Change-Relaxed Active Fairness Auditing

A. Godinot[1,2,3], E. Le Merrer[2], C. Penzo[3], F. Taïani[1], G. Tredan[4]

[1] Université de Rennes
[2] Centre Inria de l'Université de Rennes
[3] Pôle d'Expertise de la Régulation Numérique
[4] LAAS/CNRS

Abstract

The pervasive deployment of user-facing automated decisions systems raises concerns over their impact on society. The sheer amount of such online platforms and their growing complexity highlights the need for automated and robust audits to assess their impact on users. This paper focuses on a recent theoretical advance named manipulation-proofness. It aims at guaranteeing successive audits of a platform cannot be gamed by the platform, provided the labels returned on the audit dataset do not change.
While this constitutes a decisive step for reliable audits, it is too restrictive, as models naturally evolve with time in practice. This paper thus explores how manipulation-proofness can be adapted to better fit actual scenarios, by studying the effects of relaxing the constraint on the amount of change the remote model can operate while being audited. Our results on the COMPAS dataset demonstrate a request gain in one of the two models considered, while also noticing the surprisingly good performance of the random strawman approach. We believe this constitutes an interesting step for further attempts to improve reliable and manipulation proof audits.

Keywords

Artificial Intelligence, Algorithmic Auditing, Black-box models, Active Learning

1 Introduction

The pervasive deployment of user-facing automated decisions systems raises concerns over their impact on society. The sheer amount of such online platforms and their growing complexity highlights the need for automated and robust audits to assess their impact on users. The advent of highly publicized audits, such as ProPublica's story on COMPAS [12] or Reuters study on Amazon's recruiting tool [8], has led to the algorithmic audit field gaining significant traction. For the public to trust Artificial Intelligence (AI) systems, and more broadly algorithmic decision systems, we need methods to explain the decision of such systems [19, 13], certify their implementation [22, 20] and automatically and robustly detect misconduct [14, 18].
Inspired by "traditional" financial audits, we focus in this work on *external certification audits*. In this type of audit, an external auditor (e.g. a regulator, or an auditing company) is commissioned by a platform to certify some desirable property (the absence of bias, for example) of its system. The system consists in a Machine Learning (ML) model h^* (see subsection 3.1) which is accessed by users through an interface (e.g. a web-page or an Application Programming Interface). To restrict the scope of this work, we consider that h^* is a binary classifier. Furthermore, we assume that the answers presented through the interface are faithful to that of the model h^*. We assume that the platform does not give access to the weights or implementation of the model h^*. The goal of the auditor is thus to certify the system as implemented and as seen by the users. The only information the auditor knows about the audited system is the hypothesis class $\mathcal{H}$ of the model $h^* \in \mathcal{H}$. We dub this setting *remote black-box certification*. Yan and Zhang [22] recently proposed a theoretical framework to model the problem of remote black-box auditing. They provide an algorithm to select a minimal set of points S to estimate a property $\mu(h^*)$ (demographic parity for example) of the remote model h^*. While the model h^* behind the API is allowed to change after the audit, the auditor is guaranteed that the value $\mu(h)$ of any model $h \in \mathcal{H}$ that agrees with h^* on S will be close to their estimation $\widehat{\mu}(h^*)$. This new estimation problem coined *manipulation-proof estimation* by Yan and Zhang is a step towards robust auditing as it provides a framework amenable to theoretical analysis. In practice, even if the type of model stays the same, because of retraining, arrival of new users or small tweaks, models served by platforms change over time. Thus, the requirement that the output of the API on the audit points does not change is too restrictive in practice. Moreover, Yan and Zhang only experimented with linear models on small datasets. In this work, we relax this recent formalization and empirically analyze its performance.

Contributions. This paper makes the following contributions. It first reviews the algorithmic audit setup, and recalls the concept and shortcomings of manipulation proofness (Sections 1 and 3). It then proposes a relaxation (coined r-AFA+) on the tolerated audit errors. We then evaluate this relaxation in Section 5, with two model classes and the COMPAS dataset, before we conclude.

