assigns to each possible prediction). All this information can be exploited by malicious users to steal the functionality or the behavior of the predictor [40, 41], which is hard to defend against without decreasing the utility of the victim model [42].

Fraudulent provider In the two previous points, the risk came from the user abusing the predictor, this last risk comes from the provider. Because hosting and serving larger and larger ML models is costly, model providers have incentives to *overcharge* users. For example, they could present a quantified or altered model in place of a full-precision model [43], use lower-grade inference hardware [44], or in the context of chat-bots, exploits the subtleties of tokenization to charge the user more tokens than they used [45].

ᔕ

In the last four subsections, I briefly introduced the craft of ML system creation, brushed on the complexity of creating practical and useful predictors, as well as some of the risks introduced by these systems. In the next section, I will address a pressing question that I have purposefully eluded until now: how and who should evaluate those predictors?

2. Machine learning audits

Selling Machine Learning products is a balancing act between performance, cost and security. Thus, to have their interests considered, users need to be able to weight in on the scale. To that end, as independent evaluations, audits can form a base on which users and regulators can ask the platform to rectify their system or, in the absence of action, take the matter to court.

1. Birhane uses the terminology AI audit, when we used ML audit. Both describe audits of systems that provide predictions, irrespective of the actual algorithms, AI or ML, behind those decisions.

Definition 1.2 (ML audit [2]) *A ML[1] audit is defined as any independent assessment of an identified audit target via an evaluation of articulated expectations with the implicit or explicit objective of accountability.*

The goal of this section is to dissect this definition and provide some context on ML audits.

2.1. Why? *"the objective of accountability"*

Audits have a long history [46, 47], which has only recently been joined by algorithmic audits [48]. Original form of audits served as a check against accounting fraud, they held the citizen in charge of monetary transfers accountable [47].

Modern ML systems now influence our lives beyond financial considerations. Some, such as social benefits fraud detection systems, impact people directly. Some, as RealPage [49], affect society indirectly, by acting as a collusion mechanism between landlords leading to global rent increase. The objective of ML audits is to detect and publicize these effects in order to incite model providers to correct them to avoid bad publicity.

In addition to incentives for better systems by targeting the reputation of model providers, audits are also a ML governance tool [50]. The European Union (EU) has had directives (i.e. goals that have to be implemented by member states) that specifically address the notion of equal treatment between individuals since the early 2000s. Recognizing the ever increasing autonomy delegated to ML systems, the EU has produced a digital regulation (i.e. binding acts that apply as is across the EU) package, the Digital Services Act (DSA) [51], Digital Markets Act (DMA) [52] and AI Act [53], to face the new risks they introduce. The enforcement of these three regulations revolve around the idea of regular risk assessments, with more stringent requirements for services with higher risks. In this setting, ML audits act as tools for regulators to enforce these regulations. Given the legal consequences of non-compliance, it it thus crucial to ensure the robustness of ML audits.

2.2. What? *"the evaluation of an identified target"*

Nobody enjoys being reduced to a number, let alone aggregated with their peers into a bar plot. Yet these are the two fundamental assumptions underlying how we evaluate ML systems: we model users as a *distribution* and we summarize their experience through *metrics*.

Metrics can be grouped into two categories: *functional* metrics, which measure performance, and *non-functional* metrics, which measure constraints.

Functional metrics Functional metrics measure how well a system achieves its goals. For predictors, they measure what kind and how many mistakes the model made. Among them are accuracy (the proportion of correct predictions),

True Positive Rate (TPR) (proportion of correct predictions among predictions that should have been positive), False Positive Rate (FPR) (the proportion of wrong predictions among predictions that should have been negative).

Definition 1.3 (A few functional metrics) *Let $h : \mathcal{X} \to \mathcal{Y}$ be a predictor, and (X, Y) random variables following the joint (input, target) distribution. For TPR and FPR, h is assumed to be a binary classifier.*

$$\begin{aligned}
\text{Accuracy}(h) &= \mathbb{P}(h(X) = Y) \\
\text{Brier}(h) &= \mathbb{E}[(h(X) - Y)^2] \\
\text{TPR}(h) &= \mathbb{P}(h(X) = 1 \mid Y = 1) \\
\text{FPR}(h) &= \mathbb{P}(h(X) = 1 \mid Y = 0)
\end{aligned}$$

Non-functional metrics Having satisfactory performance is a requirement for a predictor to be released. However, a broad range of other factors influence the useability of a predictor in practice. These factors are grouped under the umbrella of *non-functional* metrics. Common examples are robustness, bias or privacy; in Definition 1.4 I introduce three fairness metrics for classification. For a broader overview, the reader can refer to [25].

Definition 1.4 (Group fairness metrics) *Assume each input point $x \in \mathcal{X}$ can be associated to a group $g \in \mathcal{G}$. Let $h : \mathcal{X} \to \mathcal{Y}$ be a predictor, and (X, Y, G) random variables following the joint (input, target, group) distribution.*

- *Demographic Parity is a measure of* independence *[25] between group membership and prediction.*

$$\text{DParity}(h) = \mathbb{P}(h(X) = 1 \mid G = 1) - \mathbb{P}(h(X) = 1 \mid G = 0)$$

- *Equalized Odds is a measure of* separation *[25], i.e. independence of the prediction and the group conditioned on the target $t \in \mathcal{Y}$.*

$$\begin{aligned}
\text{EOdds}_t(h) = \mathbb{P}(h(X) = 1 \mid Y = t, G = 1) \\
- \mathbb{P}(h(X) = 1 \mid Y = t, G = 0)
\end{aligned}$$

- *Group Calibration is a measure of* sufficiency *[25], i.e. independence of the target and the group, conditioned on the prediction $t \in \mathcal{Y}$.*

$$\begin{aligned}
\text{GCal}_t(h) = \mathbb{P}(Y = 1 \mid h(X) = t, G = 1) \\
- \mathbb{P}(Y = 1 \mid h(X) = t, G = 0)
\end{aligned}$$

The audit dataset Definition 1.3 and Definition 1.4 are defined in terms of probabilities with respect to an input, target, and group *distribution* (that we will call $\mathcal{D}$). However, in practice, these probabilities are not directly accessible during the audit. Thus, to run the audit, an evaluation dataset D_{test} is sampled from $\mathcal{D}$.

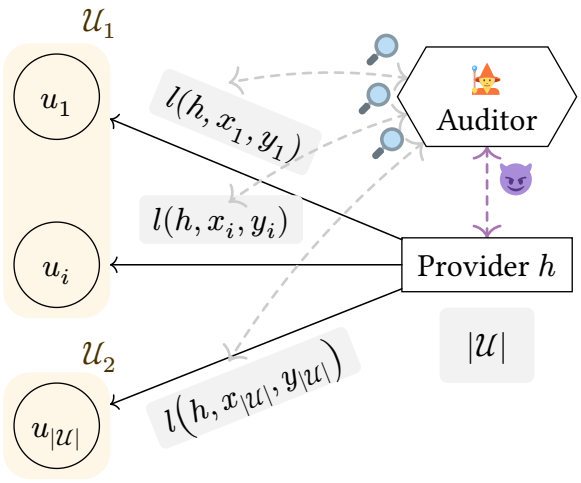


Figure 1.2: The three actors of an audit. The **user** gets utility from the ML system via the model predictions, measured by the loss l. The **model provider** gets utility proportional to its number of users. The **auditor**'s goal is to check the performance of the predictor h for the users and the performance disparities between groups (highlighted in orange). The easiest case would be the auditor being able to ask users for interaction data (⇢ arrows). However, for privacy and computational reasons, they instead scrape or query the predictor directly (⇢ arrows).

Then, the entity running the evaluation collects the predictions of h on D_{test} and aggregates them into a quantity that should be close to the *true* value of the metric. This poses an issue for external audits: how can we be sure that the collected data follows the distribution $\mathcal{D}$ assumed by the metric?

2.3. Who, and for whom? *"an independent assessment"*

The audit game gathers three players: the user, the model provider and the auditor. They all have different goals, different access and interactions with the predictor and different background knowledge about the task solved by the predictor. Before I describe the different access and interaction levels, let me introduce the three protagonists (pictured in Figure 1.2).

The user is any party (individual or organization) that derives utility from the predictor provided by the model provider. This includes active users, who initiate prediction requests (e.g., submitting queries to a model), and passive users, who experience the consequences of predictions without direct engagement (e.g., targets of algorithmic decision-making). In two-sided markets, such as digital marketplaces, the user role may extend to both buyers (benefiting from predictive recommendations) and sellers (gaining exposure through the predictor's output).

The model provider refers to the entity hosting the predictor. It could be the entity who built the predictor, collected the training data, but can also be just a

Data D	Predictions $h_\theta(x)$	Model h_θ
`D0`: None	`P0`: None	`M0`: None
`D1`: Aggregated statistics	`P1`: Raw predictions $h_\theta(x) \in \mathcal{Y}$	`M1`: model class $\mathcal{H}$
`D2`: Training set D_{train}	`P2`: Logits/probits $h_\theta(x) \in \mathbb{R}^{\lvert\mathcal{Y}\rvert}$	`M2`: weights θ
`D3`: User logs	`P3`: Explanations	`M3`: training objective l

Figure 1.3: Examples of different access levels to the data, predictions and model used in the creation of a ML system.

link in a larger ML supply chain. In this manuscript, we assume the model provider has full access to the predictor, and in particular to its code, weights and training data. The model provider sells predictions and is rewarded in proportion to the perceived accuracy of its predictions.

The auditor refers it the party that seeks to evaluate the predictor provided to the users. It includes journalists, auditing firms, regulators or even user collectives. Because of limited resources and knowledge, users cannot all run a full audit each time they suspect the predictor of wrongdoing. Therefore, the auditor serves as an agent representing the interests of the user in hope that the greater resources and expertise of the auditor will force the model provider to act more than in the case of a single user.

Access and interaction The studied predictors are often behind APIs or graphical interfaces, exposing very little of their inner workings. Because of the competition between actors and the perceived security risks the default is to disclose as little information as possible to the public and to the auditors. Yet, while users might only receive the raw predictions, auditors often have the possibility to request confidence estimates or prediction scores. To distinguish different types of access to the ML system, I list the different nuggets of information the model provider release in Figure 1.3.

In addition to *what* can be accessed, *how* it can be accessed determines which properties of the predictor can be verified and which cannot. For example, the types of analysis the auditor will conduct if they only has access to logged (query, prediction) pairs, will be different than if they could query the predictor with any input. Thus, *how* some information can be accessed defines the interaction model between the auditor and the model provider. Some examples of interaction models are listed in Figure 1.4.

Now, I review a few examples of common (access, interaction) combinations studied in the literature and this manuscript. The first one is the black-box audit.

Interaction	Description
`I1`: Sampling	The oracle chooses a random point in $x \in D$ and returns it or the prediction $h(x)$ to the caller
`I2`: Conditional sampling	The caller specifies a subset S of the dataset D, the oracle chooses a random point $x \in S$ and returns it or the prediction $h(x)$
`I3`: Query	The caller specifies a query x and the oracle returns the prediction $h(x)$
`I4`: Full	The caller has access to the entire dataset D and to the weights of the predictor h

Figure 1.4: Examples of different interaction models to access a predictor h or dataset D. The interaction is expressed between a caller and an oracle that mediate the access to the object.

Definition 1.5 (Black-box audit) *An audit is said to be a black-box audit if the auditor only has query access (Figure 1.4, `I3`) to the raw predictions (Figure 1.3, `P2`) of the studied predictor.*

To describe relaxations of the black-box audit model, the community has found a plethora of, sometimes creative, variations of the black-box. Some of them appear in [54]: black-box (Definition 1.5), grey-box (`I3` + `P3` + `M2`), white-box (`I4` for all objects in Figure 1.3), de-facto-white-box (same as white box but with copyright protection) or even outside-the-box access (same as grey-box + `D2`). To avoid talking about shades of grey and strange boxes, we will only use the term black-box as defined in Definition 1.5 and explicitly mention the additional information given to the auditor using terms defined in Figure 1.3 and 1.4.

The accessible parts of the ML system and the allowed interaction constrain the properties an external auditor will be able to audit. For example, privacy is a property obtained from the training procedure: it requires full access to the training pipeline (`D3`+`M4`) and hypothesis class (`M2`). On the other hand, measuring properties such as robustness, performance or fairness only require query access to the final predictor (`P2` or `P3`) but can be much more efficient if the auditor has more access to the model weights or training data.

The question "what minimal access" is required for ML auditing is crucial to inform public policy and governance schemes [55] to target what information to request from model providers. One contribution of this manuscript is to make some progress towards answering this question.

2.4. When? The four steps of Machine Learning audits

A ML audit is not just a matter of writing a Python script to evaluate the predictor behind and send the cavalry if the performance is too low. In fact, the computer scientist part might be the easiest one. Before even starting the audit, the auditor must know what to look for, think about how to measure it and the potential source of errors. As I will discuss in the next section, these choices are all potential targets that can be manipulated by the model provider to evade the audit. Thus, the goal of this section is All in all, ML audits can be divided in four phases: reconnaissance, systematization, measurement and monitoring.

During the **reconnaissance** phase, the auditor monitors, looks for and gathers initial evidence about the ML system. A first mechanism consists in collecting reports and complaints from users [56] or model providers themselves [57], then run continuous tests to detect if groups of the population experience issues more often than others. An alternative, implemented by the Organisation for Economic Co-operation and Development (OECD) [58] and the Responsible AI Collaborative [59] is to monitor news coverage on AI and classify the content using Natural Language Processing (NLP) techniques to detect incidents and hazards.

During the **systematization** phase, the auditor articulates a specific question and a corresponding specification (i.e. metrics and audit dataset) to test. To continue our example, the auditor might ask "is this specific report an outlier or is there a systemic issue?" by testing if the demographic parity of the predictor is below a given threshold. If the auditor has access to users, they can record the interactions of the users with the predictor as in [60]. When impossible, the auditor has to resort to variations of sock-puppet audits [61] or use public datasets to build an audit dataset. The design space at this stage is very large, the interested reader can refer to [62] for an overview of algorithm audits.

During the **measurement** phase, the auditor interacts with the model provider to gather the predictor outputs and system information required by the verification process. For example, the auditor might interact with the API to estimate the demographic parity and request additional information to the model provider to determine if bias mitigations measures are sufficient. This phase is critical to mitigate inconsistencies and blind spots in the data. To to so, it is important to gather as much information as possible, via different data sources [63], additional model artifacts [64] or through specific protocols that guarantee the integrity of the data and model [65].

During the **monitoring** phase, the auditor continues the interaction with the model provider and its users. It is important, both to ensure that the model provider took the necessary steps to address the shortcomings surfaced by the audit if any, and to check that the predictor or context does not drastically change, which would

require a new study. In the fairness example, the auditor would either pressure the platform into improving its system and continue measuring demographic parity to check that they do. While this has not yet been explored in the context of ML audits, tools range from model change detection (has the model changed?) [66, 67] to distribution shift detection (has the users' population or behavior changed?) [68, 69].

Each of these phases encompass their own challenges. Chapter 3 and Chapter 2 are concerned with the robustness of the measurement phase to deceptive model providers while Chapter 4 introduces a simple and efficient baseline for the monitoring phase.

2.5. On the user distribution

In this last subsection, I want to take a few lines to reflect on convenient assumption of an underlying user distribution and the issues it causes to ML auditing. The notion of "a probabilistic distribution determined arbitrarily by nature" [70] from which examples are drawn was introduced to the learning community by Valiant in 1984. Valiant's goal was to separate concepts that are computationally learnable using examples from concepts that would require too much computation to learn. The central notion of learnability is generalization: how does the predictor perform on examples not seen during training ?

Definition 1.6 (Generalization gap [3]) *Let $\mathcal{D}$ be a distribution on input, target pairs in $\mathcal{X} \times \mathcal{Y}$, and $l : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$ a loss function. The generalization gap of a predictor $h : \mathcal{X} \to \mathcal{Y}$ with respect to a dataset $D \subset \mathcal{X} \times \mathcal{Y}$ is defined as*

$$\Delta_{\text{gen}} = \mathbb{E}_{(x,y)\sim\mathcal{D}}[l(h, x, y)] - L(h, D)$$

When D is the training data and $\mathcal{D}$ is the distribution of new data, the generalization gap describes how representative the training error is of the real error on new data points. Assuming that both the training data D_{train} and D_{test} come from the same distribution $\mathcal{D}$ allows to leverage concentration bounds to upper-bound the generalization gap Δ_{gen} by a function of the error on the training data and thus estimate the performance of the predictor on new data *without looking at the performance on a test set.*

In the context of ML auditing, assuming that individual users can be represented as feature vectors drawn independently from a fixed joint probability distribution allows to reason about *audit validity*. Even a day, a month, or a year after the audit was conducted, as long as users can still be viewed as sampled from the same distribution, concentration bounds can be used to prove that the conclusions of the audit continue to hold.

Unfortunately, neither the auditor or the model provider have explicit description of such user distribution. Of course, there exists methods to test if a distribution has changed and if it has an impact on the predictor's performance [71] but in practice, these methods are only capable to detect change that were anticipated during their inception [68]. Does this mean that companies are bound to spend millions of euros every year on the services of audit firms even if they made no substantial changes? I do not have a definitive answer. In this manuscript, I assume that the users (or their behavior) does not significantly change and focus on the (potentially deceptive) changes of the predictor.

3. Audit manipulations

All the above discussion on how to audit ML systems and what access levels and interactions are required assumed a cooperative model provider. However, building powerful, fair, private and robust systems is hard. It involves complex tradeoffs that depend not only on technical constraints but also on the goals and policies of the model provider. Thus, when building such a system is too costly or when the model provider would rather not expose their true goals and policies, it might be easier to cheat.

Definition 1.7 (Audit manipulation) *Let $h : \mathcal{X} \to \mathcal{Y}$ be a deterministic predictor. Let h_{auditor} (resp. h_{user}) be the view of the predictor h by the auditor (resp. the user).*

- *If for all $x \in \mathcal{X}$, $h_{\text{auditor}}(x) = h_{\text{user}}(x) = h(x)$, the audit is not manipulated.*
- *Otherwise, if there exist $x_0 \in \mathcal{X}$ such that $h_{\text{auditor}}(x) \neq h_{\text{user}(x)}$, then the audit is manipulated.*

The interactions between the user, auditor and model provider are summarized in Figure 1.2. A model provider is successful at *evading* the audit if the manipulation it performed allowed them to pass the audit. That is, even though the predictor does not satisfy one of the requirements (i.e. $\mu_i(h, D_{\text{test}}) < M_i$), the audit test still accepts the predictor (i.e. $\mathcal{T}(h) = \texttt{Pass}$).

3.1. Manipulation incentives

There are a lot of different incentives for platforms to manipulate audits. A first incentive to manipulate audits is reputation. If the results of the audits are released to the public, displaying strong results can boost the reputation of the model provider and conversely, poor results can harm their revenue. Without asserting anything about the intention of the mentioned platforms, here are some examples of discrepancies observed between the information provided during an audit and separate, independent observations.

- One example is the non-disclosed specialization of predictors to a specific benchmark. This is what happened when Meta released the Llama 4 [72] models on LMArena [73] in 2025. The LMArena is a platform used to rank

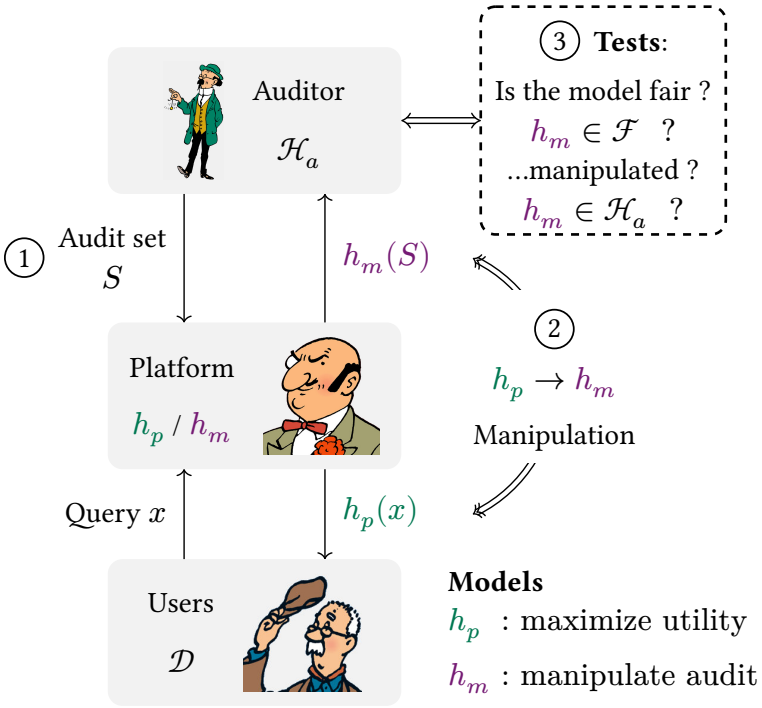


Figure 1.5: The manipulation game. The model provider exposes a model h_p to the users. To appear fair to the auditor while not deteriorating the utility for its users, the model provider manipulates its answers on the audit set S.

chat-bots from user preferences. Meta was caught [74, 75] (but never conceded) using a model optimized to be likable by humans on LMArena while releasing a more controlled version to perform better on general knowledge and reasoning benchmarks.

- Another example is the case of research APIs where non-trivial discrepancies between the auditor's API and the users' interface as been observed. In particular, this introduces significant risks for studies based on research APIs [51]. As an example, researchers reported some videos available on the TikTok website were not available in the research API, and the engagement metrics were highly underestimated by the research API [76].

A second incentive to manipulate audits appears when different criteria μ_i of the specification are (even partially) antagonistic. Beyond revealing business secrets, the choice of trade-off has to take roots in the values of the model provider and the goal they pursue. As an audit might reveal these trade-offs, model providers can be tempted to fake a given trade-off. An often discussed example is the incompatibility between fairness metrics and the associated trade-offs with accuracy. The notions of independence, separation and sufficiency (Definition 1.4) are mutually exclusive, i.e. no pair of conditions can hold at the same time [25]. Moreover, when the platform cannot collect more data, they have to resort to fairness repair

methods that often lead to tradeoffs with performance [77]. Thus, instead of carefully choosing a trade-off between the different fairness notions and the accuracy of their system, model providers can just optimize the model shown to the auditor to satisfy the audit criteria and serve the more accurate (albeit more unfair) model to the users.

Finally, a third incentive arises from the fines theoretically associated to the non-compliance with current regulations mentioned in Subsection 1.2.1.

3.2. Manipulation targets

There are a lot of opportunities in the specification/verification process for a deceptive model provider to manipulate the outcome of the audit, and black-box audits place the auditor in a delicate position. To be a meaningful governance mechanism, the specification and verification process of the audit must be as transparent as possible to avoid regulatory uncertainty [78]. This transparency can be exploited by a deceptive model provider. Per the definition of the specification and verification process, to evade the audit the model provider can manipulate four components: the specification data D_{test}, the metrics μ_i (and associated thresholds M_i), the predictor h itself and the test $\mathcal{T}$.

Manipulating data If the auditor relies on the model provider donating or giving access to internal data to define the test dataset D_{test}, the model provider can choose to hide some data in its favor. More specifically, consider a specification (D, μ, M) and a predictor M that does not meet the specification, i.e. $\mu(h, D) < M$. [79] and [80] proved that it is easy to craft a subset $\tilde{D}$ such that

- $\tilde{D}$ is indistinguishable from D with respect to a statistical test such as a Kolmogorov-Smirnov test (KS test).
- The predictor appears to satisfy the specification, i.e. $\mu\big(h, \tilde{D}\big) > M$.

Manipulating metrics If the model provider can influence the metrics that will be used to evaluate it's system, they can pick (intentionally or not) the metrics on which their system perform well [81]. This is what happened in the opposition between the investigation website ProPublica and Northpointe (now Equivant), the creator of the COMPAS recidivism prediction software. ProPublica accused COMPAS of discriminating against black people because it exhibited disparities in False-Positives between black people and the rest of the population. Northpointe refuted the accusation, claiming that their system was fair because it satisfied accuracy-parity and True-Positive parity [82].

Manipulating the predictor During the audit, the model provider can easily alter the answers to the auditor's queries or simply swap the model shown. Chapter 2 and Chapter 3 study the impact of such manipulations and how they can('t) be detected. Some works have also suggested that model providers could

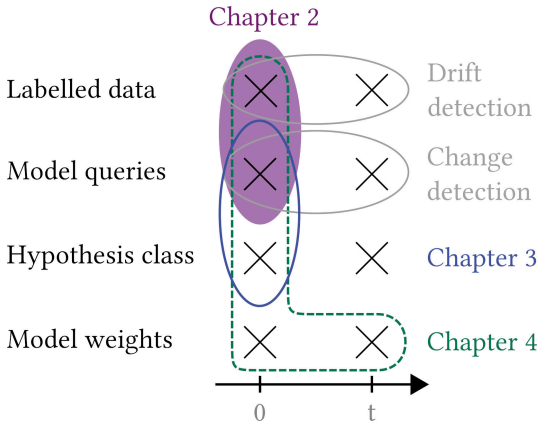


Figure 1.6: From one-time audits to monitoring, and the different kinds of information that the auditor has access to.

be asked to provide explanations alongside the predictions to improve the audit efficiency [64]. However, this additional information can also be manipulated by the model provider and used to escape the audit [83–85].

Manipulating the test In practice, because the model providers are not always forced to open their systems to auditors, then can negotiate the access to their internal systems to impose restrictions on the kind of tests that can be run on their infrastructure. An example would be to impose a query budget constraint to the auditor or force a given interface (such as a research API instead of scraping access). Fewer queries or less access to the predictor would mean a higher variance of the test, which could be exploited by the model provider to evade the audit.

4. Outline of the manuscript

All in all, designing a fair audit protocol is a balancing act. The specification has to meaningfully protect users. The verification should be lightweight for honest model providers and is still protect against deceptive ones. It requires the auditor to be knowledgeable on the prediction task solved by the model provider, to understand the socio-economic context it operates in, and pay attention to the design of the audit protocol to avoid blatant manipulations.

During the three years of my PhD, I made three main contributions towards answering the question this manuscript asks, with different assumptions about the knowledge and access of the auditor (see Figure 1.6). The first one I will present, although not the first in chronological order, deals with a proposition to model the problem and some ideas to solve it. In this chapter (Chapter 2), I introduce the framework of *Auditing with a prior*. To demonstrate the manipulation-proofness guarantees that can be obtained, I introduce a *prior* based on the labels the auditor has access to in their audit dataset D_{test}. The next chapter (Chapter 3) is an exploration of an alternative prior, based on the auditor knowing the hypothesis

class used by the model provider. It shows that while knowing the hypothesis class allows for a clever process to build the specification dataset D_{test}, the offered manipulation-proofness guarantees vanish as the model grows in complexity. Finally, in Chapter 4, I explore monitoring as a solution, complementary to the audit prior, to reduce the manipulation risks.

5. Contributions

The content of this manuscript is based on three works I have published in international conferences during my PhD. A preliminary version of [86] was published in a local French conference. Finally, I also actively contributed to open-source projects related to auditing, machine learning and scientific editing.

5.1. International conferences

- Augustin Godinot, Erwan Le Merrer, Gilles Trédan, Camilla Penzo, and Franois Taïani. "Under Manipulations, Are Some AI Models Harder to Audit?." In "2024 IEEE Conference on Secure and Trustworthy Machine Learning **(SaTML)**." Special issue, *2024 IEEE Conference on Secure and Trustworthy Machine Learning **(SaTML)***, April 2024, 644–64. https://doi.org/10.1109/SaTML59370.2024.00038
  - **Code**: https://github.com/grodino/tryphon
  - **Contributions**: I am the first author.
- Augustin Godinot, Gilles Tredan, Erwan Merrer, Camilla Penzo, and Francois Taiani. "Queries, Representation & Detection: The Next 100 Model Fingerprinting Schemes." Paper presented at The 39th Annual **AAAI** Conference on Artificial Intelligence. December 9, 2024. https://openreview.net/forum?id=rv0kUJses4
  - **Code**: https://github.com/grodino/QuRD
  - **Contributions**: I am the first author.
- Garcia Bourrée, Jade, Augustin Godinot, Sayan Biswas, et al. "Robust ML Auditing Using Prior Knowledge." In "Proceedings of the 42nd International Conference on Machine Learning." Special issue, *Proceedings of the 42nd International Conference on Machine Learning*, PMLR, October 6, 2025, 18794–810. https://proceedings.mlr.press/v267/garcia-bourree25a.html **(ICML25, Spotlight poster, top 2.6%)**
  - **Code**: https://github.com/grodino/merlin
  - **Contributions**: I am the co-first author. I proposed the project to the other collaborators, created the code-base and designed the mathematical framework. Jade and Sayan helped with the proofs and Milos and Martijn helped with the experiments.

5.2. Local conferences

- Augustin Godinot, Erwan Le Merrer, Gilles Trédan, Camilla Penzo, and François Taïani. "Change-Relaxed Active Fairness Auditing." July 6, 2023, 91. https://hal.science/hal-04395914
  - **Contributions**: I am the first author.

5.3. Open source software

- **Timm**: A python package to re-use pretrained models easily. https://github.com/huggingface/pytorch-image-models
- **max-stats**: A scraping service to collect and audit data from the French MAX-JEUNE railway pass. https://github.com/grodino/max-stats
- **Typst**: A document creation language. This manuscript is written in Typst. https://typst.app/home

Chapter 2

In search for a prior

> There is a crack, a crack in everything, that's how the light gets in.
>
> — Leonard Cohen, Anthem

This chapter presents a novel theoretical framework and a practical implementation for preventing manipulations by the model provider when the auditor only has query access to the predictor. Our analysis starts from a simple observation: auditors can readily collect labeled data reflecting the model provider's service from independent sources, a common practice whose theoretical and empirical implications remain unexplored. For example, when studying an online content moderation system, the auditor could have some evidence at hand, to confront the model under scrutiny, e.g., "A post with this content *must* pass the moderation filter, otherwise there is some bias on a protected feature of the user profile". Thus, by incorporating this dataset, the auditor can independently verify the model provider's responses, cross-referencing them against known ground truth labels. Our method enables more reliable detection of fairness violations while reducing the reliance on assumptions about the model provider's behavior. Specifically, we aim to answer the following research question:

> Can the auditor's prior knowledge of the ground truth prevent manipulations of fairness audits?

This chapter studies fairness audits of ML decision-making systems under manipulation by the model-hosting model provider. We consider binary classification problems, which is in line with related work in the domain of ML fairness analysis [1, 90]. We make the following three contributions:

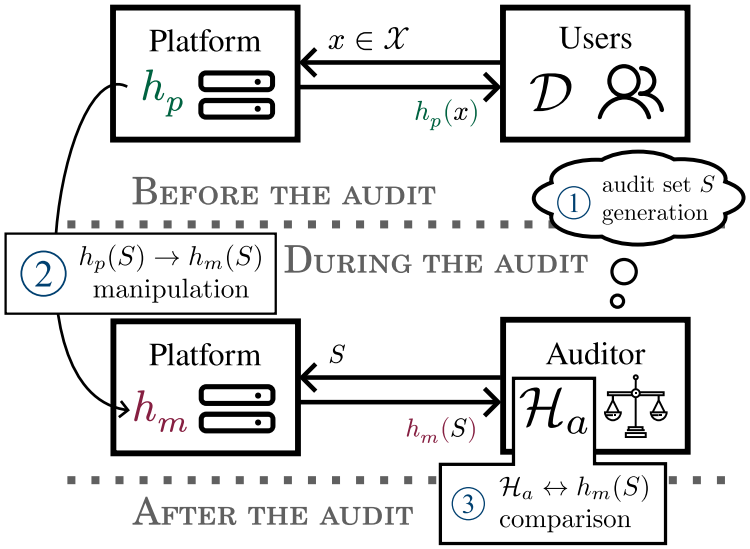


Figure 2.1: The manipulation game. The model provider exposes a model h_p to the users. To appear fair to the auditor while not deteriorating the utility for its users, the model provider manipulates its answers on the audit set S.

1. We introduce and analyze a new fairness auditing approach for black-box interactions where the auditor has access to prior knowledge about the model provider and the ML task (see Section 2.3).
2. We theoretically analyze how much unfairness a model provider can conceal given the auditor's prior knowledge. For any auditor priors, our results highlight the importance of keeping the auditor's prior knowledge private (see Section 2.3). For the dataset prior we introduce, we establish bounds on the concealable unfairness when the auditor prior remains confidential (see Section 2.4).
3. By simulating fairness audits on multiple tabular and vision datasets, we provide a more nuanced understanding of how our framework should be implemented. Our experiments offer insights into setting the detection threshold used to identify manipulations (see Section 2.5).

2. Related Work

This chapter sits at the crossroads between three facets of the literature on the evaluation of ML algorithms. First, the motivation to even consider the possibility of such manipulations is rooted in the notion of *fairwashing* and the early works of U. Aïvodji, H. Arai, O. Fortineau, S. Gambs, S. Hara, and A. Tapp on the manipulation of model explanations (Section 2.2). Second, this chapter addresses a fundamental question in the external model evaluation literature: even without manipulations, what model properties (e.g., accuracy, fairness, robustness) can be

externally verified (Section 2.2)? Finally, our method is a new leaf, on the young but growing tree of manipulation-proof external audit protocols (Section 2.2).

Fairwashing and the rationalization of ML ethics Addressing fairness issues often requires compromising model performance for advantaged groups which can discourage companies from embracing fair training practices [77, 91]. Companies have two incentives to pay attention to the impact of their system on society. The first incentive comes from regulatory efforts such as the Algorithmic Accountability Act (AAA) [92] (US) and the Digital Markets Act (DMA) [93] (EU) that impose fairness, transparency, and accountability constraints on large digital model providers. Yet, how to enforce these regulations is still an open problem [94]. The second incentive is public image. Since fairness, transparency and accountability are laudable goal, audits, investigative journalism and certifications [95] should force companies to pay attention to these objectives. However, both incentives are external: the model provider just has to *appear* fair, transparent and accountable. Therefore, a rational model provider can simply trade the fake appearance of fairness for a small risk to get caught.

