Smooth Sensitivity for Learning Differentially-Private yet Accurate Rule Lists

Timothée Ly, Julien Ferry, Marie-José Huguet, Sébastien Gambs, Ulrich Aivodji

Smooth Sensitivity for Learning Differentially-Private yet Accurate Rule Lists

1st Timothée Ly
LAAS-CNRS, Telecom Paris & KTH
Toulouse, France
tly@laas.fr

2nd Julien Ferry
Polytechnique Montréal
Montréal, Canada
julien.ferry@polymtl.ca

3rd Marie-José Huguet
LAAS-CNRS, Université de Toulouse, CNRS, INSA
Toulouse, France
huguet@laas.fr

4th Sébastien Gambs
Université du Québec à Montréal
Montréal, Canada
gambs.sebastien@uqam.ca

5th Ulrich Aïvodji
École de Technologie Supérieure
Montréal, Canada
Ulrich.Aivodji@etsmtl.ca

***Abstract*—Differentially-private (DP) mechanisms can be embedded into the design of a machine learning algorithm to protect the resulting model against privacy leakage. However, this often comes with a significant loss of accuracy due to the noise added to enforce DP. In this paper, we aim at improving this trade-off for a popular class of machine learning algorithms leveraging the Gini impurity as an information gain criterion to greedily build interpretable models such as decision trees or rule lists. To this end, we establish the smooth sensitivity of the Gini impurity, which can be used to obtain thorough DP guarantees while adding noise scaled with tighter magnitude. We illustrate the applicability of this mechanism by integrating it within a greedy algorithm producing rule list models, motivated by the fact that such models remain understudied in the DP literature. Our theoretical analysis and experimental results confirm that the DP rule lists models integrating smooth sensitivity have higher accuracy that those using other DP frameworks based on global sensitivity, for identical privacy budgets.**

***Index Terms*—Differential Privacy, Interpretability, Rule Lists, Machine Learning**

I. Introduction

Machine learning models are increasingly used for high-stakes decision-making tasks such as kidney exchange [1] or recidivism prediction [2]. As such tasks often require the use of sensitive data (*e.g.*, medical or criminal records), it is crucial to ensure that the learned models do not leak undesired information. Another important aspect is to make sure human users can verify and trust the models' decisions, which motivates the use of inherently interpretable models when possible [3]. However, such models are also vulnerable to privacy attacks such as membership inference [4], [5], in which the objective of the adversary is to infer the presence of a particular profile in the training dataset, or reconstruction attacks [6], [7], in which the aim of the adversary is to reconstruct the training set. To counter this issue, Differential Privacy (DP) [8], [9] has emerged as a *de facto* privacy standard, thoroughly bounding the amount of information any adversary can gain regarding any single individual in the dataset.

For instance, recent works [10], [11] survey the literature on existing DP variants of different typical machine learning algorithms, such as the DP versions of the Principal Component Analysis algorithm [12] and of the Stochastic Gradient Descent [13]. However, much less work has been dedicated to the DP learning of interpretable models. Nonetheless, a line of works have studied different adaptations of learning algorithms producing tree-based models (*i.e.*, mostly decision trees and random forests) [14]. These works rely on popular greedy induction algorithms such as `CART` [15]. An important challenge they had to face is that the Gini impurity, a popular information gain criterion widely used at each iteration of these greedy algorithms, has very high global sensitivity. This leads to the addition of a considerable amount of noise to comply with DP, hence harming the utility of the resulting model. In this paper, we address this issue by leveraging the notion of smooth sensitivity [16]. More precisely, we first theoretically characterize the smooth sensitivity of the Gini impurity. Then, we design a DP mechanism based on smooth sensitivity that we integrate into a greedy algorithm for learning rule list models. Depending on the considered noise distribution, our approach can provide either pure or approximate DP guarantees. Our experimental results show that the proposed DP mechanisms incur a lower accuracy loss than other mechanisms based on global sensitivity for identical privacy budgets.

The outline of the paper is as follows. First, in Section II, we recall the background on rule lists models and DP. Afterwards, in Section III, we introduce the building blocks of our approach, namely greedy learning of rule lists, Gini impurity and smooth sensitivity. Then, in Section IV, we present our main contribution on establishing the smooth sensitivity of the Gini impurity. We leverage it to design an effective algorithm for learning DP rule lists. Finally, in Section V, we empirically evaluate our proposed methods in terms of privacy-accuracy trade-offs and robustness to privacy attacks before concluding in Section VI.

II. Background

A. Rule Lists

We consider a tabular dataset $\mathcal{D}$ of n samples in which each sample s corresponds to a set of binary features and has a binary label y_s. Rule lists, originally introduced as a way to efficiently represent Boolean functions, are a common type of interpretable models [17], [18]. More precisely, a rule list RL is a sequence of $K+1$ rules $(r_1, \ldots, r_K, r_0) \in \mathscr{R}^{K+1}$ in which $\mathscr{R}$ is the set of possible rules (which, for instance, can be pre-mined). Each rule $r_i \in \mathscr{R}$ is composed of a Boolean assertion p_i called the antecedent and a label prediction $q_i \in \{0, 1\}$ named the consequent and can be denoted by $r_i = p_i \rightarrow q_i$. A sample s of dataset $\mathcal{D}$ is said to be caught by rule r_i when p_i evaluates to true for s, which leads to s being classified with label q_i. The *default rule*, $r_0 = \text{True} \rightarrow q_0$ classifies any sample not caught by the previous rules to $q_0 \in \{0, 1\}$ fixed.

Rule List 1 provides an example model trained for a recidivism prediction task. As evidenced through this illustrative example, the use of any discriminatory feature would immediately be spotted, which is an advantage of using inherently interpretable models [3]. This is in contrast with black-box models, in which such an undesired behaviour would be difficult - or even impossible [19] - to detect.

```
if Prior-Crimes≠ 0 then True
else if Juvenile-Felonies ≤3 and Juvenile-Crimes ∉ [[1, 3]]
    then False
else True
```

Rule list 1: Example rule list learnt on the `Compas` dataset [2]. The binary prediction is whether the offendent will recidivate within two years or not.

Rule lists are provably more expressive than decision trees of comparable size [17], mainly because there can be any arbitrary overlap between the supports of the different rules within the rule list (since they are ordered), while the supports of the leaves of a decision tree are inherently disjoint. Hence, any decision tree with depth d can be translated into an equivalent rule list using rules involving at most d boolean conditions, while the opposite is not true. A direct consequence from an information theory perspective is that rule lists encode more information regarding their training data than decision trees of comparable size [6]. Nevertheless, despite their advantage in terms of compactness, rule lists remain less studied than decision trees in the privacy literature, which is why we focus on them in this work. There are a few notable exceptions, although they consider different setups and have different objectives [20], [21]. On the one hand, Daniely and Feldman [21] discuss the sample complexity of learning decision lists in a non-interactive local differential privacy context. On the other hand, Thaler et al. [20] consider the problem of privately releasing a database whose rows consist of pre-defined decision lists, in the context of privacy-preserving data publishing.

Rule lists can be built either with exact methods (producing certifiably optimal models) such as the `CORELS` [22] tree-based algorithm or with heuristic approaches [23], which we specifically consider in this paper. More precisely, we chose to focus on greedy learning algorithms (as introduced in Section III-A), because this framework encompasses a wide range of interpretable models beyond rule lists, such as decision trees or random forests. This makes our results easily applicable to these other models.

B. Differential Privacy

Differential privacy (DP) is a mathematical property that yields strong privacy guarantees with respect to queries or computations performed on a database [9]. The underlying idea is the following: an algorithm is differentially private if its distribution over outputs does not change much after adding or removing one sample (corresponding to one individual with personal information). The objective of privacy-preserving machine learning is to reconcile two antagonist purposes: extracting useful correlations from data without revealing private information regarding any single individual. In particular in machine learning, DP can be integrated into the learning algorithm to ensure that the resulting model does not leak too much information with respect to the input dataset. In this context, a differentially private learning algorithm ensures that the distribution over outputs (*i.e.*, possible models) is not impacted significantly by the addition or removal of a sample from the training set.

More formally, two datasets $\mathcal{D}$ and $\mathcal{D}'$ are said to be neighbouring if they differ at most by one sample, denoted by $\|\mathcal{D} - \mathcal{D}'\|_1 \leq 1$ for $(\mathcal{D}, \mathcal{D}') \in \mathbb{N}^{|\mathscr{X}|}$ in which $\mathscr{X}$ is the finite set of all possible samples in a dataset. Similarly, the number of elements in a dataset $\mathcal{D}$ is $\|\mathcal{D}\|_1$. More details regarding these definitions and notations are provided within the Appendix A-A.