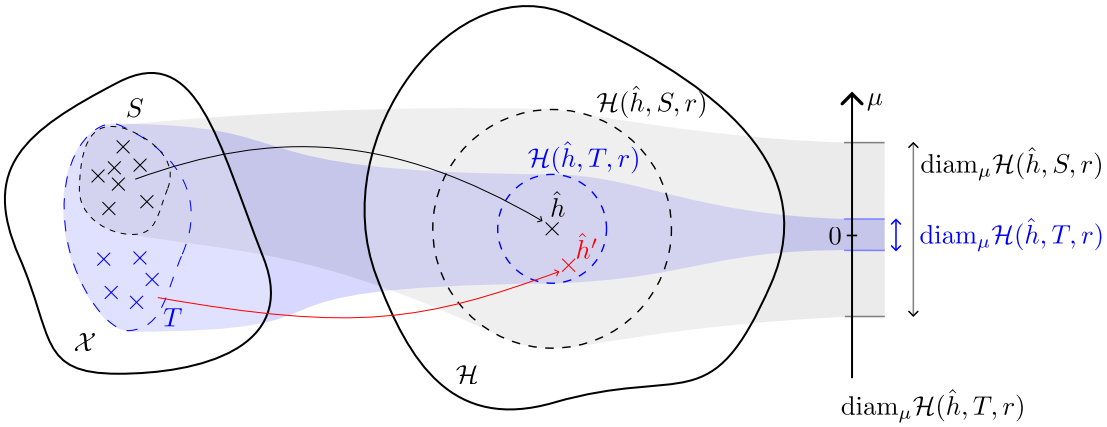

Figure 1: Schematic overview of the r-AFA+ algorithm. On the left, S and T are subsets of the input set $\mathcal{X}$. The goal of the approach is to identify a small set of inputs (left) that quickly reduce the version space (center) to models having close μ-values (right). Those inputs are iteratively improved to refine an estimate of the property of interest (μ-value).

At each outer iteration (line 2), the algorithm consists in three steps. **1. (black)** Train a surrogate $\hat{h}$ on the current labeled dataset $(S, h^*(S))$. **2. (blue)** Find dataset T that minimizes the μ-diameter of the version space. **3. (red)** Merge T in S, query the API on T to train a new surrogate $\hat{h}'$.

2 Related work

AI audits. The AI audit field seeks to understand Artificial Intelligence (AI) algorithms as part of a larger socio-technical system. Most of the published audits include two phases (see for example [2, 15]). First, the auditor analyses the context of the algorithm: the training data, the users or the team who built said system. Then, they typically perform a statistical study to discover potential biases in the algorithm's output. Recently, efforts have been made to formalize requirements dictated by regulatory bodies (such as the Data Minimization Principle [18]) and provide algorithms to help their enforcement. One challenge of remote black-box auditing is to limit the number of queries used to perform the audit. Issuing too few queries prevents any meaningful analysis but if the auditor requests large bursts of queries, they risk being blacklisted.

Robust auditing. Audits relying on statistical studies often make the simplifying assumption that the audited platform is cooperative and honest. While very practical, this is overly optimistic since there were examples of companies trying to evade high-stakes audits in the past [11]. Without this assumption, many simple audits become theoretically impossible [21]. To overcome this limit, the notion of manipulation-proof estimation has recently emerged [22]. Intuitively, this approach aims at constructing an auditing procedure that is resilient to arbitrary manipulation by the auditee while making as few assumptions as possible on the audited target.

Distribution testing. The field of tolerant distribution testing is interested in answering the question: given samples from an unknown distribution p, is this distribution ϵ_1-close ($min_{q\in\mathcal{P}}(d, q) \leq \epsilon_1$) to the set of distributions $\mathcal{P}$ or is it ϵ_2-far ($min_{q\in\mathcal{P}}(d, q) \geq \epsilon_2$) from it? Some fairness measures (such as demographic fairness) can be formulated as the independence between the output of positive labels (e.g. granting a loan, recruiting a new employee) and sensitive attributes (e.g. gender, ethnicity, religious beliefs, political views). Thus, as a specific tolerant distribution test, testing for independence could certify demographic parity. For an introduction to distribution testing and its extension to tolerant distribution testing, refer to [4] and [5]. It is however not straightforward how to certify other fairness properties.