Definition 2.1 (Fairwashing [4]) *fairwashing occurs when the actions and communication of the model provider promote the false perception that their machine learning model respects some ethical values, while failing to uphold those claims in practice.*

The work of U. Aïvodji, H. Arai, O. Fortineau, S. Gambs, S. Hara, and A. Tapp has extensively studied how model explanations can be used for fairwashing [4, 84, 96]. Independently, the problem of explanation manipulation was studied as a form of the *bouncer problem* in [85].

External evaluation of ML algorithms Fairness auditing evaluates ML models to ensure fairness and accountability, often without access to proprietary model internals [97]. This black-box auditing approach relies on querying the model and analyzing its outputs against pre-defined fairness metrics [2, 98]. Current attempts to enhance fairness audits with tangible guarantees draw inspiration from hypothesis testing [55, 99–103], online fairness auditing [104, 105], and formal methods for fairness certification [106–109]. Beyond statistical methods, the work of C. Yadav, M. Moshkovitz, and K. Chaudhuri explores the role of explanations in the auditing process [64]. Recent works also stress the importance of broadening the lens of algorithm auditing by incorporating user perspectives and sociotechnical factors [110, 111].

Attempts at robust audits Manipulating fairness audits is very simple. Auditors can be fooled by biased sampling when the decision maker is allowed to publish a labeled dataset as proof of model fairness [79]. Adversarial attacks on explanation methods, such as `LIME` and `SHAP`, can be employed to produce misleading interpre-

tations of model behavior [4, 83, 85, 96, 112–114]. model providers can also modify the output of their models to create the appearance of fairness without addressing underlying biases [1, 63, 90]. However, the challenge of designing audits that are robust to advanced manipulation strategies remains open. The idea of using auditor prior knowledge that we formalize in this work has been implicitly studied in different contexts. Based on active learning techniques work has studied how auditors could leverage knowledge about the hypothesis class [1, 90]. In a more practical setting, [115] studied using model distillation methods [115] to use prior about the ground truth and hypothesis class [115]. Finally, recent improvement in the useability of *Zero-Knowledge Proofs* have sparked interest in applications of cryptographic primitives to robust audits. Among them, Confidential-PROFITT and FairProof propose to integrate cryptographic techniques in cooperation with the model providers, to ensure the faithfulness of model provider responses during audits [116–118]. However, implementing cryptographic audit protocols is very intrusive for the model provider and technically restrictive, and thus awaits for adoption.

3. Enhancing Black-box Auditing with a Prior

Since a malicious model provider can manipulate Demographic Parity with relative ease, the auditor has to find ways to prevent these manipulations (e.g., using a different metric) or to detect them. In this section, we explore the latter. To detect manipulations, the auditor must use *prior* knowledge about what constitutes a "likely set of answers" on its audit dataset S. Then, using this prior, they would be able to estimate the likelihood that the received set of answers $h_m(S)$ has been manipulated.

3.1. Modeling the Auditor Prior

Previous work has demonstrated that prior knowledge is both a practical and an essential tool for auditing, yet the notion of an auditor prior has not been explicitly leveraged in the analysis of fairness audits. We define an *auditor prior* as follows.

Definition 2.2 (Auditor prior) *The* auditor prior *is a set of models $\mathcal{H}_a \subset \mathcal{Y}^{\mathcal{X}}$ that the auditor can reasonably expect to observe given her knowledge of the decision task by the model provider.*

For example, in [115], the authors study feature importance by training two models — one on a public dataset and another via distillation of the audited ML model — and comparing the resulting models. In this case, the prior is the set of models that are similar to the model trained on the public dataset. Using a more theoretical approach, [1] and [90] explored the case of an auditor knowing the hypothesis class of the model provider, i.e., $\mathcal{H}_a = \mathcal{H}$. [119] proposed to use an assumption about the Boolean Fourier coefficients of models in the hypothesis class $\mathcal{H}$ to

construct the prior $\mathcal{H}_a$. Finally, [63] and [96] used side-channel access (e.g., an additional API or explanations) to the ML model to define $\mathcal{H}_a$ and derive guarantees on the measured fairness. For example, in the setting of [63], the prior would be the set of models whose outputs agree with the previous predictions collected via scraping. In Section 2.4, we introduce a labeled dataset D_a that the auditor will leverage to define $\mathcal{H}_a$. Definition 2.2 captures all of the situations above and allows to formulate general results about the problem of robust auditing.

The auditing process The auditing process consists of three steps which we visualize in Figure 2.1. Here, h_p refers to the model that the model provider exposes to its users (the bottom part of Figure 2.1) and h_m refers to the model exposed to the auditor (top part of Figure 2.1). First, the auditor builds an audit set $S \subset \mathcal{X}$ and sends the queries in S to the model provider (step ①). The model provider receives S *all at once* and computes the answers using its model h_p. To appear fair if it is not, the model provider projects its labels $h_p(S)$ on the set $\mathcal{F}$ of fair models. This defines a manipulated model h_m and the answers $h_m(S)$ the model provider will send to the auditor (step ②, see Subsection 2.4.1). The auditor receives $h_m(S)$ and exploits these samples to evaluate whether the model provider is fair ($h_m \in \mathcal{F}$) and honest ($h_p = h_m$), (step ③, e.g. using Figure 2.2). Since the auditor does not have direct access to h_p, they compare h_m to their prior $\mathcal{H}_a$ to decide whether the model provider is honest or malicious. Thus, the auditor tests the two following properties of h_m:

$$\begin{aligned} \textbf{Is the model provider fair?} \quad & h_m \overset{?}{\in} \mathcal{F} \\ \textbf{Is the model provider honest?} \quad & h_m \overset{?}{\in} \mathcal{H}_a \end{aligned} \tag{2.1}$$

Note In this section, we expose general results in terms of models $h \in \mathcal{Y}^{\mathcal{X}}$. Later in Section 2.4, when considering dataset priors, the considered models will be restrictions on the audit dataset, i.e. labelings of the audit dataset. For dataset priors (i.e., when $\mathcal{H}_a$ is a ball, see Section 2.4), we draw $\mathcal{F}$ and $\mathcal{H}_a$ in Figure 2.3. Given a model h_m, the fairness audit is equivalent to checking if h_m belongs to the blue shaded area. In the example of Figure 2.3, the model provider would be flagged as malicious as h_m belongs to $\mathcal{F}$ but not to $\mathcal{H}_a$.

Online v.s. batch auditing Note that we assume that the model provider receives all audit queries at once and that it is possible to detect all the audit queries. In practice, the queries are usually issued online (that is, one-by-one) by the auditor, through web-scraping or through an API. Compared to online auditing, it is easier for the model provider to manipulate an audit if it knows all the audit queries before having to answer. On the other hand, because the auditor has to send all their queries at once, they cannot use the answers of the model provider to actively guide the generation of the audit questions (e.g., as in [1],

[90]). Ultimately, our setting is built as a worst-case analysis of the auditing game for the auditor.

Auditing axioms To avoid trivial audits, we add two modeling assumptions. The first assumption ensures that the auditors' prior is correct so that a honest model provider does not appear as lying. The second assumption asserts that an audit is necessary, otherwise the auditor could directly conclude from his prior that the model provider is unfair. That is to say, the auditor should never flag a honest model provider malicious. In particular, the auditor must have a prior that is close to the ground truth. Those assumptions are expressed as:

$$h_p \in \mathcal{H}_a \quad \text{and} \quad \mathcal{H}_a \cap \mathcal{F} \neq \emptyset \tag{2.2}$$

3.2. On Public Auditor Priors

A typical auditor proceeds in the following way. Upon examining a model provider's model h_m, the auditor must first understand the task addressed by h_m and what constitutes a "good-performing model" on this task. In our moderation example, the auditor might try to look for public moderation datasets to test the performance of h_m using a few examples. It might also look for publicly-available moderation models to compare their resulting input/output pairs with those of h_m. Unfortunately, our first result is that regardless of the prior the auditor might construct, if these models are public (or at least known by the model provider), the model provider will always be able to manipulate the audit:

Theorem 2.1 (Public prior) *Assume the model provider knows $\mathcal{H}_a$, it can then always pick $h_m \in \{\mathcal{H}_a \cap \mathcal{F}\}$ to appear both fair and honest.*

Proof (Theorem 2.1) First, recall that by definition the model provider knows $\mathcal{F}$. Assume that the dataset prior is public, the model provider also knows $\mathcal{H}_a$. Hence the model provider can compute $\mathcal{F} \cap \mathcal{H}_a$. As by assumption, $\mathcal{F} \cap \mathcal{H}_a \neq \emptyset$ (Equation (2.2)), the model provider can pick any model $h_m \in \mathcal{F} \cap \mathcal{H}_a$. ∎

In the case of [96], the model provider perfectly knows $\mathcal{H}_a$ (because the $\mathcal{H}_a$ is coming from queries of its model) so the detector is subject to this manipulation (called *Irreducibility* in the paper). In T. Yan and C. Zhang's work, $\mathcal{H}_a$ is the hypothesis class $\mathcal{H}$ of the model provider, communicated to the auditor before the audit. Theorem 2.1 provides a novel view on the impossibility results that we will cover in Chapter 3.

4. Using Labeled Datasets for More Robust Audits

In an ideal, yet unrealistic audit scenario, the auditor would have access to non-manipulated answers from the original model provider model h_p. The prior $\mathcal{H}_a$ would then be the set of models that agree with these non-manipulated answers

and would allow the auditor to detect inconsistencies between the original h_p and manipulated h_m models. Yet in general, the auditor does not have access to such non-manipulated answers.

As an alternative, we propose to study the use of a private (because of Theorem 2.1) dataset D_a, collected by the auditor to construct the auditor prior $\mathcal{H}_a$. This idea (coupled with an assumption on the hypothesis class) has been studied experimentally [115] but the more recent theoretical works on robust auditing diverged towards studying priors on the model itself rather than on the data [1, 96, 119]. In the following, we define what a dataset prior is, and study the guarantees an auditor can achieve using this prior.

Because the auditor does not have any assumption of the hypothesis class of h_m or h_p, and only observes h_m through its audit dataset D_a, **in this section all models are restrictions to the audit dataset**. Thus, h_p and h_m are labelings of D_a, and $\mathcal{F}$ is the space of fair labelings on D_a:

$$\begin{gathered} h_p : D_a \to \mathcal{Y} \quad \text{and} \quad h_m : D_a \to \mathcal{Y} \\ \mathcal{F} = \left\{ h \mid h \in \mathcal{Y}^{|D_a|} \text{ and } \mu(h, D_a) = 0 \right\}. \end{gathered} \tag{2.3}$$

Definition 2.3 (Dataset prior) *Let $D_a = \left(D_a^X, D_a^Y, D_a^G\right) \subset \mathcal{X} \times \mathcal{Y} \times \mathcal{G}$ be a labeled dataset the auditor has access to. Let $n = |D_a|$ be the cardinality of D_a and $\tau > 0$ a threshold set by the auditor. The dataset prior $\mathcal{H}_a$ is defined as the set of models restricted to D_a^X that have a reasonable risk L (see Definition 1.1) on D_a:*

$$\mathcal{H}_a = \left\{ h \in \mathcal{Y}^{|D_a|} \mid L(h, D_a) < \tau \right\}$$

Unless noted otherwise, in this section and in Section 2.5, $\mathcal{H}_a$ will denote the dataset prior. To test if the model provider is honest, the auditor needs to verify whether $h_m \in \mathcal{H}_a$, i.e., whether $L(h_m, D_a) < \tau$. The risk threshold τ thus plays a crucial role in the guarantees the auditor will be able to achieve. The final verification test we propose as an audit with a prior is presented in Figure 2.2. We discuss the impact of τ in Subsection 2.4.2 and guidelines to set its value in Subsection 2.4.3, but first, we need to discuss the definition of optimal manipulation in Subsection 2.4.1.

4.1. Optimal Manipulation

Given the audit set S and its model h_p, the objective of a manipulative model provider is to create a set of answers $h_m \in \mathcal{Y}^{|D_a|}$ that appear fair to the auditor but also do not raise suspicions. Ideally, the model provider would like to know the auditor prior $\mathcal{H}_a$ (see Theorem 2.1), but in the general case it cannot because it is not public information. As a consequence, the model provider cannot directly optimize its answers to be expectable *and* fair. However, the model provider still

Input audit dataset D_a, query access to predictor h,
threshold τ, tolerance ε

Define $D_a^t = \{(x, y) : (x, y, g) \in D_a \text{ and } g = t\}$
Collect $\hat{h} = \{h(x) : (x, y, g) \in D_a\}$
$\hat{L} \leftarrow \frac{1}{|D_a|} \sum_{(x,y)\in D_a} l(h, x, y)$
$\hat{\mu} \leftarrow \frac{1}{|D_a^1|} \sum_{(x,y)\in D_a^1} \mathbb{1}\{h(x) = 1\} - \frac{1}{|D_a^0|} \sum_{(x,y)\in D_a^0} \mathbb{1}\{h(x) = 1\}$
If $\hat{L} < \tau$ **and** $\hat{\mu} < \varepsilon$ **Return** `Pass`
Else Return `Fail`

Figure 2.2: Audit with a prior for demographic parity.

has cards up its sleeve; it already trained a model h_p on a dataset D that is close to that of the auditor D_a.

Thus, instead of searching h_m in $\mathcal{H}_a \cap \mathcal{F}$, the model provider can assume that its true model h_p is expectable – that is, $h_p \in \mathcal{H}_a$ – and try to find a fair model $h_m \in \mathcal{F}$ while flipping as few labels as possible from h_p. Therefore, the optimal manipulation is the projection of h_p on $\mathcal{F}$:

$$h_m = \text{proj}_{\mathcal{F}}(h_p) = \arg\min_{h\in\mathcal{F}} d(h, h_p). \tag{2.4}$$

The distance d in equation Equation (2.4) is the empirical risk L of h using the labels of h_p as the ground truth.

$$d(h, h_p) = \frac{1}{|D_a|} \sum_{(x,y)\in D_a} l(h, x, h_p(y)) \tag{2.5}$$

4.2. Achievable Guarantees

By construction, when there are no manipulations (i.e. $h_p = h_m$), if the predictor is fair (i.e. $\mu(h, D_{\text{test}}) < \varepsilon$), then $\mathbb{P}(\mathcal{T}_{\text{prior}}(h_m) = \texttt{Pass}) = 1$ and conversely if the predictor is unfair ($\mu(h, D_{\text{test}}) \geq \varepsilon$) , then $\mathbb{P}(\mathcal{T}_{\text{prior}}(h_m) = \texttt{Fail}) = 1$. Now, assume that the audited predictor is manipulated, i.e. $h_p \neq h_m$. We argue that the model provider has no incentive to manipulate the audit if their original predictor was fair. Thus, if $h_p \in \mathcal{F}$, then $h_p = h_m$ and $h_p \in \mathcal{H}_a \Rightarrow h_m \in \mathcal{H}_a$. This means that our test $\mathcal{T}_{\text{prior}}$ test has no false positives, and the main quantity of interest to the auditor is the *manipulation detection rate.*

Definition 2.4 (Manipulation detection rate) *The probability P_d, over the randomness of the provider's manipulation strategy, that the auditor correctly detects a manipulative model provider with optimal manipulation is*

$$P_d = P(h_m \notin \mathcal{H}_a \mid h_p \in \mathcal{H}_a).$$

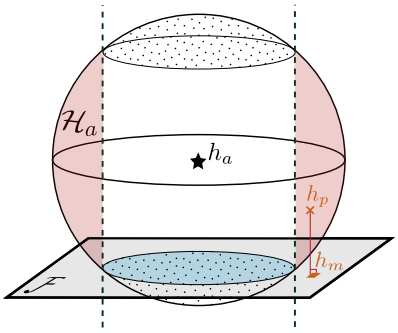


Figure 2.3: Representation of the auditor prior $\mathcal{H}_a$, the honest model provider model h_p and a corresponding malicious model h_m on the fair $\mathcal{F}$ plane. The red area represents the area where model providers optimal manipulations are detected as dishonest: they fall outside of the blue region of $\mathcal{F}$

Estimating or computing P_d requires the knowledge of the distribution of labelings in $\mathcal{H}_a$. Unfortunately, unless they have access to the training pipeline of the model provider, this model distribution is inaccessible to the auditor. To overcome this issue, we make the assumption of an *uninformative prior*: since the auditor does not know the model distribution in $\mathcal{H}_a$, they must assume it is uniform.

Theorem 2.2 (Prior-Uniform detection rate) *Under the dataset prior of definition Definition 2.3 with $l(h, x, y) = \ell_2(h(x) - y)$ the ℓ_2 norm, and the* uninformative prior *assumption, the probability that the auditor correctly detects a malicious model provider trying to be fair is*

$$P_d = 1 - \frac{1}{W_n}\left(\int_0^{\arccos\left(\frac{\delta}{\tau}\right)} \sin^n(\theta)d\theta - \frac{\delta}{\tau}\left(1 - \frac{\delta^2}{\tau^2}\right)^{\frac{n-1}{2}}\right).$$

with D_a^Y the labels of the samples in D_a, $\delta = d(D_a^Y, \mathcal{F})$, the distance of D_a^Y to $\mathcal{F}$ and W_n the n-term of Wallis' integrals.

To gain intuition about the proof, we represent the audit case for $|S| = 3$ in Figure 2.3. By definition of the dataset prior, $\mathcal{H}_a$ is a ball of radius τ, centered on D_a^Y, the labels given in the audit dataset D_a. The manipulation of a model h_p can be detected only if the resulting model is outside of $\mathcal{H}_a$, as shown in orange on Figure 2.3. The probability of detection is thus 1 minus the volume of original models h_p whose projection on $\mathcal{F}$ lies outside of $\mathcal{H}_a$. This volume is highlighted in red in Figure 2.3. The detailed proof of Theorem 2.2 is deferred to Appendix A.

Theorem 2.2 highlights two key parameters to the auditor's success: the unfairness of the prior $\delta = d(D_a, \mathcal{F})$ and the expectability threshold τ. There are three cases of interest: if the audit dataset D_a is perfectly fair $\delta = 0$, if the distance of the

audit dataset to the set of fair predictions is equal to the threshold $\delta = \tau$ and what happens in between.

As exposed in Corollary 2.1, if the dataset prior is perfectly fair (i.e., $\delta = 0$), then the auditor has no chance to detect a manipulated model as non-expectable, i.e. $P_d = 0$.

Corollary 2.1 (Balanced dataset D_a) *If $\mathcal{H}_a$ is a ball centered in the ground-truth D_a^Y that is fair, then the auditor has a probability zero to correctly detect a malicious model provider trying to be fair. $\mathcal{H}_a = B(D_a^:\tau) \wedge D_a^\in \partial\mathcal{F}$ implies $P_d = 0$.*

Proof (Corollary 2.1) If $D_a^\in \partial\mathcal{F}$ then $\delta = 0$ and $\arccos\left(\frac{\delta}{\tau}\right) = \arccos(0) = \frac{\pi}{2}$ in the formula of Theorem 2.2. Thus, $P_d = 1 - \frac{1}{W_n}(W_n - 0) = 0$. ∎

On the other hand, if $\tau = \delta$[2], that is the $\mathcal{H}_a$ ball is tangent to the set of fair predictions $\mathcal{F}$ then $P_d = 1$ and any attempt at gaming the audit will be detected.

Corollary 2.2 (Tangent prior) *If the prior $\mathcal{H}_a$ is tangent to the hyperplane of fair predictions $\mathcal{F}$, then the auditor has a probability one to correctly detect a malicious model provider trying to be fair. That is,*

$$\tau = \delta \Rightarrow P_d = 1.$$

Proof (Corollary 2.2) If $\mathcal{H}_a$ is tangent to $\mathcal{F}$ then $\delta = \tau$. Thus, $\arccos\left(\frac{\delta}{\tau}\right) = \arccos(1) = 0$ and $\int_0^{\arccos\left(\frac{\delta}{\tau}\right)} \sin^{n(\theta)} d\theta = 0$.

Following Theorem 2.2 with $\frac{\delta}{\tau} = 1$, $P_d = 1 - \frac{1}{W_n}(0 - 0) = 1$. ∎

Finally, in the case $0 < \delta < \tau$, we derive a lower bound on the probability P_d to detect manipulations in Corollary 2.3. The ratio $\frac{\delta}{\tau}$ that maximizes the detection rate lower bound decreases (and tends to 0) as the budget increases. Thus, while the audit dataset D_a should not be perfectly balanced (i.e. $\delta = 0$), as the audit budget increases it should be close to perfect parity ($\delta \gtrsim 0$).

Corollary 2.3 (Detection rate lower bound) *If n is even, the probability of detecting manipulations is lower bounded by*

$$\frac{1}{W_n}\frac{\delta}{\tau}\left(1 - \frac{\delta^2}{\tau^2}\right)^{\frac{n-1}{2}} \leq P_d \leq 1.$$

Moreover, the lower bound is maximized when $\frac{\delta}{\tau} = \frac{\sqrt{n+3}-\sqrt{n-1}}{2}$.

The proof of Corollary 2.3 is deferred to Appendix B.

2. Per our first axiom in Equation (2.2), we have that $\delta \leq \tau$.

4.3. Practical Considerations and Discussion

In practice, τ is determined by the task difficulty, and the amount of data available to solve the task. One possibility to tune the value of τ is to use the error rate of current state-of-the-art models that solve the task at hand as a minimum value. We empirically explore this option in Subsection 2.5.4. If the auditor has the resources, an alternative would be to train a set of models on the task and use them to calibrate τ. We leave further exploration of the calibration of τ to future work.

On the other hand, the value of δ is determined by the audit set sampling procedure. In most cases, the audit set is sampled i.i.d. from a pre-specified audit distribution. In this case, the value of δ is fully determined by the resulting sample S. To regain some control over δ, the auditor has to allow other audit set sampling strategies, at the expense of potential statistical bias in the fairness and accuracy estimations.

5. Empirical Evaluation

We now empirically quantify the extent to which the model provider can manipulate the unfairness of its ML model. To that end, we study the *concealable unfairness*: the maximum level of unfairness a model provider can hope to hide before being detected as malicious. First, we evaluate the effectiveness of different manipulation strategies and determine the optimal one. Since any practical fairness repair method can be used as a manipulation methods, we explore in Subsection 2.5.3 **(RQ1)** What is the best manipulation strategy implementation? Then, we study in Subsection 2.5.4 the dynamics of the concealable unfairness when the audit budget $|S|$ increases: **(RQ2)** Can the auditor always find an audit budget that prevents the model provider from hiding any unfairness, i.e., that always allows to flag the model provider if malicious?

5.1. Experimental Setup

We conduct our experiments on tabular and vision modalities. The tabular dataset comes from the ACSEmployment task for the state of Minnesota in 2018, which is derived from US Census data and provided in folktables [120]. The objective of this task is to predict whether an individual between the age of 16 and 90 is employed or not. As input features of the model h_p, we consider several attributes of the individual, including gender, race, and age. The fairness of the models is evaluated along the race attribute given in the dataset: one group consists of individuals identified as "white alone", while the other includes all remaining individuals.

For the vision modality, we study CelebA [121], which consists of images of celebrities along with several binary attributes associated with each image, such as whether the person in the photo is blond, smiling, or if the photo is blurry. As input to a vision model, we use the image to predict one of the associated

attributes. The target attribute varies across experiments and will be specified accordingly. Demographic Parity is evaluated along the gender attribute given in the dataset. For the ACSEmployment dataset, we train Gradient Boosted Decision Tree (GBDT) and Logistic Regression (Log. Reg.) models, while for CelebA, we train a LeNet convolutional neural network [122]. GBDT and Log. Reg. are trained using the default parameters of their respective implementations in `Scikit-Learn`. Meanwhile, LeNet is trained irrespective of the target attribute using the Adam optimizer with a learning rate of $\gamma = 0.001$, a batch size of 32, and for two epochs, which is sufficient for the model to converge on all features. The code to run the experiments is available online.[3]

5.2. Implementing Optimal Audit Manipulations

In practice, computing the optimal manipulation $h_m = \text{proj}_{\mathcal{F}}(h_p)$ amounts to solving

$$\begin{aligned} h_m(S) \in \arg\min & \; L\left(h, D_{h_p}\right) \\ \text{s.t. } & \hat{\mu}(h, S) < \tau \end{aligned} \tag{2.6}$$

with $D_{h_p} = \left\{\left(x, h_p(x)\right) : x \in S\right\}$. We note that this problem is the same problem solved by in-processing and post-processing fairness repair methods [123]. Thus, ironically, computing the optimal manipulation is equivalent to choosing the optimal fairness repair method. The only difference being on which set the fairness constraints and accuracy objectives are defined: the audit set S instead of the training dataset. Thus, since any practical fairness repair method can be repurposed for manipulation, we adapted four classical fairness repair methods: ROC Mitigation (ROC) [124], Optimal Label Transport (OT-L) [125], Linear Relaxation (LinR) [126] and Threshold Manipulation (ThreshOpt) [127].

5.3. Quantifying the Concealable Unfairness For Different Detection Scores

We next quantify the amount of fairness that a model provider can hide for the different manipulation strategies above. This *concealable unfairness* $\Delta_\mu(h_p, h_m)$ is defined as the DP gap between the manipulated and honest models.

$$\Delta_\mu(h_p, h_m) = \left|\hat{\mu}(h_m, S) - \hat{\mu}(h_p, S)\right| \tag{2.7}$$

To decide whether the model observed during the audit is manipulated, the auditor has to decide whether $h_m \in \mathcal{H}_a$ or not. To do so, the auditor estimates $L(h_m, D_a)$ by computing the *detection score* $\text{Detect}(h_m, S)$.

3. See https://github.com/grodino/merlin.

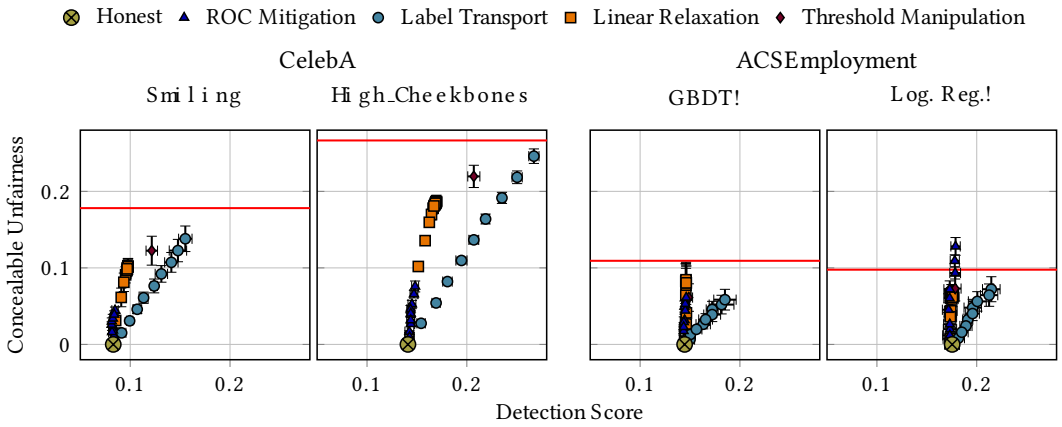


Figure 2.4: The concealable unfairness by the model provider for different detection scores and manipulation strategies. We highlight this for two features of the CelebA dataset (left) and for two different ML models trained on the ACSEmployment dataset (right). The horizontal red line indicates the Demographic Parity of the most unfair model without manipulation.

$$\text{Detect}(h_m, S) = \sum_{(x,y)\in S} \mathbb{1}(h_m(x) \neq y) \tag{2.8}$$

To build (h_p, h_m) model pairs, we consider manipulation methods among ROC [124], OT-L [125], LinR [126] and ThreshOpt [127], varying hyperparameter values when applicable. In Figure 2.4, we plot the value of the concealable unfairness $\Delta_\mu(h_p, h_m)$ against the detection score $\text{Detect}(h_m, S)$ computed by the auditor. We show the results of LeNet models trained on two CelebA targets (first and second subplots), and GBDT and Log. Reg. models trained on ACSEmployment (third and fourth subplots). The horizontal red lines indicates the DP of the most unfair model without manipulation.

First, we observe that for all the datasets, the model provider can conceal significant amounts of unfairness: from 10 to 20 points differences between the two protected groups. Comparing the concealable unfairness values with the DP of the most unfair honest model (red horizontal line), we observe that the manipulation strategies almost all able to totally conceal the original model unfairness. Then, focusing on the x axis, the difference in $\text{Detect}(h_m, S)$ between the different honest models highlights the impact the performance of the model provider's model should have on the detection threshold τ. In fact, depending on the dataset and on the model, $\text{Detect}(h_p, S)$ varies from ~ 0.1 to ~ 0.2. In Subsection 2.5.4, we explore a solution to setup the threshold.

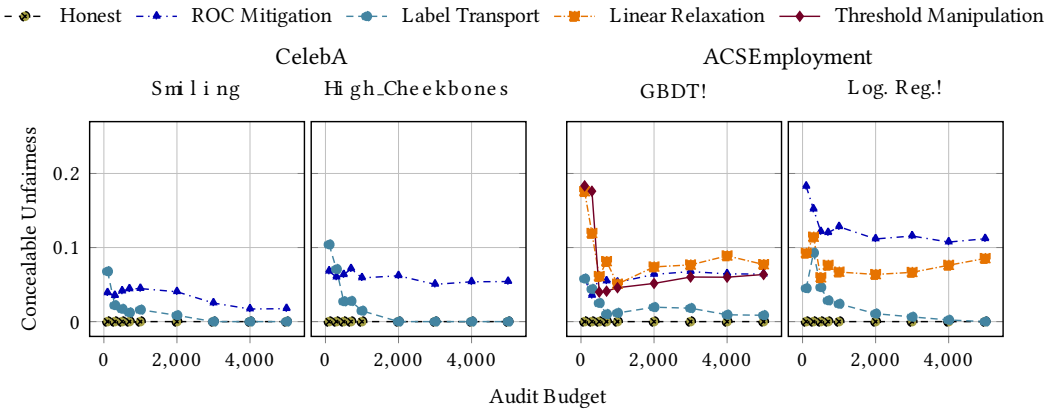


Figure 2.5: The concealable unfairness for different audit budgets (i.e., data samples from the labeled dataset). We highlight this for two features of the CelebA dataset (left) and for two different ML models trained on the ACSEmployment dataset (right).

5.4. Dynamics of the Concealable Unfairness as The Audit Budget Increases

The probability of detecting manipulations (via the the detection score) should intuitively increase as the auditor gains access to a larger number of data samples (i.e., has a higher audit budget) since this allows for a more accurate comparison of h_m with the data prior $\mathcal{H}_a$. In this experiment, we explore how well this intuition holds in practice. For this purpose, we fix the hyperparameters for each manipulation method by selecting those that result in the highest concealable unfairness for a given base model, as discussed in Subsection 2.5.3. Then, for each base model–target attribute pair, we determine the maximum concealable unfairness that a model provider can achieve while ensuring that its detection score (see eq. Equation (2.8)) remains below the detection threshold. As proposed in Section 2.4 the threshold for each model is set to $1 - x$, where x represents the maximum accuracy achieved when training a set of models on the corresponding target. This process is repeated for audit budgets ranging from 100 to $5,000$.

The results of this experiment are shown in Figure 2.5. The two plots on the left display the results for CelebA using the same base model but different target attributes, while the two plots on the right show results for ACSEmployment using the same target attribute but different base models. These results reveal two distinct cases. In the first case (CelebA `Smiling` in Figure 2.5), the concealable unfairness converges to zero as the audit budget increases. In the second case (all the other facets of Figure 2.5), the concealable unfairness remains nonzero despite an increasing budget. We hypothesize that this separation arises from the low aleatoric uncertainty associated to the `Smiling` target compared to that of `High Cheekbones` and ACSEmployment. Detecting a smile on a picture is a well defined

task. On the other hand, determining if a person has "high cheekbones" is less well defined (the notion of "high" can vary from annotator to annotator), leading to more label noise. Similarly, guessing the income of a person simply from a few demographic attributes is also more subject to label noise as similar people (even with similar job positions) can have very different salaries. All in all, because the accuracy range of models trained on `Smiling` is narrower, the detection threshold τ can be tighter, which makes the manipulations harder to implement.

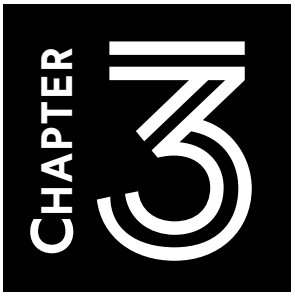
As a consequence, while our dataset prior prevents manipulations for tasks with low aleatoric uncertainty, model providers can still game audits if they can hide in the label noise. This answers **(RQ2)**. In response to **(RQ1)**, we observe from Figure 2.4 and 2.5 that the Linear Relaxation and ROC Mitigation manipulation strategies are the most effective for a manipulative model provider.

6. Conclusion and Discussion

We investigated, both theoretically and experimentally, the conditions under which an audit can or cannot be manipulated when auditing with a prior. We introduced an empirical method for tuning the manipulation detection threshold to maximize the auditor's probability of detecting malicious model providers.

While our work offers regulators a framework for defending against audit manipulations, the path to accountability extends much further. A significant gap remains between audit evaluations and the actual mitigation of identified issues [128], [129]. Moreover, one-time audits are inherently limited, as model providers can alter their models in harmful ways after the audit has concluded. Addressing these challenges in future work will require the development of continuous or adaptive auditing mechanisms, potentially incorporating auditor priors, to ensure sustained accountability and fairness.