An algorithm $\mathcal{M} : \mathbb{N}^{|\mathscr{X}|} \mapsto \mathscr{Y}$ is (ε, δ)-differentially private if $\forall\ S \subseteq \mathscr{Y}, \forall\ (\mathcal{D}, \mathcal{D}') \in (\mathbb{N}^{|\mathscr{X}|})^2, ||\mathcal{D} - \mathcal{D}'||_1 \leq 1$, we have:

$$\mathbb{P}(\mathcal{M}(\mathcal{D}) \in S) \leq \exp(\varepsilon)\mathbb{P}(\mathcal{M}(\mathcal{D}') \in S) + \delta \text{ [9].}$$

The parameter ε controls the level of privacy of the algorithm as it defines how much the probability of an output can vary when adding or removing a sample. Typically, $\varepsilon = 1$ is considered a reasonable value in terms of provided protection [24]. *Pure DP* refers to when $\delta = 0$, while *approximate DP* corresponds to values of $\delta > 0$. Indeed, δ can be interpreted as a probability of "total privacy failure". One possible instance of this δ-failure could be that with probability $1 - \delta$, the model will behave like pure DP while with probability δ (*i.e.*, the failure probability), there will be no privacy guarantees at all. $\delta \ll \frac{1}{\|\mathcal{D}\|_1}$ is considered to be an absolute requirement since a δ of order $\mathcal{O}(1/\|\mathcal{D}\|_1)$ could enable the total release of some samples of the dataset.

Intuitively, differentially private mechanisms often revolve around the idea of adding noise of a magnitude of order close to how steep the output function can change with slight variations of the input. More precisely, for a given function $f \in \mathbb{R}^k$, its *global sensitivity* precisely quantifies how much the value of f can differ between any two neighbouring datasets. Formally,

the global sensitivity of a function $f : \mathbb{N}^{|\mathscr{X}|} \mapsto \mathbb{R}^k$ for norm l_p (usually l_1 or l_2), denoted by $\Delta_p f$, is defined for any neighbouring datasets $\mathcal{D}$ and $\mathcal{D}'$ as follows:

$$\Delta_p f = \max_{\substack{\mathcal{D},\mathcal{D}' \in \mathbb{N}^{|\mathscr{X}|} \\ ||\mathcal{D}-\mathcal{D}'||_1 = 1}} ||f(\mathcal{D}) - f(\mathcal{D}')||_p.$$

One of the shortcomings of global sensitivity is that it does not take into account the position in the latent space of the considered datasets. Nevertheless, it can effectively be used to obtain thorough DP guarantees. A straightforward approach to achieve DP is by adding noise to the output of the given function f. Two popular differentially private mechanisms [9] are based on this principle: the *Laplace mechanism* $\mathscr{M}^{\Delta_1}_{LAPLACE}(\mathcal{D}, f, \varepsilon)$ and the *Gaussian mechanism* $\mathscr{M}^{\Delta_2}_{GAUSS}(\mathcal{D}, f, \varepsilon, c)$. In these mechanisms, for each component j of the function f, the amplitude of the added noise N_j is scaled to its global sensitivity.

The Laplace mechanism $\mathscr{M}^{\Delta_1}_{LAPLACE}(\mathcal{D}, f, \varepsilon)$ is based on Δ_1 (global sensitivity with l_1 norm) along with Laplace noise: $\forall j \in \{1, \ldots, k\}, N_j \sim Lap(\Delta_1 f/\varepsilon)$. The probability density function of the Laplace distribution is $Lap(x \mid b) = \frac{1}{2b} \exp\left(\frac{-|x|}{b}\right)$ and the Laplace mechanism satisfies $(\varepsilon, 0)$-DP.

The Gaussian mechanism $\mathscr{M}^{\Delta_2}_{GAUSS}(\mathcal{D}, f, \varepsilon, \delta)$ satisfies (ε, δ)-DP. It uses Δ_2 (global sensitivity with l_2 norm) together with Gaussian noise: $\forall j \in \{1, \ldots, k\}, N_j \sim \mathcal{N}\left(\mu = 0, \sigma = \frac{c \cdot \Delta_2 f}{\varepsilon}\right)$ with $c^2 > 2\log(\frac{1.25}{\delta})$.

Another common DP mechanism, called the *Exponential mechanism* $\mathscr{M}^{\Delta u}_{EXP}(\mathcal{D}, u, \mathcal{V})$, considers a (discrete) set of possible outputs $\mathcal{V}$ and randomly samples one of them with respect to a utility metric. Thus in this mechanism, the noise added to comply with DP is introduced through a random sampling of the candidate outputs rather than added on the outputs themselves. More precisely, let $u : (\mathcal{D}, v) \mapsto u(\mathcal{D}, v)$ denote the utility function of element v with respect to dataset $\mathcal{D}$. The global sensitivity of this utility function is defined as:

$$\Delta u = \max_{v \in \mathcal{V}} \max_{\substack{\mathcal{D},\mathcal{D}' \in \mathbb{N}^{|\mathscr{X}|} \\ ||\mathcal{D}-\mathcal{D}'||_1 \leq 1}} |u(\mathcal{D}, v) - u(\mathcal{D}', v)|$$

The Exponential mechanism $\mathscr{M}^{\Delta u}_{EXP}(\mathcal{D}, u, \mathcal{V})$ samples an element $v \in \mathcal{V}$ with probability $p \propto \exp(\frac{\varepsilon \cdot u(\mathcal{D}, v)}{2 \cdot \Delta u})$ and satisfies $(\varepsilon, 0)$-DP [9].

Finally, we will also use the *Noisy Max Report* mechanism [9], which satisfies (ε, δ)-DP. Considering a noisy mechanism $\mathscr{M}_{noisy}$ satisfying (ε, δ)-DP, the Noisy Max Report mechanism returns: $\operatorname{argmax}_{v \in \mathcal{V}} \mathscr{M}_{noisy}(\mathcal{D}, u(\cdot, v), \varepsilon)$.

Among others, DP comes with two fundamental properties. First, the *post-processing* property ensures that DP guarantees are not affected by post-processing the output of a DP mechanism. Second, the *composability* property enables the composition of different differentially private mechanisms (sequentially or in parallel) and the computation of the global privacy budget. In a nutshell, sequential composition states that if several DP mechanisms are applied on overlapping datasets, their privacy budgets sum up, whereas parallel composition considers the application of several DP mechanisms on disjoints datasets, in which case there is no need to sum up their budgets. More details regarding these properties are provided in the Appendix A-B.

III. Building Blocks

A. A Greedy Algorithm for Learning Rule Lists

Greedy algorithms are widely used for learning decision tree models. For instance, the commonly used `CART` algorithm [15] iteratively builds a decision tree in a top-down manner, by successively selecting the feature (and split value) yielding the best information gain value according to some pre-defined criterion. While algorithms for learning rule lists in a greedy manner are far less popular than their counterparts for learning decision trees, some implementations exist in the literature. For instance, the `imodels` [1] library [23] contains algorithms for learning different types of interpretable models, including rule lists (denoted `GreedyRL`). More precisely, `GreedyRL` iteratively calls `CART` to build a depth-one decision tree at each level of the rule list, optimizing a given information gain criterion. Just like for decision trees, greedy algorithms for building rule lists successively select the best rule $r_i = p_i \rightarrow q_i$ given some information gain criterion. Thus, at each level of the rule list being built, the `GreedyRL` algorithm iterates through all possible rules and keeps the one leading to the best information gain value.

B. Gini Impurity for Rule Lists

In this paper, we consider the Gini impurity index originally used in the `CART` [15] algorithm as a measure of the information gain. In a nutshell, this index quantifies how well a rule separates the data into two categories with respect to different labels, with the value of zero being reached when the examples are perfectly separated. The algorithm stops when all the samples are classified, but other stopping criteria can be implemented such as a maximum length on the list of rules or a minimum support condition on each rule (*i.e.*, number of points left to be classified).