Active learning. Active learning is a form of interactive learning where the learning algorithm can iteratively select the examples to train on. At each training step, the learning algorithm can use the performance of the trained model and past training examples to decide which training example to select next in order to optimize the learning process. The literature on active learning proved that interacting with the trained model could dramatically reduce learning sample complexity [10]. Even if the trained model is treated as a black box, it is possible to iteratively select training points based on the model's outputs to reduce the number of training points needed [7]. These methods cannot be directly applied to black-box remote auditing because they need to probe the model on the whole dataset at each iteration, whereas we try to minimize the number of queries to said model. Yet, the method we present in this work builds on this idea to select the audit dataset by interacting with a *surrogate* of the API instead of the API itself. (We discuss this notion of surrogate model in more detail in Section 3.)

3 Manipulation proofness with AFA

In this section, we present in more detail the notion of *manipulation-proof estimation* introduced by Yan and Zhang [22] in the context of *remote black-box auditing*. We first introduce some key notations and assumptions and formalize the auditing process as a game between the auditor and the platform. We then define the notion of manipulation-proof estimation and present the general intu-

ition behind the AFA algorithm proposed by Yan and Zhang to solve the auditing game in a manipulation-proof manner.

3.1 Notations and assumptions

We consider a platform that seeks to solve a classification task (e.g. whether or not to grant a loan) based on user features (some information regarding the prospective borrower) grouped in a vector $x \in \mathcal{X}$. We assume that the space of all possible inputs—the *sample space* $\mathcal{X}$—is finite. Should $\mathcal{X}$ not be finite, it is suffices to sample a fixed number of instances in $\mathcal{X}$ and treat them as a finite sample space. As explained in [7], it is then possible to adapt the bounds obtained for a finite $\mathcal{X}$.

When the platform trains its model, it effectively chooses a hypothesis $h^* \in \mathcal{H}$ in a set of possible (deterministic) models—the *hypothesis space* $\mathcal{H}$. The hypothesis space could for example be the set of linear binary classifiers on $\mathcal{X}$. Since the platform solves a classification task, the set of possible outputs—the *output space* $\mathcal{Y}$—is also finite. Therefore, because $\mathcal{X}$ and $\mathcal{Y}$ are finite, the space $\mathcal{Y}^{\mathcal{X}}$ of all functions from $\mathcal{X}$ to $\mathcal{Y}$ (and by extension $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$) is also finite. For any set S, we write $|S|$ its cardinal and $\mathcal{P}(S)$ the set of all its subsets.

The auditor seeks to test whether the model h^* used by the platform respects some desirable property $\mu : (\mathcal{P}(\mathcal{X}), \mathcal{H}) \to \mathbb{R}$. For simplicity, we use the notation abuse $\mu(h) = \mu(\mathcal{X}, h)$. Define the *demographic parity* μ_{DP} as

$$\mu_{\text{DP}}(S,h) = \frac{1}{|S \cap A|} \sum_{x \in S \cap A} \mathbb{1}\{h(x) = 1\} - \frac{1}{|S \cap A^c|} \sum_{x \in S \cap A^c} \mathbb{1}\{h(x) = 1\} \quad (1)$$

where $\mathbb{1}\{P\}$ is the indicator function for the predicate P, A is the set of samples in $\mathcal{X}$ with a positive sensitive attribute and A^c its complementary in $\mathcal{X}$.