Leveraging knowledge on the hypothesis class



> The laws of mathematics are very commendable, but the only law that applies in Australia is the law of Australia.
>
> — Malcom Turnbull, Prime Minister

The previous chapter introduced the audit-with-a-prior framework and studied the case when the prior is defined by a labeled dataset (Definition 2.3). Among the previous attempts at formalizing robust auditing [1, 64, 104], Yan et al. [1] have shown that the knowledge of the hypothesis class used by the model provider can *potentially* reduce the required number of audit queries to reach a given robustness level. Their method is based on disagreement-based active learning [130] which requires training surrogates of the model provider's model. However, they only demonstrated their proposed audit algorithm with linear models on small datasets (StudentPerf [131] and COMPAS [132]). Furthermore, they prove that quantifying the potential improvement (in terms of query complexity) of their algorithm over a simple random baseline is computationally intractable. In this chapter, we investigate whether the model provider can engineer models that simultaneously achieve a high utility and evade the audit and ask

> Can the auditor's prior knowledge of the hypothesis class prevent manipulations of fairness audits?

To that end, we compare the manipulation-proofness guarantees of a simple uniform random audit algorithm (Algorithm 1) against the best guarantees a regulator could hope for. Our contributions are three-fold.

1. We first consider those hypothesis classes that can perfectly reproduce any labeling of the dataset. This covers two practical cases: either the model provider has a model with a very high capacity, or the auditor's prior on the

model provider's model is uninformative. We prove in Theorem 3.1 (Subsection 3.4.1) that no audit method —whether active or passive—can deliver a better performance than random sampling. We also prove in Corollary 3.1 that this impossibility holds even if the hypothesis class can only imperfectly reproduce any labeling of the dataset with a bounded error rate.

2. To uncover what properties of the hypothesis class influence its auditability, in Subsection Subsection 3.4.2 we analyze the simple class of dictionary models, whose manipulation guarantees can be analytically derived. We identify regimes in which the hypothesis class cannot be audited more efficiently than by random sampling.

3. To build a practical understanding of our theoretical results, we formally define the notion of *manipulability under random audits* and *capacity* in Subsection 3.5.1. We then evaluate the manipulability under random audits of classical ML models for tabular data. We empirically confirm the strong connection between the classical *Rademacher complexity* and the manipulability of manipulation-proof auditing. Since modern ML hypothesis classes tend to exhibit larger and larger capacities, we argue that our work brings up the limits of the current formulation of manipulation-proof auditing.

2. Related work

The problem of manipulation-proof auditing and more generally black-box, remote, and robust property verification of ML platforms arises from the need to enforce regulations. As an example, consider the European Union. Classical fairness regulation of online ML models mainly comes from the *Racial Equality Directive* [133], the *Framework Equality Directive* [134] and the *Gender Equality Directives* [135, 136]. Recently, the EU set out to create regulations specific to online platforms. These are the *AI Act* [137], the *Digital Services Act* [51] and the *Digital Markets Act* [52]. These directives provide a legal framework that prescribes what online platforms may and may not do, but offer little to verify that these rules are respected in practice. The manipulation-proof framework is a first attempt to provide operational solutions that can detect when platforms do not follow the law.

In addition, our results are mostly related to the following lines of work.

Algorithm auditing The field of algorithm auditing is interested in understanding the impact of algorithms on the lives or the people impacted by those algorithms' decisions. In practice, auditing algorithms *in vivo* (that is as they are deployed in online services) is challenging because they constantly evolve, mostly without records [138]. For a survey on examples of published academic audits of decision systems, refer to [139]. Moreover, because it is impossible for researchers

or regulators to audit each automated decision system, it has been observed that most of the recent discoveries of problematic algorithm behavior have surfaced thanks to users of those systems [111, 140]. Again, after a problematic algorithm behavior has been detected and after a court decision has been made, we still need to be able to monitor that this decision is respected.

Audit metrics and audit design With the advent of broadly publicized algorithm audits such as COMPAS [132] or Reuters' study on Amazon's recruiting tool [141], there has been an effort to devise metrics and their interpretations to better understand the impact of algorithms on their users. Most of the effort has been directed towards the operationalization of fairness values into the ML framework [25]. Classical fairness measures include Demographic Parity [142], Equalized Odds [127], Equal Opportunity [127] or Predictive Parity [143]. All of these measures encompass different visions of fairness and choosing one versus the other has political implications on the considered notion of fairness [144, 145]. While still marginal, some works are interested in other aspects of the audit of AI algorithms. For example, [146] is interested in the verification that online platforms comply with the Data Minimization Principle. Another interesting work [147] considers the problem of automatically auditing the privacy guarantees offered by AI algorithms. However, most of the presented works do not yet consider the possibility of the model provider gaming their audit.

Robust verification The literature on robust auditing is still in its infancy. The manipulation-proof [1] framework has only recently been introduced. However, with its goal of efficiently choosing the next audit query based on previous queries and the associated outputs of the API, the manipulation-proof framework exhibits clear links with the active learning literature [148, 149]. With the aim of finding methods to ease the audit, [64] showed that the explanation provided by the model provider can greatly improve the robustness of audits. For example, they show that for linear classifiers, a single result along its counterfactual explanation allows to totally characterize the model. Our work does not assume that the auditor has access to explanations. It is likely that faithful explanation could lead to audit algorithms with increased MP guarantees. On another line of works, [117] and [150] suggest instantiating an audit protocol in which both the model provider and the auditor would be active, drawing inspiration from zero-knowledge proofs and interactive verification protocols.

Benign overfitting and model capacity As we proved in this work, manipulability under random audits has deep connections with model capacity and their ability to perfectly fit arbitrary datasets. Classical metrics that capture the notion of model capacity include the VC-dimension [151] or the Rademacher Complexity (which we used for its usability in practice) [152]. Moreover, our experiments on the link between auditability and model capacity have been motivated by the

recent finding that larger models can fit the training dataset perfectly while still showing good generalization properties [153]. This effect has been observed for linear models [154], Support Vector Machines [155] and Decision Trees [156]. In the manipulation-proof audit setting, we show that this type of behavior is very problematic. In fact, if a model is able to fit any audit set and yet keep its generalization performance, platforms do not even have to lie to the auditor. They just have to train their model to give the answers the auditor expects on their audit set. Then, the model provider can define any objective for the rest of the input space, even if it does not align with the auditor's metric.

Interestingly, the connection between model capacity and audit query complexity is not limited to manipulation-proof estimation of parity measures. In their work on certified feature sensitivity auditing [64], Yadav et al. provide an algorithm to audit feature sensitivity for decision trees whose query complexity grows linearly with the capacity (number of nodes) of the tree.

3. Auditing and manipulation-proof estimation

During a typical audit, the auditor defines a measure of interest μ with an associated threshold τ_μ. Classical measures used by auditors are statistical parity indicators [25] focusing on independence (e.g. demographic parity, group fairness), separation (e.g. balance for positive/negative class, equalized odds) and sufficiency (e.g. calibration, predictive parity). Given that demographic parity does not require any ground truth labels and since it is often used as the archetypal example in the literature, we use it as the measure μ throughout this paper. While the results we present refer specifically to demographic parity, it is straightforward to extend them to any parity measure of the form

$$\begin{aligned}\mu(h, S) = \mathbb{P}(h(X) = 1 \mid X \in S, E) \\ -\mathbb{P}\big(h(X) = 1 \mid X \in S, \overline{E}\big)\end{aligned} \tag{3.1}$$

with E an event defined with respect to the random variables X, X_A and Y, where X represents the input, Y the ground truth label, and $X_A \in \{0, 1\}$ is the sensitive attribute of interest for the auditor. For example, for demographic parity, $E = (X_A = 1)$. We would like to stress that for other less common measures that can nonetheless present an interest for auditors (e.g. level of privacy [147] or the degree of compliance with data minimization [146]), MP remains an open problem.

3.1. Threat model

We describe the interaction between the auditor and the model provider in the threat model diagram (Figure 3.1). Before the audit, the model provider discloses the hypothesis space $\mathcal{H}$ they use (decision trees for example) to the auditor. Then, during the auditing phase, the auditor interacts with the (unknown) model $h \in$

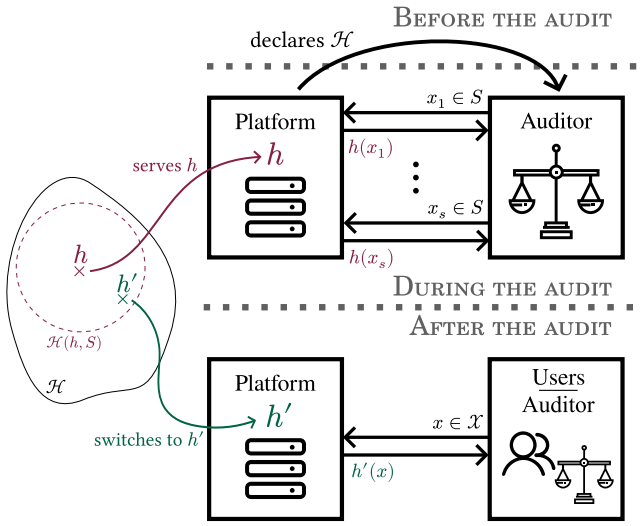


Figure 3.1: Security game of the manipulation-proof auditing framework. Before the audit, the model provider declares the hypothesis space $\mathcal{H}$ to the auditor. During the audit, the model provider serves the model $h \in \mathcal{H}$ and the auditor queries h on S. After the audit, the model provider can change its model to h' with the constraint that $\forall x \in S, h'(x) = h(x)$ or equivalently, $h' \in \mathcal{H}(h, S)$.

$\mathcal{H}$ exposed by the model provider to iteratively build an audit set $S \subset \mathcal{X}$. The manipulation-proof framework acknowledges the possibility for a model provider to try to *evade* the audit by showing a fair model h to the auditor, then switching to a more accurate but potentially unfair model h'. The only assumption on how the model provider may choose the new model h' is that it should be *consistent* with h. The consistency constraint requires h' to have the same outputs as h on the audit set S, otherwise the auditor could easily check that the model provider changed its model after the audit by re-querying it on S. We now formalize the capabilities and knowledge of the model provider and the auditor in the MP framework.

- *Auditor capabilities*: The auditor can send adaptive queries to the model provider to build an audit set $S \subset \mathcal{X}$.
- *Auditor knowledge*: The auditor knows the hypothesis class $\mathcal{H}$ implemented by the model provider and the value of the sensitive attribute x_A of all the points in the input space $\mathcal{X}$. However, the auditor does not know the specific hypothesis $h \in \mathcal{H}$ implemented by the model provider.
- *Platform capabilities*: The model provider can change its model from $h \in \mathcal{H}$ to $h' \in \mathcal{H}$ after the audit as long as h' respects the consistency constraint $\forall x \in S, h(x) = h'(x)$.
- *Platform knowledge*: The model provider knows the property μ (e.g. Demographic Parity) being measured by the auditor. As the auditor, it knows the value of the sensitive attribute x_A of all the points in the input space $\mathcal{X}$.

3.2. Machine Learning notations

Except when noted, we will consider a binary classification task as in [149], with finite *input space* $\mathcal{X}$ and output space $\mathcal{Y} = \{0,1\}$.[4] $\mathcal{Y}^{\mathcal{X}}$ denotes the space of functions $\mathcal{X} \to \mathcal{Y}$. For any sample $x \in \mathcal{X}$, we refer to its sensitive attribute (e.g., gender, ethnicity, religion) as $x_A \in \{0,1\}$. The sensitive attribute of the points in $\mathcal{X}$ induces a partition of the input space. We note $\mathcal{X}_A = \{x \in \mathcal{X} : x_A = 1\}$ and remark that $\mathcal{X}_{\overline{A}} = \overline{\mathcal{X}_A}$. For any set V, $\mathcal{P}(V)$ denotes the set of all subsets of V and $\mathcal{U}(V)$ denotes the uniform distribution on V. By training the classification model, the model provider effectively chooses a model h in some hypothesis class $\mathcal{H}$. The auditor defines a measure $\mu : \mathcal{H} \times \mathcal{P}(\mathcal{X}) \to \mathbb{R}_+$, which is known by the model provider. For any subset $V \subset \mathcal{H}$ and $S \subset \mathcal{X}$, we define *the diameter of* V *with respect to the measure* μ as

$$\operatorname{diam}_{\mu(\cdot,S)} V = \max_{h,h' \in V} |\mu(h,S) - \mu(h',S)|, \tag{3.2}$$

when S is the entire input space $\mathcal{X}$, we abuse the notation and write $\operatorname{diam}_{\mu(\cdot,S)} V = \operatorname{diam}_{\mu} V$. Finally, define for any subset $V \subset \mathcal{H}$, sample $x \in \mathcal{X}$ and label $y \in \{0,1\}$ the set $V[x,y] = \{h \in V : h(x) = y\}$. The cost $\operatorname{Cost}(V)$ of a subset $V \subset \mathcal{H}$ is defined in Equation (3.3). Note that when the context is clear, we elide the ε for simplicity.

$$\operatorname{Cost}_{\varepsilon(V)} = \begin{cases} 0 & \text{if } \operatorname{diam}_{\mu} V < \varepsilon \\ 1 + \min_{x \in \mathcal{X}} \max_{y \in \{0,1\}} \operatorname{Cost}_{\varepsilon}(V[x,y]) & \text{else} \end{cases} \tag{3.3}$$

Before we formally define the *capacity* of a hypothesis class in Subsection 3.5.2, we will use the term capacity loosely. Intuitively the capacity of a hypothesis class $\mathcal{H}$ is related to the ability for any labeling of the input space $\mathcal{X}$ to find a hypothesis $h \in \mathcal{H}$ that realizes this labeling. More details on the notion of capacity can be found in Section 3.2.

3.3. What is an active auditing algorithm?

An audit algorithm $\mathcal{A}$ with label budget s is a sequence of (possibly randomized) $s+1$ functions $(f_0, ..., f_s)$. For each iteration i, the function $f_i : (\mathcal{X} \times \{0,1\})^{i+1} \to \mathcal{X}$ chooses the next sample $x_i = f_i((x_0, h(x_0)), ..., (x_{i-1}, h(x_{i-1})))$ to query and add to the audit set. After the query budget has been spent, the end result of the algorithm is the audit set $S = \mathcal{A}(h)$. Note that most published black-box audits of web platforms are not active [139]. In this case, an audit algorithm reduces to a single (possibly randomized) function f_s which does not depend on the answers provided by the model provider.

4. Should $\mathcal{X}$ be infinite, [149] notes that it suffices to sample a finite i.i.d. subset $\tilde{\mathcal{X}}$ and extend all the following bounds by classical generalization bounds.

3.4. The manipulation-proof auditing framework

Following the framework of Yan & Zhang [1], the model provider is assumed to be *self-consistent*, i.e. when the model provider returns a given output $y = h(x)$ to an auditor's query x, the model provider commits to this value and cannot return a different answer $y' = h(x)$ if x is queried again at a later moment in time. Furthermore, as explained in the threat model Figure 3.1, it is assumed that the auditor knows the *hypothesis class* $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ of the model implemented by the model provider. The self-consistency of the model provider together with the knowledge of the hypothesis class defines a subset of "plausible" models in $\mathcal{H}$ that have the same answers as the model provider on the current audit set S. This subset is called the *version space* [130, 157].

Definition 3.1 (Version space) *Define the model $h \in \mathcal{H}$ and audit set $S \subset \mathcal{X}$. The version space of h induced by S is*

$$\mathcal{H}(h, S) = \{h' \in \mathcal{H} : \forall x \in S, h'(x) = h(x)\}.$$

We assume that the model provider seeks to maximize its profits, which is not necessarily aligned with the property that the regulator seeks to enforce. During the audit process, the auditor incrementally builds an audit set $S \subseteq \mathcal{X}$ based on their previous queries and the answers of the model provider. The goal of the auditor is to produce an estimate $\hat{\mu}$ as close as possible to the real value while being robust to the potential manipulations implemented by the model provider. We now formulate the two requirements of the *manipulation-proof (MP)* auditing problem, as introduced in [1].

$$\begin{aligned} &\text{Create an algorithm } \mathcal{A} \text{ with smallest budget s such that,} \\ &\qquad\qquad \text{(fidelity)} \quad |\mu(h, \mathcal{A}(h)) - \mu(h, \mathcal{X})| < \varepsilon \\ &\text{(manipulation-proofness)} \quad \operatorname{diam}_{\mu} \mathcal{H}(h, \mathcal{A}(h)) < \varepsilon \end{aligned} \tag{3.4}$$

Fidelity is the classical estimation constraints. It requires the estimated value $\hat{\mu} = \mu(h^*, S)$ to be close to the real value $\mu(h, \mathcal{X})$. In addition, *manipulation-proofness* requires that if the model provider changes its implemented instance from h to h' while respecting the self-consistency constraint $h' \in \mathcal{H}(h, S)$, the difference between the previous $\mu(h, \mathcal{X})$ and new $\mu(h', \mathcal{X})$ values of μ must be bounded. Therefore, the μ-diameter is the biggest change in the value of μ the auditor would accept if the model provider changed to another (consistent) hypothesis.

3.5. Comparing manipulation-proof auditing algorithms

There are two ways to compare two audit algorithms $\mathcal{A}$ and $\mathcal{A}'$. Either fix a target manipulation-proofness guarantee ε and evaluate the number of queries needed

Algorithm	Query complexity
Random sampling (Algorithm 1)	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\log\lvert\mathcal{H}\rvert\right)$
Optimal deterministic [1]	$\mathrm{Cost}_{\varepsilon}(\mathcal{H})$
Oracle based approximation (AFA) [1]	$\mathcal{O}(\log\lvert\mathcal{H}\rvert\log\lvert\mathcal{X}\rvert\ \mathrm{Cost}(\mathcal{H}))$

Table 3.1: The query complexity of different auditing algorithms in the manipulation-proof framework, extracted from Yan et al. [1]

by $\mathcal{A}$ and $\mathcal{A}'$, or fix the audit budget s and evaluate the μ-diameter of the audit sets built by $\mathcal{A}$ and $\mathcal{A}'$.

Yan & Zhang [1] focused on the former: the study of the *query complexity* of different audit algorithms. For general hypothesis classes, they introduced three auditing algorithms. The first one is the baseline random audit algorithm. This audit algorithm consists in sampling among points with positive and negative sensitive attributes, and computing the empirical frequencies of the events $(h(X) = 1 \mid X_A = 1)$ and $(h(X) = 1 \mid X_A = 0)$ (see Algorithm Algorithm 1). To capture the minimal query complexity attainable by deterministic audit algorithms, they introduced a second algorithm based on the recursive minimization of $\mathrm{Cost}(\mathcal{H})$. Finally, T. Yan and C. Zhang introduced a third, oracle-based, algorithm that we coin AFA. We summarize the query complexities proved by [1] in Table 3.1.

Motivated by the implementation of MP audit algorithms, we choose to focus on the second comparison approach: fixing an audit budget and evaluating the μ-diameter. This approach is better suited to our situation since in practice, auditors have a limited query budget that would be agreed upon with the model provider prior to the audit.

3.6. The computational complexity of manipulation-proof auditing

As exposed in Table 3.1, the best attainable query complexity, as well as the query complexity of the more practical AFA algorithm depend on the value of $\mathrm{Cost}(\mathcal{H})$. In addition, the computational complexity of AFA [1] is the time to train a model from the hypothesis class $\mathcal{H}$ multiplied by the query complexity. However, T. Yan and C. Zhang prove that $\mathrm{Cost}(\mathcal{H})$ is hard to compute, hard to approximate and hard to optimize [1]. Thus, not only it prevents practical implementations of the optimal deterministic algorithm, it also prevents practical analysis of the query complexity and computational complexity of AFA for large models that are costly to train.

4. The competitive effectiveness of random audits

Current state-of-the-art models for tabular data (see Figure 3.5) and image data (see e.g. [153]) are able to fit very large train sets with close to perfect accuracy

while retaining good generalization properties. In our setting this would mean that these models can represent any binary classification function $f : \mathcal{X} \to \{0, 1\}$ of the input space. As we saw in Subsection 3.3.6, the only tractable algorithm (`AFA` [1]) that was proposed to solve the MP auditing task (Equation (3.4)) is still too computationally intense to audit large models because it requires to be able to train a lot of copies efficiently. Moreover, while [1] experimented on small datasets with linear models, there exists no implementations or practical experiments on larger models. Thus, the potential gains brought by `AFA` are hard to predict. Yet, for `AFA` to be used in practice, it would be necessary to balance the extra cost induced by auditing with `AFA` with the added guarantees of `AFA`. Thus, a natural practical question arises. **Is the added manipulation-proofness guarantee worth paying the computational toll?**

To answer this question, instead of analyzing $\mathrm{Cost}(\mathcal{H})$ (which is hard to compute and derive) as [1], we directly express the value of $\mathrm{diam}_\mu \mathcal{H}(h, S)$ for specific hypothesis classes. Identifying hypothesis classes $\mathcal{H}$ wherein the value of $\mathrm{diam}_\mu(h, S)$ remains constant across all audit sets S allows us to find scenarios in which enhancing manipulation-proofness guarantees beyond that of a random baseline is impossible.

In this section, we consider three typical but insightful forms of hypothesis space $\mathcal{H}$ to better understand this balance between computational cost and added robustness. We prove in Subsection 3.4.1 that for hypothesis classes shattering the whole input space, all the audit algorithms have the same performance as random sampling. Next, to understand what happens for classes that are only able to fit a part of the dataset, we consider the illustrative class $\mathcal{H}_m^{\mathrm{dict}}$ of dictionaries of size m. We derive the exact value of their μ-diameter in Subsection 3.4.2 and show the link between the memory as an intuitive notion of the capacity and the MP guarantees obtainable when auditing dictionary models. Last but not least, building on the results of Subsection 3.4.1 and Subsection 3.4.2, we introduce a formal notion of the capacity of a binary classification hypothesis class as the maximum number of samples a model provider can interpolate while still retaining good generalization performance. Under this definition, we prove in Subsection 3.4.3 that large capacity models cannot be audited more efficiently than by the random baseline.

Require: Proportions β_1, β_2, budget s
Ensure: audit dataset S with $|S| = s$
$s^+ \leftarrow \lfloor \beta_1 |\mathcal{X}_A| \rfloor$, $s^- \leftarrow \left\lfloor \beta_2 \left|\overline{\mathcal{X}_A}\right| \right\rfloor$
$S^+ \leftarrow$ sample s^+ points in $\mathcal{X}_A$ without replacement
$S^- \leftarrow$ sample s^- points in $\overline{\mathcal{X}_A}$ without replacement
Return $S = S^+ \sqcup S^-$

Algorithm 1: The random sampling audit strategy

4.1. Hypothesis classes that can fit the dataset entirely

To build intuition on the following theorems, let us first consider classes able to fit any labeling of $\mathcal{X}$. This corresponds to the case of a model provider with a very large, over-parametrized hypothesis class $\mathcal{H}$ able to fit any labeling of the whole input space $\mathcal{X}$.[5] This assumption is equivalent to considering the hypothesis class $\mathcal{H} = \{0,1\}^{\mathcal{X}}$. Because all the functions from the input space to the output space are possible, the answer of the model provider on a query x does not give any information on the possible answers to the other queries in $\mathcal{X}$. It follows, that no matter how the points are iteratively chosen, only the number of points (and the value of their associated sensitive attribute) will matter in the computation of the μ-diameter. We now formalize this intuition.

Theorem 3.1 (No need to aim) *Let $\mathcal{H} = \{0,1\}^{\mathcal{X}}$. For any audit set $S \subseteq \mathcal{X}$ and hypothesis $h \in \mathcal{H}$,*

$$\operatorname{diam}_\mu \mathcal{H}(h, S) = 2 - (\mathbb{P}(X \in S \mid X_A = 1) + \mathbb{P}(X \in S \mid X_A = 0))$$

Proof sketch. The first step in proving Theorem 3.1 relies on the fact that all the instances $h' \in \mathcal{H}(h, S)$ have the same value of $\mu(h', S)$. After decomposing the μ-diameter on S and $\overline{S}$, we use this fact to separate the μ-diameter into the difference between a maximization and a minimization problem. The optima of these problems rely on the existence of hypotheses $h^\uparrow, h^\downarrow \in \mathcal{H}(h, S)$ that exactly fit the sensitive attribute (resp. its negation) on $\overline{S}$. Since $\mathcal{H}$ is the space of all functions, it is always possible to find such $h^\uparrow$ and $h^\downarrow$. Finally, we find these optima and simplify their expressions to reach that of Theorem 3.1. A complete proof is provided in Appendix C.

The values $\mathbb{P}(X \in S \mid X_A = 1)$ and $\mathbb{P}(X \in S \mid X_A = 0)$ are aggregated quantities that depend only on the relative proportion of sensitive ($x_A = 1$) and non-sensitive ($x_A = 0$) samples in the audit set S. Therefore, for any pair $(\mathbb{P}(X \in S \mid X_A = 1), \mathbb{P}(X \in S \mid X_A = 0))$, one can design a random sampling scheme

5. This does not contradict the No Free Lunch theorem since here, the input space $\mathcal{X}$ is finite.

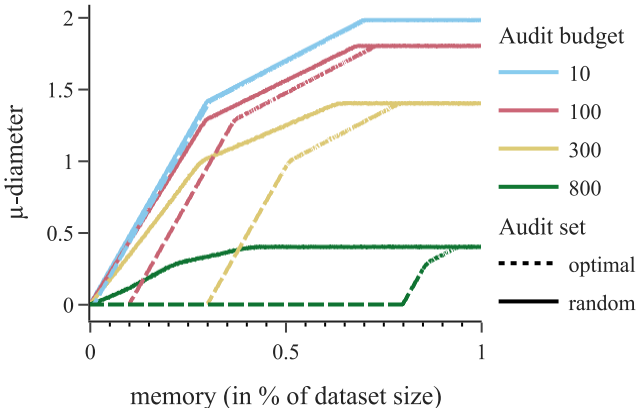


Figure 3.2: The diameter (vertical axis) resulting from the amount of memory (horizontal axis) of the dictionary model studied in subsection 3.2. The various audit budgets are represented by different curve colors, while the optimal audit set appears as dashed curves, and the random baseline audit sets as plain lines.

that achieves the desired relative proportions. We expose such algorithm in Algorithm 1. Since the auditor by definition knows the sensitive attribute of each sample, the idea is to sample points from $\mathcal{X}_A$ and $\mathcal{X}_{\overline{A}}$ with the right proportions (β_1, β_2) in S_{random}. Setting $(\beta_1, \beta_2) = (\mathbb{P}(X \in S \mid X_A = 1), \mathbb{P}(X \in S \mid X_A = 0))$ in Algorithm 1 yields $(\mathbb{P}(X \in S_{\text{random}} \mid X_A = 1), \mathbb{P}(X \in S_{\text{random}} \mid X_A = 0)) = (\mathbb{P}(X \in S \mid X_A = 1), \mathbb{P}(X \in S \mid X_A = 0))$. Following Theorem 3.1, any audit set S with the same relative proportions $(\mathbb{P}(X \in S \mid X_A = 1), \mathbb{P}(X \in S \mid X_A = 0))$ yields the same μ-diameter. Since any couple $(\mathbb{P}(X \in S \mid X_A = 1), \mathbb{P}(X \in S \mid X_A = 0))$ is also attainable by the random sampling algorithm described in Algorithm 1, **when the hypothesis class can perfectly fit any arbitrary label distribution, all audit algorithms –active or not– have at most the same manipulation-proofness guarantees as random sampling**.

As a side note, removing the assumption that the auditor knows the hypothesis class implemented by the model provider is equivalent to assuming $\mathcal{H} = \{0,1\}^{\mathcal{X}}$. In this sense, by proving that random sampling is optimal when the hypothesis class is unknown, Theorem 3.1 demonstrates that knowing $\mathcal{H}$ is necessary (but not sufficient) to design more efficient manipulation-proof auditing methods.

4.2. An illustrative example with dictionaries

It is unlikely in practice that any hypothesis class can fit the entire input space $\mathcal{X}$. We now relax this assumption to pursue our analysis of the achievable manipulation-proofness guarantees of models with a large capacity. To that end, we introduce the class $\mathcal{H}_m^{\text{dict}}$ of dictionary models. A dictionary $d \in \mathcal{H}_m^{\text{dict}}$ is built by choosing a set of $m \in [\![n]\!]$ samples in $\mathcal{X}$ and storing the corresponding labels.

When the dictionary is asked to label a sample that it did not store, it returns 0 as a default value. Define, for any set of vectors $V \subseteq R^d$, $\mathfrak{S}(V)$ the set of vectors obtained from V by including all permutations of the coefficients of each $v \in V$. The hypothesis class of dictionaries of memory m is formally introduced in Definition 3.2.

Definition 3.2 (Dictionary hypothesis class) *Consider an input space $\mathcal{X}$, $n = |\mathcal{X}|$. The class of dictionaries of memory $m \in [\![n]\!]$ is defined as*

$$\mathcal{H}_m^{\text{dict}} = \mathfrak{S}(\{0,1\}^m \times \{0_{R^{n-m}}\})$$

While such a hypothesis class is not likely to be used in a practical context (as it will typically fail to generalize beyond the encountered examples, exhibiting a blatant overfitting) it is simple enough to support an analysis of the MP guarantees for both randomized and optimal approaches. Moreover, its main parameter (the memory m) directly influences its capacity. The exact value of the μ-diameter of dictionary hypothesis classes is exposed in Theorem 3.2. The proof can be found in Appendix D.

Theorem 3.2 (Memory and auditability) *Consider $S \subseteq \mathcal{X}$, $d \in \mathcal{H}_m^{\text{dict}}$. Note $m' = m - |x \in S : d(x) = 1|$. The μ-diameter of $\mathcal{H}_m^{\text{dict}}(d, S)$ is given by*

$$\operatorname{diam}_\mu \mathcal{H}_m^{\text{dict}}(d,S) = \frac{\min\left(\left|\mathcal{X}_A \cap \overline{S}\right|, m'\right)}{|\mathcal{X}_A|} + \frac{\min\left(\left|\overline{\mathcal{X}_A} \cap \overline{S}\right|, m'\right)}{\left|\overline{\mathcal{X}_A}\right|}$$

Proof sketch. The proof relies on the same development of the diameter as in the proof of Theorem 3.1 but instead of finding $h^\uparrow$ and $h^\downarrow$, we are able to give the values of the optima thanks to the structure of $\mathcal{H}_m^{\text{dict}}$. The complete proof is exposed in Appendix D.

We are interested in the high memory m, low audit budget $|S|$ regime. In this situation, there exist couples (S, m) such that $\left|\mathcal{X}_A \cap \overline{S}\right| \leq m'$ and $\left|\overline{\mathcal{X}_A} \cap \overline{S}\right| \leq m'$. Thus, in this regime, the μ-diameter does not depend on the memorized points of the particular dictionary d chosen by the model provider. Therefore, as for the case $\mathcal{H} = \{0,1\}^{\mathcal{X}}$, in the high memory, low audit budget regime, all audit algorithms – active or not – have at most the same manipulation-proofness guarantees as random sampling.

Simulation of the impact of memory over diameter The expression of the μ-diameter exposed in Theorem 3.2 is piecewise linear in the memory m. To gain intuition, we plot the value of $\operatorname{diam}_\mu \mathcal{H}_{m(d,S)}^{\text{dict}}$ in Figure 3.2 for a setting where $|\mathcal{X}| = 1000$, $\mathbb{P}(X_A = 1) = 0.3$ and the μ-diameter of the random strategy is averaged over 100 realizations of S. We first observe the drastic impact of dictionary memory on an audit of a fixed budget: for instance, with an audit budget

of 300 (representing nearly one-third of the whole input space) an optimal audit set barely achieves a μ-diameter of 1 when auditing dictionaries with memory $m = 500$. Furthermore, given a fixed audit budget, the gap between randomized and optimal audit sets shrinks as the memory grows. This is especially striking in low audit budget regimes, that correspond to a typical audit situation. Moreover, for an audit budget of 100 and memory values larger than 70% the random and optimal audit strategies have the same μ-diameter. This observation hints that Theorem 3.1's conclusions should hold for a broader set of hypothesis classes.

4.3. Tying it all together: large capacity and auditability

We derived in Subsection 3.4.2 the exact expression of the μ-diameter for toy models able to memorize part of the input space. Motivated by the *benign overfitting* phenomenon [153, 156, 158, 159], we now consider the case of a hypothesis class that is able to perfectly fit any subset $S \subseteq \mathcal{X}$ of reasonable size, but require in addition that the resulting hypothesis h^* maintains good accuracy on the rest of the dataset.

It has been observed that contrary to common knowledge on the bias-variance tradeoff, large ML models can exhibit good generalization properties while perfectly fitting the train data. This benign overfitting phenomenon (also related to *double descent*), is observed in models that are largely overparametrized compared to the training data available at hand. Nevertheless, we show in Figure 3.3 that trees and GBDTs can reach the maximum capacity, indicating that they also can interpolate the training data. Drawing intuition from the empirical characterization of benign overfitting in [153, 156, 158, 159], we derive the formal definition of a large capacity hypothesis class in Definition 3.3.

Definition 3.3 (Benign Overfitting Hypothesis class) *Consider an input space $\mathcal{X}$, a hypothesis class $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$ and a labeling $c \in \{0,1\}^{\mathcal{X}}$. $\mathcal{H}$ is said to exhibit benign overfitting with respect to labeling c if there exists $d_0 \in N_*$ and $\varepsilon \in [0,1)$ such that*

$$\forall d \leq d_0, S \subseteq \mathcal{X}, \sigma \in \{0,1\}^d, \exists h \in \mathcal{H},$$
$$\begin{cases} \forall x_i \in S, h(x_i) = \sigma_i & \text{(fit any train set)} \\ \mathbb{P}\Big(h(X) = c(X) \mid X \in \overline{S}\Big) = 1 - \varepsilon & \text{(with low error on } c\text{)} \end{cases}$$

As is stands, Definition 3.3 is tightly linked to the notion of version space. If $\mathcal{H}$ exhibits overfitting, we are guaranteed that all the version spaces $\mathcal{H}(h^*, S)$ (such that $|S| \leq d_0$) derived from $\mathcal{H}$ contain a hypothesis that generalizes well on the whole dataset. Moreover, Definition 3.3 is the literal formalization of the notion of benign overfitting considered in [153] and [158]: models that can fit any labeling

–even random– of the train set while still having a good test performance when evaluated on the target distribution.