Consider a given rule r for a specific node of a pre-existing list of rules, which means that some samples were already captured by previous rules and are not accounted for. Let $C(r) \subset \mathcal{D}$ be the subset of samples captured by rule r, in which $n_c(r)$ is the number of samples in $C(r)$ and $n_l(r)$ the number of samples not captured by rule r. For a rule list $RL = (r_1, \ldots, r_K, r_0)$, and a given position j in the sequence, let $\tilde{n}(j)$ be the number of samples not captured by previous rules $r_1 \ldots r_{j-1}$. In particular, this means that $\tilde{n}(j) = n_c(r_j) + n_l(r_j) = n - \sum_{i=1}^{j-1} n_c(r_i)$. In addition, let $\hat{y}_c(r)$ denote the average outcome (*i.e.* the predicted label) of the rule r, $\hat{y}_c(r) = \frac{1}{n_c(r)} \sum_{s \in C(r)} y_s$. Similarly, the average outcome of the remaining samples is

$$\hat{y}_l(r) = \frac{1}{n_l(r)} \sum_{s \in \mathcal{D} \setminus \left(\cup_{i=1}^{j-1} C(r_i) \,\cup\, C(r)\right)} y_s$$

[1] https://github.com/csinva/imodels

The Gini impurity reduction with respect to rule r is denoted as $\mathscr{G}(r)$. It can be divided into two terms $\mathscr{G}_c(r)$ and $\mathscr{G}_l(r)$, respectively for the samples caught and the ones not caught by the rule: $\mathscr{G}(r) = \mathscr{G}_c(r) + \mathscr{G}_l(r)$. Note that we not only consider the samples caught by the rule (through $\mathscr{G}_c(r)$) but also those which are not (through $\mathscr{G}_l(r)$) as it matters for the following rules in the rule list. For binary classification, the Gini impurity reduction for a rule r at position j is given by:

$$\mathscr{G}_c(r) = \frac{n_c(r)}{\tilde{n}(j)}\big(1 - \hat{y}_c(r)^2 - (1 - \hat{y}_c(r))^2\big) \tag{1}$$

$$\mathscr{G}_l(r) = \frac{n_l(r)}{\tilde{n}(j)}\big(1 - \hat{y}_l(r)^2 - (1 - \hat{y}_l(r))^2\big) \tag{2}$$

C. Smooth Sensitivity

The DP mechanisms described in Section II-B rely on the notion of global sensitivity. However, some functions only display a very loose bound for their global sensitivity. For instance, the global sensitivity of the Gini impurity is 0.5, irrespective of the actual number of samples left to be classified. This is considerably high, given that the Gini impurity takes values in $[0, 1]$. To address this limit, previous work [16] proposed a way to compel a tighter bound on the added noise. Firstly, they have introduced the notion of the *local sensitivity* of a function $f : \mathbb{N}^{|\mathscr{X}|} \mapsto \mathbb{R}^k$ at a dataset $\mathcal{D}$, denoted $LS_f(\mathcal{D})$, as: $\max_{\substack{\mathcal{D}' \in \mathbb{N}^{|\mathscr{X}|}: \\ ||\mathcal{D}-\mathcal{D}'||_1 = 1}} ||f(\mathcal{D}) - f(\mathcal{D}')||_1$. However, replacing directly the global sensitivity by local sensitivity does not yield privacy guarantees.

This motivated the formulation of a refined sensitivity notion denoted as *smooth sensitivity*. This notion exploits a *smooth upper bound* of $LS_f(\mathcal{D})$, denoted by $S_{f,\beta}(\mathcal{D})$, as follows. For $\beta > 0$, $S_{f,\beta}(\mathcal{D}) : \mathbb{N}^{|\mathscr{X}|} \mapsto \mathbb{R}^+$ is a β-smooth upper bound on the local sensitivity of f if it satisfies :

$$\begin{aligned} &\forall \mathcal{D} \in \mathbb{N}^{|\mathscr{X}|}, \forall \mathcal{D}' \in \mathbb{N}^{|\mathscr{X}|} \; s.t. \; ||\mathcal{D} - \mathcal{D}'||_1 = 1, \\ &S_{f,\beta}(\mathcal{D}) \geq LS_f(\mathcal{D}) \quad \text{and} \quad S_{f,\beta}(\mathcal{D}) \leq e^{\beta} S_{f,\beta}(\mathcal{D}') \end{aligned} \tag{3}$$

The smallest function to satisfy Equation (3) is called the *smooth sensitivity* and denoted $S^*_{f,\beta}(\mathcal{D})$:

$$\text{For } \beta > 0, S^*_{f,\beta}(\mathcal{D}) = \max_{\mathcal{D}' \in \mathbb{N}^{|\mathscr{X}|}} LS_f(\mathcal{D}') e^{-\beta ||\mathcal{D} - \mathcal{D}'||_1}$$

Nissim et al. [16] proposed an iterative computation of the smooth sensitivity (Lemma III.1) considering datasets than can vary up to k samples rather than 1. Let $\mathcal{T}_k$ denote the local sensitivity of f at distance k:

$$\mathcal{T}_k(\mathcal{D}) = \max\big\{LS_f(\mathcal{D}') \;\big|\; ||\mathcal{D}' - \mathcal{D}||_1 \leq k\big\}.$$

Lemma III.1. [16] $S^*_{f,\beta}(\mathcal{D}) = \max\big\{e^{-\beta k}\mathcal{T}_k(\mathcal{D}) \;\big|\; k \in \mathbb{N}\big\}$. (proof recalled in Appendix A-C)

As stated by [25]–[27], smooth sensitivity is a very powerful tool to replace global sensitivity for differentially private machine learning. However, finding a closed form for $S^*_{f,\beta}(\mathcal{D})$ is difficult and sometimes requires to make stronger assumptions on the model. Nonetheless, two DP mechanisms were proposed by [16] based on the smooth sensitivity.

The first one is based on Cauchy noise and uses an additional parameter γ:

$$\mathscr{M}^{S^*_{\cdot,\beta}}_{CAUCHY}(\mathcal{D}, f, \varepsilon) : \mathcal{D} \mapsto f(\mathcal{D}) + \frac{2(\gamma+1)S^*_{f,\beta}(\mathcal{D})}{\varepsilon} \cdot \eta$$

with $\beta \leq \frac{\varepsilon}{2(\gamma+1)}$, $\gamma > 1$ and $\eta \sim h(z) \propto \frac{1}{1+|z|^\gamma}$ the Cauchy noise. This mechanism satisfies (ε,0)-DP.

The second one uses Laplace noise and satisfies (ε, δ)-DP:

$$\mathscr{M}^{S^*_{\cdot,\beta}}_{LAPLACE}(\mathcal{D}, f, \varepsilon, \delta) : \mathcal{D} \mapsto f(\mathcal{D}) \; + \; \frac{2 \cdot S^*_{f,\beta}(\mathcal{D})}{\varepsilon} \cdot \eta$$

with $\beta \leq \frac{\varepsilon}{2\log(2/\delta)}$ and $\eta \sim Lap(1)$, the Laplace noise.

Note that in contrast to global sensitivity, adding Laplace noise within the framework of smooth sensitivity does not yield pure DP anymore but approximate one. Furthermore, to obtain pure DP guarantees along with smooth sensitivity, heavy-tailed noise distributions must be considered, such as the Cauchy distribution. In the following, we fix $\beta = \frac{\varepsilon_{node}}{2\log(2/\delta_{node})}$ for the β-smooth upper bound.

IV. A differentially private Greedy Learning Algorithm for Rule Lists

We now introduce our framework for learning differentially private rule lists leveraging smooth sensitivity. Unlike [25] who integrate smooth sensitivity to determine the majority class for a leaf in a tree, we integrate it to determine the rule with the best Gini impurity.

A. Establishing the Smooth Sensitivity of the Gini Impurity

The local sensitivity for the Gini impurity has been characterized in [28]. Considering the support $\tilde{n}(j)$ of the jth rule, it is defined by:

$$LS_{\mathcal{G}}(\tilde{n}(j)) = 1 - \Big(\frac{\tilde{n}(j)}{\tilde{n}(j)+1}\Big)^2 - \Big(\frac{1}{\tilde{n}(j)+1}\Big)^2$$

Given a minimal support Λ imposed for each selection of rule (*i.e.,* a minimum number of samples that a rule must capture in order to be considered for inclusion within the built rule list), we have derived in Theorem IV.1 a method to compute the smooth sensitivity of the Gini impurity.

Theorem IV.1 (Smooth Sensitivity of the Gini impurity)**.** *Let $\Lambda \in \mathbb{N}^*$ be the given minimum support. By inverting the parameter k and the variable $\mathcal{D}$ in the function $\mathcal{T}_k(\mathcal{D})$, we define the following function :*

$$\xi_{\mathcal{D},\beta}(k) : \left| \begin{array}{ccc} \mathbb{N} & \longrightarrow & \mathbb{R}^+ \\ k & \longmapsto & e^{-k\beta} \cdot g\big[\max(\Lambda, \|\mathcal{D}\|_1 - k)\big] \end{array} \right.$$

in which

$$g : \left| \begin{array}{ccc} \mathbb{R}^+ & \longrightarrow & [0,1] \\ x & \longmapsto & 1 - \Big(\frac{x}{x+1}\Big)^2 - \Big(\frac{1}{x+1}\Big)^2 \end{array} \right.$$

The smooth sensitivity of a rule with a dataset $\mathcal{D}$ of points that remain to classify is :

$$S^*_{\mathcal{G},\beta}(\mathcal{D}) = \max\Big[\xi_{\mathcal{D},\beta}(0),\, \xi_{\mathcal{D},\beta}(\lfloor t \rfloor),\, \xi_{\mathcal{D},\beta}(\lceil t \rceil),\, \xi_{\mathcal{D},\beta}(\|\mathcal{D}\|_1 - \Lambda)\Big]$$

with $t = \|\mathcal{D}\|_1 - \dfrac{1-\beta-\sqrt{(1-\beta)^2-4\beta}}{2\beta}$ *if well defined and otherwise* 0.