We define the Hamming distance $d_H(h(S), h^*(S)) = |x \in S : h(x) \neq h^*(x)|$. Simply put, the Hamming distance is the number of points in S on which two hypotheses (or models) h and h^* disagree. Finally, for any subset of the hypothesis class $V \subset \mathcal{H}$, the *μ-diameter* $\text{diam}_\mu(V)$ is the largest difference in the value of μ between any two models in V.

$$\text{diam}_\mu(V) = \max_{h,h' \in V} |\mu(h) - \mu(h')| \quad (2)$$

3.2 The Auditing Game

The auditing process can be modeled as a game between the auditor and the audited platform. First, the auditor decides on the fairness measure $\mu : (\mathcal{P}(\mathcal{X}), \mathcal{H}) \to \mathbb{R}$ on which they want to evaluate the platform. We assume the auditor can directly query the platform's output on a given input $x \in \mathcal{X}$, either through an API or through scraping. By gathering outputs on well-chosen inputs, the auditor seeks to construct an *audit dataset* $S \subseteq \mathcal{X}$, which the auditor will then use to estimate how well the platform respects the fairness measure μ, by using $\widehat{\mu} = \mu(S, h^*)$ as an estimation of the true value $\mu(\mathcal{X}, h^*)$. In practice, platforms regularly retrain their model h^*, for instance to account for new users or to improve it. As a result, h^* is likely to evolve after it has been audited. Thus, constructing a robust estimator intuitively means constructing an estimator that does not change too much even if the model is slightly modified after the audit. More formally, this auditing game can be described as follows:

Phase 1. At time t_0, the auditor constructs an audit dataset $S \subset \mathcal{X}$ to build its estimator $\widehat{\mu}(S, h^*_{t_0})$ by interacting with the model $h^*_{t_0}$ served by the platform. The time needed to construct the audit dataset is supposed to be negligible and the remote model is assumed not to change during this phase.

Phase 2. At any time $t > t_0$ after the audit, and for any reason (retraining, new user or even adversarial change), we allow the model to change slightly. By re-querying the new model h^*_t on the same dataset S, we verify that answers to queries in S have not changed $d_H\left(h^*_t(S), h^*_{t_0}(S)\right) = 0$. The auditor's goal is to detect through their estimation $\widehat{\mu}(S, h^*_t)$ when $\mu(\mathcal{X}, h^*_t)$ deviates too much from some target boundary, in which case the certificate must be revoked.

3.3 Manipulation-proof estimation.

The auditor's algorithm solves the above auditing game if it can produce an auditing set S such that for all $h \in \mathcal{H}$, if $d(h(S), h^*_t(S)) = 0$, then the μ-value of h cannot be at a distance larger than ϵ to $h^*_{t_0}$. To formalize the auditor's goal, we first define the notion of *version space*. It is the set of models h whose output agree with that of h^* on S.

$$\mathcal{H}(S, h^*) = \{h \in \mathcal{H} \ : \ d(h(S), h^*(S)) = 0\} \quad (3)$$

Then, an estimator $\widehat{\mu}(S, h^*)$ of $\mu(\mathcal{X}, h^*)$ is said (r, ϵ)-*manipulation-proof* i.i.f.

$$\text{diam}_\mu \mathcal{H}(S, h^*) < \epsilon \quad (4)$$

The auditor only queries the labels $h^*(x)$ of points $x \in S$ therefore, they can only base their estimation $\widehat{\mu}(S, h^*)$ of μ on $(S, h^*(S))$. Multiple models in $\mathcal{H}$ can have the same answers on S and the auditor does not have any means to know which one of them is behind the API. Thus, there is an uncertainty on the true value $\mu(\mathcal{X}, h^*)$. The μ-diameter evaluates how well different audit datasets S might lead to a smaller/larger uncertainty on $\mu(\mathcal{X}, h^*)$. In their paper, Yan and Zhang frame the auditing game as a minimax game and prove a lower-bound on the number of queries required to reach ϵ-manipulation proofness. Inspired by the *Multiplicative Weight Update* method [1], they provide a randomized approximate algorithm AFA (Active Fairness Auditing, see our adapted version algorithm 1) to compute a solution with a query competitive ratio of $\mathcal{O}(\log(\mathcal{H}) \log(\mathcal{X}))$.