This definition of large capacity models enables the same analysis as in Theorem 3.1, without the requirement that the hypothesis class $\mathcal{H}$ spans the entire set of functions $\{0,1\}^{\mathcal{X}}$.

Corollary 3.1 (Benign overfitting and auditability) *Let $\mathcal{X}$ and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any input space and hypothesis class. Assume that $\mathcal{H}$ exhibits benign overfitting with respect to the sensitive attribute X_A and its opposite $1 - X_A$[6], then for any $d \leq d_0$, and $S \in \mathcal{X}^d$,*

$$\begin{aligned}\operatorname{diam}_{\mu}\mathcal{H}(h^*, S) \geq \mathbb{P}(X \in S \mid X_A = 1) + \mathbb{P}(X \in S \mid X_A = 0) \\ -2\,\mathbb{P}(X \in S) - 2\varepsilon(1 - \mathbb{P}(X \in S))\end{aligned}$$

The proof of Corollary 3.1 is deferred to Appendix E Observe that lower bound on the μ-diameter given by Corollary 3.1 only depends on the aggregated quantities $\mathbb{P}\big(X \in \overline{S}\big)$, $\mathbb{P}\big(X \in \overline{S} \mid X_A = 1\big)$ and $\mathbb{P}\big(X \in \overline{S} \mid X_A = 1\big)$. As for Theorem 3.1, this implies that no audit method, active or not can perform better than a simple random sampling baseline (Algorithm 1) with the right proportions β_1 and β_2. The term $\mathbb{P}(X \in S \mid X_A = 1) + \mathbb{P}(X \in S \mid X_A = 0) - 2\mathbb{P}(X \in S)$ indicates the importance of the relative proportion of these two audited groups in the audit set as in Theorem 3.1. The term $2\varepsilon(1 - \mathbb{P}(X \in S))$ indicates that as expected, the larger the error rate ε gets, the smaller the μ-diameter will be. Thus, **when the hypothesis class exhibits benign overfitting, all audit algorithms –active or not– have at most the same manipulation-proofness guarantees as random sampling.** This shows that, under manipulations, large models currently used in production are not auditable more efficiently than by random sampling.

5. Manipulability under random audits and model capacity

As shown in Section 3.4, the random audit baseline is optimal when the model has a large capacity, but has no guarantee of optimality when the hypothesis class is constrained to lower capacities. To compare ML algorithms in practice, we now introduce a measure of *manipulability under random audits* and a measure of *model capacity*. We will use these methods to empirically evaluate the manipulability of auditing several models of increasing capacities in Section 3.6.

6. That is, Definition 3.3 holds for $c = x_A$ and $c = 1 - x_A$

5.1. Measuring the manipulability under random audits of practical models

The manipulability of a hypothesis class $\mathcal{H}$, is defined (Equation (3.5)) as the μ -diameter obtained and averaged over audit datasets S sampled by the random audit baseline Algorithm 1 with budget s.

$$\text{Manipulability}(\mathcal{H}, s) = \mathbb{E}_{S,h^*}\left[\text{diam}_\mu \mathcal{H}(S, h^*)\right] \tag{3.5}$$

The manipulability under random audits is a lower bound of the auditor "power" In a perfect situation, for any budget $s = |S|$, the auditor would be able to select the audit set S^* that attains the minimum μ-diameter, whatever the hypothesis class $\mathcal{H}$ and chosen hypothesis $h^* \in \mathcal{H}$ are. As explained in Subsection 3.3.6, this is not possible in practice for computational reasons and thus cannot be simulated. Thus we evaluate the manipulability under random audits with the baseline random audit strategy (Algorithm Algorithm 1). Taking the expectation of $\text{diam}_\mu \mathcal{H}(h^*, S)$ over random audits allows to upper bound the value of the minimum attainable μ-diameter $\min_{\mathcal{A}} \text{diam}_\mu \mathcal{H}(h^*, \mathcal{A}(h^*))$.

The manipulability under random audits is a lower bound of the model provider "power" In a fully adversarial setting, whatever the hypothesis class $\mathcal{H}$, the model provider would choose the hypothesis h^* that maximizes $\text{diam}_\mu \mathcal{H}(h^*, S)$ for most of the audit sets S the auditor could come up with. While this would effectively be the worst case for the auditor, it is however unlikely to happen in practice since the model provider would have to balance the maximization of the accuracy with the maximization of the μ-diameter. Therefore, we consider the more practical situation in which the model provider can freely choose the hypothesis class $\mathcal{H}$ but the implemented instance h^* minimizes a classical loss L adapted to the model being trained (e.g. cross-entropy or ℓ_2 norm). This can be seen as a lower bound of the adversarial "power" of the model provider.

5.2. Measuring the capacity of practical models

There are multiple operationalizations of the notion of capacity, from theoretically-rooted metrics such as the VC dimension [151] or Rademacher complexity [152], to more empirical definition such as the number of iterations until overfitting [153]. The interplay between VC-dimension and manipulability under random audits is already pointed out in [1], where it is observed that models of VC-dimension higher than $1,600$ have a high manipulability under random audits.

Unfortunately, the VC dimension of a class is difficult to estimate in practical settings. Instead, the empirical Rademacher complexity (Equation (3.6)) is leveraged to quantify the capacity of the studied hypothesis classes. Informally in our setting, a hypothesis class has a high Rademacher complexity if whatever the labels and size of an audit set S, there exists an instance $h_S \in \mathcal{H}$ that fits those labels on

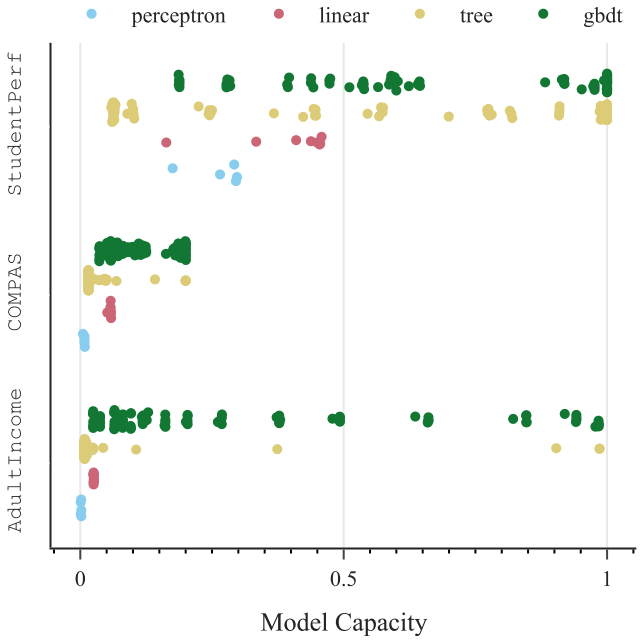


Figure 3.3: Distribution of the capacity (horizontal axis) for different hyperparameters choices on the three datasets (vertical axis). Each model is trained with different hyperparameter values with each couple (model, hyperparameter) representing a different hypothesis class $\mathcal{H}$. For each (model, hyperparameter) couple, the empirical Rademacher values $R_m(\mathcal{H} \circ D)$ are averaged over 15 realizations of D and σ_i before computing the model capacity.

S with high accuracy. To avoid threshold effects in our experiments, we average the complexity over different sizes of D considered in the Rademacher metric (Equation (3.7)). Formally:

$$R_m(\mathcal{H} \circ D) = \frac{1}{m}\mathbb{E}_{\boldsymbol{\sigma}\sim\{0,1\}^m}\left[\sup_{h\in\mathcal{H}}\sum_{x_i\in D}\sigma_i h(x_i)\right] \tag{3.6}$$

$$\text{Capacity}(\mathcal{H}) = \mathbb{E}_{D\sim\mathcal{X}^m, m\sim[\![|\mathcal{X}|]\!]}\left[R_{m(\mathcal{H}\circ D)}\right] \tag{3.7}$$

6. Experiments

In this section, we explore the relation of the manipulability under random audits (Equation (3.5)) with the capacity of hypothesis classes (Equation (3.7)). The following experiments were run on three tabular datasets: StudentPerf [131], COMPAS [132] and AdultIncome [120]. Dataset statistics and considered tasks are presented in Table 3.2. Neural methods on tabular data are still outperformed by tree methods [160]. We thus choose to focus our study on the four following models: linear models, perceptrons, decision trees and gradient-boosted trees.

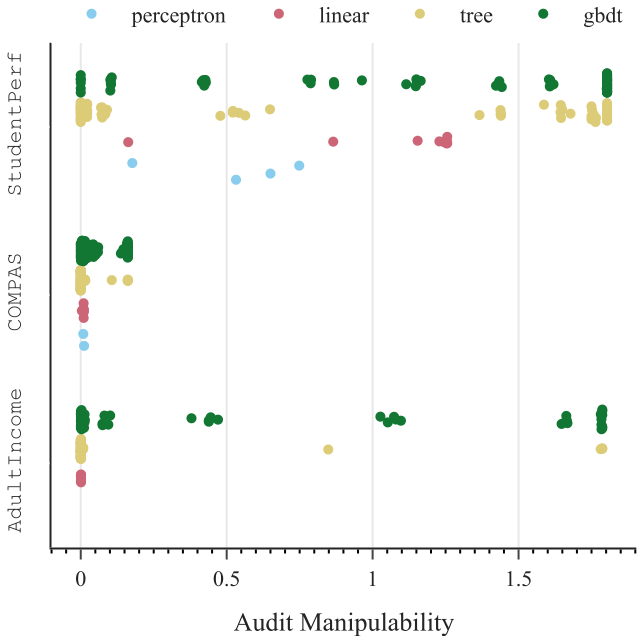


Figure 3.4: Distribution of the Manipulability (manipulability under random audits) values (horizontal axis) of different models $\mathbb{H}$ on a selection of datasets (vertical axis). Each bar represents a different model $\mathbb{H}$ (trees, linear models, ...). Each model is trained with different hyperparameter values with each couple (model, hyperparameter) representing a different hypothesis class $\mathcal{H}$. For each dataset, the size of the audit set is set to 10% of the dataset size: $|S| = 0.1|\mathcal{X}|$. For each (model, hyperparameter) couple, the μ-diameter are averaged over 15 audit datasets before computing the manipulability.

Similar to [160], we selected a range of hyperparameters for each model and sampled a total of 500 hyperparameters over the 4 models. In previous sections, we stated results with respect to a given hypothesis class $\mathcal{H}$. In the following experiments, a hypothesis class $\mathcal{H}$ represents a couple (model, hyperparameters). Thus, a model represents a family of hypothesis classes $\mathbb{H} = \left(\mathcal{H}_1, ..., \mathcal{H}_f\right)$, each hypothesis class $\mathcal{H}_i$ being associated with a hyperparameters tuple.

The hyperparameters and their value range are listed in Appendix H (Table 6.1). For each model, we created a grid with all the possible combinations of hyperparameter values and ran our experiments on all the resulting (model, hyperparameter) couples.

6.1. Simulating hypothesis spaces with a broad range of manipulability and capacity

In Figure 3.4, we plot the manipulability under random audits of different hypothesis classes. These classes are constructed by using multiple hyperparameters for each family $\mathbb{H}$ listed in Table 6.1; each dot then represents a specific (family,

dataset	Size n	Features d	Task
StudentPerf	395	43	Predict if students pass the exam
COMPAS	6172	21	Predict subject recidivism
AdultIncome	22, 268	10	Predict if income is $\geq$50,000

Table 3.2: Datasets stats

hyperparameter set) couple. On one hand, for large datasets (such as AdultIncome and COMPAS), we observe that simpler models (linear, perceptron) have a very low manipulability, no matter the hyperparameter set used. On the other hand, for smaller datasets (such as StudentPerf), smaller models (such as linear models or perceptrons) can also fit the data hence also becoming harder to audit.

Similarly, in Figure 3.3, we plot the capacity of the simulated hypothesis classes on AdultIncome, COMPAS and StudentPerf. As discussed before, it can be observed that for AdultIncome and StudentPerf datasets, tree-based models reach the maximum capacity value of 1. However, on the COMPAS dataset all hypothesis classes exhibit capacity values that do not exceed 0.2 points. This has been observed before [161] and does not affect our main argument on the link between model capacity and manipulability.

6.2. Measuring the μ-diameter in practice

As originally defined in [1] and following the definition of the μ-diameter, the evaluation of $\text{diam}_{\mu(S,h^*)}$ requires to solve the following optimization problem:

$$\begin{aligned} &\max_{h,h'} \ |\mu(h,S) - \mu(h')| \\ &\text{subject to } h(x) = h'(x) = h^*(x) \ \forall x \in S \end{aligned} \tag{3.8}$$

This problem be separated in two optimization problems: the maximization/minimization over $h \in \mathcal{H}$ of $\mu(h,S)$ under the constraint that $\forall x \in S, h(x) = h^*(x)$.

$$\begin{aligned} &\max_h / \min_h \ \mu(h,S) \\ &\text{subject to } h(x) = h^*(x) \ \forall x \in S \end{aligned} \tag{3.9}$$

As proposed by [1], we use the method introduced by [162] to reframe this constrained optimization problem as a sequence of weighted classification tasks. Then, we use off-the-self estimators from scikit-learn and XGBoost to perform the optimization with the appropriate weights.

6.3. Model capacity conditions manipulability

In Subsection 3.5.1 we compared different models and how difficult they were to audit, depending on the chosen hyperparameters. We now take a closer look at the impact of a model's capacity on its manipulability under random audits, in

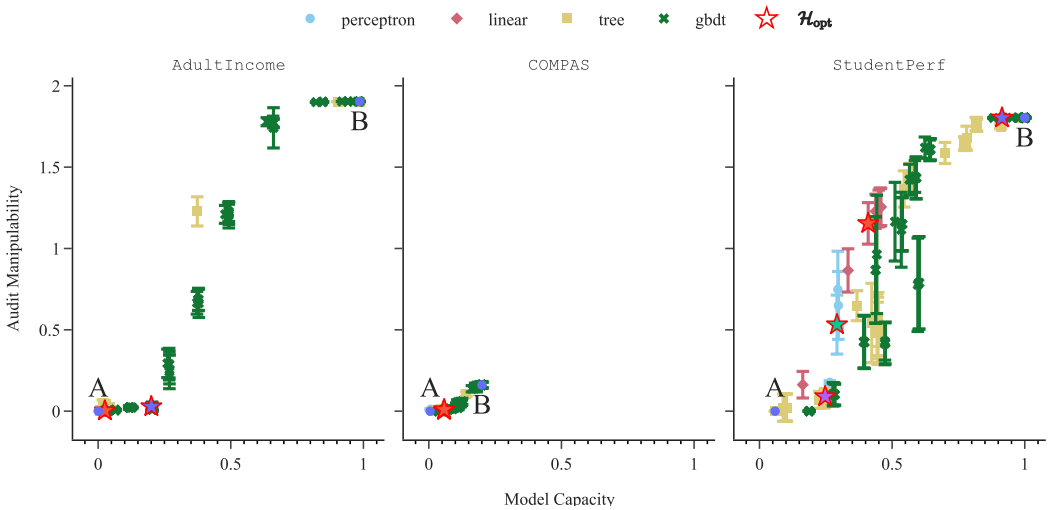


Figure 3.5: Distribution of the manipulability under random audits values (vertical axis) of different models versus their capacity (horizontal axis) on a selection of datasets. Each point represents a couple (model, hyperparameter). For each dataset, the size of the audit set is set to 10% of the dataset size: $|S| = 0.1|\mathcal{X}|$. For each (model, hyperparameter) couple, the Manipulability is averaged over 15 audit datasets, and the capacity is computed over 30 randomizations of the dataset labels. The error bars represent the standard deviation.

an attempt to confirm the link between both concepts. We plot in Figure 3.5 the relation between the capacity of a hypothesis class and its manipulability under random audits. Points also represent (model, hyperparameter) couples, while the vertical error bars represent the standard deviation of the μ-diameter values for different random audit sets S.

Consistent with the intuition and results developed until now, we observe that for all the datasets, the manipulability under random audits increases with the capacity of the hypothesis class. While on both AdultIncome and StudentPerf, the μ-diameter reaches the maximum capacity value at almost 2, for COMPAS, the effect is not as dramatic. To highlight the connection between the results exposed in Section 3.4 and the empirical relation found between model capacity and manipulability under random audits, we focus next on two specific points, marked with the letters A and B in Figure 3.5.

First, consider the point $A = (\text{Capacity} \approx 0, \text{Manipulability} \approx 0)$. For a hypothesis class to have a null capacity, it has to have null Rademacher complexity on any subset of the sample space. This is verified by models that perform no better than random labels generation. Since the value of $\mu(h, \mathcal{X})$ of any instance of such hypothesis class is only determined by the ratio of samples with a positive sensitive attribute, the μ-diameter of such hypothesis class is null. This is why in

Figure 3.5, models with near-zero capacity have a very low (if not null) manipulability under random audits.

The second notable point is $B = (\text{Capacity} = \text{Capacity}_{\max}, \text{Manipulability} = \text{Manipulability}_{\max})$. Any hypothesis with a unitary capacity has a unitary Rademacher complexity for any dataset size s and thus shatters any subset of $\mathcal{X}$. Therefore, at point B, Theorem 3.1′s hypothesis $\mathcal{H} = \{0,1\}^{\mathcal{X}}$ holds. This means that hypothesis classes that are characterized by this point cannot be audited more efficiently than by a random audit strategy. It follows that (at least on StudentPerf and AdultIncome) the model provider can always choose a hypothesis class that cannot be audited efficiently by any strategy, forcing the auditor to prompt most of the input space to obtain robustness guarantees.

Generalization versus diameter We saw that by choosing the right hypothesis class (that is, the right set of hyperparameters), the model provider can easily evade the audit. However, in practice the choice of hypothesis class is also guided by a classical train-dev-test separation, choosing the hyperparameter set that generalizes best. What is the typical μ-diameter of hypotheses classes that generalize well? To answer this question, we simulate a 5-fold hyperparameter optimization procedure. For each family of models, we denote $\mathcal{H}_{\text{opt}}$ the hypothesis class with the set of hyperparameters that minimize the 5-fold average test loss in its model family $\mathbb{H}$. For each model family, $\mathcal{H}_{\text{opt}}$ is differentiated in Figure 3.5 by a star marker with red edges. Interestingly, for COMPAS and AdultIncome datasets and for all model families, the generalization-optimal hypothesis classes $\mathcal{H}_{\text{opt}}$ have a relatively low capacity compared to the maximum achievable capacity, especially for tree-based models. For the StudentPerf dataset, the results are more nuanced, most likely because the dataset has a limited size, which implies that it is simpler to reach high capacity values.

As a glimmer of hope, from point A to B, there is a range of hypothesis classes for which the random strategy could be improved as seen by the size of the y -axis error bars. Overall, the hypothesis classes that are most likely to be implemented by faithful platforms (the hypothesis classes that generalize well) are already straightforward to audit (they have a Manipulability ≈ 0). Yet, unfaithful platforms wanting to game the audit can always choose a hypothesis class that forces the auditor to issue a lot of queries to reach higher manipulation-proofness guarantees.

6.4. The cost of exhausting the auditor

We observed in Section 3.4 and Subsection 3.5.1 that the hypothesis classes that are the hardest to audit are those with the largest capacity. Yet, we also observed that the hypothesis classes most likely to be implemented (i.e. the ones with the highest generalization) have a low μ-diameter and are not those with high capac-

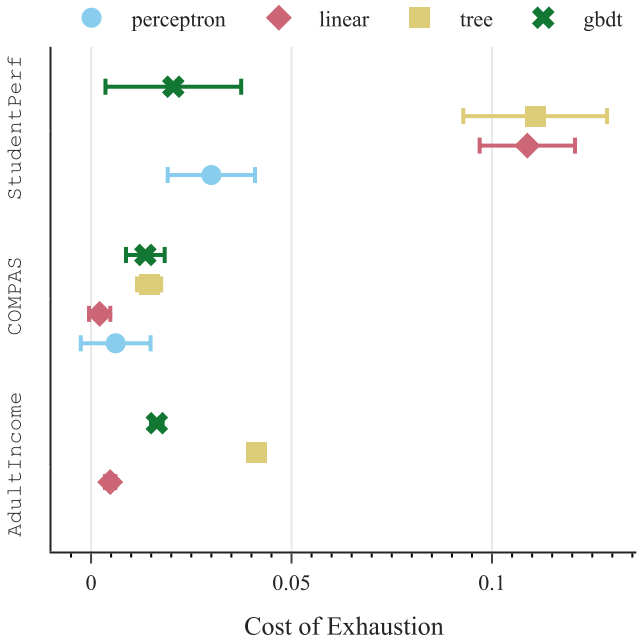


Figure 3.6: Distribution of the *cost of exhaustion* for the four model families (perceptron, linear, tree and GBDT) on the three considered datasets. The error bars show the 95% confidence interval on the values of the difference of $\text{Accuracy}_{\text{test}}$ for the best hypotheses in $\mathcal{H}^{\text{acc}}$ and $\mathcal{H}^{\mu}$. For all models, on all datasets (except for trees and linear models on StudentPerf), the cost of exhaustion is below 1%. Trees are the models with the highest cost of exhaustion, while for all the other models, the cost of exhaustion remains relatively low (in particular for the large capacities GDBTs), indicating a negligible accuracy cost for audit evasion.

ity. In the manipulation-proof framework of [1] we operate in, the model provider chooses the hypothesis class without constraints before disclosing it to the auditor. Therefore, when choosing a specific model family $\mathbb{H}$, a malicious model provider would have the possibility to trade performance (i.e. generalization capability) with the ability to attempt audit evasion. To understand the trade-offs involved in balancing these two objectives, we introduce the notion of $\text{CostOfExhaustion}(\mathbb{H})$ of a model family $\mathbb{H}$.

A model family $\mathbb{H} = \{\mathcal{H}_1, ..., \mathcal{H}_F\}$ is a set of hypothesis classes. The family $\mathbb{H}$ of decision trees where each hypothesis class $\mathcal{H}_i$ corresponds to a maximum depth value i is an example of model family. To define the CostOfExhaustion metric, we first introduce two particular hypothesis ($\mathcal{H}^{\text{acc}}$ and $\mathcal{H}^{\mu}$) classes of $\mathbb{H}$. $\mathcal{H}^{\text{acc}}$ is the hypothesis class in $\mathbb{H}$ with the best trained test accuracy:

$$\mathcal{H}^{\text{acc}} = \arg\max_{\mathcal{H}\in\mathbb{H}} \max_{h\in\mathcal{H}} \text{Accuracy}_{\text{test}(h,\mathcal{X})}. \tag{3.10}$$

Assuming that an honest model provider chooses its hypothesis class based on generalization capabilities, $\mathcal{H}^{\text{acc}}$ is the hypothesis class an honest model provider would actually choose. Then, define the hypothesis class in $\mathbb{H}$ with the largest manipulability (for a fixed audit budget s):

$$\mathcal{H}^{\mu} = \arg\max_{\mathcal{H}\in\mathbb{H}} \text{Manipulability}(\mathcal{H}, s). \tag{3.11}$$

Should a model provider try to escape audits at a low cost, they would try to find a hypothesis class whose optimal hypothesis h^* leads to a high μ-diameter. Thus, the cost of exhaustion is the accuracy cost of using the hypothesis class $\mathcal{H}^{\mu}$ compared to using $\mathcal{H}^{\text{acc}}$:

$$\begin{aligned}\text{CostOfExhaustion}(\mathbb{H}) = &\max_{h\in\mathcal{H}^{\text{acc}}} \text{Accuracy}_{\text{test}(h,\mathcal{X})} \\ &- \max_{h\in\mathcal{H}^{\mu}} \text{Accuracy}_{\text{test}(h,\mathcal{X})}\end{aligned} \tag{3.12}$$

The cost of exhaustion is plotted in Figure 3.6, for the four model families already considered, on the three datasets. The error bars show the 95% confidence interval on the values of the difference of $\text{Accuracy}_{\text{test}}$ for the best hypotheses in $\mathcal{H}^{\text{acc}}$ and $\mathcal{H}^{\mu}$. For all models, on all considered datasets (except for trees and linear models on the dataset StudentPerf), the cost of exhaustion is below 1%. Trees are the models with the highest cost of exhaustion. In fact, as we observed in Figure 3.5, given enough capacity, trees can reach the maximum manipulability under random audits. Yet, it is known that without regularization, complex trees can easily overfit the training data, thus lowering the max test accuracy of the $\mathcal{H}^{\mu}$ class compared to the max test accuracy of $\mathcal{H}^{\text{acc}}$. On the other hand, the models with the lowest cost of exhaustion (except on StudentPerf) are linear models. As observed on Figure 3.3, for all datasets, linear models span a small portion of the capacity range (around .1 points for StudentPerf and less than .01 points for COMPAS and AdultIncome), compared to larger models (e.g. GBDTs) which cover almost the entire capacity range on StudentPerf and AdultIncome. This result is challenging for the existence of efficient audits in the manipulation-proof framework. In fact, the witnessed low cost of exhaustion for larger capacity models indicates that platforms may evade audits at the cost of a minor loss in accuracy.

6.5. Effects of the audit set size

In this section, we experiment with different sizes of audit dataset and show that our conclusions do not change with the change in dataset size (we had $|S| = .1|\mathcal{X}|$ in previous experiments). To do so, we select three different hypotheses classes for each model family. We choose the hypothesis class that generalizes best $\mathcal{H}_{\text{opt}}$, the hypothesis class with the lowest capacity $\mathcal{H}_{-}$ and with the highest capacity $\mathcal{H}_{+}$. In Figure 3.7 we show the audit manipulability of each hypothesis class against the

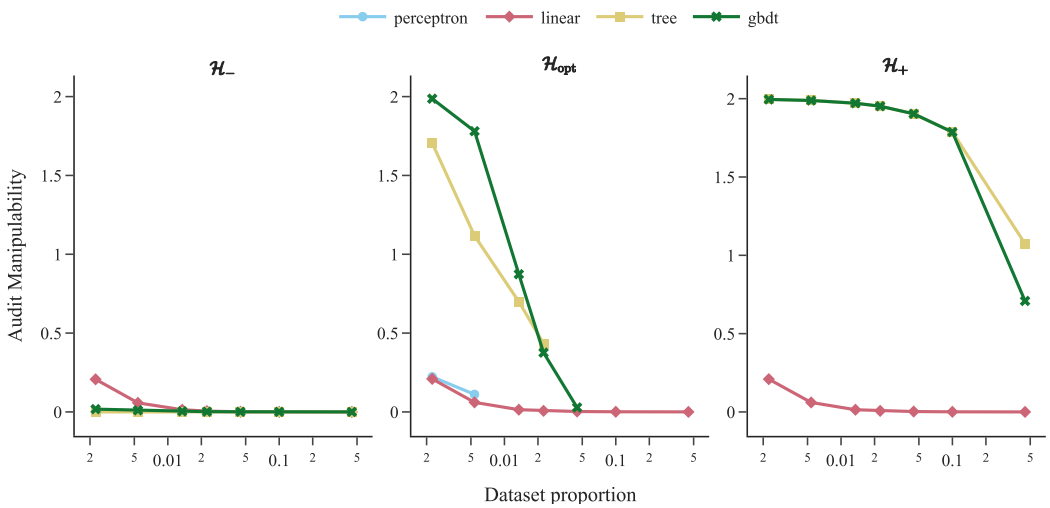


Figure 3.7: Evolution of the μ-diameter with the size of the audit set S represented as a proportion of the total dataset size for the AdultIncome dataset. Each line represents an audited model, whose hyperparameters are either tuned for the best generalization, either tuned for the highest capacity or tuned for the lowest capacity. For each (model, hyperparameter) couple, the μ-diameter is averaged over 15 audit datasets.

size of the audit dataset $|S|$. The results indicate that there is no significant inversion of the manipulability under random audits between the various hypotheses in the range of interest. Results in Figure 3.7 are shown only for the AdultIncome dataset. T he results for the other datasets are showed in Appendix F, in Figure 6.1 and 6.2, which both lead to the same conclusion.

7. Conclusion and discussions

The introduction of the *manipulation-proofness* framework [1] has certainly been an important step for auditors to start understanding that algorithmic audits can suffer from model provider manipulations and what cost that brings along.

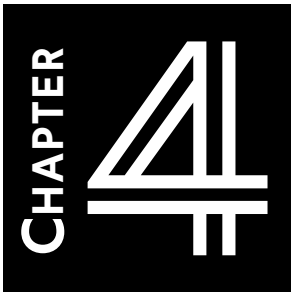
In this work, we conducted a thorough exploration of the concept of manipulation-proofness. We derived theoretical conditions on the hypothesis class implemented by the model provider for the impossibility of efficient manipulation-proof audits. We carried out a thorough experimental validation on the *manipulability under random audits* of state-of-the-art models for tabular data. Our results draw a connection between the capacity of the audited model and the manipulability of the audit task.

We now discuss some countermeasures to improve the audit robustness. A promising line of work is to require platforms to provide certificates. Since the goal of certificates is to provide a cheap verification procedure (at the cost of a

potentially high certificate generation cost), this would shift the computational burden to the model provider. One example of a fairness certificate was provided in [117]. Such extended assumptions (over mere black-box audits) are certainly an interesting research line for future works.

In the end, when implementing large-capacity models, a model provider can always game the audit without sacrificing too much accuracy. We believe that this demonstrates the limitations of black-box auditing for regulation, even when the hypothesis class used by the model provider is known to the regulator. We claim that regulators should be given more than black-box access to AI models as part of the audit procedure or that they should explore certification-based audits such as [117]. Therefore, we urge the community to participate in the search for audit frameworks that are both exploitable in practice and also supported by theoretical guarantees.

Efficiently monitoring model changes



> Pour saisir le monde d'aujourd'hui, nous usons d'un langage qu fut établi pour le monde d'hier. Et la vie du passé nous semble mieux répondre à notre nature pour la seule raison qu'elle répond mieux à notre langage.
>
> — Terre des hommes, Antoine de Saint-Exupéry

In the two previous chapters, the goal was to prevent manipulations during the audit by using some prior information available to the auditor. In this chapter, I take a different approach and explore methods to detect model manipulations after the audit. In this setting, the auditor can use any audit method to measure the metrics they are interested in, and then periodically check that the predictor they observed has not changed too much after the audit. One technique that can be used for model change detection is *model fingerprinting*.

Similarly to how image fingerprints can analyze the provenance of a picture by identifying artefacts due to the compression scheme, the specific sensor technology, or even the up-scaling method [163], model fingerprints analyze the outputs of a ML model h to extract artefacts that are characteristic of h itself. Model fingerprints were originally introduced to verify model provenance [164]. A model owner would extract a unique representation Z_h, the fingerprint, from the output of their model h. Later, he fingerprint Z_h would be compared with the fingerprint $Z_{h'}$ extracted from another model h', which is suspected to be a stolen copy of h. Finally, based on the comparison between Z_h and $Z_{h'}$, the model owner would decide if they need to flag h' as stolen and take further action.

The initial goal of this work was to select the model fingerprint most adapted to our auditing use-case. However, this chapter instead presents a surprising artefact of fingerprinting evaluation. Fingerprinting evaluation consists in generating *positive* and *negative* model pairs (h, h'), where positive model pairs consist in

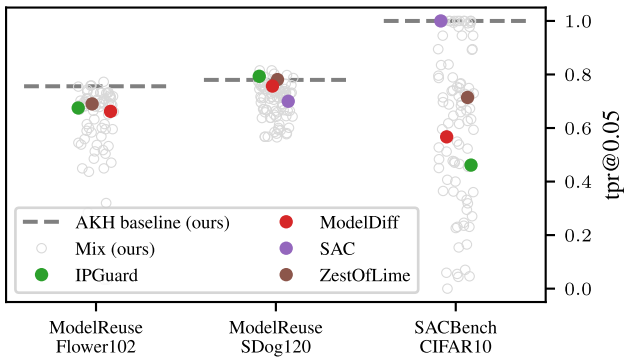


Figure 4.1: The TPR@5% of most of the fingerprinting schemes proposed in the literature is at best as good as the simple baseline we introduce. Each colored dot represents the performance of an existing fingerprinting scheme evaluated on a given benchmark. The gray dots are fingerprinting schemes we created using our Query, Representation and Detection (QuRD) decomposition.

a victim model h and a model h' stolen from h (e.g. through model extraction), while for negative model pairs, h and h' are totally unrelated (e.g. trained on a different dataset). A collection of such positive and negative pairs is called *benchmark*. Figure 4.1 displays the True Positive Rate (TPR@5%, see Paragraph *Fingerprint evaluation* for the exact definition) of existing fingerprints on two existing benchmarks, ModelReuse [165] and SACBench [166]. Figure 4.1 demonstrates that *the simple baseline that we introduce (gray dashed lines) performs on par with existing state-of-the-art fingerprinting schemes (coloured dots), which are much more complex.* Thus, in this work, we seek to understand the reasons behind this result from two angles:

> Why does a simple baseline match complex fingerprints performance on existing benchmarks?

As in most of the model fingerprinting literature, we consider image classification models. Note that, contrary to model watermarking methods, fingerprinting does not provide any theoretical guarantees on the false alarm rate (e.g. false positives). Thus, a strong empirical evaluation of model fingerprinting schemes is paramount to ensure their empirical soundness. Our contributions will be the following.

1. We introduce a simple yet powerful baseline and provide theoretical guarantees on its performance. Albeit on a simple model copy detection task, this constitutes the first theoretical analysis of the guarantees of a model fingerprinting scheme.
2. We survey and compare existing fingerprinting schemes for classification tasks. Our novel queries-representation-calibration decomposition (hereafter

we coin QuRD) enables us to systematize and thus uncover new and unexplored fingerprinting schemes. The novelty of QuRD lies in its mix of geometrical (distance between fingerprints leads to distance between models) and statistical insights (the fingerprint is then used to perform a statistical property test).

3. We compare existing benchmarks and investigate their differences in both the way the pair of test models (h, h') are generated and the distinguishability of the victim h and suspected h' models. Our work constitutes the first systematic comparison of classifier fingerprinting benchmarks, and reveals insights into how to build more informative and challenging benchmarks. All the code required to re-run our experiments, implement new benchmarks and evaluate new fingerprints is available online.[7].