Proof. The detailed proof is provided in Appendices B-A and B-B in which we first prove it for $\Lambda = 1$ and generalize the proof for $\Lambda \in \mathbb{N}^*$. Crucially, recall that the smooth sensitivity is the same for any rule at a given position since we have proven that the smooth sensitivity of the Gini impurity only takes into account the number of elements left to be classified (and not how the rule captures them or not). The proof is a proof by exhaustion, in which the smooth sensitivity is computed as $S^*_{\mathcal{G},\beta}(x) = \max_{k\in\mathbb{N}} e^{-k\beta}\mathcal{T}_k(x)$. We first determine the function $\mathcal{T}_k(x)$ in which x is a dataset. We observe that it does not depend on the actual value of the dataset but solely the number of samples it contains. We obtain $\mathcal{T}_k(x) = g(\max(1, ||x||_1 - k))$. Given that we managed to obtain a closed form for $\mathcal{T}_k(x)$ in which k directly intervenes, we now consider x to be a parameter and put k as a variable of our function hence the introduction of function $\xi_{x,\beta}(t) = e^{-k\beta}\mathcal{T}_k(x)$. Rather, we study this function on $\mathbb{R}^+$ since it is differentiable. Through the cancellation of the derivative, we are able to find the minima and the maxima of the function. However, since these are $\mathbb{R}$-valued maxima, we finally truncate them to the closest higher and inferior integers to obtain the smooth sensitivity. Note that since we associate the sign of the derivative to a polynomial, it gives us extra control over the monotony of the function ξ since we know a polynomial takes the sign of the highest degree coefficient outside of the roots (granted that they exist) and the opposite inside the roots. For certain values of β such maxima may not exist because they are computed as the roots of a polynomial whose values depend on β. For this reason, we state that the formula should encompass t only if it is well defined (*i.e.*, the value inside the square root is not negative) and otherwise replace it by 0. Proving that the smooth sensitivity only depends on $||x||$ is a core result of our approach. Thanks to this, at each iteration we only need to query the number of elements left to classify and apply the same smooth sensitivity to all rules (the Gini impurity depends on the split made by the rule but its smooth sensitivity is independent of it). □

Figure 1 gives an overview on the amount of noise one has to add to the computed Gini impurity to get a target DP guarantee, using either global or smooth sensitivity. More precisely in this figure, we display the noise distortion generated for a fixed $\varepsilon = 1$ by each DP mechanism as a function of the number of samples captured by the rule. Importantly, we observe how the use of smooth sensitivity allows to scale down the generated noise when considering more samples. This is not the case for global sensitivity, which is dataset-independent. Overall, the noise added using smooth sensitivity is far inferior to the noise generated by a global sensitivity mechanism for a similar level of a privacy - which is a promising preliminary result for the implementation of a private model relying on the smooth sensitivity of the Gini Impurity.

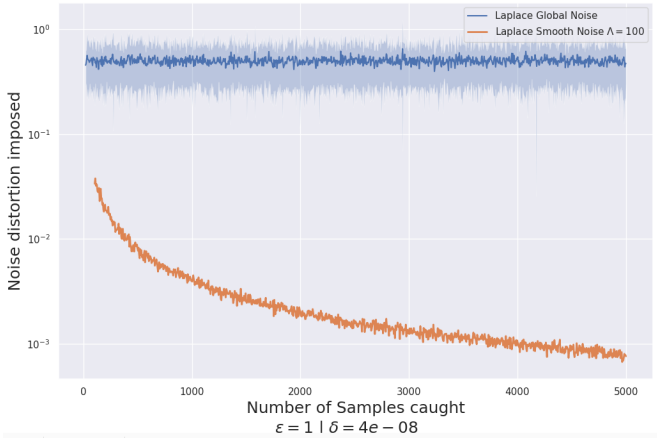


Fig. 1: Comparison of the amplitude (log scale) of the noise added by the Laplace mechanism scaled to either the Smooth or Global Sensitivities.

Many learning algorithms use a regularization parameter scaling with the size or complexity of the model to enhance interpretability and reduce overfitting [29]. In our case, apart from being a key factor for the smooth sensitivity computation, the minimum support also leads to a better comprehensibility of the resulting rule list by encouraging its sparsity, as there can only be as many as $\frac{1}{\lambda}$ rules where λ is the relative minimum support defined as $\lambda = \frac{\Lambda}{\|x\|_1}$. It also plays the role of a regularization parameter as it helps the model to not overfit. Note that other types of rule lists learning algorithms, such as the `CORELS` exact method, also consider a regularization term based on the number of rules within the built rule list.

B. Greedy Learning of Differentially Private Rule Lists Leveraging Smooth Sensitivity

The proposed differentially private algorithm for learning rule lists, called `sm-Laplace`, is detailed in Algorithm 1. This algorithm is based on the smooth sensitivity established in the previous subsection and uses Laplace noise. It takes as input a set of rules $\mathscr{R}$ that is assumed to be publicly known and is not obtained as a computation from the data. Note that this assumption is consistent with the literature. For a set of parameters (Gini impurity computation, rule list size and minimum support, and privacy guarantees) this greedy algorithm iteratively adds a new rule to the rule list RL. At each step, it checks whether the support in the current remaining dataset X_{rem} verifies the minimum support condition (Line 4), including the confidence threshold computed once for all (Line 2). For each rule $r \in \mathscr{R}$, its noisy Gini impurity is computed using our proposed Laplace mechanism based on smooth sensitivity at Line 11 and the rule $R^\star$ with lowest noisy Gini is returned. $R^\star$ is then added to RL with its DP prediction $q^\star$ (Line 14) and removed from $\mathscr{R}$. The main loop is stopped when (1) the rule list reaches the maximum length, (2) the support condition is not verified anymore or (3) adding a rule does not improve the Gini impurity value.

Algorithm 1 Approximate $(\varepsilon, \delta)-$DP-Greedy Rule List with Smooth Sensitivity

Input: Dataset $\mathcal{D} \in \mathbb{N}^{|\mathscr{X}|}$, Rule set $\mathscr{R}$
Parameters: Min support of the dataset λ, Max length of a rule list K, DP budget (ε, δ), Confidence $\mathcal{C}$
Output: Rule List RL (and noisy counts c_0, c_1)

1: $X_{rem} \leftarrow \mathcal{D}$, $R_{rem} \leftarrow \mathscr{R}$, $RL \leftarrow []$, {Initialisation}
$\Lambda \leftarrow \lfloor \|x\|_1 \times \lambda \rfloor$, Stop $\leftarrow$ False
2: $\mathcal{T} \leftarrow$ **confidence_threshold**$(\mathcal{C})$
3: **while** $RL \cdot$ **size**$() < K$ **and** $\neg$Stop **do**
4: **if** $\mathcal{M}^{\Delta_1}_{LAPLACE}(X_{rem}, count(X_{rem}, \cdot), \varepsilon_{node}) < \Lambda + \mathcal{T}$ **then**
5: Stop $\leftarrow$ True
6: **else**
7: $\mathcal{G}_{bound} \leftarrow \mathcal{M}^{S^*_{\cdot,\beta}}_{LAPLACE}(\emptyset, \mathscr{G}_{X_{rem}}(\cdot), \varepsilon_{node}, \delta_{node})$
8: $\mathcal{G}^\bigstar \leftarrow \mathcal{G}_{bound}$ {no rule added gini}
9: $R^\bigstar \leftarrow \emptyset$, $q^\bigstar \leftarrow$ **pred_DP**$(\emptyset, X_{rem})$
10: **for** $r \in R_{rem}$ **do**
11: $\mathcal{G} \leftarrow \mathcal{M}^{S^*_{\cdot,\beta}}_{LAPLACE}(r, \mathscr{G}_{X_{rem}}(\cdot), \varepsilon_{node}, \delta_{node})$
12: **if** $\mathcal{G} < \mathcal{G}^\bigstar$ **then**
13: $\mathcal{G}^\bigstar \leftarrow \mathcal{G}$, $R^\bigstar \leftarrow r$
14: $q^\bigstar \leftarrow$ **pred_DP**(r, X_{rem})
15: **end if**
16: **end for**
17: **if** $R^\bigstar = \emptyset$ **then**
18: Stop $\leftarrow$ True
19: **else**
20: $RL \cdot$ **append**$(R^\bigstar, q^\bigstar)$
21: **update**$_{DB}(X_{rem} \leftarrow X_{rem} \setminus \mathcal{C}(R^\bigstar))$
22: **end if**
23: **end if**
24: **end while**

DP computation of the rules' predictions (Line 14). In Algorithm 1, it is necessary to determine the prediction for each rule in a differentially private manner. Indeed, in the non-DP setup, the prediction is computed as the majority class among the samples caught by the rule. However, such a deterministic selection of the best prediction is not compatible with DP. For instance, consider two neighbouring datasets $\mathcal{D}$ and $\mathcal{D}'$. Let r be a rule picked from the rule list built on $\mathcal{D}$. If $\mathcal{D}'$ is $\mathcal{D}$ deprived from one element that would flip the outcome of r, the probability of observing this outcome in the built rule list is also flipped from 1 to 0 breaking any DP guarantee. Thus, the rules' predictions have to be determined using DP-protected counts. In our implementation (Algorithm 2), we use the Laplace mechanism based on the global sensitivity to compute the counts for each rule that are later used to determine the rule's prediction. Thus, Algorithm 2 provides pure DP guarantees. Note that since counting queries have sensitivity 1 while their output takes values in $[0, n]$, this is a reasonably low value and the mechanisms based on global sensitivity usually yield good utility in this setting.