We present in algorithm 1 the core structure of the algorithm proposed in [22] with the modifications discussed in Section 4. The intuition behind this algorithm is to use the black-box teaching algorithm introduced in [7]. To avoid probing the API on the entire $\mathcal{X}$, we assume that we have access to an oracle $\mathcal{O} : \mathcal{P}(\mathcal{X}) \to \mathcal{H}$ providing surrogates of

h^* trained on S (line 4). The oracle $\mathcal{O}$ is assumed to be mistake bounded, that is there exits $M > 0$ such that for any sequence $(x_i)_i$ of points from $\mathcal{X}$ and their corresponding labels $(y_i)_i$, $\sum_{k=1}^{+\infty} \mathbb{1}\{\mathcal{O}((x_1, \ldots, x_k))(x_k) \neq y_k\} \leq M$. Example of such oracles include the perceptron algorithm [16] and the halving algorithm [3]. This surrogate is then used with the black-box teaching algorithm in [7] to find a subset T of $\mathcal{X}$ maximizing $\text{diam}_\mu \mathcal{H}(\mathcal{O}(S), T)$.

4 Robust auditing in practice: giving some slack on the radius

Because our goal is to systematically analyze the performance of [22] in practice, we had to modify it to account for more realistic settings. The original AFA algorithm requires that after the audit, the labels of the queried points must not change. We argue that this assumption needs to be relaxed for two reasons. First, as we said for practical reasons the model might change slightly over time, modifying a small fraction of labels. Second, the auditor might not have access to an exact description of $\mathcal{H}$. This implies that the hypothesis space used for the audit $\mathcal{H}_{\text{surrogate}}$ does not match the one used by the platform $\mathcal{H}_{\text{API}}$. Because of this mismatch, the return condition line 19 might never be met. For these reasons, we relaxed the condition to $d(h(S), h^*(S)) < r$ in the definition of the version space (Equation 3) and adapted the algorithm accordingly. We name this method r-AFA+ (r-radius Active Fairness Auditing) and provide the pseudo-code in algorithm 1.

$$\mathcal{H}(S, h^*, r) = \{h \in \mathcal{H} \ : \ d(h(S), h^*(S)) \leq r\} \qquad (5)$$

Theoretically, it is still unclear how these allowed errors might influence the query complexity of r-AFA+ compared to AFA. On one hand it definitely increases the cardinal of the version space, potentially increasing the μ-diameter for a given budget, requiring a larger T at each inner iteration. On the other hand, the exit condition of the algorithm (line 19) is less restrictive and decreases the number of outer iterations (and thus the total number of queries in S). As this is still preliminary work, we leave a more in-depth analysis of r-AFA+ for future work, and discuss the empirical results that follow from this relaxation in the next section.

5 Evaluation

We now quantify the impact of relaxing AFA with a tolerance radius of r changes, with r-AFA+.

Implementation Based on the notebooks provided in the supporting material of [22] we reimplemented algorithm 1 (with our modifications discussed in section 4). The implementation differs from the pseudocode in two ways. First, we modify the termination of the algorithm. The joint requirements for termination of estimated μ-diameter smaller that ϵ (line 10) and surrogate/API agreement (line 19) are replaced by the condition that $|S|$ does not exceed the budget. Second, instead of querying all the points in T (line 17) we only query one of them and re-enter the inner loop.

Algorithm 1 Remote black-box certification with r-AFA+

Require: Hypothesis class $\mathcal{H}$, mistake M-bounded oracle $\mathcal{O}$, target error ϵ, property μ, confidence δ, radius r