1. Background and Setting

Stealing ML models The possibilities for an adversary to steal a given model are endless. They could break into the infrastructure of their victim [167], perform black-box model extraction attacks [168, 169] or just use the output of the victim's model to train their own. In this work, we consider adversaries seeking to steal the functionality of the victim's model.

Detecting IP violation via model fingerprinting The dominant approach to model fingerprinting is based on comparing the outputs of models on adversarial queries, as in AFA [170], TAFA [171], IPGuard [172], ModelDiff [165], FUAP [173], FCAE [174], DeepFoolF [175], and DeepJudge [176]. Other approaches leverage the sensitivity of ML models at random points sampled from the train set (e.g. SSF [164], ModelGif [177]), some explanations generated from the victim model h ZestOfLIME [178] or even train classifiers to distinguish stolen from benign model MetaV [179]. Some other works explore the use of natural images (images in the training/validation set) to craft their query set S, as in FBI [180] or SAC [166]. All of these works try to detect model stealing, however comparison among them and the assumptions they make are rarely taken into consideration. In this work, we introduce a framework to compare and evaluate these fingerprints.

Problem setting Consider an input space $\mathcal{X}$, a space of labels $\mathcal{Y} = \{1, ..., C\}$ with C classes, a data distribution $\mathcal{D}$ on $\mathcal{X}$ and a ground truth concept $c \in \{1, ..., C\}^{\mathcal{X}}$. A first party called the *victim* trains a model h on a classification task $\mathcal{C}$, then deploys this model in production. A second party called the *adversary* wishes to recreate a model h' that is close to identical to h ($h' \approx h$) to deploy it at a low cost. The task of checking whether a *suspected model* h' is a copy of the *victim model* h is modeled as a property test [181].

7. https://github.com/grodino/QuRD

Definition 4.1 (Model fingerprint) *A model fingerprint $\mathcal{T}$ is a (randomized) algorithm that takes two models h and h' as input and returns* 1 *with high probability if h' is stolen from h,* 0 *else.*

$$\begin{cases} \text{if } \; h = h' \text{ then } \; \mathbb{P}(\mathcal{T}(h, h') = 1) > \frac{2}{3} \; \textit{Copied model !} \\ \text{if } \; h \neq h' \text{ then } \; \mathbb{P}(\mathcal{T}(h, h') = 0) > \frac{2}{3} \; \textit{Just an other model} \end{cases}$$

The fingerprint (a.k.a. the property test) should be *effective*, *robust* and *unique*. We also require the fingerprint to be *efficient* in terms of queries and samples.

1. *Effectiveness*: if $h' = h$, then the suspected model is flagged by the victim with high probability.
2. *Robustness*: if h' is a slightly modified version of h (via fine-tuning, pruning, model extraction ...), then the suspected model should still be flagged.
3. *Uniqueness*: Original models $h' \neq h$ are not flagged.
4. *Efficiency*: the test uses few queries to the suspected model h' and few samples x from the data distribution.

Accessibility of data and models The type of fingerprinting scheme that can be used by the victim depends on the access the victim has to the suspected model h'. We will assume that the victim can freely query the suspected model h'. Yet, the output of the suspected model will range from label-only query access, to top-K labels query access, probits or logits query access and even to gradients query access. Following the fingerprinting literature, it is assumed that the victim has full access to its training data and model h.

2. Filling the gaps with the AKH baseline

The first contribution of this paper is the proposal and analysis of a simple yet powerful baseline, which, as we observed in Figure 4.1 performs at least as well as State-Of-the-Art fingerprinting schemes.

It is assumed that the victim has access to samples from the input distribution, for example the test set they used to validate their model. The baseline refers to Tolstoy's Anna Karenina principle that states "All happy families are alike; each unhappy family is unhappy in its own way". Thus, instead of using random samples for the input space $\mathcal{X}$, we look for points that are mis-classified by h and compare the victim and suspected models on those points. Our baseline, coined the *Anna Karenina Heuristic* (AKH, Figure 4.2), proceeds as follows. First, the victim chooses a negative input: a point $x \sim \mathcal{D}$ such that h wrongly classifies x: $h(x) \neq c(x)$. We write $\overline{\mathcal{D}_h}$ the resulting negative inputs distribution. Then, the victim queries the suspected model h' on x. Finally, if $h'(x) = h(x)$ the suspected model h' is flagged as stolen, otherwise h' is deemed benign.

Requires Sampling access to $\mathcal{D}$, white-box h, black-box h'

Draw $x \sim \overline{\mathcal{D}}_h$ (distribution on $\mathcal{X}$ such that $h(x) \neq c(x)$)

If $h(x) = h'(x)$

  Return 1 (`Stolen`)

Else Return 0 (`Benign`)

Figure 4.2: The proposed baseline, AKH.

Proposition 4.1 (AKH guarantees) *Consider $h, h' \in \mathcal{Y}^{\mathcal{X}}$ two models and $\alpha = \mathbb{P}(h(x) = c(x))$ (resp. $\alpha' = \mathbb{P}(h'(x) = c(x))$) their accuracy. Let $\delta = d_{H(h,h')}$ be the relative Hamming distance between h and h' and $\delta_C = \mathbb{P}(h(x) \neq h'(x) \mid h(x) \neq c(x))$. The property test $\mathcal{T}_b$ defined by AKH enjoys the following guarantees:*

$$\text{If } \; h = h', \; \mathbb{P}(\mathcal{D})[\mathcal{T}_b(h, h') = 1] = 1$$

$$\text{If } \; h \neq h', \; \mathbb{P}(\mathcal{D})[\mathcal{T}_b(h, h') = 0] = \delta_C \geq \frac{\delta - (1 - \alpha')}{1 - \alpha}$$

The proof of Proposition 4.1 is deferred to Appendix G. Proposition 4.1 establishes that AKH is a one-sided error test. Thus, in the favorable scenario where h' is copied (i.e. not tampered with), $\mathcal{T}_b$ will always detect it. To simplify the analysis, we defined AKH using only one query to the suspected model. To further decrease the False Negative Rate, one should run the baseline multiple times. A majority vote among the values returned by $\mathcal{T}_b$ decreases the False Negative Rate exponentially [181]. If instead of selecting negative examples (points $x \in \mathcal{X}$ that are wrongly classified by h), the victim was to use random samples according to $\mathcal{D}$, the test would still have a one-sided error but the True Negative Rate $\mathbb{P}(\mathcal{D})[\mathcal{T}(h, h') = 0]$ would be equal to the hamming distance δ between h and h'. This gives us an idea on when AKH can outperform schemes based on random sampling: either when the error rate $1 - \alpha$ of the victim model h is low or when the error rate $1 - \alpha'$ of the suspected classifier h' is low compared to $1 - \alpha$.

The experimental TPR@5% of AKH is displayed in Figure 4.1 in gray dashed lines. On ModelReuse (SDog120 dataset) and on SACBench, AKH performs on par with the best existing fingerprints. On ModelReuse (Flower102 dataset), AKH even performs better than the best existing fingerprints. In the two following sections we explore the reasons behind this observation by looking at the two players of Figure 4.1: the fingerprints and the benchmarks used to compare them.

3. Query, Representation & Detection: the QuRD framework

The literature on model fingerprinting does not provide a unified definition of model stealing detection. Most works focus on particular transformations of the stolen model, which they seek to detect. Only a few works [172, 173, 180] are based on a rigorous formulation of the problem. Some fingerprinting schemes (e.g. ZestOfLIME or ModelGif) are described from a geometrical point of view: the goal is to create a distance between models to distinguish stolen models from unrelated models. On the other hand, some works are described from a statistical point of view: the goal is to test whether $h' = h$ or not. Thus, comparing and categorizing existing fingerprints is not trivial. As a second contribution to this paper, we propose an original decomposition of the existing (and future) fingerprinting schemes into three core components:

1. **Query Sampling**, which generates the query set $S \subset \mathcal{X}$ on which to query h and h', e.g. selecting a subset of the victim model training set h.
2. **Representation**, which computes a compact representation $Z_h = g(Y_h)$ and $Z_{h'} = g(Y_{h'})$ of the answers $Y_h = \{h(x) : x \in S\}$ and $Y_{h'} = \{h'(x) : x \in S\}$ that are returned by the two models h and h' on the sample S. A basic strategy is to use the raw answers as a representation, that is, $Z_h = Y_h, Z_{h'} = Y_{h'}$.
3. **Detection**, which uses the two fingerprints Z_h and $Z_{h'}$, and possibly a set of calibration fingerprints $\{Z_i\}_i$, to decide whether h' is a stolen version of h or not.

3.1. Query Sampling (Q)

Existing approaches use four main techniques to build the query set when generating fingerprints, *Uniform sampling*, *Adversarial sampling*, *Negative sampling*, and *Subsampling* (see Table 4.1). Query Sampling (Q) methods are based on the transformation of a seed query set S_{seed}, which is either the training set or the test set used by the victim when generating h (both assumed to follow the same data distribution $\mathcal{D}$), or images composed of random pixel values.

Uniform sampling The easiest way to generate S is to sample uniformly from the data distribution or from a seed set $S_{\text{seed}} \subset \mathcal{X}$.

$$S \sim \mathcal{D} \quad \text{or} \quad S \sim \mathcal{U}(S_{\text{seed}}) \tag{4.1}$$

Adversarial sampling Adversarial sampling exploits the intuition that models tend to be characterized by their decision-boundary [165, 172, 182]. Compared to uniform sampling, adversarial sampling leads to a better detection rate for a lower query budget s. Starting from a set of seed inputs $S_{\text{seed}} \subset \mathcal{X}$, adversarial sampling

Seed set S_{seed}	**Uniform**	**Adversarial**	**Negative**	**Subsampling**	**Joint detector training**
input space	∅	IPGuard	∅	∅	MetaV
test set	∅	DeepJudge, FCAE	FBI	∅	∅
train set	ModelGif	ModelDiff, FUAP, IPGuard, AFA, ModelGif, DeepJudge, SSF[1]	**SAC**	**ZestOfLIME**, **SAC**	FUAP

Table 4.1: Type of seed set S_{seed} (rows), Query Sampling (Q) (columns), model access (emphasis) and Representation (R) (decorations) used. Adversarial sampling dominates the fingerprinting literature. Fingerprinting scheme appearing in multiple cells either require or can accommodate both Sampling/seed types. The text decoration stands for the access required to the remote suspected model h': no decoration = label access, underline = probits access, dashed underline = label or probit access, wavy underline = gradients access. The text emphasis indicate the type of Representation: no emphasis = raw model outputs, *italicized* = pairwise representation, **bold** = listwise representation. [1]SSF actually uses sensitive samples instead of adversarial samples.

computes a set of samples S_{adv}, targeted or not, using the following optimization procedure.

$$S_{\text{adv}} = \left\{ \arg\max_{u, \|x-u\|<\varepsilon} d(h(x), h(u)), x \in S_{\text{seed}} \right\} \tag{4.2}$$

Common methods used for solving Equation (4.2) include Projected Gradient Descent [183] or DeepFool [184]. Finally, the final query set is the concatenation of the seed and adversarial samples $S = (S_{\text{seed}}, S_{\text{adv}})$.

Negative sampling As for adversarial sampling, negative sampling [166] enjoys better detection rates for a given query budget. However, it does not need to compute gradients of h, it just needs query access to h, which can dramatically speed up the generation of the query set S. The core intuition follows that if h' makes the same mistakes as h, there is a high probability that the adversary stole h.

$$S \subset S_{\text{seed}} \quad \text{subject to} \quad \forall x \in S, h(x) \neq c(x) \tag{4.3}$$

Subsampling Subsampling exploits domain knowledge to create new samples $V(x) = \{x_j\}_j$ in the vicinity of a seed point x. Compared to negative and adversarial sampling, subsampling allows to create a large query-set with few samples from the data distribution.

$$S = \left(S_{\text{seed}}, \{V(x)\}_{x \in S_{\text{seed}}}\right). \tag{4.4}$$

[178] uses the super-pixel sampling technique of LIME [185] to generate images around each image in a seed set S_{seed}.

3.2. Representation (R)

Once the model h and h' have been queried on a sample of data points, the resulting outputs Y_h and $Y_{h'}$ must be recorded using some representation. We have identified three strategies in the literature: *Raw Labels/Logits*, *Pairwise correlation*, and *Listwise correlation*.

Raw labels/logits The simplest representation of the set of answers collected from the two models would be the set of answers themselves (labels or logits). However, depending on the way h' was constructed (or not) from h, different representations are more suitable.

$$Z_h = Y_h \in \left(\mathbb{R}^C\right)^s \text{ (logits) or } \{1, ..., C\}^s \text{ (labels)} \tag{4.5}$$

Pairwise correlation When the audit set S consists of pairs of samples (x, u) that have a specific meaning (e.g. u is an adversarial version of x as in ModelDiff), it is interesting to use these pairwise comparisons as the representation of the model.

$$Z_h = (d(h(x), h(u)))_{(x,u) \in S} \in \mathbb{R}^{\frac{s}{2}} \tag{4.6}$$

Listwise correlation Generalizing the idea of pairwise correlation, if the audit samples are not specifically paired but comparison is still meaningful, the victim can compute the similarity between all pairs of answers and use the resulting similarity matrix as representation. This is what is used by SAC.

$$Z_h = (d(h(x), h(u)))_{x \in S, u \in S} \in \mathbb{R}^{s \times s} \tag{4.7}$$

3.3. Detection (D)

Finally, once the victim has generated the fingerprints of their model and that of the suspected model (Z_h and $Z_{h'}$), the last step is to compare Z_h and $Z_{h'}$ to decide whether to flag h' or not.

There exists two approaches to Detection (D): directly compute a distance (e.g. hamming as in AFA or mutual information as in FBI) between the generated fingerprints or learn a classifier that takes the two fingerprints and outputs a theft probability score as in MetaV. In both cases, the victim needs access to its own pool of fingerprints from unrelated models $\mathcal{G} = \left\{G_1, ..., G_{|\mathcal{G}|}\right)$, to calibrate the detection threshold.

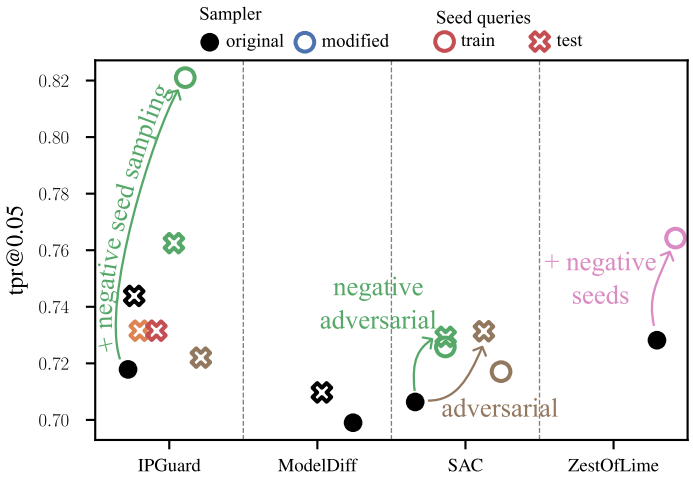


Figure 4.3: TPR@5% gains on ModelReuse obtained by modifying the sampler of existing fingerprints. The sampler can be modified in two ways: drawing seed queries from the train vs test set (materialized as circles vs crosses) or using a different queries sampler (materialized as a different color). Selecting negative seed inputs for adversarial generation instead of the original seeds can lead to improvements on the order of 10 points (+14%).

3.4. The next 100 fingerprints

In this subsection, we highlight the benefits of our novel QuRD decomposition for creating new and improved fingerprinting schemes and compare the existing fingerprints on a previously under-explored axis: the query budget.

Fingerprint evaluation The *Effectiveness*, *Robustness* and *Uniqueness* of fingerprints are evaluated by computing the Receiver-Operator Curve (ROC). The final Detection (D) step consists in threshing a distance or the output of a classifier based on the fingerprints Z_h and $Z_{h'}$. The ROC shows the relationship between the True Positive Rate (TPR), which is the proportion of positive pairs (h, h') that are flagged as positive by the fingerprint, and the False Positive Rate (FPR), which is the proportion of negative pairs (h, h') that are flagged as positive by the fingerprint. The Receiver-Operator Curve (ROC) captures the trade-off between the cost to the victim of missing a stolen model compared to the cost of wrongly flagging a model as stolen. Recognizing the high cost of False Positives for the victim, we will report the TPR such that the FPR is below a threshold of 5%: TPR@5%, averaged over 5 runs with independent random seeds.

Creating new fingerprints using the QuRD framework Following our QuRD framework, Table 4.1 categorizes exiting fingerprints (listed previously in Background and Setting). Table 4.1 shows that a large part of the literature focused on fingerprints based on adversarial sampling. Several QuRD combinations have not been explored yet by the literature. Moreover, the schemes always focus on using

	Stealing and obfuscation methods	ModelReuse Flower102	ModelReuse SDog120	SACBench CIFAR10
Model leak	**same**	✓	✓	✓
	quantize	✓	✓	×
	finetune	×	×	✓
	transfer	×	×	✓
	prune	✓	✓	✓
Model extraction	**probits**	✓	✓	✓
	label	✓	✓	✓
	adversarial (labels)	×	×	✓

Table 4.2: Stealing and obfuscation methods implemented by different benchmarks.

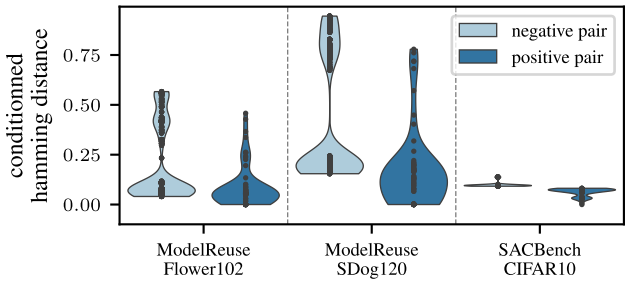


Figure 4.4: Distribution of the conditioned Hamming distance $d_{C(h,h')}$ between the models of each positive/negative (h, h') pair.

only one type of Query Sampling (Q) but very rarely explore chaining or mixing, e.g.using negative samples as the seeds for generating adversarial examples. Thus, to explore the space of QuRD combinations, we re-implemented the Query Sampler, Representation, and Detection of four existing fingerprints: ModelDiff, SAC, IPGuard and ZestOfLIME. We mixed them to create $\sim$ 100 new fingerprints. In Figure 4.1, gray-edged dots represent such QuRD combinations. Of course, not all new combinations are worth considering, as many QuRD combinations exhibit lower TPR@5% than existing fingerprints. Thus, in Figure 4.3 we show the potential improvements that can be reached by modifying the Query Sampler (Q) and/or the seed set S_{seed} of existing schemes on ModelReuse. Figure 4.3 shows that it is possible to increase the TPR@5% of IPGuard by 10 points (+14%) simply by choosing negative seed samples as the starting points for the generation of adversarial examples.

Comparing apples to apples: a focus on the query budget Although not displayed in Table 4.1, the query budget required by the existing fingerprints

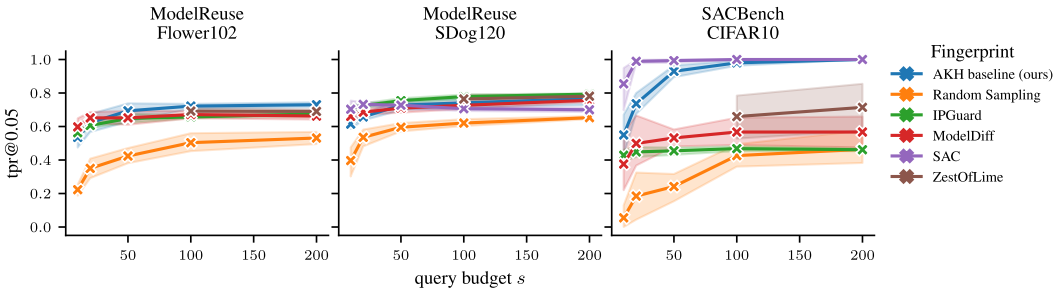


Figure 4.5: The effect of the query budget s on the *Efficiency* and *Robustness* of existing fingerprints, as measured by TPR@5%.

can vary greatly. For example, ZestOfLIME requires from 1000 to 128000 queries while FBI only requires ~ 100 queries to reach the advertised performance. In Figure 4.5 we show the TPR@5% of existing fingerprints along our AKH baseline and selected QuRD variations. Keeping a small query budget is of paramount importance, mainly to remain stealthy against potential defenses [186], but also to avoid disrupting the remote service with (tens to hundreds of) thousands of queries. Once more, we observe that fingerprints based on negative sampling equal or outperform fingerprints based on adversarial sampling. From 0 to 100 queries for SACBench and 0 to 50 for ModelReuse, most fingerprints exhibit notable improvements at each query budget increment. After 100 (or 50) queries, most fingerprints show a plateau. Thus, it appears that there exists an optimal query budget, dependent on the benchmark but not on the fingerprinting scheme. Finally, schemes based on negative sampling appear to suffer a lower variance than adversarial-based fingerprints, especially on SACBench.

Although the performance of most fingerprints plateau after 50-100 queries, the performance of some fingerprints (e.g. ModelDiff and SAC) suffers when the query budget increases from 100 to 400 queries. This phenomenon is observable only for schemes whose representations are based on a pairwise or a listwise comparison. We believe that when the number of query points is increased, the self-correlation increases regardless of the fact that a pair is positive or negative. Thus, the gap between the positive pair distance and the negative pair distance decreases with budget, which in turn decreases the performance of the fingerprint.

4. Fingerprinting benchmarks

Because there are no strong guarantees regarding *Effectiveness* and *Robustness* of fingerprinting schemes, proper empirical evaluation is critical to assessing their performance. The main difficulty of evaluation lies in the definition (and implementation) of realistic *positive* ($h' = h$) and *negative* ($h' \neq h$) model pairs. To do this, we need to separate how the adversary steals the model (how to achieve $h' =$

	Model leak				Probit extraction		Label extraction	
Fingerprint	**same**	**quantize**	**finetune**	**transfer**	**prune**	**vanilla**	**vanilla**	**adversarial**
IPGuard	**1.0** ± 0	**1.0** ± 0	**1.0** ± 0	**1.0** ± 0	**0.94** ± .01	0.64 ± .02	0.12 ± 0	0.02 ± .01
ModelDiff	**1.0** ± 0	**1.0** ± 0	**1.0** ± 0	**1.0** ± 0	**0.94** ± .01	0.59 ± .05	0.14 ± .02	0.16 ± .07
Random	**1.0** ± 0	0.93 ± .03	**1.0** ± 0	0.48 ± .2	0.71 ± .02	0.46 ± .01	0.07 ± .02	0.06 ± .05
SAC	**1.0** ± 0	**1.0** ± 0	**1.0** ± 0	**1.0** ± 0	0.92 ± 0	**0.81** ± 0	**0.59** ± .02	**0** ± 0
ZestOfLime	**1.0** ± 0	**1.0** ± 0	**1.0** ± 0	0.78 ± .17	0.86 ± 0	0.74 ± .02	0.38 ± .05	0.29 ± .11
AKH (ours)	**1.0** ± 0	**1.0** ± 0	**1.0** ± 0	**1.0** ± 0	0.91 ± .01	0.78 ± .01	0.46 ± .01	0.92 ± .03

Table 4.3: TPR@0.05 of the existing fingerprints with a budget of 100 queries. For each task, the best performance are highlighted.

h) and how the adversary tries to conceal their theft by modifying the stolen model to avoid detection by the victim).

Stealing a model **1) Model leak**: the adversary directly steals the architecture and weights of the model h and uses them to solve the same task. This can happen via an internal leak [187] or an attack on the company infrastructure [167]. **2) (Adversarial) model extraction** The adversary only has query access to the source model and trains their model based on the probits or the labels of the source model. The model extraction can either be probits or labels-based [168, 169]. In addition, depending on the threat model, the architecture trained by the attacker is not always the same as the victim model h and the adversary might not have access to samples from the input domain [169].

Stolen model obfuscation Once an attacker has stolen the model h, they will try obfuscating their model to hide their theft. To avoid detection by model fingerprinting, the adversary may act on a combination of three aspects of the model inference process. **1) Model/weights tampering** As first approach, the adversary can directly modify the model itself to remove potential watermarks embedded in the weights of the model: weights pruning [188, 189], model quantization and finetuning or transferring the model to a small private dataset [165]. **2) Input modifications** The second concealment trick is to apply transformations to the inputs fed to the model to limit the effect of adversarial inputs [180]: JPEG compression, equalization, or posterization. **3) Output noise**: Finally, to avoid giving away too much information, the adversary can try to slightly alter the outputs of the model, e.g. returning only the Top-K labels, averaging the outputs over a neighbourhood of the input [190] or implementing model-stealing defences [191, 192].

4.1. The majority of benchmarked tasks are solved

The performance shown previously in Figure 4.1, Figure 4.4, Figure 4.5 were all aggregated at a benchmark level. In this section, we separate the performance of the fingerprints with respect to the model-stealing and obfuscation methods. We will seek to answer the question **What type of stealing and obfuscation methods can be considered as resolved issues and, hence, on which ones should practitioners focus?** Positive pairs are grouped by task, i.e., how the copied model h' was created from h, along with their corresponding negative pairs. Each task corresponds to the combination of a stealing and an obfuscation method. This decomposition is especially interesting since, as we will observe, a large portion of the tasks are solved by all the fingerprints, while the rest, and more complicated tasks, allows to discriminate the different fingerprints much more clearly.

As for benchmark-aggregated performance discussed in the QuRD Section, Table 4.3 shows that AKH is on par or surpasses all the previously introduced schemes. More interestingly, Table 4.3 reveals that a large part of the tasks considered by ModelReuse and SACBench (namely the same, quantization, finetuning, and transfer tasks) are completely solved by existing fingerprints, as well as by AKH. The remaining unsolved tasks consist of model stealing by model extraction, using no obfuscation attempts. Surprisingly, adversarial label extraction is easily detected by fingerprints based on negative sampling but not by adversarial, random, or subsampling-based fingerprints. Model extraction detection is, thus, a hard subtask of model stealing detection.

The results of Table 4.3 highlight an issue with the current benchmarks: trying to detect if a suspected model h' is the same as the victim's h up to small model perturbations (pruning, quantization, etc.) is fundamentally different from detecting model extraction. These two objectives differ in difficulty to be detected (as we mentioned earlier), but they also differ greatly in the efforts the adversary has to consent to reach the same accuracy.

4.2. Why does SACBench look so easy?

As we observed in Figure 4.1, the performance of fingerprints varies greatly from one benchmark to another. In this section, we try to uncover the reasons for this variability. A fingerprinting benchmark is essentially a procedure to generate positive and negative model pairs (h, h') by varying the model stealing and obfuscation methods. In the following, we investigate the properties of positive and negative pairs for each benchmark, in order to better understand the reasons why the various benchmarks seem to be unable to discriminate proposed fingerprint schemes and are beaten by the simple baseline presented in the previous section. ModelReuse and SACBench employ the same set of model stealing and obfusca-

tion methods with two exceptions: ModelReuse uses model quantization as an obfuscation strategy, while SACBench performs adversarial model extraction. This explains the inferior performance of fingerprints based on adversarial sampling (ModelDiff and IPGuard) on SACBench.

However, the slight choice difference of the stealing and obfuscation methods included in ModelReuse compared to SACBench does not explain the exceptional performance of AKH and SAC compared to the other fingerprints. To that end, in Figure 4.4 we show the value of the conditioned Hamming distance δ_C (see Proposition 4.1) for all model pairs (h, h'). We note that the variability of the distance between h and h' is much higher for ModelReuse than for SACBench. This indicates that SACBench's process for creating the positive and negative pairs may not introduce enough diversity in the generated models, which could lead to overestimating the performance of its fingerprints. However, as observed in Figure 4.1, except SAC, all fingerprints have a comparable TPR@5% on SACBench and ModelReuse. To explain the difference in performance of AKH and SAC, we need to consider the separation between the distribution of $\delta_{C(h,h')}$ for the positive and negative model pairs (h, h'). Figure 4.4 shows a better separation between $\delta(h, h')$ for positive and negative pairs in SACBench. On the other hand, both datasets of ModelReuse show a large overlap in the distributions of distances of positive and negative pairs. Thus, since SAC is based on negative sampling, it appears that the generated positive and negative pairs of SACBench are especially well suited to the SAC fingerprint they introduce.

5. Related works

Model-theft proactive defenses An alternative to fingerprinting is for the victim to choose a proactive solution consisting in *watermarking* their model (see, e.g., [193, 194] for an overview), or by defending it using defenses implemented at training or inference time [186, 191].

Connections with tampering detection A problem closely related to model fingerprinting is *tampering* detection. The goal is to detect if a model served by a platform is the intended model originally sent by the owner, or if the model has been tampered with [43, 164], by backdoor attacks [195] for instance.

Connections with interpretable model distance To debug model creation and to help ML audits, a body of work is interested in *interpretable* model distances. Instead giving a single distance value, it also gives an explanation such as domains on where the models differ the most [196] or a simple approximation of the difference of the two models [197].

6. Conclusion

Our systematic analysis of the existing model fingerprinting schemes and benchmarks revealed a concerning evaluation artifact: the benchmarks studied are either not discriminative or solved by our simple AKH baseline. Firstly, most tasks are solved with almost any fingerprint. Secondly, the created victim/stolen model pairs are too easy to distinguish from victim/benign model pairs. Moreover, our QuRD framework reveals that schemes based on adversarial sampling are brittle compared to schemes using natural images.

While some of the tasks of model stealing detection can now be considered solved, several open challenges remain. One key issue is ensuring the robustness of fingerprinting techniques against adaptive adversaries who may actively attempt to evade detection. Furthermore, the development of effective fingerprints for other modalities than images would require further exploration.

Conclusion

> Quand la terre claquera dans l'espace comme une noix sèche, nos œuvres n'ajouteront pas un atome à la poussière. [...] et dire que nous le savons et que notre orgueil s'acharne !
>
> — L'Œuvre, Émile Zola

In this manuscript, I studied black-box audits, in which the auditor only has query-access to the studied predictor. Aiming to find the minimal additional information required to derive guarantees in the case of a deceptive model provider, I introduced the notion of audit priors (Chapter 2 and 3) and suggested to use model fingerprinting techniques for efficient model monitoring (Chapter 4).

1. Summary of contributions

A common model provider defense to lobby against access to their data is that only the outputs of the predictor matter, for they are the only user-facing part of the system. Chapter 3 revealed however that without knowledge about the training data, even with full access to the hypothesis class of the predictor, audits are easily manipulated by deceptive model providers, especially as the capacity of the underlying model grows. Having access to (train, test, or external) data is thus necessary for robust audits. Yet, Chapter 2 exposed that, while necessary, this data access is not sufficient. Labeled data can reduce the magnitude of the manipulations that can remain undetectable. Unfortunately, the experiments exposed that model providers can still game audits by exploiting label noise inherent to a lot of prediction tasks involving tabular data. Finally, in Chapter 4, I explored how the new "vetted researcher" white-box access mandated by the DSA, could be used to improve external ML systems monitoring. Experiments revealed that proper evaluation and comparison of model change detection is subtle: it requires

to simulate the changes we want to detect in a realistic fashion. In the end, the performance of the baseline I introduced advocates for simple methods.

Going back to the four steps of AI audits –reconnaissance, systematization, measurement, and monitoring (Subsection 1.2.4)– the methods and tools presented in this manuscript provide *defense* mechanisms to the two last steps. The defenses worked by exploiting labeled data (Chapter 2), model structure (Chapter 3) or model weights (Chapter 4). All in all, **audits are a matter of information gain**.

2. Discussion

This manuscript exposed results for a specific type of predictor, namely binary classifiers[8], with a set of worst-case assumptions for the auditor: a single, easily manipulatable, audit metric and a requirement to detect the smallest model change possible. The applicability of the auditing framework presented in this manuscript can thus be broadened in the following directions.

Beyond binary classifiers Extending our manipulation defenses to the evaluation of other prediction tasks such as generative modelling or regression is a direct avenue for future work. In fact, it might even be easier than it seems because of how these systems are evaluated in practice. For example, the typical academic evaluation of text generation models consists in finding difficult tasks and reducing them to classification problems to calculate accuracy scores (e.g. MMLU [198], GSM1k [199] or MathQA [200]). Thus, since the evaluation is reduced to a (multi-class) classification task, it should possible to directly adapt the methods presented in this manuscript.

Multiple audit metrics The audit setting I considered was voluntarily simple: a single metric, measured by one auditor. To extend the results to a broader class of separation and sufficiency metrics (recall Definition 1.4), we can build on the insights of Chapter 2 and 3 that any fairness repair method can be transformed into a manipulation strategy to leverage the recent advances in fairness without demographics [201, 202]. Moreover, it was always implicitly assumed that the auditor only measured one metric. A direct follow-up would be to audit whether a model h is on the Pareto front of a set of metrics, for example that it satisfies a predefined tradeoff between accuracy, fairness and privacy.

Incentive-aware change detection The performance of our very simple baseline compared to much more complex methods in Chapter 4 revealed that the current evaluation of model change detection is too simplistic. Moreover, there is an issue of hyper-parameter tuning: all the methods require to tune a threshold by simulating the model changes that are expected. This means that there are

8. except in Chapter 4 where we also considered multi-class classifiers

no guarantees to detect model changes that have not been anticipated. The `QuRD` python fingerprint library I introduced in Chapter 4 is a good starting point to investigate these issues. Finally, when trying to verify if a model improved after an audit, not all changes need to be detected, only those that affect the audit metric. To that end, building on the simple AKH baseline, it would be possible to leverage influence functions [203] to further refine the selection of the fingerprint query set to the points that are most responsible for a change in the audit metric.

3. Perspectives

Concluding this manuscript, I would like to address some blind spots of ML auditing I encountered during my PhD: the user and its assumed passivity, the cost of auditing to our institutions and the growing impossibility to simply refuse to be subjected to AI decisions.