Confidence threshold for minimum support (Line 2).

Algorithm 2 Function **pred_DP** :

Input: Rule r, Remaining samples X_{rem}
Parameters: DP budget (ε, δ)
Output: Prediction q, Counts (c_0 and c_1)

$c_0 \leftarrow \mathcal{M}^{\Delta_1}_{LAPLACE}(r,$ **count_0**$(X_{rem}, \cdot), \varepsilon_{node})$
$c_1 \leftarrow \mathcal{M}^{\Delta_1}_{LAPLACE}(r,$ **count_1**$(X_{rem}, \cdot), \varepsilon_{node})$
$q \leftarrow 0$ **if** $c_0 > c_1$ **else** 1

One remaining issue with the proposed smooth sensitivity framework is that the minimum support requirement may jeopardize the DP guarantees. For instance, consider $\mathcal{D}$ a dataset and a fixed minimum support Λ, and let r a rule. Suppose that after applying rule r, the number of points remaining for classification $n_l(r)$ is exactly equal to Λ. Let also $\mathcal{D}'$ be a dataset neighbouring $\mathcal{D}$ that misses one of the samples not caught by r in $\mathcal{D}$. Then, the support of $\mathcal{D}'$ after applying rule r is strictly smaller than Λ so any rule will necessarily be discarded because it is a stopping condition. Again, this breaks any DP guarantees, as the resulting model may change significantly due to the absence of a single sample in the dataset. To solve this issue in the proposed algorithm, we consider a threshold for minimum support that in most cases preserves the DP guarantees. Knowing that counting queries have a global sensitivity of 1, after each split of the dataset, we add Laplace noise $\sim Lap\big(\frac{\Delta_1 f=1}{\varepsilon}\big)$ to the noisy support. If the noisy support is under a given predefined threshold then we stop here and use the default classification, while otherwise we keep adding rules. To determine the threshold, assume that Λ and ε are fixed and we want a confidence $\mathcal{C} = 0.98$. When the added noise is negative (*i.e.*, the noisy support is lower than the exact support), the algorithm does not add any rule even if the smooth sensitivity computation remains exact. However, when the noisy support is above the exact support, we need to assess how large the added noise can be. This can be done by studying the distribution of the Laplace noise to determine at what value t it will be above the confidence $\mathcal{C}$. More precisely, we search for $t > 0$ such that : $\int_{-\infty}^{t} Lap(x|b)\, dx \geq \mathcal{C} \iff t \geq -\frac{\log(2)+\log(1-\mathcal{C})}{\varepsilon}$. The confidence threshold is $\mathcal{T} = 1 + \lfloor t \rfloor$ (Algorithm 3).

Algorithm 3 Function **confidence_threshold** :

Input: Confidence $\mathcal{C}$
Parameters: DP budget (ε, δ)
Output: Threshold $\mathcal{T}$

$$\mathcal{T} = \Big\lfloor -\frac{\log(2) + \log(1-\mathcal{C})}{\varepsilon_{node}} \Big\rfloor + 1$$

For instance, with $\varepsilon = 0.1$, and $\mathcal{C} = 0.98$, we obtain $t = \lfloor 6.733 \rfloor + 1 = 7$. This means that we can claim with a confidence of 0.98 that if the algorithm decides to add rules, then it respects the minimal support constraint. In practice, the confidence $\mathcal{C}$ will only apply to the later rules of the rule list when the number of samples left becomes scarce.

Privacy budget. Let (ε, δ) be the total privacy budget allocated to the algorithm. Using the sequential and parallel composition for DP mechanisms, we must determine the fraction of the privacy budget to allocate per node (*i.e.*, how much privacy budget should be allocated for the choice of each rule). We will denote these quantities by ε_{node} and δ_{node}. Let K the maximum length of a rule list. While it is common for tree-based models to display the counts for each leaf (*i.e.*, in our case for each rule), this information should also be made differentially private. First in Line 4, the minimum support condition is verified by applying the Laplace mechanism with global sensitivity (satisfying $(\varepsilon, 0)$-DP). Then, the computation of the Gini impurity (Line 11) is made inside the dataset for each candidate rule and only the rule corresponding to the maximum of these noisy Gini impurity values is returned to the algorithm, which is the *Noisy Max Report* mechanism that only accounts for one DP query (as introduced in Section II-B). Computing the two noisy counts of the chosen rule (Algorithm 2) also counts only for one query since the sets of samples caught and not caught are disjoint, which leads to the application of the parallel composition. Finally, with sequential composition, it gives us 3 operations per node, with 2 achieving pure DP. For the default rule, only noisy counts are used and no Gini impurity is computed. Therefore, $\varepsilon_{node} = \frac{\varepsilon}{3K-1}$ and $\delta_{node} = \frac{\delta}{K-1}$.

A variant satisfying pure DP. While our proposed `sm-Laplace` algorithm satisfies approximate (ε, δ)-DP, it is worth observing that the only operation not satisfying pure DP is the noisy max report using the Laplace mechanism along with the smooth sensitivity of the Gini impurity (line 11). This operation satisfies pure DP if the Laplace mechanism is replaced by the Cauchy mechanism within this smooth sensitivity framework, *i.e.,* by replacing line 11 with $\mathcal{G} \leftarrow \mathcal{M}^{S^*_{\cdot,\beta}}_{CAUCHY}(r, \mathscr{G}_{X_{rem}}(\cdot), \varepsilon_{node})$. In such a case, the privacy budget analysis remains unchanged, apart from the δ parameter which is now 0, allowing the whole algorithm to yield pure $(\varepsilon, 0)$-DP guarantees. We coin the resulting variant `sm-Cauchy`. In a nutshell, it is also based on our smooth sensitivity framework, but adds noise from the Cauchy distribution to the computed Gini impurity values to provide pure DP guarantees while still leveraging our smooth sensitivity framework.

V. Experimental Evaluation

In this section, we assess experimentally the effect of smooth sensitivity on the resulting models' accuracy when compared to other approaches based on global sensitivity, for comparable privacy guarantees (pure or approximate DP). We evaluate the performances of the built rule lists in terms of predictive accuracy, robustness to privacy attacks as well as preservation of features' importance.

A. Experimental Settings

For our experiments, we consider three popular datasets: `German Credit`, `Compas` and `Adult` in their binarized versions. Sensitive attributes were removed as their use is prohibited to avoid disparate treatment. The set of rules $\mathscr{R}$ used in these experiments is publicly available. It is made up of conjunctions of up to two Boolean attributes or their negation.

In `German Credit` [30], the classification task is to predict whether individuals have a good or bad credit score. Features are binarized using one-hot encoding for categorical ones and quantiles (2 bins) for numerical ones. The resulting dataset contains $1,000$ samples and we consider 49 premined rules. For `Compas` [2], the objective is to predict whether an individual will re-offend within two years or not. Features are binarized using one-hot encoding for categorical ones and quantiles (with 5 bins) for numerical ones. The resulting dataset contains $6,150$ samples and we have 18 rules. The classification task in `Adult` [30] is to predict whether an individual earns more than $50,000$\$ per year. Categorical attributes are one-hot encoded and numerical ones are discretized using quantiles (3 bins). The resulting dataset contains $48,842$ samples and we use 47 rules (attributes or their negation).

In our experiments, we build upon the baseline `GreedyRL` implementation available in the literature [6][2] and further modify their code to implement our proposed DP mechanisms within the `sm-Laplace` and `sm-Cauchy` algorithms[3]. For each value of ε, we average our results over 100 runs with different random seeds to account for train/test distribution (*i.e.*, train/test split of 70/30) and the randomization due to the application of DP. The value of δ was set $\frac{1}{\|\mathcal{D}\|_1^2}$ and the maximum length for rule lists was set to $K = 5$ as we empirically observed that lower values could impede the model accuracy and higher values do not substantially increase accuracy. Importantly, these trends were confirmed by extensive preliminary experiments and were consistent over all methods and datasets, as further discussed in Section V-D. The other hyperparameters of the proposed algorithm were fixed with preliminary grid search leading to $\mathcal{C} = 0.99$, $\lambda = 0.12$ for `German Credit` and $\lambda = 0.05$ for `Compas` and `Adult`. All our experiments are run on an Intel CORE I7-8700 @3.20GHz CPU.