Ensure: audit dataset S

1: $S \leftarrow \emptyset$
2: **while** true **do**
3: $\quad T \leftarrow \emptyset$
4: $\quad \hat{h} \leftarrow \mathcal{O}(S)$
5: $\quad w(x) \leftarrow \frac{1}{|\mathcal{X}|}, \ \forall x \in \mathcal{X}$
6: $\quad \tau(x) \sim \text{Exp}\left(\ln\left(\frac{M}{\delta}\,|\mathcal{H}|^2\right)\right)$
7: $\quad$ **while** true **do**
8: $\quad\quad$ ▷ *Estimate the μ-diameter of the current version space* ◁
9: $\quad\quad (h_{\min}, h_{\max}) \leftarrow \arg\min / \max_h \mu(h)$ s.t. $d(h(T), \hat{h}(T)) \leq r$
10: $\quad\quad$ **if** $\mu(h_{\max}) - \mu(h_{\min}) < \epsilon$ **then**
11: $\quad\quad\quad$ **break**
12: $\quad\quad \Delta(h_{\max}, h_{\min}) = \left\{x \in \mathcal{X} \ : \ h_{\max}(x) \neq \hat{h}(x) \text{ or } h_{\min}(x) \neq \hat{h}(x)\right\}$
13: $\quad\quad$ ▷ *Multiplicative weight update* ◁
14: $\quad\quad$ **while** $\sum_{x \in \Delta(h_{\max}, h_{\min})} w(x) \leq 1$ **do**
15: $\quad\quad\quad w(x) \leftarrow 2\, w(x), \ \forall x \in \Delta(h_{\max}, h_{\min})$
16: $\quad\quad T \leftarrow \{x \in \mathcal{X} \ : \ w(x) \geq \tau(x)\}$
17: $\quad$ query h^* on T
18: $\quad S \leftarrow S \cup T$
19: $\quad$ **if** $\hat{h} \in \mathcal{H}(h^\star, S, r)$ **then**
20: $\quad\quad$ **return** S

Dataset We run our experiments on three datasets : student performance [6], COMPAS [12] and the reconstructed adults dataset [9]. In this preliminary version of our work, we only showcase results on the COMPAS dataset. COMPAS is a tool used by the US Department of Justice to evaluate the risk of recidivism among defendants, based on individual features such as age, gender, localization, origins amongst others. The COMPAS dataset consists in a list of 6172 defendants with their individual features and recidivism status.

Classifier model We run our experiments with multiple API hypotheses classes adapted to the classification task on tabular data: linear regression, support vector machines, decision trees and gradient boosted decision trees. Again, because it is a preliminary version of our work, we only analyse here the case of decision trees and linear classifiers, as implemented in `scikit-learn` [17]. We perform classical hyperparameter optimization with 5-fold validation to train the model behind the API.

Auditing algorithms The simplest algorithm we test is a random sampling baseline. Given a budget b, b points are uniformly sampled in $\mathcal{X}$ without replacement to form the audit dataset S. The second baseline is the AFA algorithm. Then we test our method r-AFA+ with two values of r.

Evaluation results In Figure 2, we plot the value of the μ-diameter $\text{diam}_\mu \mathcal{H}(h^*, S, 5)$ against the audit budget $|S|$. On