3.1. Users back in the governance loop

External audits are a way to delegate the estimation of the users' risk/utility tradeoff of a system to an entity (the auditor) with greater resources and expertise. My PhD work, and to a greater extent, the robust auditing literature, focuses on the auditor and primarily treats users as passive agents to protect, or merely as data providers, never as active agents that can participate in the audit process. Realizing that they can collectively have a great impact on the model through their data (used for training), the field of collective [204] action has initiated tools for users to influence the AI systems they use.

A first solution to bring users back to the governance loop are **reporting databases** [56]. Regulators or a mandated entity actively collect user reports from users and continuously tests for systematic issues or disparate treatment evidence. The key challenges with this approach are the same as online user reviews: the entity collecting the reports must handle fake users and the fact that users will have different harm threshold before reporting an issue. This problem is similar to the issues in online reviews. Thus, to tackle these issues, bringing tools from the field of robust sparse voting [205] might be a fruitful direction in defending against malicious users.

An other solution is to increase the collaboration between users and auditors. Users can share (potentially privatized) personal data to help the auditor create their audit dataset. Conversely, by transparently sharing the conclusions of their investigations, auditors can help users relate their experience to more systematic analyses of the systems they use. A promising direction to allow users or auditors aggregate these (potential noisy) sources of information, lies in the recent developments in the fields of **testing with (private) advice** [206] and **prediction-powered inference** [207]. The idea is to use the auxiliary, potentially unreliable

data, to decrease the variance of the audit test statistic, thus reducing the number of necessary audit queries. In light of the results in this manuscript, it would be also relevant to study the resistance of these methods to adversarial manipulations of the predictions by the model provider.

Finally, before implementing the proposed auditing strategies, it is necessary to study their impact on users, both as a path to accountability (do the added auditing guarantees draws a path to better accountability ?) and as a data release mechanism that could negatively impact those whose data will be used for the audit (e.g. delivery platform workers penalized for donating their data to journalists scrutinizing the platform).

3.2. Shifting the auditing cost to providers

In the theoretical literature on auditing, and testing in general, there is a strong emphasis on the query budget: the number of queries to issue to the model provider to achieve guarantees on what is measured. The reasons are two-fold: less queries means a stealthier audit and lower data acquisition costs. This formalization of auditing places the burden on the auditor to issue the queries, come-up with computational methods to minimize the query budget and design tests that are manipulation-proof, powerful, with a low False Positive Rate. Meanwhile, the model provider only has to operate as usual, albeit with a few additional queries from the auditor.

A promising avenue to shift the cost of ML governance to providers is the use of cryptographic primitives to go from a harm-detection based governance to a system certification scheme. While still early and computationally expensive, primitives based on **zero-knowledge proofs** [208] and **homomorphic encryption** [209] can certify some properties such as fairness or accuracy of a model whose verification only requires to check the model's signature. See Section 2.2 for examples of schemes applied to auditing. Therefore, instead of tasking auditors to monitor AI systems at the expense of taxpayers that might not even use said systems, regulators could force high-risk systems to implement these scheme and allow users to check the model signature to verify that they are using the right model. However, implementing this in practice is not yet feasible because of the computational toll of these cryptographic certification. Thus it would be fruitful to investigate what kind of metrics can cheaply be certified cryptographically and what kind of metrics require audit-type governance.

3.3. A broader outlook on algorithmic decision systems

Despite the apparent progress in the craft of ML predictors, the software systems we use every day are more and more concentrated in the hands of a few companies, feel arguably less and less useful and increasingly difficult to avoid.

First observed and theorized by Foucault [210] when studying how the current form of prison and surveillance techniques came to be, the growth of communication technology has resulted in an intensification of power relations. Since then, this link has also been observed in the history of the development of the internet [211], which enabled a much broader, un-targeted surveillance of the population, and in the recent developments of computer vision [212], which fuels surveillance and identification technologies. It turns out that this power is currently held in large parts by the tech companies providing the technologies we marvel at and the online platforms we use everyday.

Moreover, the phenomenon of *enshittification* suggests that the interests of those companies are not aligned with users'. Coined by Doctorow [213] enshittification describes the life-cycle of online platforms: "First, they are good to their users; then they abuse their users to make things better for their business customers; finally, they abuse those business customers to claw back all the value for themselves.". While it is the most visible part [214], online platforms are not the only victims, other sectors such as enterprise software or aviation [215] are also concerned. This would not be an issue if users and businesses could easily get off those platforms, but they have become so instrumental and pervasive in our everyday lives that we are often not given a choice, especially when incorporated into government services.

Therefore, beyond auditing methods, we need methods and tools to help users hold the platforms they (are sometimes forced to) use accountable, not only for their mistakes, but also for failing to take the documented harms they cause into proper consideration when they design the tradeoffs of their systems.

Appendix

A. Proof of Theorem 2.2

Notations

$\mathcal{H}$	Hypothesis class
$\mathcal{F}$	Set of fair models
$\mathcal{H}_a$	Set of expectable models
D_a^Y	Audit set ground truth
δ	Distance between the groundtruth and the set of expectable model
h_p	Original model of the model provider
h_m	Manipulated model of the model provider
$\mathcal{X}$	Input space
$\mathcal{D}$	Data distribution
x	Sample from input space
$\mathcal{Y}$	Output space
y	Sample from output space
$\mathcal{A}$	Protected feature
z	Sample
n	dimension of $\mathcal{Z}$

Let $D_a = \left(D_a^X, D_a^Y, D_a^G\right) \subset \mathcal{X} \times \mathcal{Y} \times \mathcal{G}$ be the audit dataset used to create the labeled-dataset prior (as in Definition 2.3). For the two groups $0, 1 \in \mathcal{G}$, define $G_i = \{x, y : (x, y, g) \in D_a \text{ and } g = i\}$. Let $\mathcal{Z} = \mathcal{Y}^{\mathcal{X}}|_{D_a^X}$ be the restriction of the set of models to the audit set. For any model h restricted to D_a^X, the (relaxed) demographic parity gap is $\mu(h, D_a) = \frac{1}{|G_1|}\sum_{x,y \in G_1} h(x) - \frac{1}{|G_0|}\sum_{x,y G_0} h(x)$. If h outputs labels, then the μ is the demographic parity gap, otherwise it is the relaxed demographic parity gap. Recall that $\mathcal{F} = \{h \in \mathcal{Z} : \mu(h, D_a) = 0\}$

As $\mathcal{F}$ is the kernel of the linear transformation μ, $\mathcal{F}$ is a hyperplane of $\mathcal{Z}$. As $\mathcal{F}$ is a hyperplane of $\mathcal{Z}$ of finite dimension, it is closed in $\mathcal{Z}$.

Having an hyperplane lead to the natural definition of (hyper)cylinder, that we use in the following theorem.

Definition 6.1 (Right cylinder) *A right cylinder $C(H, B)$ is the set of all points whose orthographic projection on a hyperplane H lies in a set B with B a subset of the boundary of H. B is called the base of the cylinder.*

Theorem 6.1 (Detection probability, general) *The probability P_d that the auditor correctly detects a malicious model provider trying to be fair is $\mathbb{P}(\mathcal{H}_a \setminus C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F}) \mid \mathcal{H}_a)$.*

Proof (Theorem 6.1) The auditor correctly detects a malicious model provider trying to be fair if and only if the manipulated model is fair but not expectable. The manipulated model is fair but not expectable if and only if the orthographic projection ${h_m}^*$ of h_p in $\mathcal{F}$ is not in $\mathcal{H}_a \cap \partial\mathcal{F}$. Thus, the manipulated model is fair but not expectable if and only if $h_p \notin C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F})$ (following Definition Definition 6.1). As by assumption $h_p \in \mathcal{H}_a$ (Equation (2.2)), it means that $h_p \in \mathcal{H}_a \setminus C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F})$. ∎

We now restate Theorem 2.2 and prove it.

Theorem 6.2 (Prior-Uniform detection rate) *Under the dataset prior of definition Definition 2.3 with $l(h, x, y) = \ell_2(h(x) - y)$ the ℓ_2 norm, and the* uninformative prior *assumption, the probability that the auditor correctly detects a malicious model provider trying to be fair is*

$$P_d = 1 - \frac{1}{W_n}\left(\int_0^{\arccos\left(\frac{\delta}{\tau}\right)} \sin^n(\theta)d\theta - \frac{\delta}{\tau}\left(1 - \frac{\delta^2}{\tau^2}\right)^{\frac{n-1}{2}}\right).$$

with D_a^Y the labels of the samples in D_a, $\delta = d(D_a^Y, \mathcal{F})$, the distance of D_a^Y to $\mathcal{F}$ and W_n the n-term of Wallis' integrals.

Proof (Theorem 2.2) As established in Theorem 6.1, $P_d = P(\mathcal{H}_a \setminus C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F}) \mid \mathcal{H}_a)$.

The probability $P(\mathcal{H}_a \setminus C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F}) \mid \mathcal{H}_a)$ is the probability to be in the ball $\mathcal{H}_a$ without the probability to be in the intersection between the ball $\mathcal{H}_a$ and the cylinder $C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F})$. In the following, we denote $V_{\text{dashed}}(\tau, \delta, n)$ this quantity.

As $\mathcal{H}_a$ is a ball, its volume is:

$$V_{\text{ball}}(\tau, n) = \frac{\pi^{\frac{n}{2}}\tau^n}{\Gamma\left(\frac{n+2}{2}\right)}$$

with $\Gamma(z) = \int_0^{\inf} t^{z-1}e^{-t}\,\mathrm{d}t$ [216].

The volume of the intersection between the cylinder and the ball is the sum of the three following volumes:

- the solid cylinder with height between $-\delta$ and δ
- the spherical cap of $\mathcal{H}_a$ that is *above* the previous cylinder (i.e. the part of $\mathcal{H}_a$ with height between δ and τ)
- the spherical cap of $\mathcal{H}_a$ that is *bellow* the previous cylinder (i.e. the part of $\mathcal{H}_a$ with height between $-\delta$ and $-\tau$)

According to [217], the volume of each spherical cap is

$$V_{\text{cap}}(\tau, \delta, n) = \frac{\pi^{\frac{n-1}{2}}\tau^n}{\Gamma\left(\frac{n+1}{2}\right)} \int_0^{\arccos\left(\frac{\delta}{\tau}\right)} \sin^{n(\theta)}\, d\theta$$

And the volume of the cylinder of height 2δ is

$$V_{\text{cyl}}(\tau, \delta, n) = 2\delta V_{\text{ball}}\left(\sqrt{\tau^2 - \delta^2}, n-1\right)$$

Thus,

$$\begin{aligned} V_{\text{dashed}}(\tau, \delta, n) &= V_{\text{ball}}(\tau, n) - 2V_{\text{cap}}(\tau, \delta, n) - V_{\text{cyl}}(\tau, \delta, n) \\ &= \frac{\pi^{\frac{n}{2}}\tau^n}{\Gamma\left(\frac{n+2}{2}\right)} - 2\frac{\pi^{\frac{n-1}{2}}\tau^n}{\Gamma\left(\frac{n+1}{2}\right)} \int_0^{\arccos\left(\frac{\delta}{\tau}\right)} \sin^{n(\theta)}\, d\theta - 2\delta\frac{\pi^{\frac{n-1}{2}}\left(\sqrt{\tau^2 - \delta^2}\right)^{n-1}}{\Gamma\left(\frac{n+1}{2}\right)} \end{aligned}$$

According to Theorem 6.1, the probability that the auditor correctly detects a malicious model provider trying to be fair is $P(\mathcal{H}_a \setminus C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F}) \mid \mathcal{H}_a)$. That is to say, it is the ratio of $V_{\text{dashed}}(\tau, \delta, n)$ over $V_{\text{ball}}(\tau, n)$:

$$\begin{aligned} P_d &= P(\mathcal{H}_a \setminus C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F}) \mid \mathcal{H}_a) \\ &= \frac{V_{\text{dashed}}(\tau, \delta, n)}{V_{\text{ball}}(\tau, n)} \\ &= 1 - 2\frac{\Gamma\left(\frac{n+2}{2}\right)}{\Gamma\left(\frac{n+1}{2}\right)}\frac{\pi^{\frac{n-1}{2}}}{\pi^{\frac{n}{2}}} \int_0^{\arccos\left(\frac{\delta}{\tau}\right)} \sin^{n(\theta)}\, d\theta - 2\delta\frac{\left(\tau^2 - \delta^2\right)^{\frac{n-1}{2}}}{\tau^n}\frac{\Gamma\left(\frac{n+2}{2}\right)}{\Gamma\left(\frac{n+1}{2}\right)}\frac{\pi^{\frac{n-1}{2}}}{\pi^{\frac{n}{2}}} \\ &= 1 - \frac{2}{\sqrt{\pi}}\frac{\Gamma\left(\frac{n+2}{2}\right)}{\Gamma\left(\frac{n+1}{2}\right)} \int_0^{\arccos\left(\frac{\delta}{\tau}\right)} \sin^{n(\theta)}\, d\theta - \frac{2\delta}{\sqrt{\pi}}\frac{\left(\tau^2 - \delta^2\right)^{\frac{n-1}{2}}}{\tau^n}\frac{\Gamma\left(\frac{n+2}{2}\right)}{\Gamma\left(\frac{n+1}{2}\right)} \\ &= 1 - \frac{2}{\sqrt{\pi}}\frac{\Gamma\left(\frac{n+2}{2}\right)}{\Gamma\left(\frac{n+1}{2}\right)} \left(\int_0^{\arccos\left(\frac{\delta}{\tau}\right)} \sin^{n(\theta)}\, d\theta - \delta\frac{\left(\tau^2 - \delta^2\right)^{\frac{n-1}{2}}}{\tau^n} \right) \end{aligned}$$

The function Γ can be written with Wallis' integrals as: $W_n = \frac{\sqrt{\pi}}{2}\frac{\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{n+2}{2}\right)}$ with $\forall n, W_n = \int_0^{\frac{\pi}{2}} \sin^{n(\theta)}\, d\theta$.

In the other hand,

$$\delta\frac{\left(\tau^2-\delta^2\right)^{\frac{n-1}{2}}}{\tau^n} = \frac{\delta}{\tau}\frac{\left(\tau^2-\delta^2\right)^{\frac{n-1}{2}}}{\tau^{n-1}}$$
$$= \frac{\delta}{\tau}\left(\frac{\tau^2-\delta^2}{\tau^2}\right)^{\frac{n-1}{2}}$$
$$= \frac{\delta}{\tau}\left(1-\frac{\delta^2}{\tau^2}\right)^{\frac{n-1}{2}}$$

Thus, $P_d = 1 - \frac{1}{W_n}\left(\int_0^{\arccos\left(\frac{\delta}{\tau}\right)} \sin^{n(\theta)}\, d\theta - \frac{\delta}{\tau}\left(1-\frac{\delta^2}{\tau^2}\right)^{\frac{n-1}{2}}\right)$. ∎

B. Proof of Corollary 2.3

Corollary 6.3 (Detection rate lower bound) *If n is even, the probability of detecting manipulations is lower bounded by*

$$\frac{1}{W_n}\frac{\delta}{\tau}\left(1-\frac{\delta^2}{\tau^2}\right)^{\frac{n-1}{2}} \leq P_d \leq 1.$$

Moreover, the lower bound is maximized when $\frac{\delta}{\tau} = \frac{\sqrt{n+3}-\sqrt{n-1}}{2}$.

Proof (Corollary 2.3) By definition, $P_d \leq 1$. We now prove the lower bound. Following Theorem 2.2, we have:

$$P_d = 1 - \frac{1}{W_n}\left(\int_0^{\arccos\left(\frac{\delta}{\tau}\right)} \sin^n(\theta)d\theta - \frac{\delta}{\tau}\left(1-\frac{\delta^2}{\tau^2}\right)^{\frac{n-1}{2}}\right)$$

Decomposing W_n (for n even) and using the positivity of sin on $\left[0, \frac{\pi}{2}\right]$.

$$W_n = \int_0^{\arccos\left(\frac{\delta}{\tau}\right)} \sin^n(\theta)d\theta + \int_{\arccos\left(\frac{\delta}{\tau}\right)}^{\frac{\pi}{2}} \underbrace{\sin^n(\theta)}_{\geq 0}\, d\theta$$

$$\Rightarrow 1 \quad \geq \frac{1}{W_n}\int_0^{\arccos\left(\frac{\delta}{\tau}\right)} \sin^n(\theta)d\theta$$

Finally, $P_d \geq \frac{1}{W_n}\frac{\delta}{\tau}\left(1-\frac{\delta^2}{\tau^2}\right)^{\frac{n-1}{2}}$.

We now set out to prove that the lower bound according is maximized when $\frac{\delta}{\tau} = \frac{\sqrt{n+3}-\sqrt{n-1}}{2}$. First, define $f(x) = \frac{x}{W_n}\left(1-x^2\right)^{\frac{n-1}{2}}$ with the change of variable $x = \frac{\delta}{\tau}$. We are interested in cases where $\tau > \delta$, *i.e.* $0 < x < 1$.

f has an extremum iff $f'(x) = 0$ somewhere in $[0, 1]$. The derivative $f'(x), x \in [0, 1]$ is expressed as

$$\begin{aligned} W_n f'(x) &= (1-x^2)^{\frac{n-1}{2}} - (n-1)x^2(1-x^2)^{\frac{n-3}{2}} \\ &= (1-x^2)^{\frac{n-3}{2}}\left(x^2 + \sqrt{n-1}x - 1\right)\left(x^2 - \sqrt{n-1}x - 1\right) \end{aligned}$$

Thus, on $[0,1]$, $f'(x) = 0$ for the following elements:

$$x_0 = 1 \quad \text{and} \quad x_1 = \frac{-\sqrt{n-1} + \sqrt{n+3}}{2}$$

From a quick analysis of the second derivative $f''(x)$ shows $f(x_0) \leq 0$ and $f(x_1) > 0$. Thus f is maximized at x_1, i.e. $\frac{\delta}{\tau} = \frac{-\sqrt{n-1}+\sqrt{n+3}}{2}$. ∎

C. Proof of Theorem 3.1

We now restate and prove Theorem 3.1

Theorem 6.1 (No need to aim) *Let $\mathcal{H} = \{0,1\}^{\mathcal{X}}$. For any audit set $S \subseteq \mathcal{X}$ and hypothesis $h \in \mathcal{H}$,*

$$\operatorname{diam}_{\mu}\mathcal{H}(h,S) = 2 - (\mathbb{P}(X \in S \mid X_A = 1) + \mathbb{P}(X \in S \mid X_A = 0))$$

Proof (Theorem 3.1) The proof is executed in 4 steps: decomposition of the value of $\mu(h,\mathcal{X})$ on S and $\overline{S}$, decomposition of the μ-diameter on S and $\overline{S}$, solving the optimization on the decomposed problems and conclusion.

Step 1: Decompose μ
For any $h \in \mathcal{H}$, $S \subseteq \mathcal{X}$

$$
\begin{aligned}
\mu(h, \mathcal{X}) &= \mathbb{P}(h(X) = 1 \mid X_A = 1) - \mathbb{P}(h(X) = 1 \mid X_A = 0) \\
&= \mathbb{P}(h(X) = 1 \mid X_A = 1, X \in S) \underbrace{\mathbb{P}(X \in S \mid X_A = 1)}_{\alpha} \\
&\quad + \mathbb{P}\Big(h(X) = 1 \mid X_A = 1, X \in \overline{S}\Big) \underbrace{\mathbb{P}\Big(X \in \overline{S} \mid X_A = 1\Big)}_{1-\alpha} \\
&\quad - \mathbb{P}(h(X) = 1 \mid X_A = 0, X \in S) \underbrace{\mathbb{P}(X \in S \mid X_A = 0)}_{\alpha - \delta} \\
&\quad - \mathbb{P}\Big(h(X) = 1 \mid X_A = 0, X \in \overline{S}\Big) \underbrace{\mathbb{P}\Big(X \in \overline{S} \mid X_A = 0\Big)}_{1-\alpha+\delta} \\
&= \alpha \underbrace{(\mathbb{P}(h(X) = 1 \mid X_A = 1, X \in S) - \mathbb{P}(h(X) = 1 \mid X_A = 0, X \in S))}_{\mu(h,S)} \\
&\quad + (1-\alpha) \underbrace{\Big(\mathbb{P}\Big(h(X) = 1 \mid X_A = 1, X \in \overline{S}\Big) - \mathbb{P}\Big(h(X) = 1 \mid X_A = 0, X \in \overline{S}\Big)\Big)}_{\mu\left(h,\overline{S}\right)} \\
&\quad + \delta\Big(\mathbb{P}(h(X) = 1 \mid X_A = 0, X \in S) - \mathbb{P}\Big(h(X) = 1 \mid X_A = 0, X \in \overline{S}\Big)\Big) \\
&= \alpha\mu(h, S) + (1-\alpha)\mu\Big(h, \overline{S}\Big) \\
&\quad + \delta\Big(\mathbb{P}(h(X) = 1 \mid X_A = 0, X \in S) - \mathbb{P}\Big(h(X) = 1 \mid X_A = 0, X \in \overline{S}\Big)\Big)
\end{aligned}
$$

Step 2: Decompose the μ-diameter
For any $h, h' \in \mathcal{H}(h^*, S)$,

$$
\begin{aligned}
\mu(h, \mathcal{X}) - \mu(h', \mathcal{X}) = \alpha \underbrace{(\mu(h, S) - \mu(h', S))}_{=0 \text{ since } h(S)=h'(S)=h^*(S)} &+ (1-\alpha)\Big(\mu\Big(h, \overline{S}\Big) - \mu\Big(h', \overline{S}\Big)\Big) \\
+\delta \underbrace{(\mathbb{P}(h(X) = 1 \mid X_A = 0, X \in S) - \mathbb{P}(h'(X) = 1 \mid X_A = 0, X \in S))}_{=0 \text{ since } h(S)=h'(S)=h^*(S)} & \\
+\delta\Big(\mathbb{P}\Big(h'(X) = 1 \mid X_A = 0, X \in \overline{S}\Big) - \mathbb{P}\Big(h(X) = 1 \mid X_A = 0, X \in \overline{S}\Big)\Big) &
\end{aligned}
$$

Using the definition and separability of the μ-diameter, we have

$$
\operatorname{diam}_\mu(h^*, S) = \max_{h \in \mathcal{H}(h^*, S)} \mu(h, S) - \min_{h' \in \mathcal{H}(h^*, S)} \mu(h, S)
$$

Therefore, by grouping the terms that depend on h and h' in the previous development:

$$
\begin{aligned}
&\operatorname{diam}_{\mu}(h^*, S) \\
&= \max_{h \in \mathcal{H}(h^*,S)} \left[(1-\alpha)\mu\left(h, \overline{S}\right) - \delta \mathbb{P}\left(h(X) = 1 \mid X_A = 0, X \in \overline{S}\right)\right] \\
&- \min_{h' \in \mathcal{H}(h^*,S)} \left[(1-\alpha)\mu\left(h', \overline{S}\right) - \delta \mathbb{P}\left(h'(X) = 1 \mid X_A = 0, X \in \overline{S}\right)\right]
\end{aligned}
$$

Step 3: Solve each optimization problem

To solve the two optimization problems, we come back to the definition of μ.

$$
\begin{aligned}
= \max_{h \in \mathcal{H}(h^*,S)} &\Big\{(1-\alpha)\mathbb{P}\left(h(X) = 1 \mid X_A = 1, X \in \overline{S}\right) \\
&-(1-\alpha+\delta)\mathbb{P}\left(h(X) = 1 \mid X_A = 0, X \in \overline{S}\right)\Big\} \\
= \max_{h \in \mathcal{H}(h^*,S)} &\Big\{(1-\alpha)\mathbb{P}\left(h(X) = 1 \mid X_A = 1, X \in \overline{S}\right) \\
&+(1-\alpha+\delta)\mathbb{P}\left(h(X) = 0 \mid X_A = 0, X \in \overline{S}\right)\Big\} - (1-\alpha+\delta)
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
= \min_{h \in \mathcal{H}(h^*,S)} &\Big\{(1-\alpha)\mathbb{P}\left(h(X) = 1 \mid X_A = 1, X \in \overline{S}\right) \\
&-(1-\alpha+\delta)\mathbb{P}\left(h(X) = 1 \mid X_A = 0, X \in \overline{S}\right)\Big\} \\
= \min_{h \in \mathcal{H}(h^*,S)} &\Big\{(1-\alpha)\mathbb{P}\left(h(X) = 1 \mid X_A = 1, X \in \overline{S}\right) \\
&+(1-\alpha+\delta)\mathbb{P}\left(h(X) = 0 \mid X_A = 0, X \in \overline{S}\right)\Big\} - (1-\alpha+\delta)
\end{aligned}
$$

We write $h^{\uparrow}$ (resp. $h^{\downarrow}$) the minimizer of (resp.).

$$
h^{\uparrow}(x) = \begin{cases} 1 & \text{if } x_A = 1 \text{ and } x \in \overline{S} \\ 0 & \text{if } x_A = 0 \text{ and } x \in \overline{S} \\ 0 & \text{else} \end{cases} \quad h^{\downarrow}(x) = \begin{cases} 1 & \text{if } x_A = 0 \text{ and } x \in \overline{S} \\ 0 & \text{if } x_A = 1 \text{ and } x \in \overline{S} \\ 0 & \text{else} \end{cases}
$$

The optimizers $h^{\uparrow}$ and $h^{\downarrow}$ yield the optima

$$
\begin{aligned}
\text{[orange]} &= -(1-\alpha+\delta) + (1-\alpha)\underbrace{\mathbb{P}\Big(h^{\uparrow(X)} = 1 \mid X_A = 1, X \in \overline{S}\Big)}_{=1} \\
&\quad + (1-\alpha+\delta)\underbrace{\mathbb{P}\Big(h^{\uparrow(X)} = 0 \mid X_A = 0, X \in \overline{S}\Big)}_{=1} \\
&= 1-\alpha \\
\text{[blue]} &= -(1-\alpha+\delta) + (1-\alpha)\underbrace{\mathbb{P}\Big(h^{\downarrow(X)} = 1 \mid X_A = 1, X \in \overline{S}\Big)}_{=0} \\
&\quad + (1-\alpha+\delta)\underbrace{\mathbb{P}\Big(h^{\downarrow(X)} = 0 \mid X_A = 0, X \in \overline{S}\Big)}_{=0} \\
&= -(1-\alpha+\delta)
\end{aligned}
$$

Step 4: Conclusion

$$
\begin{aligned}
\operatorname{diam}_\mu \mathcal{H}(h^*, S) &= \text{[orange]} - \text{[blue]} \\
&= (1-\alpha) + (1-\alpha+\delta) \\
&= 2 - (\mathbb{P}(X \in S \mid X_A = 1) + \mathbb{P}(X \in S \mid X_A = 0))
\end{aligned}
$$

∎

D. Proof of Theorem 3.2

We now restate and prove Theorem 3.2

Theorem 6.2 (Memory and auditability) *Consider $S \subseteq \mathcal{X}$, $d \in \mathcal{H}_m^{\text{dict}}$. Note $m' = m - |x \in S : d(x) = 1|$. The μ-diameter of $\mathcal{H}_m^{\text{dict}}(d, S)$ is given by*

$$
\operatorname{diam}_\mu \mathcal{H}_m^{\text{dict}}(d, S) = \frac{\min\Big(\big|\mathcal{X}_A \cap \overline{S}\big|, m'\Big)}{|\mathcal{X}_A|} + \frac{\min\Big(\big|\overline{\mathcal{X}_A} \cap \overline{S}\big|, m'\Big)}{\big|\overline{\mathcal{X}_A}\big|}
$$

Proof (Theorem 3.2) In the proof of Theorem 3.1, we established the following identity (for any hypothesis class thus for $\mathcal{H}_m^{\text{dict}}$, and for any S and d^*):

$$\begin{aligned}
&\operatorname{diam}_{\mu}\mathcal{H}^{\text{dict}}(d^*, S) \\
&= \max_{d\in\mathcal{H}^{\text{dict}}(d^*,S)} \Big\{\mathbb{P}\Big(X \in \overline{S} \mid X_A = 1\Big)\mathbb{P}\Big(d(X) = 1 \mid X_A = 1, X \in \overline{S}\Big) \\
&\qquad\qquad +\mathbb{P}\Big(X \in \overline{S} \mid X_A = 0\Big)\mathbb{P}\Big(d(X) = 0 \mid X_A = 0, X \in \overline{S}\Big)\Big\} \\
&- \min_{d\in\mathcal{H}^{\text{dict}}(d^*,S)} \Big\{\mathbb{P}\Big(X \in \overline{S} \mid X_A = 1\Big)\mathbb{P}\Big(d(X) = 1 \mid X_A = 1, X \in \overline{S}\Big) \\
&\qquad\qquad +\mathbb{P}\Big(X \in \overline{S} \mid X_A = 0\Big)\mathbb{P}\Big(d(X) = 0 \mid X_A = 0, X \in \overline{S}\Big)\Big\}
\end{aligned}$$

First, observe that in the two optimization problems, the value of the objective function does not depend on the values of d on S. Moreover, the choices of the labels $d(x)$ for $x \in \overline{S}$ can be made freely as long as d does not have more than $m' = m - |x \in S : d^*(x) = 1|$ "1"s (because it has to use $|x \in S : d^*(x) = 1|$ slots of memory to store the answers of d^* on S).

Therefore, the dictionary that optimizes is built by storing as many "1"s in d on the entries of $x \in \mathcal{X}_A \cap \overline{S}$ within the limits of the m' slots left. This leads to

$$= \mathbb{P}\Big(X \in \overline{S} \mid X_A = 1\Big)\frac{\min\Big(\big|\mathcal{X}_A \cap \overline{S}\big|, m'\Big)}{\big|\mathcal{X}_A \cap \overline{S}\big|} + \mathbb{P}\Big(X \in \overline{S} \mid X_A = 0\Big) * 1$$

Next, rewriting as a maximization problem, we get

$$\begin{aligned}
&= \mathbb{P}\Big(X \in \overline{S} \mid X_A = 1\Big) + \mathbb{P}\Big(X \in \overline{S} \mid X_A = 0\Big) \\
&\quad - \min_{d\in\mathcal{H}^{\text{dict}}(d^*,S)} \Big\{\mathbb{P}\Big(X \in \overline{S} \mid X_A = 1\Big)\mathbb{P}\Big(d(X) = 0 \mid X_A = 1, X \in \overline{S}\Big) \\
&\qquad\qquad +\mathbb{P}\Big(X \in \overline{S} \mid X_A = 0\Big)\mathbb{P}\Big(d(X) = 1 \mid X_A = 0, X \in \overline{S}\Big)\Big\}
\end{aligned}$$

Similar to the case of , the dictionary that optimizes is built by storing as many "1"s in d on the entries of $x \in \overline{\mathcal{X}_A} \cap \overline{S}$ withing the limits of the m' slots left. This leads to

$$\begin{aligned}
&= \mathbb{P}\Big(X \in \overline{S} \mid X_A = 1\Big) + \mathbb{P}\Big(X \in \overline{S} \mid X_A = 0\Big) \\
&\quad -\mathbb{P}\Big(X \in \overline{S} \mid X_A = 1\Big) * 1 - \mathbb{P}\Big(X \in \overline{S} \mid X_A = 0\Big)\frac{\min\Big(\big|\overline{\mathcal{X}_A} \cap \overline{S}\big|, m'\Big)}{\big|\overline{\mathcal{X}_A} \cap \overline{S}\big|}
\end{aligned}$$

Composing the expressions of and , we get

$$\operatorname{diam}_\mu \mathcal{H}^{\text{dict}}(d^*, S) = \mathbb{P}\Big(X \in \overline{S} \mid X_A = 1\Big) \frac{\min\Big(\big|\mathcal{X}_A \cap \overline{S}\big|, m'\Big)}{\big|\mathcal{X}_A \cap \overline{S}\big|} + \mathbb{P}\Big(X \in \overline{S} \mid X_A = 0\Big) \frac{\min\Big(\big|\overline{\mathcal{X}_A} \cap \overline{S}\big|, m'\Big)}{\big|\overline{\mathcal{X}_A} \cap \overline{S}\big|}$$

Here, it is important to understand that in the notations $\mathbb{P}(X \in S)$ or $\mathbb{P}(d(X) = 1)$, X is a random variable taking values in $\mathcal{X}$ with a uniform probability. Therefore, $\mathbb{P}\Big(X \in \overline{S} \mid X_A = 1\Big) = \frac{|\mathcal{X}_A \cap \overline{S}|}{|\mathcal{X}_A|}$ and $\mathbb{P}\Big(X \in \overline{S} \mid X_A = 0\Big) = \frac{|\overline{\mathcal{X}_A} \cap \overline{S}|}{|\overline{\mathcal{X}_A}|}$, which simplifies the previous equation

$$\operatorname{diam}_\mu \mathcal{H}^{\text{dict}}(d^*, S) = \frac{\min\Big(\big|\mathcal{X}_A \cap \overline{S}\big|, m'\Big)}{|\mathcal{X}_A|} + \frac{\min\Big(\big|\overline{\mathcal{X}_A} \cap \overline{S}\big|, m'\Big)}{\big|\overline{\mathcal{X}_A}\big|}$$

∎

E. Proof of Corollary 3.1

We now restate and prove Corollary 3.1

Corollary 6.1 (Benign overfitting and auditability) *Let $\mathcal{X}$ and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any input space and hypothesis class. Assume that $\mathcal{H}$ exhibits benign overfitting with respect to the sensitive attribute X_A and its opposite $1 - X_A$*[9]*, then for any $d \leq d_0$, and $S \in \mathcal{X}^d$,*

$$\operatorname{diam}_\mu \mathcal{H}(h^*, S) \geq \mathbb{P}(X \in S \mid X_A = 1) + \mathbb{P}(X \in S \mid X_A = 0) - 2\,\mathbb{P}(X \in S) - 2\varepsilon(1 - \mathbb{P}(X \in S))$$

Proof (Corollary 3.1) Note $\alpha_1 = \mathbb{P}\Big(X \in \overline{S} \mid X_A = 1\Big)$ and $\alpha_0 = \mathbb{P}\Big(X \in \overline{S} \mid X_A = 0\Big)$. In the proof of Theorem 3.1, we established the following equality:

$$\begin{aligned}
&\operatorname{diam}_\mu \mathcal{H}(h^*, S) \\
&= \max_{h \in \mathcal{H}(h^*, S)} \Big\{ \alpha_1 \mathbb{P}\Big(h(X) = 1 \mid X_A = 1, X \in \overline{S}\Big) + \alpha_0 \mathbb{P}\Big(h(X) = 0 \mid X_A = 0, X \in \overline{S}\Big) \Big\} \\
&- \min_{h \in \mathcal{H}(h^*, S)} \Big\{ \alpha_1 \mathbb{P}\Big(h(X) = 1 \mid X_A = 1, X \in \overline{S}\Big) + \alpha_0 \mathbb{P}\Big(h(X) = 0 \mid X_A = 0, X \in \overline{S}\Big) \Big\}
\end{aligned}$$

And

9. That is, Definition 3.3 holds for $c = x_A$ and $c = 1 - x_A$

$$= \alpha_1 + \alpha_0 - \max_{h \in \mathcal{H}(h^*, S)} \Big\{ \alpha_1 \mathbb{P}\Big(h(X) = 0 \mid X_A = 1, X \in \overline{S}\Big) + \alpha_0 \mathbb{P}\Big(h(X) = 1 \mid X_A = 0, X \in \overline{S}\Big) \Big\}$$

Since $\mathcal{H}$ exhibits benign overfitting with respect to the sensitive attribute and $|S| \leq d_0$, there exists $h \in \mathcal{H}(h^*, S)$ such that $\mathbb{P}\Big(h(X) = X_A \mid X \in \overline{S}\Big) = 1 - \varepsilon$. Moreover,

$$\begin{aligned}
\mathbb{P}\Big(h(X) = X_A \mid X \in \overline{S}\Big) &= \mathbb{P}\Big(h(X) = 1 \mid X \in \overline{S}, X_A = 1\Big)\mathbb{P}\Big(X_A = 1 \mid X \in \overline{S}\Big) \\
&\quad + \mathbb{P}\Big(h(X) = 0 \mid X \in \overline{S}, X_A = 0\Big)\mathbb{P}\Big(X_A = 0 \mid X \in \overline{S}\Big) \\
&= \alpha_1 \frac{\mathbb{P}(X_A = 1)}{\mathbb{P}\Big(X \in \overline{S}\Big)} \mathbb{P}\Big(h(X) = 1 \mid X \in \overline{S}, X_A = 1\Big) \\
&\quad + \alpha_0 \frac{\mathbb{P}(X_A = 0)}{\mathbb{P}\Big(X \in \overline{S}\Big)} \mathbb{P}\Big(h(X) = 1 \mid X \in \overline{S}, X_A = 0\Big)
\end{aligned}$$

Since $\mathbb{P}(X_A = 0) + \mathbb{P}(X_A = 1) = 1$, $\mathbb{P}(X_A = 0) \geq 0$ and $\mathbb{P}(X_A = 1) \geq 0$, we have

$$\begin{aligned}
&\alpha_1 \mathbb{P}\Big(h(X) = 1 \mid X \in \overline{S}, X_A = 1\Big) + \alpha_0 \mathbb{P}\Big(h(X) = 1 \mid X \in \overline{S}, X_A = 0\Big) \\
&\geq \mathbb{P}(X_A = 1)\alpha_1 \mathbb{P}\Big(h(X) = 1 \mid X \in \overline{S}, X_A = 1\Big) \\
&\quad + \mathbb{P}(X_A = 0)\alpha_0 \mathbb{P}\Big(h(X) = 1 \mid X \in \overline{S}, X_A = 0\Big) \\
&= (1 - \varepsilon)\mathbb{P}\Big(X \in \overline{S}\Big)
\end{aligned}$$

Therefore, $\geq (1 - \varepsilon)\mathbb{P}\Big(X \in \overline{S}\Big)$.