B. Considered Baselines: DP Greedy Rule Lists Algorithms based on global sensitivity

To assess the effectiveness of our proposed framework leveraging smooth sensitivity, we will compare it with baseline algorithms based on global sensitivity. These baselines are global sensitivity-based variants of our DP greedy rule lists algorithm, and to ensure a fair comparison, the aim of this subsection is to determine their best performing version. More precisely, in these experiments, we compare two different versions of the greedy rule lists learning algorithm using global sensitivity.

Noisy Gini. The first version simply replaces the smooth sensivity of the Gini impurity with its global sensitivity. More

[2]https://github.com/ferryjul/ProbabilisticDatasetsReconstruction

[3]The source code will be released publicly upon acceptance.

precisely, in Line 11, Laplace noise scaled to global sensitivity is added to the Gini Impurity and there is no need to compute the confidence threshold (Line 2) to comply with DP. Thus, some privacy budget can be saved during that step.

Noisy counts. The second version leverages the global sensitivity of counting queries (equal to 1) rather than using the global sensitivity of the Gini impurity which is very high (related to the range of possible values for these queries, i.e., $[0, 1]$ for Gini impurity values and $[0, n]$ for counting queries). We first access the counts $n_c(r)$ and $n_l(r)$ for each rule (number of elements caught / not caught by rule r) in a differentially private way (through the Laplace mechanism). Using Equations (1) and (2), along with the obtained noisy counts, we compute the Gini impurity for each rule, and only keep the rule minimizing it. According to the post-processing property, this quantity remains differentially private. Nonetheless, this access is not a *Noisy Max Report* mechanism anymore but a regular access to all counts for each rule. This means that the privacy budget per node needs to be further split for each rule of the ruleset $\mathscr{R}$, which leads to a factor of $1/2|\mathscr{R}|$ in the denominator.

Note that both considered variants (noisy Gini and noisy counts) satisfy pure DP since they both rely on the Laplace mechanism along with global sensitivity. Alternatively, one could imagine deriving the best Gini impurity using a noisy max report over the counts (i.e., for saving on the privacy budget) but there is no guarantee that the counts returned will belong to the same rule. As such, it becomes impossible to determine the rule with the best Gini using only the noisy max report mechanisms over the counts.

In the following experiments, we consider the non-private greedy rule lists algorithm (`GreedyRL`) as baseline and the two versions of the pure DP greedy rule lists algorithm based on global sensitivity and Laplace noise (namely, noisy Gini and noisy counts). Figure 2 displays the training accuracy of these three algorithms when the privacy parameter ε varies from 10^{-1} to 10^{+4}. Note that while such high values do not provide meaningful privacy guarantees (and are not used later in our experiments), they allow verifying the asymptotic behaviour of the two compared variants of global sensitivity-based approaches. In particular, they confirm that both approaches eventually converge towards the non-private variant when ϵ becomes sufficiently large. The privacy regime of interest to determine which of the two versions performs the best lies within $\varepsilon \in [0.1, 20]$. Indeed, when ε goes over 20, it becomes hard to quantify how the theoretical guarantees apply on realistic settings while a value under 0.1 leads to poorly performing models.

Figure 2 shows that in the considered privacy regime, a rule list model built using the noisy Gini version (that is, using the global sensitivity of the Gini impurity) performs better than a model learnt based on noisy counts. However, it is interesting to note that for very large values of the privacy budget ε, the noisy Gini version is slower to reach the accuracy of the baseline model obtained with `GreedyRL`. When ε is high enough, the noise added is so low that the Gini impurity

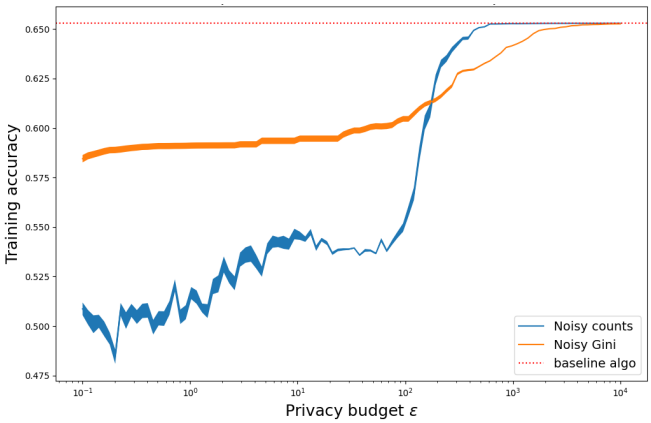


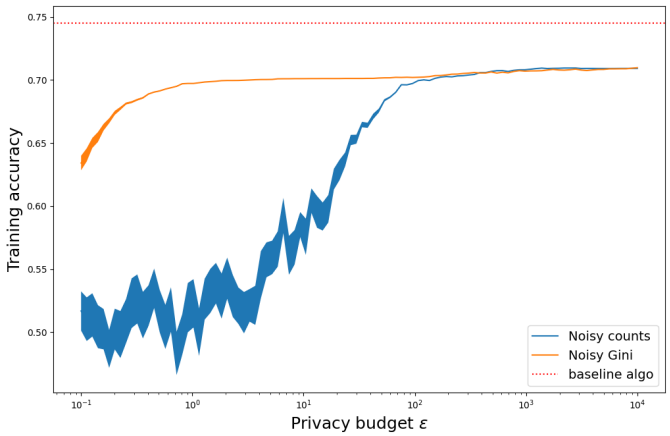
Fig. 2: Comparison of Noisy counts and Noisy Gini versions using global sensitivity (log-scaled), applied on the `Compas` dataset.

scores are ranked according to their original value hence a consistent result with `GreedyRL`. The model using only the noisy counts remains nonetheless interesting in a setting in which the mined ruleset is pre-processed beforehand to a small cardinality (*e.g.* less than a hundred). Indeed, the privacy budget is inversely proportional to the cardinal of the ruleset, so better performances can be expected on smaller instances of rulesets.

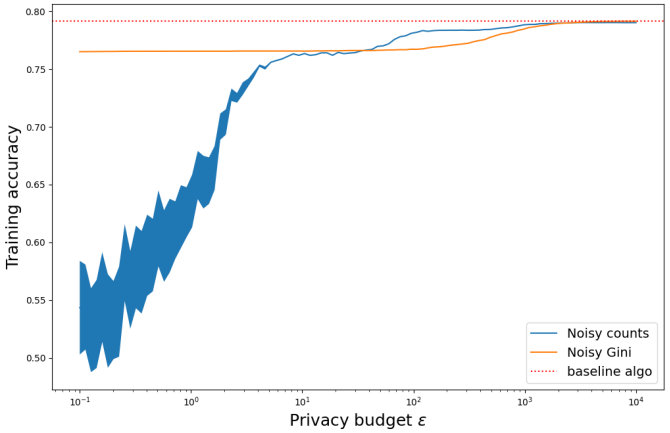
The same trends are observed in Figure 3 for the `German credit` and `Adult` datasets when comparing the two versions leveraging global sensitivity to output the best rule. The method using the global sensitivity of the Gini impurity (noisy gini) remains the best choice for the considered privacy regimes. Finally, we now focus on this version for the remaining of the experiments. This method will be coined as `gl-Laplace`, while its variant replacing the Laplace mechanism by the Gaussian mechanism for computing the noisy Gini impurity based on global sensitivity will be coined as `gl-Gaussian`. Hence, recall that `gl-Laplace` satisfies pure DP while `gl-Gaussian` satisfies approximate DP.

C. Prediction Performance

We now compare the test accuracy of rule list models obtained by Algorithm 1 along with different DP mechanisms. More precisely, we consider two mechanisms based on smooth sensitivity and either Cauchy (`sm-Cauchy`) or Laplace (`sm-Laplace`) noise as well as two mechanisms based on global sensitivity and Gaussian (`gl-Gaussian`) or Laplace (`gl-Laplace`) noise. Finally, we also implemented the Exponential mechanism using the Gini impurity as the utility function for sampling the best rule at each node (`gl-Exponential`). For these experiments, we thus consider three pure DP algorithms: `gl-Laplace`, `gl-Exponential`, and `sm-Cauchy` and two approximate DP algorithms: `gl-Gaussian` and `sm-Laplace`. The baseline test accuracy is given by the non-private `GreedyRL` algorithm.