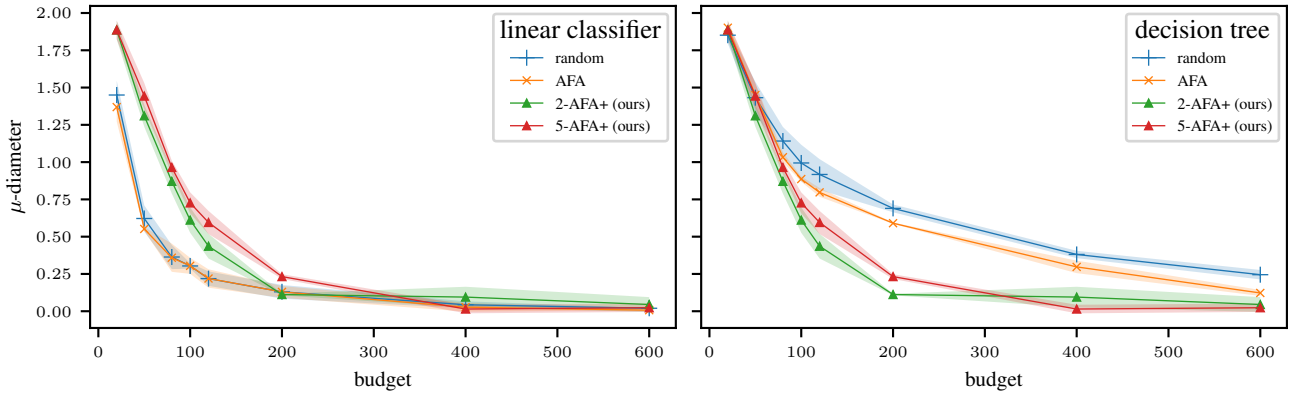

Figure 2: μ-diameter of the version space for budgets ranging from 200 to 600 requests. On the left, $\mathcal{H}$ is the set of linear classifiers. On the right, $\mathcal{H}$ is the set of decision trees. The figure compares two baselines (random sampling and original AFA) against r-AFA+ for two radius values ($r = 2$ and $r = 5$).

the right, $\mathcal{H}$ is the set of decision trees. The figure compares the two baselines (random sampling and original AFA) and our method. Note that we are evaluating the audit algorithms in the more realistic setting with $r = 5 > 0$.

In the case of linear classifiers, both our methods and the baselines tend to a null μ-radius. Our methods even reduce slightly the convergence speed of the μ-radius to 0 (2-AFA+ needs $\sim$ 50 more queries that random and AFA to reach a μ-diameter of 0.25). Yet, after 200 queries, all methods become equivalent in terms of μ-diameter. In addition, this plot highlights the performance of the simplest random baseline: it is the best performing method on this (dataset, api model) combination.

The second situation gives a totally different picture of the comparison between AFA and r-AFA+. While AFA performs as well as the random baseline, our methods allow to save up to 400 queries ($\sim$ 66%) to reach a μ-diameter of 0.25. The intuition behind the performance gap of r-AFA+ between the decision tree and linear APIs is linked to the regularity of the decision function. If the decision boundary is smoother (as is the case for linear models), two models that do not agree on a given set of points would not agree on the remaining points with a high probability. On the other hand, if the decision boundary is very irregular (as is the case for decision trees), two models that do not agree on a set of points might still be very close on the other points. Thus in this case, it seems that allowing for more disagreement (a.k.a. increasing r) between $\hat{h}$ and the μ-optimal model h in line 9 helps to include models similar to the API h^* even if $\hat{h}$ is far from it in the beginning.

The takeways from this evaluation are that i) the gains in terms of budget are highly hypothesis dependant, 2) AFA is never substantially better than random, which questions its utility (high complexity w.r.t. random selection), and 3) r-AFA+ is at least as competitive as random and AFA on the long run (i.e. for small diameters).

6 Conclusion

Being robust to slight model changes is a practical requirement to take into account the practices of deployed ML systems that often evolve. In this context, the promising auditing approach of producing one-shot certificates might frequently require auditors to re-audit the target model after each slight update. This paper explores how certificates can be designed to be robust to such modifications and presented preliminary results that support this direction.

We have empirically shown that the r-AFA+ relaxation can provide an interesting gain over AFA in one scenario, and that the random and computationnaly cheap strawman approach is also to be considered. We leave to futurework a full characterization of the model families on which these observations generalize. Futurework also includes the study of the impact of removing the assumption that the hypothesis class is known by the auditor. More precisely, allowing for a restricted hypothesis space while preserving the accuracy of audits seems like an important next step for reliable and practical audits.

As a final remark, throughout this work, we used the term "Active Auditing" coined by the authors of [22]. Yet since, this algorithm guarantees that *if* the platform does not change $\mathcal{H}$ *then* we can "easily" verify that our estimated value still holds. Thus, a more accurate term would be "active certification". This splits the goal of algorithmic auditing: trying to build certificates for platforms to defend themselves, or finding estimators that are able to uncover misconduct robust to concealment attempts from the platform.

Acknowledgements

We would like to thank Tom Yan (co-author of [22]) for the exchange we had upon implementing their algorithm.