With the same arguments, we prove $\leq \alpha_0 + \alpha_1 - (1 - \varepsilon)\mathbb{P}\Big(X \in \overline{S}\Big)$.

To conclude, $\operatorname{diam}_\mu \mathcal{H}(h^*, S) \geq 2(1 - \varepsilon)\mathbb{P}\Big(X \in \overline{S}\Big) - (\alpha_0 + \alpha_1)$. ∎

F. Effect of the audit dataset size

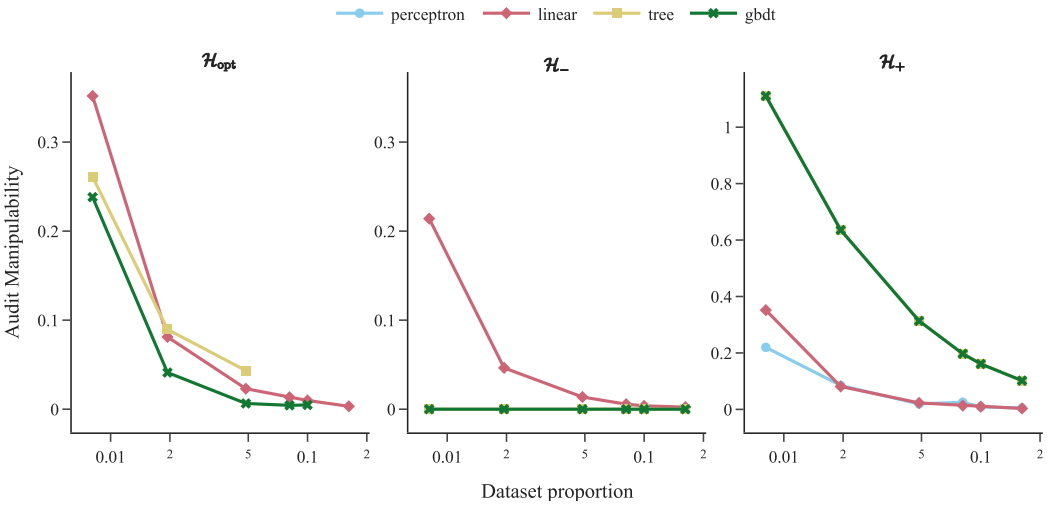


Figure 6.1: Evolution of the μ-diameter with the size of the audit set S represented as a proportion of the total dataset size for the COMPAS dataset. Each line represents an audited model, whose hyperparameters are either tuned for the best generalization, either tuned for the highest capacity or tuned for the lowest capacity. For each (model, hyperparameter) couple, the μ-diameter is averaged over 15 audit datasets.

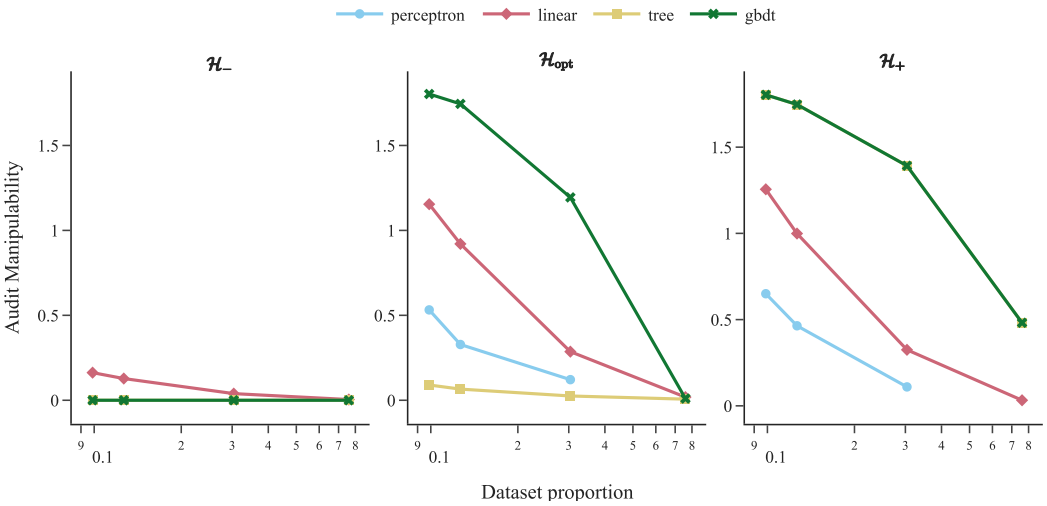


Figure 6.2: Evolution of the μ-diameter with the size of the audit set S represented as a proportion of the total dataset size for the StudentPerf dataset. Each line represents an audited model, whose hyperparameters are either tuned for the best generalization, either tuned for the highest capacity or tuned for the lowest capacity. For each (model, hyperparameter) couple, the μ-diameter is averaged over 15 audit datasets.

G. Proof of Proposition 4.1

We now restate and prove Proposition 4.1

Proposition 6.1 (AKH guarantees) *Consider $h, h' \in \mathcal{Y}^{\mathcal{X}}$ two models and $\alpha = \mathbb{P}(h(x) = c(x))$ (resp. $\alpha' = \mathbb{P}(h'(x) = c(x))$) their accuracy. Let $\delta = d_{H(h,h')}$ be the relative Hamming distance between h and h' and $\delta_C = \mathbb{P}(h(x) \neq h'(x) \mid h(x) \neq c(x))$. The property test $\mathcal{T}_b$ defined by AKH enjoys the following guarantees:*

$$\text{If } \; h = h', \; \mathbb{P}(\mathcal{D})[\mathcal{T}_b(h, h') = 1] = 1$$

$$\text{If } \; h \neq h', \; \mathbb{P}(\mathcal{D})[\mathcal{T}_b(h, h') = 0] = \delta_C \geq \frac{\delta - (1 - \alpha')}{1 - \alpha}$$

Proof (Proposition 4.1) Case $h = h'$
In this case, $\forall x \in \mathcal{X}, h(x) = h'(x)$. Thus, $\mathcal{T}$ will always return 1.

Case $h \neq h'$

$$\begin{aligned}
\mathbb{P}\left(\mathcal{T}^{\mathcal{D}(h,h')} = 0\right) &= \mathbb{P}\left(b : x \sim \overline{\mathcal{D}_h}\right)[h(x) \neq h'(x)] \\
&= \mathbb{P}(h(x) \neq h'(x) \quad \text{given} \quad h(x) \neq c(x)) \\
&= \frac{\mathbb{P}(h(x) \neq h'(x), h(x) \neq c(x))}{\underbrace{\mathbb{P}(h(x) \neq c(x))}_{1-\alpha}}
\end{aligned}$$

We now decompose the event $h(x) \neq h'(x)$ on the partition $(h(x) = c(x), h(x) \neq c(x))$.

$$\begin{aligned}
\mathbb{P}(h(x) \neq h'(x)) = &\mathbb{P}(h(x) \neq h'(x), h(x) \neq c(x)) \\
&+ \mathbb{P}(h(x) \neq h'(x), h(x) = c(x))
\end{aligned}$$

Using the inclusion $\{h(x) \neq h'(x), h(x) = c(x)\} \subset \{h'(x) \neq c(x)\}$

$$\mathbb{P}(h(x) \neq h'(x), h(x) = c(x)) \leq \underbrace{\mathbb{P}(h'(x) \neq c(x))}_{1-\alpha'}$$

Thus,

$$
\begin{aligned}
&\mathbb{P}\Big(\mathcal{T}^{\mathcal{D}(h,h')}=0\Big)\\
&=\frac{\mathbb{P}(h(x)\neq h'(x),h(x)\neq c(x))}{\mathbb{P}(h(x)\neq c(x))}\\
&=\frac{\overbrace{\mathbb{P}(h(x)\neq h'(x))}^{\delta}-\overbrace{\mathbb{P}(h(x)\neq h'(x),h(x)=c(x))}^{\leq 1-\alpha'}}{\underbrace{\mathbb{P}(h(x)\neq c(x))}_{1-\alpha}}\\
&\geq\frac{\delta-(1-\alpha')}{1-\alpha}.
\end{aligned}
$$

∎

H. Evaluation Setup

The fingerprints we re-implemented are IPGuard, ModelDiff, SAC and ZestOfLIME. We based our implementation on the descriptions of the schemes in their respective papers and re-used part of the authors' code when available. We choose two benchmarks – ModelReuse and SACBench – spanning three common vision datasets: Stanford Dogs [218], Oxford Flowers [219] and CIFAR10 [220] which we abbreviate as SDog120, Flower102 and CIFAR10. We used the model weights released by the authors of the respective benchmarks. For each experiment, we report the average (and standard deviation) over five runs for each setting. The experiments were run on a compute cluster. The nodes were based on an Intel Cascade Lake 6248 processor with 16Go Nvidia Tesla V100 SXM2 GPUs.

I. Details on the computation of the True and False Positive Rate

The True Positive Rate and False Postive Rate are computed as follows. Consider a fingerprint (as defined in the problem setting section) $\mathcal{T}:\left(y^{x},y^{x}\right)\rightarrow\{0,1\}$. Define $\mathbb{V}$ to be a set of victim models and for each victim model $h\in\mathbb{V}$, $\mathbb{S}(h)$ is a set of models stolen from h and $\mathbb{U}(h)$ is a set of models unrelated to h. A benchmark is a triplet $\mathbb{B}=(\mathbb{V},(\mathbb{S}(h))_{h\in\mathbb{V}},(\mathbb{U}(h))_{h\in\mathbb{V}})$. The True and False positive Rate reported in the paper are computed as follows.

$$
\begin{aligned}
\mathrm{TPR}(\mathbb{B})&=\frac{1}{|\mathbb{V}|}\sum_{h\in\mathbb{V}}\frac{\sum_{h'\in\mathbb{S}(h)}\mathbb{1}(\mathcal{T}(h,h')=1)}{|\mathbb{S}(h)|}\\
\mathrm{FPR}(\mathbb{B})&=\frac{1}{|\mathbb{V}|}\sum_{h\in\mathbb{V}}\frac{\sum_{h'\in\mathbb{U}(h)}\mathbb{1}(\mathcal{T}(h,h')=1)}{|\mathbb{U}(h)|}
\end{aligned}
\tag{6.1}
$$

Model	Hyperparameters	Value range
LINEAR	penalty	None, l2
	C	0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000
PERCEPTRON	penalty	None, l2
	alpha	1e-06, 1e-05, 0.0001, 0.001, 0.01
TREE	max_depth	2, 4, 8, 16, 32, 64, 128
	ccp_alpha	0.001, 0.003, 0.005, 0.007, 0.01, 0.05, 0.1, 0.2, 0.5, 0.0
GBDT	max_depth	1, 2, 4, 8
	n_estimators	100, 200, 500
	reg_lambda	0.0, 1e-6, 1e-3, 0.1, 1.0, 1e6, 1e7
	max_leaves	0
	learning_rate	0.3
	gamma	0.0
	min_child_weight	0.0
	max_delta_step	0.0
	subsample	1.0
	reg_alpha	0.0
	early_stopping_rounds	None

Table 6.1: Value range for the hyperparameters of the models used in the experiments.

Bibliography

[1] T. Yan and C. Zhang, "Active Fairness Auditing," in *Proceedings of the 39th International Conference on Machine Learning*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds., in Proceedings of Machine Learning Research, vol. 162. PMLR, June 2022, pp. 24929–24962. Accessed: Oct. 12, 2023. [Online]. Available: https://proceedings.mlr.press/v162/yan22c.html

[2] A. Birhane, R. Steed, V. Ojewale, B. Vecchione, and I. D. Raji, "AI Auditing: The Broken Bus on the Road to AI Accountability," in *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2024, pp. 612–643. doi: 10.1109/SaTML59370.2024.00037.

[3] M. Hardt and B. Recht, *Patterns, predictions, and actions: Foundations of machine learning*. Princeton University Press, 2022. [Online]. Available: https://mlstory.org/

[4] U. Aïvodji, H. Arai, O. Fortineau, S. Gambs, S. Hara, and A. Tapp, "Fairwashing: the risk of rationalization," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, K. Chaudhuri and R. Salakhutdinov, Eds., in Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 161–170.

[5] B. Wronski *et al.*, "Handheld multi-frame super-resolution," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 28:1–28:18, July 2019, doi: 10.1145/3306346.3323024.

[6] I. Price *et al.*, "Probabilistic weather forecasting with machine learning," *Nature*, vol. 637, no. 8044, pp. 84–90, Jan. 2025, doi: 10.1038/s41586-024-08252-9.

[7] L. Raynaud, C. Brochet, and G. Moldovan, "ML for weather prediction at Météo-France: current status and future plans," in *EGU General Assembly Conference Abstracts*, 2024, p. 1675. Accessed: Oct. 20, 2025. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2024EGUGA..26.1675R/abstract

[8] X. Zhao, A. Cox, and L. Cai, "ChatGPT and the digitisation of writing," *Humanities and Social Sciences Communications*, vol. 11, no. 1, p. 482, Apr. 2024, doi: 10.1057/s41599-024-02904-x.

[9] J. Buolamwini and T. Gebru, "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification," in *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, PMLR, Jan. 2018, pp. 77–91. Accessed: Nov. 24, 2025. [Online]. Available: https://proceedings.mlr.press/v81/buolamwini18a.html

[10] H. Lycklama, A. Viand, N. Küchler, C. Knabenhans, and A. Hithnawi, "Holding secrets accountable: auditing privacy-preserving machine learning," in *Proceedings of the 33rd USENIX Conference on Security Symposium*, in SEC '24. USA: USENIX Association, Aug. 2024, pp. 1975–1992.

[11] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, "Universal and Transferable Adversarial Attacks on Aligned Language Models." Accessed: Nov. 24, 2025. [Online]. Available: http://arxiv.org/abs/2307.15043

[12] J. Schmid, T. Sytsma, and A. Shenk, "Evaluating Natural Monopoly Conditions in the AI Foundation Model Market," Sept. 2024. Accessed: Nov. 19, 2025. [Online]. Available: https://www.rand.org/pubs/research_reports/RRA3415-1.html

[13] M. A. Lemley and J. Noti-Victor, "Anticompetitive Acquiescence." Accessed: Nov. 19, 2025. [Online]. Available: https://papers.ssrn.com/abstract=5320353

[14] A. Korinek and J. Vipra, "Concentrating intelligence: scaling and market structure in artificial intelligence," *Economic Policy*, vol. 40, no. 121, pp. 225–256, Jan. 2025, doi: 10.1093/epolic/eiae057.

[15] M. Von Thun, "Monopoly Power Is the Elephant in the Room in the AI Debate," *TechPolicy.Press*, Oct. 2023, Accessed: Nov. 19, 2025. [Online]. Available: https://www.techpolicy.press/monopoly-power-is-the-elephant-in-the-room-in-the-ai-debate/

[16] "Artificial intelligence, data and competition," OECD Artificial Intelligence Papers 18, May 2024. doi: 10.1787/e7e88884-en.

[17] A. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, July 1959, doi: 10.1147/rd.33.0210.

[18] R. Mahmood, J. Lucas, J. M. Alvarez, S. Fidler, and M. T. Law, "Optimizing Data Collection for Machine Learning," *Journal of Machine Learning Research*, vol. 26, no. 38, pp. 1–52, 2025, Accessed: Oct. 20, 2025. [Online]. Available: http://jmlr.org/papers/v26/23-0292.html

[19] J. Ferry, R. Fukasawa, T. Pascal, and T. Vidal, "Trained random forests completely reveal your dataset," in *Proceedings of the 41st International Conference on Machine Learning*, in ICML'24, vol. 235. Vienna, Austria: JMLR.org, July 2024, pp. 13545–13569.

[20] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramèr, "Membership Inference Attacks From First Principles," in *2022 IEEE Symposium on Security and Privacy (SP)*, May 2022, pp. 1897–1914. doi: 10.1109/SP46214.2022.9833649.

[21] R. Webster and T. Furon, "Multi-modal Identity Extraction," presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025, pp. 10797–10806. Accessed: Nov. 20, 2025. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2025/html/Webster_Multi-modal_Identity_Extraction_ICCV_2025_paper.html

[22] R. Geirhos *et al.*, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, Nov. 2020, doi: 10.1038/s42256-020-00257-z.

[23] D. Thiel and J. Hancock, "Identifying and Eliminating CSAM in Generative ML Training Data and Models," *Stanford Digital Repository*, 2023, doi: https://doi.org/10.25740/kh752sm9123.

[24] A. Birhane, S. Dehdashtian, V. Prabhu, and V. Boddeti, "The Dark Side of Dataset Scaling: Evaluating Racial Classification in Multimodal Models," in *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, Rio de Janeiro Brazil: ACM, June 2024, pp. 1229–1244. doi: 10.1145/3630106.3658968.

[25] S. Barocas, M. Hardt, and A. Narayanan, *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, 2023. Accessed: Jan. 04, 2024. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=ouawEAAAQBAJ&oi=fnd&pg=PA83&dq=Fairness+and+Machine+Learning&ots=2kDVfTXV8P&sig=uYNciZV1R1c7X_LakbSmwN_YJYk

[26] J. Hayes *et al.*, "Measuring memorization in language models via probabilistic extraction," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, L. Chiruzzo, A. Ritter, and L. Wang, Eds., Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 9266–9291. doi: 10.18653/v1/2025.naacl-long.469.

[27] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang, "Quantifying Memorization Across Neural Language Models," presented at the

The Eleventh International Conference on Learning Representations, Sept. 2022. Accessed: Nov. 21, 2025. [Online]. Available: https://openreview.net/forum?id=TatRHT_1cK

[28] A. Cooper and J. Grimmelmann, "The Files are in the Computer: On Copyright, Memorization, and Generative AI," *Chicago-Kent Law Review*, vol. 100, no. 1, p. 141, Aug. 2025, [Online]. Available: https://scholarship.kentlaw.iit.edu/cklawreview/vol100/iss1/9

[29] A.-L. Cauchy, "Méthode générale pour la résolution des systèmes d'équations simultanées," *Compte Rendu des Séances de l'Académie des Sciences*, vol. 25, pp. 536–538, 1847, [Online]. Available: https://gallica.bnf.fr/ark:/12148/bpt6k2982c.image.f540.pagination.langEN

[30] F. Bach, *Learning Theory from First Principles*. in Adaptive Computation and Machine Learning series. Cambridge, MA, USA: MIT Press, 2024. Accessed: Oct. 20, 2025. [Online]. Available: https://mitpress.mit.edu/9780262049443/learning-theory-from-first-principles/

[31] H. Robbins and S. Monro, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, Sept. 1951, doi: 10.1214/aoms/1177729586.

[32] E. Francazi, M. Baity-Jesi, and A. Lucchi, "A Theoretical Analysis of the Learning Dynamics under Class Imbalance," in *Proceedings of the 40th International Conference on Machine Learning*, PMLR, July 2023, pp. 10285–10322. Accessed: Nov. 24, 2025. [Online]. Available: https://proceedings.mlr.press/v202/francazi23a.html

[33] F. Bachoc, J. Bolte, R. Boustany, and J.-M. Loubes, "When majority rules, minority loses: bias amplification of gradient descent." Accessed: Nov. 24, 2025. [Online]. Available: http://arxiv.org/abs/2505.13122

[34] I. Shumailov *et al.*, "Manipulating SGD with Data Ordering Attacks," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2021, pp. 18021–18032. Accessed: Nov. 24, 2025. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/959ab9a0695c467e7caf75431a872e5c-Abstract.html

[35] A. Q. Jiang *et al.*, "Mixtral of Experts." Accessed: Oct. 21, 2025. [Online]. Available: http://arxiv.org/abs/2401.04088

[36] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A Survey of Quantization Methods for Efficient Neural Network Inference," *Low-Power Computer Vision*. Chapman, Hall/CRC, 2022.

[37] J. Liu, P. Cui, and H. Namkoong, "Modeling and Exploiting Data Heterogeneity under Distribution Shifts," New Orleans, Dec. 11, 2023. Accessed: Oct. 21, 2025. [Online]. Available: https://neurips.cc/virtual/2023/73953

[38] R. Umbach, N. Henry, G. F. Beard, and C. M. Berryessa, "Non-Consensual Synthetic Intimate Imagery: Prevalence, Attitudes, and Knowledge in 10 Countries," in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, in CHI '24. New York, NY, USA: Association for Computing Machinery, May 2024, pp. 1–20. doi: 10.1145/3613904.3642382.

[39] N. Singh, "Teenager arrested after explicit AI photos of 50 schoolgirls appear online," *The Independent*, June 2024, Accessed: Nov. 24, 2025. [Online]. Available: https://www.independent.co.uk/news/world/australasia/australia-schoolgirls-ai-nude-photos-scandal-b2561104.html

[40] N. Carlini *et al.*, "Stealing part of a production language model," in *Proceedings of the 41st International Conference on Machine Learning*, in ICML'24, vol. 235. Vienna, Austria: JMLR.org, July 2024, pp. 5680–5705.

[41] M. Nasr *et al.*, "Scalable Extraction of Training Data from Aligned, Production Language Models," presented at the The Thirteenth International Conference on Learning Representations, Oct. 2024. Accessed: Oct. 21, 2025. [Online]. Available: https://openreview.net/forum?id=vjel3nWP2a

[42] A. Dziedzic, N. Dhawan, M. A. Kaleem, J. Guan, and N. Papernot, "On the Difficulty of Defending Self-Supervised Learning against Model Extraction," in *Proceedings of the 39th International Conference on Machine Learning*, PMLR, June 2022, pp. 5757–5776. Accessed: Nov. 24, 2025. [Online]. Available: https://proceedings.mlr.press/v162/dziedzic22a.html

[43] E. Le Merrer and G. Trédan, "TamperNN: Efficient Tampering Detection of Deployed Neural Nets," in *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*, Oct. 2019, pp. 424–434. doi: 10.1109/ISSRE.2019.00049.

[44] C. Zhang, H. Foerster, R. D. Mullins, Y. Zhao, and I. Shumailov, "Hardware and Software Platform Inference," presented at the Forty-second International Conference on Machine Learning, June 2025. Accessed: Nov. 24, 2025. [Online]. Available: https://openreview.net/forum?id=kdmjVF1iDO

[45] A. A. Velasco, S. Tsirtsis, and M. Gomez-Rodriguez, "Auditing Pay-Per-Token in Large Language Models." Accessed: Nov. 24, 2025. [Online]. Available: http://arxiv.org/abs/2510.05181

[46] D. Flint, *Philosophy and Principles of Auditing: An Introduction*. Basingstoke: Palgrave Macmillan, 1988.

[47] L. Teck-Heang and A. Md Ali, "The evolution of auditing: An analysis of the historical development," pp. 1548–6583, Dec. 2008.

[48] B. Vecchione, K. Levy, and S. Barocas, "Algorithmic Auditing and Social Justice: Lessons from the History of Audit Studies," in *Proceedings of the 1st ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, in EAAMO '21. New York, NY, USA: Association for Computing Machinery, Nov. 2021, pp. 1–9. doi: 10.1145/3465416.3483294.

[49] D. McCabe, "RealPage Agrees to Settle Federal Rent-Collusion Case," *The New York Times*, Nov. 2025, Accessed: Nov. 26, 2025. [Online]. Available: https://www.nytimes.com/2025/11/24/technology/realpage-doj-settlement.html

[50] A. Reuel *et al.*, "Open Problems in Technical AI Governance," *Transactions on Machine Learning Research*, Nov. 2024, Accessed: Nov. 26, 2025. [Online]. Available: https://openreview.net/forum?id=1nO4qFMiS0

[51] "Regulation (EU) 2022/2065 of the European Parliament and of the Council of 19~October 2022 on a Single Market For Digital Services and Amending Directive~2000/31/EC (Digital Services Act) (Text with EEA Relevance)." Accessed: Oct. 12, 2023. [Online]. Available: http://data.europa.eu/eli/reg/2022/2065/oj/eng

[52] "Regulation (EU) 2022/1925 of the European Parliament and of the Council of 14~September 2022 on Contestable and Fair Markets in the Digital Sector and Amending Directives (EU) 2019/1937 and (EU) 2020/1828 (Digital Markets Act) (Text with EEA Relevance)." Accessed: Oct. 12, 2023. [Online]. Available: http://data.europa.eu/eli/reg/2022/1925/oj/eng

[53] "Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (Text with EEA relevance)." Accessed: Nov. 26, 2025. [Online]. Available: http://data.europa.eu/eli/reg/2024/1689/oj

[54] S. Casper *et al.*, "Black-Box Access Is Insufficient for Rigorous AI Audits," in *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, in FAccT '24. New York, NY, USA: Association for Computing Machinery, June 2024, pp. 2254–2272. doi: 10.1145/3630106.3659037.

[55] S. H. Cen and R. Alur, "From Transparency to Accountability and Back: A Discussion of Access and Evidence in AI Auditing." 2024.

[56] J. Dai, P. Gradu, I. D. Raji, and B. Recht, "From Individual Experience to Collective Evidence: A Reporting-Based Framework for Identifying Systemic Harms," in *Proceedings of the 42nd International Conference on Machine Learning*, PMLR, Oct. 2025, pp. 12063–12083. Accessed: Nov. 18, 2025. [Online]. Available: https://proceedings.mlr.press/v267/dai25g.html

[57] L. Gailmard, D. Spence, C. Lawrence, and D. E. Ho, "Known Unknowns and Unknown Unknowns: Designing a Scalable Adverse Event Reporting System for AI," *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, vol. 8, no. 2, pp. 1004–1017, Oct. 2025, doi: 10.1609/aies.v8i2.36607.

[58] OECD, "Towards a common reporting framework for AI incidents," *OECD Artificial Intelligence Papers*, Feb. 2025, doi: 10.1787/f326d4ac-en.

[59] S. McGregor, "Preventing Repeated Real World AI Failures by Cataloging Incidents: The AI Incident Database," in *Proceedings of the AAAI Conference on Artificial Intelligence*, May 2021, pp. 15458–15463. doi: 10.1609/aaai.v35i17.17817.

[60] S. Chouaki, A. Chakraborty, O. Goga, and S. Zannettou, "What News Do People Get on Social Media? Analyzing Exposure and Consumption of News through Data Donations," in *Proceedings of the ACM Web Conference 2024*, in WWW '24. New York, NY, USA: Association for Computing Machinery, May 2024, pp. 2371–2382. doi: 10.1145/3589334.3645399.

[61] P. Bouchaud and P. Ramaciotti, "Auditing the audits: evaluating methodologies for social media recommender system audits," *Applied Network Science*, vol. 9, no. 1, pp. 1–20, Dec. 2024, doi: 10.1007/s41109-024-00668-6.

[62] J. Bandy, "Problematic Machine Behavior: A Systematic Literature Review of Algorithm Audits," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CSCW1, pp. 74:1–74:34, Apr. 2021, doi: 10.1145/3449148.

[63] J. Garcia Bourrée, E. Le Merrer, G. Tredan, and B. Rottembourg, "On the Relevance of APIs Facing Fairwashed Audits." 2023.

[64] C. Yadav, M. Moshkovitz, and K. Chaudhuri, "XAudit : A Theoretical Look at Auditing with Explanations." Accessed: Oct. 12, 2023. [Online]. Available: http://arxiv.org/abs/2206.04740

[65] O. Franzese, A. S. Shamsabadi, C. Luck, and H. Haddadi, "Secure and Confidential Certificates of Online Fairness." Accessed: Nov. 26, 2025. [Online]. Available: http://arxiv.org/abs/2410.02777

[66] Y. Bu, J. Lu, and V. V. Veeravalli, "Model Change Detection with Application to Machine Learning," in *ICASSP 2019 - 2019 IEEE International Conference*

on Acoustics, Speech and Signal Processing (ICASSP), May 2019, pp. 5341–5346. doi: 10.1109/ICASSP.2019.8682153.

[67] S. Shekhar and A. Ramdas, "Reducing sequential change detection to sequential estimation." Accessed: Dec. 21, 2023. [Online]. Available: http://arxiv.org/abs/2309.09111

[68] S. Rabanser, S. Günnemann, and Z. Lipton, "Failing Loudly: An Empirical Study of Methods for Detecting Dataset Shift," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2019. Accessed: Nov. 10, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/846c260d715e5b854ffad5f70a516c88-Abstract.html

[69] A. Malinin *et al.*, "Shifts: A Dataset of Real Distributional Shift Across Multiple Large-Scale Tasks," *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, vol. 1, Dec. 2021, Accessed: Nov. 14, 2022. [Online]. Available: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/ad61ab143223efbc24c7d2583be69251-Abstract-round2.html

[70] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, Nov. 1984, doi: 10.1145/1968.1972.

[71] S. Zhao, A. Sinha, Y. He, A. Perreault, J. Song, and S. Ermon, "Comparing Distributions by Measuring Differences that Affect Decision Making," presented at the International Conference on Learning Representations, Jan. 2022. Accessed: June 02, 2023. [Online]. Available: https://openreview.net/forum?id=KB5onONJIAU

[72] M. AI, "The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation." Accessed: Oct. 13, 2025. [Online]. Available: https://ai.meta.com/blog/llama-4-multimodal-intelligence/

[73] W.-L. Chiang *et al.*, "Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference," in *Proceedings of the 41st International Conference on Machine Learning*, PMLR, July 2024, pp. 8359–8388. Accessed: Oct. 13, 2025. [Online]. Available: https://proceedings.mlr.press/v235/chiang24b.html

[74] K. Wiggers, "Meta's benchmarks for its new AI models are a bit misleading." Accessed: Oct. 13, 2025. [Online]. Available: https://techcrunch.com/2025/04/06/metas-benchmarks-for-its-new-ai-models-are-a-bit-misleading/

[75] lmarena.ai, "lmarena on X.com: "We've seen questions from the community about the latest release of Llama-4 on Arena. [...]"." Accessed: Oct. 13, 2025. [Online]. Available: https://x.com/arena/status/1909397817434816562

[76] G. D. Pearson, N. A. Silver, J. Y. Robinson, M. Azadi, B. A. Schillo, and J. M. Kreslake, "Beyond the margin of error: a systematic and replicable audit of the TikTok research API," *Information, Communication & Society*, vol. 28, no. 3, pp. 452–470, Feb. 2025, doi: 10.1080/1369118X.2024.2420032.