(a) German credit

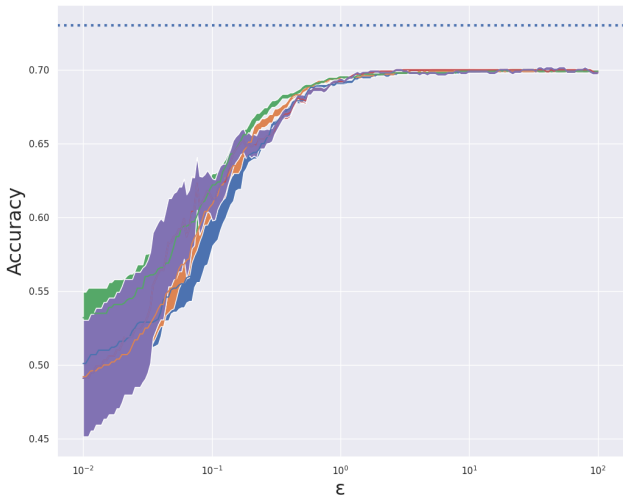


(b) Adult

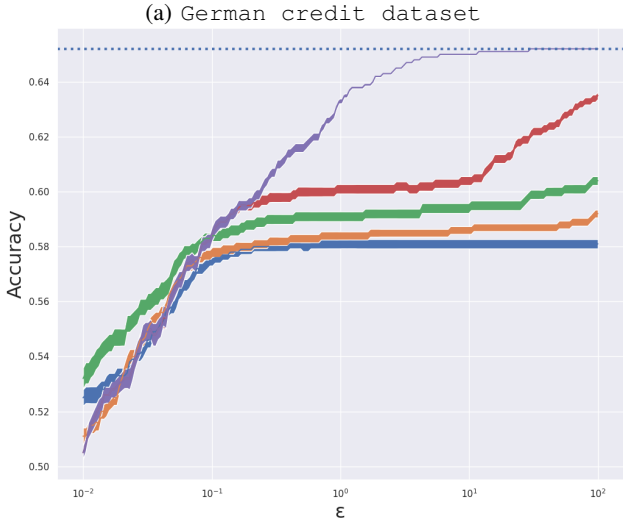
Fig. 3: Comparison of Noisy counts and Noisy Gini versions using global sensitivity (log-scaled) on the German credit and Adult datasets.

In our experiments, we consider privacy budgets $\varepsilon \in [0.01, 100]$ for a total of 200 values with ε uniformly distributed across the logarithmic scale. The results, averaged over 100 runs as described in Section V-A, are displayed in Figure 4 and the test accuracy for $\varepsilon = 10$ is reported in the right part of Table II. Note that we use a logarithmic scale for the values of ε on the x-axis of Figure 4 as is often the case in the literature [13], in order to avoid hiding the tightest privacy budgets (*e.g.*, $\varepsilon \leq 1$).

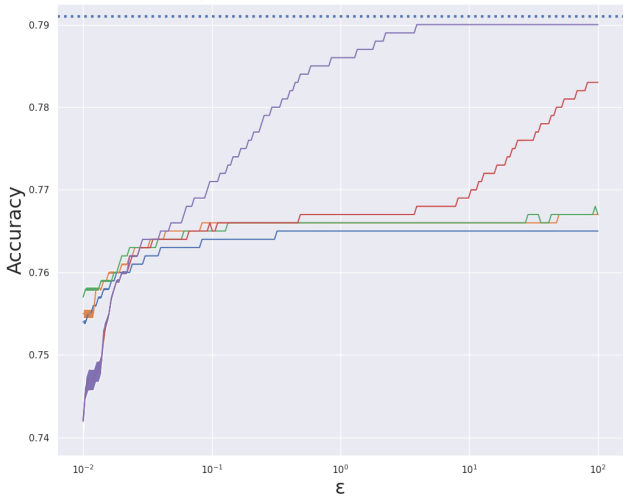
As shown in Figure 4, the two variants based on the smooth sensitivity framework perform particularly well for relatively large datasets (Compas and Adult). In addition, for Compas and Adult, the convergence of the approaches based on smooth sensitivity to the baseline model is very steep. In contrast, DP mechanisms based on global sensitivity usually converge around $\varepsilon \approx 10^3$. For $\varepsilon \geq 0.1$, the mechanisms based on smooth sensitivity either match or outperform the standard global DP approaches. Importantly, the sm-Laplace mechanism consistently and largely outperforms all other approaches on a wide range of privacy budgets of interest. Focusing on pure DP methods, the sm-Cauchy mechanism consistently



(a) German credit dataset



(b) COMPAS dataset



(c) UCI Adult Income dataset

Fig. 4: Comparison based on the test accuracy of different DP rule list algorithms.

performs better than the global sensitivity based frameworks. These experiments confirm the theoretical analysis: for both approximate and pure DP, and for a wide range of privacy budgets, the use of smooth sensitivity in place of global sensitivity allows for better accuracy-privacy trade-offs.

We now compare the two variants based on smooth sensitivity, namely `sm-Cauchy`, satisfying pure DP, and `sm-Laplace`, providing approximate DP guarantees. The `sm-Cauchy` accuracy is much slower to converge than the latter, as can be observed on the `Compas` and `Adult` datasets. The Cauchy distribution has a polynomial decaying tail, which is much heavier than the exponential decaying tail of the Laplace distribution. Thus, out of the many random noise values generated at each step of the algorithm, a few might end up far from the average amplitude, which might deteriorate significantly the accuracy. As a consequence, although the `sm-Cauchy` mechanism provides a good alternative to pure DP mechanisms based on the global sensitivity, we advise to replace it by its Laplace counterpart even if the provided privacy guarantees are slightly weaker.

Compared to the differentially private random forest proposed in [25], we incur at $\varepsilon = 1$ a significantly lower accuracy loss with respect to the non-private model. For this level of privacy, our smooth sensitivity-based `sm-Laplace` algorithm has less than 0.5 absolute accuracy decrease (the accuracy of the proposed DP algorithm is 78.7% vs 79.1% for the non private version) against at least 1.0 for theirs (the accuracy of the DP algorithm is 82% and the non DP version is about 83%).

We now focus on the three best performing methods, namely our proposed smooth sensitivity based `sm-Laplace` and `sm-Cauchy` mechanisms as well as the global sensitivity baseline `gl-Laplace`. We compute and display the standard error of the empirical mean estimator θ whose formula is as follows:

$$err(\theta) = \sqrt{\frac{\text{Var}(\theta)}{\Gamma}}$$

where Γ is the number of observations for a given set of hyperparameters.

Table I presents the standard error of the empirical mean estimator with respect to the random seeds at given values of ε. We observe a high standard error on accuracy at low ε for the smooth sensitivity based models (under 1e-2, standard error can be considered low because it means the accuracy hovers by less than ± 0.01). Global sensitivity models yield a similar standard error at very low ε. As ε values goes up, the standard errors decreases which means that the models' behaviour becomes more deterministic. We also observe that the error decreases when the size of the dataset is higher, although by a small margin between `Compas` and `Adult` but the standard error is five times higher at $\varepsilon = 0.01$ on the `german-credit` dataset. Overall, the smooth sensitivity model using Laplace noise consistently has an equivalent or lower standard error than the global sensitivity model for $\varepsilon \geq 1$. The discrepancy observed for the error of the Cauchy noise model is most likely based on the previous observation about Cauchy distribution's wide tail.

TABLE I: Standard error of the empirical mean estimator across datasets for global and smooth sensitivity based DP mechanisms with respect to test set accuracy.

(a) Standard error for the `sm-Laplace` algorithm.

ε	German	Compas	Adult
0.01	2.0e-02	5.0e-03	4.2e-03
0.1	1.2e-02	5.0e-03	1.0e-03
1	3.6e-03	2.9e-03	4.7e-04
10	3.3e-03	1.2e-03	4.3e-04
100	3.4e-03	1.1e-03	4.1e-04

(b) Standard error for the `sm-Cauchy` algorithm.

ε	German	Compas	Adult
0.01	2.0e-02	5.0e-03	4.2e-03
0.1	1.2e-02	5.0e-03	8.4e-04
1	3.6e-03	4.1e-03	8.5e-04
10	3.1e-03	4.0e-03	9.9e-04
100	3.1e-03	3.6e-03	8.0e-04

(c) Standard error for the `gl-Laplace` algorithm.