References

[1] Sanjeev Arora, Elad Hazan, and Satyen Kale. “The Multiplicative Weights Update Method: A Meta-

Algorithm and Applications". In: *Theory of Computing* 8.6 (May 1, 2012), pp. 121–164.

[2] Solon Barocas, Moritz Hardt, and Arvind Narayanan. "Fairness in Machine Learning". In: *Nips tutorial* 1 (2017).

[3] Ja M. Barzdin. "On the Prediction of General Recursive Functions". In: *Soviet Mathematics Doklady*. Vol. 13. 1972, pp. 1224–1228.

[4] Clément L. Canonne. *A Survey on Distribution Testing: Your Data Is Big. But Is It Blue?* Graduate Surveys 9. Theory of Computing Library, Aug. 15, 2020. 100 pp.

[5] Clement L. Canonne et al. "The Price of Tolerance in Distribution Testing". In: *Proceedings of Thirty Fifth Conference on Learning Theory*. Conference on Learning Theory. PMLR, June 28, 2022, pp. 573–624.

[6] Paulo Cortez and Alice Silva. "Using Data Mining to Predict Secondary School Student Performance". In: *EUROSIS* (Jan. 1, 2008).

[7] Sanjoy Dasgupta et al. "Teaching a Black-Box Learner". In: *Proceedings of the 36th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, May 24, 2019, pp. 1547–1555.

[8] Jeffrey Dastin. "Amazon Scraps Secret AI Recruiting Tool That Showed Bias against Women". In: *Reuters. Retail* (Oct. 10, 2018).

[9] Frances Ding et al. "Retiring Adult: New Datasets for Fair Machine Learning". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 6478–6490.

[10] Steve Hanneke. "Theory of Disagreement-Based Active Learning". In: *Foundations and Trends® in Machine Learning* 7.2-3 (June 11, 2014), pp. 131–309. ISSN: 1935-8237, 1935-8245.

[11] Russell Hotten. "Volkswagen: The Scandal Explained". In: *BBC News. Business* (Sept. 22, 2015).

[12] Jeff Larson et al. "How We Analyzed the COMPAS Recidivism Algorithm". In: *ProPublica* (May 23, 2016).

[13] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[14] J. Nathan Matias, Austin Hounsel, and Nick Feamster. "Software-Supported Audits of Decision-Making Systems: Testing Google and Facebook's Political Advertising Policies". In: *Proceedings of the ACM on Human-Computer Interaction* 6 (CSCW1 Apr. 7, 2022), 118:1–118:19.

[15] Danaë Metaxa et al. "Auditing Algorithms: Understanding Algorithmic Systems from the Outside In". In: *Foundations and Trends® in Human–Computer Interaction* 14.4 (2021), pp. 272–344. ISSN: 1551-3955, 1551-3963.

[16] Albert B. Novikoff. *On Convergence Proofs for Perceptrons*. STANFORD RESEARCH INST MENLO PARK CA, 1963.

[17] Fabian Pedregosa et al. "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. ISSN: 1533-7928.

[18] Bashir Rastegarpanah, Krishna Gummadi, and Mark Crovella. "Auditing Black-Box Prediction Models for Data Minimization Compliance". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 20621–20632.

[19] Marco Tulio Ribeiro, Sameer Singh, and Carlos 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*. KDD '16. New York, NY, USA: Association for Computing Machinery, Aug. 13, 2016, pp. 1135–1144. ISBN: 978-1-4503-4232-2.

[20] Ali Shahin Shamsabadi et al. "Confidential-PROFITT: Confidential PROof of FaIr Training of Trees". In: The Eleventh International Conference on Learning Representations. Feb. 1, 2023.

[21] Ali Shahin Shamsabadi et al. "Washing The Unwashable : On The (Im)Possibility of Fairwashing Detection". In: Advances in Neural Information Processing Systems. Oct. 31, 2022.

[22] Tom Yan and Chicheng Zhang. "Active Fairness Auditing". In: *Proceedings of the 39th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, June 28, 2022, pp. 24929–24962.