[77] D. Zietlow *et al.*, "Leveling Down in Computer Vision: Pareto Inefficiencies in Fair Deep Classifiers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, IEEE, 2022, pp. 10400–10411. doi: 10.1109/CVPR52688.2022.01016.

[78] M. Yaghini, P. Liu, A. Magnuson, N. Dullerud, and N. Papernot, "Trustworthy ML Regulation as a Principal-Agent Problem," in *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, in FAccT '25. New York, NY, USA: Association for Computing Machinery, June 2025, pp. 3291–3302. doi: 10.1145/3715275.3732211.

[79] K. Fukuchi, S. Hara, and T. Maehara, "Faking Fairness via Stealthily Biased Sampling," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, AAAI Press, 2020, pp. 412–419.

[80] V. Lafargue, A. L. Monteiro, E. Claeys, L. Risser, and J.-M. Loubes, "Exposing the Illusion of Fairness: Auditing Vulnerabilities to Distributional Manipulation Attacks." Accessed: Oct. 14, 2025. [Online]. Available: http://arxiv.org/abs/2507.20708

[81] K. Meding and T. Hagendorff, "Fairness Hacking: The Malicious Practice of Shrouding Unfairness in Algorithms," *Philosophy & Technology*, vol. 37, no. 1, p. 4, Jan. 2024, doi: 10.1007/s13347-023-00679-8.

[82] A. Washington, "How To Argue With An Algorithm: Lessons From The Compas-Propublica Debate," *Colorado Technology Law Journal*, vol. 17, no. 1, p. 131, Jan. 2018, [Online]. Available: https://scholar.law.colorado.edu/ctlj/vol17/iss1/4

[83] C. J. Anders, P. Pasliev, A.-K. Dombrowski, K.-R. Müller, and P. Kessel, "Fairwashing explanations with off-manifold detergent," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, in Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 314–323.

[84] U. Aïvodji, H. Arai, S. Gambs, and S. Hara, "Characterizing the risk of fairwashing," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021,*

December 6-14, 2021, virtual, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 14822–14834.

[85] E. Le Merrer and G. Trédan, "Remote Explainability Faces the Bouncer Problem," *Nature Machine Intelligence*, vol. 2, no. 9, pp. 529–539, 2020, doi: 10.1038/s42256-020-0216-z.

[86] A. Godinot, E. Le Merrer, G. Trédan, C. Penzo, and F. Taïani, "Under manipulations, are some AI models harder to audit?," in *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, Apr. 2024, pp. 644–664. doi: 10.1109/SaTML59370.2024.00038.

[87] A. Godinot, G. Tredan, E. Merrer, C. Penzo, and F. Taiani, "Queries, Representation & Detection: the next 100 model fingerprinting schemes," presented at the The 39th Annual AAAI Conference on Artificial Intelligence, Dec. 2024. Accessed: Dec. 01, 2025. [Online]. Available: https://openreview.net/forum?id=rv0kUJses4

[88] J. Garcia Bourrée *et al.*, "Robust ML Auditing using Prior Knowledge," in *Proceedings of the 42nd International Conference on Machine Learning*, PMLR, Oct. 2025, pp. 18794–18810. Accessed: Nov. 05, 2025. [Online]. Available: https://proceedings.mlr.press/v267/garcia-bourree25a.html

[89] A. Godinot, E. L. Merrer, G. Trédan, C. Penzo, and F. Taïani, "Change-Relaxed Active Fairness Auditing," presented at the RJCIA 2023 - 21e Rencontres des Jeunes Chercheurs en Intelligence Artificiel, July 2023, p. 91. Accessed: Dec. 01, 2025. [Online]. Available: https://hal.science/hal-04395914

[90] A. Godinot, E. Le Merrer, G. Trédan, C. Penzo, and F. Taïani, "Under Manipulations, Are Some AI Models Harder to Audit?," in *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2024, pp. 644–664. doi: 10.1109/SaTML59370.2024.00038.

[91] H. Zhao and G. J. Gordon, "Inherent Tradeoffs in Learning Fair Representations," *J. Mach. Learn. Res.*, vol. 23, pp. 57:1–57:26, 2022.

[92] U. Congress, "Algorithmic Accountability Act of 2022." 2022.

[93] E. Union, "Regulation (EU) 2022/1925 of the European Parliament and of the Council of 14 September 2022 on Contestable and Fair Markets in the Digital Sector (Digital Markets Act)." 2022.

[94] J. Crémer *et al.*, "Enforcing the Digital Markets Act: institutional choices, compliance, and antitrust," *Journal of Antitrust Enforcement*, vol. 11, no. 3, pp. 315–349, 2023.

[95] S. Costanza-Chock, I. D. Raji, and J. Buolamwini, "Who Audits the Auditors? Recommendations from a field scan of the algorithmic auditing ecosystem," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022, pp. 1571–1583.

[96] A. S. Shamsabadi *et al.*, "Washing The Unwashable : On The (Im)possibility of Fairwashing Detection," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022.

[97] A. Ng, "Can Auditing Eliminate Bias from Algorithms?." 2021.

[98] M. de Vos *et al.*, "Fairness auditing with multi-agent collaboration," *ECAI 2024*. IOS Press, pp. 1116–1123, 2024.

[99] N. Si, K. Murthy, J. H. Blanchet, and V. A. Nguyen, "Testing Group Fairness via Optimal Transport Projections," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, M. Meila and T. Zhang, Eds., in Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 9649–9659.

[100] B. Taskesen, J. Blanchet, D. Kuhn, and V. A. Nguyen, "A Statistical Test for Probabilistic Fairness," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, in FAccT '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 648–665. doi: 10.1145/3442188.3445927.

[101] C. DiCiccio, S. Vasudevan, K. Basu, K. Kenthapadi, and D. Agarwal, "Evaluating Fairness Using Permutation Tests," in *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds., ACM, 2020, pp. 1467–1477.

[102] J. J. Cherian and E. J. Candès, "Statistical Inference for Fairness Auditing," *Journal of Machine Learning Research*, vol. 25, no. 149, pp. 1–49, 2024.

[103] C. Bénesse, F. Gamboa, J.-M. Loubes, and T. Boissin, "Fairness Seen as Global Sensitivity Analysis," *Machine Learning*, vol. 113, no. 5, pp. 3205–3232, 2024, doi: 10.1007/s10994-022-06202-y.

[104] B. Chugg, S. Cortes-Gomez, B. Wilder, and A. Ramdas, "Auditing Fairness by Betting," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A.

Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., Nov. 2023. Accessed: Jan. 17, 2024. [Online]. Available: https://openreview.net/forum?id=EEVpt3dJQj

[105] P. Maneriker, C. Burley, and S. Parthasarathy, "Online Fairness Auditing through Iterative Refinement," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, A. K. Singh, Y. Sun, L. Akoglu, D. Gunopulos, X. Yan, R. Kumar, F. Ozcan, and J. Ye, Eds., ACM, 2023, pp. 1665–1676. doi: 10.1145/3580305.3599454.

[106] A. Albarghouthi, L. D'Antoni, S. Drews, and A. V. Nori, "FairSquare: Probabilistic Verification of Program Fairness," *Proc. ACM Program. Lang.*, vol. 1, no. OOPSLA, pp. 80:1–80:30, 2017, doi: 10.1145/3133904.

[107] B. Ghosh, D. Basu, and K. S. Meel, "Justicia: A Stochastic SAT Approach to Formally Verify Fairness," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, AAAI Press, 2021, pp. 7554–7563.

[108] B. Ghosh, D. Basu, and K. S. Meel, "Algorithmic Fairness Verification with Graphical Models," in *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, AAAI Press, 2022, pp. 9539–9548.

[109] G. Borca-Tasciuc, X. Guo, S. Bak, and S. Skiena, "Provable Fairness for Neural Network Models Using Formal Verification." 2022.

[110] M. S. Lam, A. Pandit, C. H. Kalicki, R. Gupta, P. Sahoo, and D. Metaxa, "Sociotechnical Audits: Broadening the Algorithm Auditing Lens to Investigate Targeted Advertising," *Proc. ACM Hum.-Comput. Interact.*, vol. 7, no. CSCW2, pp. 360:1–360:37, 2023, doi: 10.1145/3610209.

[111] W. H. Deng, B. Guo, A. Devrio, H. Shen, M. Eslami, and K. Holstein, "Understanding Practices, Challenges, and Opportunities for User-Engaged Algorithm Auditing in Industry Practice," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, A. Schmidt, K. Väänänen, T. Goyal, P. O. Kristensson, A. Peters, S. Mueller, J. R. Williamson, and M. L. Wilson, Eds., in CHI '23. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 1–18. doi: 10.1145/3544548.3581026.

[112] H. Fokkema, R. de Heide, and T. van Erven, "Attribution-Based Explanations That Provide Recourse Cannot Be Robust," *Journal of Machine Learning Research*, vol. 24, no. 360, pp. 1–37, 2023.

[113] G. Laberge, U. Aïvodji, S. Hara, M. Marchand, and F. Khomh, "Fooling SHAP with Stealthily Biased Sampling," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, OpenReview.net, 2023.

[114] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Fooling LIME and SHAP: Adversarial Attacks on Post Hoc Explanation Methods," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, in AIES '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 180–186. doi: 10.1145/3375627.3375830.

[115] S. Tan, R. Caruana, G. Hooker, and Y. Lou, "Distill-and-Compare: Auditing Black-Box Models Using Transparent Model Distillation," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, in AIES '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 303–310. doi: 10.1145/3278721.3278725.

[116] C. Yadav, A. R. Chowdhury, D. Boneh, and K. Chaudhuri, "FairProof : Confidential and Certifiable Fairness for Neural Networks," in *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, OpenReview.net, 2024.

[117] A. S. Shamsabadi *et al.*, "Confidential-PROFITT: Confidential PROof of FaIr Training of Trees," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, OpenReview.net, Feb. 2023. Accessed: Mar. 08, 2023. [Online]. Available: https://openreview.net/forum?id=iIfDQVyuFD

[118] S. Waiwitlikhit, I. Stoica, Y. Sun, T. Hashimoto, and D. Kang, "Trustless Audits without Revealing Data or Models," in *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, OpenReview.net, 2024.

[119] A. Ajarra, B. Ghosh, and D. Basu, "Active Fourier Auditor for Estimating Distributional Properties of ML Models." 2024.

[120] F. Ding, M. Hardt, J. Miller, and L. Schmidt, "Retiring Adult: New Datasets for Fair Machine Learning," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., Curran Associates, Inc., 2021, pp. 6478–6490. Accessed: Feb. 08, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/32e54441e6382a7fbacbbbaf3c450059-Abstract.html

[121] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, IEEE Computer Society, 2015, pp. 3730–3738. doi: 10.1109/ICCV.2015.425.

[122] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[123] S. Caton and C. Haas, "Fairness in Machine Learning: A Survey," *ACM Comput. Surv.*, vol. 56, no. 7, pp. 166:1–166:38, 2024, doi: 10.1145/3616865.

[124] F. Kamiran, A. Karim, and X. Zhang, "Decision theory for discrimination-aware classification," in *2012 IEEE 12th international conference on data mining*, 2012, pp. 924–929.

[125] R. Jiang, A. Pacchiano, T. Stepleton, H. Jiang, and S. Chiappa, "Wasserstein Fair Classification," in *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, A. Globerson and R. Silva, Eds., in Proceedings of Machine Learning Research, vol. 115. AUAI Press, 2019, pp. 862–872.

[126] M. Lohaus, M. Perrot, and U. von Luxburg, "Too Relaxed to Be Fair," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, in Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 6360–6369.

[127] M. Hardt, E. Price, and N. Srebro, "Equality of Opportunity in Supervised Learning," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, Eds., in NIPS'16. Red Hook, NY, USA: Curran Associates Inc., Dec. 2016, pp. 3323–3331.

[128] D. Raji, E. Denton, E. M. Bender, A. Hanna, and A. Paullada, "AI and the Everything in the Whole Wide World Benchmark," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, J. Vanschoren and S. Yeung, Eds., 2021.

[129] G. Mukobi, "Reasons to doubt the impact of AI risk evaluations," *ArXiv preprint*, 2024.

[130] S. Hanneke, "Theory of Disagreement-Based Active Learning," *Foundations and Trends® in Machine Learning*, vol. 7, no. 2–3, pp. 131–309, June 2014, doi: 10.1561/2200000037.

[131] P. Cortez and A. Silva, "Using Data Mining to Predict Secondary School Student Performance," *EUROSIS*, Jan. 2008.

[132] J. Larson, S. Mattu, L. Kirchner, and J. Angwin, "How We Analyzed the COMPAS Recidivism Algorithm," *ProPublica*, May 2016, Accessed: Mar. 06, 2023. [Online]. Available: https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm

[133] "Council Directive 2000/43/EC of 29 June 2000 Implementing the Principle of Equal Treatment between Persons Irrespective of Racial or Ethnic Origin." Accessed: Oct. 12, 2023. [Online]. Available: http://data.europa.eu/eli/dir/2000/43/oj/eng

[134] "Council Directive 2000/78/EC of 27 November 2000 Establishing a General Framework for Equal Treatment in Employment and Occupation." Accessed: Oct. 12, 2023. [Online]. Available: http://data.europa.eu/eli/dir/2000/78/oj/eng

[135] "Council Directive 2004/113/EC of 13 December 2004 Implementing the Principle of Equal Treatment between Men and Women in the Access to and Supply of Goods and Services." Accessed: Oct. 12, 2023. [Online]. Available: http://data.europa.eu/eli/dir/2004/113/oj/eng

[136] "Directive 2006/54/EC of the European Parliament and of the Council of 5~July 2006 on the Implementation of the Principle of Equal Opportunities and Equal Treatment of Men and Women in Matters of Employment and Occupation (Recast)." Accessed: Oct. 12, 2023. [Online]. Available: http://data.europa.eu/eli/dir/2006/54/oj/eng

[137] "Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS." Accessed: Oct. 12, 2023. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206

[138] D. Metaxa *et al.*, "Auditing Algorithms: Understanding Algorithmic Systems from the Outside In," *Foundations and Trends® in Human–Computer Interaction*, vol. 14, no. 4, pp. 272–344, 2021, doi: 10.1561/1100000083.

[139] J. Bandy, "Problematic Machine Behavior: A Systematic Literature Review of Algorithm Audits," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CSCW1, pp. 74:1–74:34, Apr. 2021, doi: 10.1145/3449148.

[140] A. DeVos, A. Dhabalia, H. Shen, K. Holstein, and M. Eslami, "Toward User-Driven Algorithm Auditing: Investigating Users' Strategies for Uncovering Harmful Algorithmic Behavior," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, in CHI '22. New York, NY,

USA: Association for Computing Machinery, Apr. 2022, pp. 1–19. doi: 10.1145/3491102.3517441.

[141] J. Dastin, "Amazon Scraps Secret AI Recruiting Tool That Showed Bias against Women," *Reuters*, Oct. 2018, Accessed: Mar. 06, 2023. [Online]. Available: https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G

[142] T. Calders, F. Kamiran, and M. Pechenizkiy, "Building Classifiers with Independency Constraints," in *2009 IEEE International Conference on Data Mining Workshops*, Dec. 2009, pp. 13–18. doi: 10.1109/ICDMW.2009.83.

[143] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq, "Algorithmic Decision Making and the Cost of Fairness," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, in KDD '17. New York, NY, USA: Association for Computing Machinery, Aug. 2017, pp. 797–806. doi: 10.1145/3097983.3098095.

[144] H. Heidari, M. Loi, K. P. Gummadi, and A. Krause, "A Moral Framework for Understanding Fair ML through Economic Models of Equality of Opportunity," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, in FAT* '19. New York, NY, USA: Association for Computing Machinery, Jan. 2019, pp. 181–190. doi: 10.1145/3287560.3287584.

[145] "Tutorial: 21 Fairness Definitions and Their Politics." Accessed: Oct. 12, 2023. [Online]. Available: https://www.youtube.com/watch?v=jIXIuYdnyyk

[146] B. Rastegarpanah, K. Gummadi, and M. Crovella, "Auditing Black-Box Prediction Models for Data Minimization Compliance," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., Curran Associates, Inc., 2021, pp. 20621–20632. Accessed: Oct. 12, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/ac6b3cce8c74b2e23688c3e45532e2a7-Abstract.html

[147] F. Lu *et al.*, "A General Framework for Auditing Differentially Private Machine Learning," presented at the Advances in Neural Information Processing Systems, Dec. 2022, pp. 4165–4176. Accessed: Aug. 16, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/hash/1add3bbdbc20c403a383482a665eb5a4-Abstract-Conference.html

[148] S. Hanneke, "Teaching Dimension and the Complexity of Active Learning," in *Learning Theory*, N. H. Bshouty and C. Gentile, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2007, pp. 66–81. doi: 10.1007/978-3-540-72927-3_7.

[149] S. Dasgupta, D. Hsu, S. Poulis, and X. Zhu, "Teaching a Black-Box Learner," in *Proceedings of the 36th International Conference on Machine Learning*, PMLR, May 2019, pp. 1547–1555. Accessed: Dec. 12, 2022. [Online]. Available: https://proceedings.mlr.press/v97/dasgupta19a.html

[150] S. Goldwasser, G. N. Rothblum, J. Shafer, and A. Yehudayoff, "Interactive Proofs for Verifying Machine Learning," in *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, J. R. Lee, Ed., in Leibniz International Proceedings in Informatics (LIPIcs), vol. 185. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021, pp. 41:1–41:19. doi: 10.4230/LIPIcs.ITCS.2021.41.

[151] V. N. Vapnik and A. Y. Chervonenkis, "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities," *Theory of Probability & Its Applications*, vol. 16, no. 2, pp. 264–280, Jan. 1971, doi: 10.1137/1116025.

[152] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, 1st ed. Cambridge University Press, 2014. doi: 10.1017/CBO9781107298019.

[153] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding Deep Learning (Still) Requires Rethinking Generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, Feb. 2021, doi: 10.1145/3446776.

[154] P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler, "Benign Overfitting in Linear Regression," *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30063–30070, Dec. 2020, doi: 10.1073/pnas.1907378117.

[155] K. Wang, V. Muthukumar, and C. Thrampoulidis, "Benign Overfitting in Multiclass Classification: All Roads Lead to Interpolation," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2021, pp. 24164–24179. Accessed: Oct. 12, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/caaa29eab72b231b0af62fbdff89bfce-Abstract.html

[156] L. Arnould, C. Boyer, and E. Scornet, "Is Interpolation Benign for Random Forest Regression?," in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, PMLR, Apr. 2023, pp. 5493–5548. Accessed: Oct. 03, 2023. [Online]. Available: https://proceedings.mlr.press/v206/arnould23a.html

[157] T. M. Mitchell, "Generalization as Search," *Artificial Intelligence*, vol. 18, no. 2, pp. 203–226, Mar. 1982, doi: 10.1016/0004-3702(82)90040-6.

[158] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling Modern Machine-Learning Practice and the Classical Bias–Variance Trade-Off," *Proceedings*

of the National Academy of Sciences, vol. 116, no. 32, pp. 15849–15854, Aug. 2019, doi: 10.1073/pnas.1903070116.

[159] S. Buschjäger and K. Morik, "There Is No Double-Descent in Random Forests." Accessed: Oct. 03, 2023. [Online]. Available: http://arxiv.org/abs/2111.04409

[160] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why Do Tree-Based Models Still Outperform Deep Learning on Typical Tabular Data?," in *Advances in Neural Information Processing Systems*, Dec. 2022, pp. 507–520. Accessed: May 10, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/hash/0378c7692da36807bdec87ab043cdadc-Abstract-Datasets_and_Benchmarks.html

[161] J. Dressel and H. Farid, "The Accuracy, Fairness, and Limits of Predicting Recidivism," *Science Advances*, vol. 4, no. 1, p. eaao5580, Jan. 2018, doi: 10.1126/sciadv.aao5580.

[162] A. Agarwal, A. Beygelzimer, M. Dudik, J. Langford, and H. Wallach, "A Reductions Approach to Fair Classification," in *Proceedings of the 35th International Conference on Machine Learning*, PMLR, July 2018, pp. 60–69. Accessed: Jan. 17, 2023. [Online]. Available: https://proceedings.mlr.press/v80/agarwal18a.html

[163] U. Ojha, Y. Li, and Y. J. Lee, "Towards Universal Fake Image Detectors That Generalize Across Generative Models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24480–24489.

[164] Z. He, T. Zhang, and R. Lee, "Sensitive-Sample Fingerprinting of Deep Neural Networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 4724–4732. doi: 10.1109/CVPR.2019.00486.

[165] Y. Li, Z. Zhang, B. Liu, Z. Yang, and Y. Liu, "ModelDiff: Testing-Based DNN Similarity Comparison for Model Reuse Detection," in *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, in ISSTA 2021. New York, NY, USA: Association for Computing Machinery, July 2021, pp. 139–151. doi: 10.1145/3460319.3464816.

[166] J. Guan, J. Liang, and R. He, "Are You Stealing My Model? Sample Correlation for Fingerprinting Deep Neural Networks," in *Advances in Neural Information Processing Systems*, Dec. 2022, pp. 36571–36584.

[167] H. Ben-Sasson and S. Tzadik, "Isolation or Hallucination? Hacking AI Infrastructure Providers for Fun and Weights." Las Vegas, Aug. 2024.

[168] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, "High Accuracy and High Fidelity Extraction of Neural Networks," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 1345–1362.

[169] J.-B. Truong, P. Maini, R. J. Walls, and N. Papernot, "Data-Free Model Extraction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4771–4780.

[170] J. Zhao, Q. Hu, G. Liu, X. Ma, F. Chen, and M. M. Hassan, "AFA: Adversarial Fingerprinting Authentication for Deep Neural Networks," *Computer Communications*, vol. 150, pp. 488–497, Jan. 2020, doi: 10.1016/j.comcom.2019.12.016.

[171] X. Pan, M. Zhang, Y. Lu, and M. Yang, "TAFA: A Task-Agnostic Fingerprinting Algorithm for Neural Networks," in *Computer Security – ESORICS 2021*, E. Bertino, H. Shulman, and M. Waidner, Eds., Cham: Springer International Publishing, 2021, pp. 542–562. doi: 10.1007/978-3-030-88418-5_26.

[172] X. Cao, J. Jia, and N. Z. Gong, "IPGuard: Protecting Intellectual Property of Deep Neural Networks via Fingerprinting the Classification Boundary," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, in ASIA CCS '21. New York, NY, USA: Association for Computing Machinery, June 2021, pp. 14–25. doi: 10.1145/3433210.3437526.

[173] Z. Peng, S. Li, G. Chen, C. Zhang, H. Zhu, and M. Xue, "Fingerprinting Deep Neural Networks Globally via Universal Adversarial Perturbations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13430–13439.

[174] N. Lukas, Y. Zhang, and F. Kerschbaum, "Deep Neural Network Fingerprinting by Conferrable Adversarial Examples," in *International Conference on Learning Representations*, Oct. 2020.

[175] S. Wang and C.-H. Chang, "Fingerprinting Deep Neural Networks - a DeepFool Approach," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2021, pp. 1–5. doi: 10.1109/ISCAS51556.2021.9401119.

[176] J. Chen *et al.*, "Copy, Right? A Testing Framework for Copyright Protection of Deep Learning Models," in *2022 IEEE Symposium on Security and Privacy (SP)*, May 2022, pp. 824–841. doi: 10.1109/SP46214.2022.9833747.

[177] J. Song, Z. Xu, S. Wu, G. Chen, and M. Song, "ModelGiF: Gradient Fields for Model Functional Distance," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 6125–6135.

[178] H. Jia, H. Chen, J. Guan, A. S. Shamsabadi, and N. Papernot, "A Zest of LIME: Towards Architecture-Independent Model Distances," in *Interna-*

tional Conference on Learning Representations, Mar. 2022. Accessed: Jan. 12, 2023. [Online]. Available: https://openreview.net/forum?id=OUz_9TiTv9j

[179] X. Pan, Y. Yan, M. Zhang, and M. Yang, "MetaV: A Meta-Verifier Approach to Task-Agnostic Model Fingerprinting," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, in KDD '22. New York, NY, USA: Association for Computing Machinery, Aug. 2022, pp. 1327–1336. doi: 10.1145/3534678.3539257.

[180] T. Maho, T. Furon, and E. Le Merrer, "Fingerprinting Classifiers With Benign Inputs," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 5459–5472, 2023, doi: 10.1109/TIFS.2023.3301268.

[181] O. Goldreich, *Introduction to Property Testing*, 1st ed. Cambridge University Press, 2017. doi: 10.1017/9781108135252.

[182] E. Le Merrer, P. Pérez, and G. Trédan, "Adversarial Frontier Stitching for Remote Neural Network Watermarking," *Neural Computing and Applications*, vol. 32, no. 13, pp. 9233–9244, July 2020, doi: 10.1007/s00521-019-04434-z.

[183] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," in *International Conference on Learning Representations*, Feb. 2018.

[184] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574–2582.

[185] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, in KDD '16. New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 1135–1144. doi: 10.1145/2939672.2939778.

[186] D. Oliynyk, R. Mayer, and A. Rauber, "I Know What You Trained Last Summer: A Survey on Stealing Machine Learning Models and Defences," *ACM Computing Surveys*, vol. 55, no. 14s, pp. 324:1–324:41, July 2023, doi: 10.1145/3595292.

[187] C. Franzen, "Mistral CEO Confirms `Leak' of New Open Source AI Model Nearing GPT-4 Performance." Jan. 2024.

[188] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks," in *Research in Attacks, Intrusions, and Defenses*, M. Bailey, T. Holz, M. Stamatogiannakis, and S.

Ioannidis, Eds., Cham: Springer International Publishing, 2018, pp. 273–294. doi: 10.1007/978-3-030-00470-5_13.

[189] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning Filters for Efficient ConvNets," in *International Conference on Learning Representations*, 2017.

[190] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified Adversarial Robustness via Randomized Smoothing," in *Proceedings of the 36th International Conference on Machine Learning*, PMLR, May 2019, pp. 1310–1320.

[191] M. Tang *et al.*, "MODELGUARD: Information-Theoretic Defense Against Model Extraction Attacks," in *33rd USENIX Security Symposium (USENIX Security 24)*, Philadelphia, PA: USENIX Association, Aug. 2024.

[192] T. Orekondy, B. Schiele, and M. Fritz, "Prediction Poisoning: Towards Defenses Against DNN Model Stealing Attacks," in *International Conference on Learning Representations*, Sept. 2019.

[193] F. Boenisch, "A Systematic Review on Model Watermarking for Neural Networks," *Frontiers in Big Data*, vol. 4, p. 729663, Nov. 2021, doi: 10.3389/fdata.2021.729663.

[194] F. Regazzoni, P. Palmieri, F. Smailbegovic, R. Cammarota, and I. Polian, "Protecting Artificial Intelligence IPs: A Survey of Watermarking and Fingerprinting for Machine Learning," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 2, pp. 180–191, 2021, doi: 10.1049/cit2.12029.

[195] T. Gu, B. Dolan-Gavitt, and S. Garg, "BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain," no. arXiv:1708.06733. arXiv, Mar. 2019.

[196] A. Rida, M.-J. Lesot, X. Renard, and C. Marsala, "Dynamic Interpretability for Model Comparison via Decision Rules," no. arXiv:2309.17095. arXiv, Sept. 2023. doi: 10.48550/arXiv.2309.17095.

[197] R. Nair, M. Mattetti, E. Daly, D. Wei, O. Alkan, and Y. Zhang, "What Changed? Interpretable Model Comparison," in *Twenty-Ninth International Joint Conference on Artificial Intelligence*, Aug. 2021, pp. 2855–2861. doi: 10.24963/ijcai.2021/393.

[198] D. Hendrycks *et al.*, "Measuring Massive Multitask Language Understanding," presented at the International Conference on Learning Representations, Oct. 2020. Accessed: Nov. 18, 2025. [Online]. Available: https://openreview.net/forum?id=d7KBjmI3GmQ

[199] H. Zhang *et al.*, "A Careful Examination of Large Language Model Performance on Grade School Arithmetic," in *Advances in Neural Information Processing Systems*, Dec. 2024, pp. 46819–46836. doi: 10.52202/079017-1485.

[200] A. Amini, S. Gabriel, S. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi, "MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 2357–2367. doi: 10.18653/v1/N19-1245.

[201] P. Lahoti *et al.*, "Fairness without demographics through adversarially reweighted learning," *Advances in neural information processing systems*, vol. 33, pp. 728–740, 2020, Accessed: Nov. 29, 2025. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/hash/07fc15c9d169ee48573edd749d25945d-Abstract.html

[202] P. J. Kenfack, S. E. Kahou, and U. Aïvodji, "A Survey on Fairness Without Demographics," *Transactions on Machine Learning Research*, Feb. 2024, Accessed: Aug. 22, 2024. [Online]. Available: https://openreview.net/forum?id=3HE4vPNIfX

[203] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, in ICML'17. Sydney, NSW, Australia: JMLR.org, Aug. 2017, pp. 1885–1894.

[204] M. Hardt, E. Mazumdar, C. Mendler-Dünner, and T. Zrnic, "Algorithmic Collective Action in Machine Learning," in *Proceedings of the 40th International Conference on Machine Learning*, PMLR, July 2023, pp. 12570–12586. Accessed: May 12, 2025. [Online]. Available: https://proceedings.mlr.press/v202/hardt23a.html

[205] Y. Allouah, R. Guerraoui, L.-N. Hoang, and O. Villemaud, "Robust Sparse Voting." Accessed: June 15, 2023. [Online]. Available: http://arxiv.org/abs/2202.08656

[206] M. Aliakbarpour, A. Burudgunte, C. Canonne, and R. Rubinfeld, "Better Private Distribution Testing by Leveraging Unverified Auxiliary Data," in *Proceedings of Thirty Eighth Conference on Learning Theory*, PMLR, July 2025, pp. 22–63. Accessed: Nov. 18, 2025. [Online]. Available: https://proceedings.mlr.press/v291/aliakbarpour25a.html

[207] V. Kilian, S. Cortinovis, and F. Caron, "Anytime-valid, Bayes-assisted, Prediction-Powered Inference," presented at the The Thirty-ninth Annual

Conference on Neural Information Processing Systems, Oct. 2025. Accessed: Nov. 18, 2025. [Online]. Available: https://openreview.net/forum?id=IhOgbtClHL&referrer=%5Bthe+profile+of+Francois+Caron%5D%28%2Fprofile%3Fid%3D%7EFrancois_Caron1%29

[208] J. Thaler, "Proofs, Arguments, and Zero-Knowledge," *Foundations and Trends® in Privacy and Security*, vol. 4, no. 2, pp. 117–660, Dec. 2022, doi: 10.1561/3300000030.

[209] J. Hoffstein, J. Pipher, and J. H. Silverman, *An Introduction to Mathematical Cryptography*. in Undergraduate Texts in Mathematics. New York, NY: Springer, 2014. doi: 10.1007/978-1-4939-1711-2.

[210] M. Foucault, *Surveiller et punir. Naissance de la prison*. Gallimard, 2014. Accessed: Nov. 18, 2025. [Online]. Available: https://shs.cairn.info/surveiller-et-punir-naissance-de-la-prison%E2%80%939782070729685

[211] F. Tréguer, "Pouvoir et résistance dans l'espace public : une contre-histoire d'Internet (XVe -XXIe siècle)," Theses, 2017. Accessed: Nov. 18, 2025. [Online]. Available: https://shs.hal.science/tel-01631122

[212] P. R. Kalluri, W. Agnew, M. Cheng, K. Owens, L. Soldaini, and A. Birhane, "Computer-vision research powers surveillance technology," *Nature*, vol. 643, no. 8070, pp. 73–79, July 2025, doi: 10.1038/s41586-025-08972-6.

[213] C. Doctorow, "The 'Enshittification' of TikTok," *Wired*, Jan. 2023, Accessed: Nov. 13, 2025. [Online]. Available: https://www.wired.com/story/tiktok-platforms-cory-doctorow/

[214] J. Naughton, "Users, advertisers – we are all trapped in the 'enshittification' of the internet," *The Guardian*, Mar. 2023, Accessed: Nov. 13, 2025. [Online]. Available: https://www.theguardian.com/commentisfree/2023/mar/11/users-advertisers-we-are-all-trapped-in-the-enshittification-of-the-internet

[215] M. Tkacik, "Crash Course," *The New Republic*, Sept. 2019, Accessed: Nov. 13, 2025. [Online]. Available: https://newrepublic.com/article/154944/boeing-737-max-investigation-indonesia-lion-air-ethiopian-airlines-managerial-revolution

[216] NIST, "NIST Digital Library of Mathematical Functions." 2013.

[217] S. Li, "Concise formulas for the area and volume of a hyperspherical cap," *Asian Journal of Mathematics & Statistics*, vol. 4, no. 1, pp. 66–70, 2010.

[218] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel Dataset for Fine-Grained Image Categorization," in *First Workshop on Fine-Grained*

Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, June 2011.

[219] M.-E. Nilsback and A. Zisserman, "Automated Flower Classification over a Large Number of Classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, Dec. 2008, pp. 722–729. doi: 10.1109/ICVGIP.2008.47.

[220] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," technical report, Apr. 2009.

Défense à venir, le 10 février 2026 à RENNES.

Jury

Présidente	Mme Aline **Roumy** *INRIA*	Directrice de Recherche
Rapporteur	M. Damien **Garreau** *University of Würzburg*	Professor
Rapporteur	M. Aurélien **Bellet** *INRIA*	Directeur de Recherche
Examinateur	M. Nicolas **Papernot** *University of Toronto*	Assistant Professor
Directeur de thèse	M. Gilles **Trédan** *LAAS-CNRS*	Directeur de Recherche
Directeur de thèse	M. Erwan **Le Merrer** *INRIA*	Chercheur

Invités

Directeur de thèse	M. François **Taïani** *Université de Rennes, IRISA*	Professeur
Encadrante	Mme. Gohar **Dashyan** *PEReN*	Lead Recherche et Éthique