ε	German	Compas	Adult
0.01	1.3e-02	6.2e-03	1.7e-03
0.1	8.8e-03	4.5e-03	8.0e-04
1	3.3e-03	4.2e-03	8.0e-04
10	3.1e-03	4.2e-03	8.0e-04
100	3.0e-03	4.0e-03	8.9e-04

We can provide an explanation for this deviation compared to global sensitivity models, and it will partly answer why the smooth sensitivity models have poorer performances at low ε. Indeed, the confidence threshold becomes exceedingly high for these privacy values and the minimum support condition is therefore more likely to fail, which causes the model to output only one rule. Naturally, in that case of underfitting, the smooth sensitivity models cannot perform as well as the classic DP models. This issue could eventually be tackled by assigning more privacy budget to the minimum support condition and less to the noisy Gini impurity computation. This asymptotic behavior however disappears quickly, especially for larger datasets because the confidence threshold variable is independent of the actual value of the minimum support and its value gets relatively smaller as the size of the dataset increases.

D. Hyperparameters and Fine-Tuning

We now provide some insights regarding the influence of the hyperparameters of Algorithm 1 on the accuracy of the resulting rule list models. Consequently, we explain how these hyperparameters' values were chosen.

Naturally, with rules of higher cardinality (i.e., a higher number of conditions on the attributes), we can expect higher accuracy since the splits would be more refined. Observe that it would not affect the privacy budget of the model since the *Noisy Max Report* is independent of the number of elements from which the `argmax` is searched. However, it yields an exponential increase in time complexity.

Optimizing the maximum number K of rules in the rule list proves to be interesting. Indeed, the maximum number of rules K heavily influences the privacy budget per node, but it is also dependent on the minimum support condition λ. Namely, there can be no more than $\min(K, \lfloor \frac{1}{\lambda} \rfloor)$ rules in the output rule list. Decreasing λ enables the inclusion of more rules, but there is a trade-off with the precision of the Laplace noise using smooth sensitivity (the higher λ the less noise added). Overall, our smooth sensitivity method consistently beats the global sensitivity methods for any value of K. A value of $K = 5$ was on average the best performing for all models. For $K = 7$ this value was most of the time not reached as the minimum support condition was not achieved anymore. Models using global sensitivity were also terminated before reaching this depth since the algorithm stops when the Gini is not improved anymore.

E. Robustness to Privacy Attacks

The protection provided by DP aims at hiding the contribution of any individual example to the output of a computation. Then, it is natural to evaluate it in practice using Membership Inference Attacks (MIAs) [4], whose objective is to determine whether an individual was part of a given model's training set or not. Indeed, performing such attacks on both the original greedy rule lists and their DP counterparts, and comparing the MIA success rate, empirically quantifies the effectiveness of the DP protection. However, this approach has two main drawbacks. First, one has to select which MIA(s) to run, and different attacks can come with different success rates. Second, we implemented and used several popular attacks from the literature, and they struggled attacking even the original (non-DP) model, as reported in the Appendix C. An intuitive explanation lies in the simplicity of our considered models: while the output of a deep neural network is a numerical value which can virtually take any value, a rule list classifies an example using one of K rules in which K is reasonably small. While this constitutes an important argument in favor of the use of rule list models, it also makes the empirical assessment of DP more difficult. Thus, for our empirical analysis of the rule lists' robustness to privacy attacks, we rather leverage the (model-agnostic) notion of *distributional overfitting* of a model, introduced by [31]. In a nutshell, *distributional overfitting* aims at quantifying how the model output distribution varies between samples inside and outside the training set. It is thus highly correlated to the vulnerability of a model to MIAs, and can be seen as an upper-bound over their success. Since rule lists are interpretable models, it is entirely possible to know what rule caught a given sample solely by iterating through the successive rules until one evaluates to true for the designated sample. For this reason, we have slightly modified the formula for distributional overfit computation to account for the knowledge that the adversary could get from the structure of the model. More precisely, we define the distributional-overfitting distance with respect to label y as:

TABLE II: Test accuracy and overall vulnerability of the greedy rule lists algorithm and its DP counterpart over 100 runs. Notation 0.507^+ indicates that the non truncated value was greater than the displayed one, and 0.507^- indicates it was smaller.

Dataset	Method	Vulnerability	Accuracy
`Compas`	`GreedyRL`	$0.507^+ \pm 4 \times 10^{-6}$	$0.660 \pm 8 \times 10^{-5}$
`Compas`	`sm-Laplace`	$0.507^- \pm 4 \times 10^{-6}$	$0.658 \pm 1 \times 10^{-4}$
`German`	`GreedyRL`	$0.524 \pm 3 \times 10^{-5}$	$0.711 \pm 5 \times 10^{-4}$
`German`	`sm-Laplace`	$0.516 \pm 5 \times 10^{-5}$	$0.683 \pm 1 \times 10^{-3}$
`Adult`	`GreedyRL`	$0.502^+ \pm 7 \times 10^{-7}$	$0.798 \pm 1 \times 10^{-5}$
`Adult`	`sm-Laplace`	$0.502^- \pm 6 \times 10^{-7}$	$0.795 \pm 1 \times 10^{-5}$

$$\tau(y) = \frac{1}{2} \sum_{r \in RL} \Big| \mathbb{P}[r|y, M = 1] - \mathbb{P}[r|y, M = 0] \Big|$$

in which $\mathbb{P}[r|y, M]$ is the probability that a sample with label y (from the training set ($M = 1$) or outside ($M = 0$)) is captured by rule $r \in RL$.

The overall vulnerability of a model introduced in [31] is then computed as the average of distributional-overfitting distances:

$$V = \frac{1}{2} + \frac{1}{2} \sum_{y \in \{0,1\}} \mathbb{P}[y] \times \tau(y)$$

Intuitively, when measured on finite training and test sets, it measures how much the proportions of samples from each possible label differ among the different rules. If the model's outputs have the exact same distributions inside and outside the training set, the vulnerability is 0.5 indicating that the expected success of a MIA is that of a random guess. We report in Table II the overall vulnerabilities measured on rule lists built with or without the use of DP. Consistent with our preliminary observations that the greedily-built rule lists are resilient to MIAs, the vulnerabilities of both the DP and non-DP models are very low. Nevertheless, we observe that non-DP models consistently exhibit slightly higher vulnerability values than their DP counterparts. This is particularly the case for the smallest dataset, namely `German Credit`, which highlights the relevance of a DP protection for such low-data regime.

F. Preservation of Feature Importance

In order to assess the effect of DP on feature importance, we use the methodology proposed by Dai et al. [32]:

1) Consider a reference model RL_{ref} trained with the `GreedyRL` baseline and a DP model RL_{DP} obtained using one of our proposed algorithms. For each of them, compute their top-k-features using Feature Permutation Importance [33], which calculates how much a feature is correlated to the output of the model. We denote the resulting sets $\text{top}(k, RL_{ref})$ and $\text{top}(k, RL_{DP})$
2) Compute their intersection ratio:
$$I_k = \frac{\big|\text{top}(k, RL_{ref}) \cap \text{top}(k, RL_{DP})\big|}{k}$$

Higher values for I_k indicate a smaller distortion of the feature importance values, hence a better preservation of this property despite the application of DP. We focus on the `Adult` dataset, considering the top-k features for $k = 7$ and maximum length of $K = 5$ for the learnt rule lists. We evaluate how the features selected by `GreedyRL` are conserved when the rule lists are built by `sm-Laplace` and `gl-Laplace`. Intuitively, our objective is to assess if the noise added to comply with DP significantly distorts the most influential features, and whether this trend is different for smooth sensitivity and global sensitivity based frameworks.

We report our results in Table III for two different privacy budgets, namely $\varepsilon = 1$ and $\varepsilon = 10$. For both considered values of ε, the `gl-Laplace` model has poor feature intersection ratio, not exceeding 0.3. In comparison, the smooth sensitivity based `sm-Laplace` has a feature intersection ratio of at least 0.5 and we observe that as ε goes up (*i.e.* when the privacy guarantees are lower), the feature intersection ratio also increases to reach nearly 0.7 for $\varepsilon = 10$. The results consistently show that the smooth sensitivity based `sm-Laplace` yields higher feature intersection than the global sensitivity based `gl-Laplace` in both cases and we infer that this model better conserves feature importance values, less distorting explainability.

TABLE III: Feature Importance Analysis (`GreedyRL` being the baseline method) over 100 runs.

ε	Method	Feature Intersection Ratio
1	`sm-Laplace`	0.542 ± 0.03
1	`gl-Laplace`	0.316 ± 0.04
10	`sm-Laplace`	0.684 ± 0.03
10	`gl-Laplace`	0.308 ± 0.04

Since the noise added by the smooth sensitivity model is much lower than using global sensitivity, the results are expected as the same rules tend to be selected for `GreedyRL` and the smooth `sm-Laplace`. Overall, the feature intersection ratio follows a tendency of increasing when ε goes up.

VI. Discussion

In this paper, we have proposed a new mechanism for learning interpretable models with DP guarantees, leveraging the smooth sensitivity of the Gini impurity. This work directly addresses a key challenge pointed out in the literature [14]. Our experiments illustrated that this new method, with equivalent privacy guarantees, offers a considerable reduction of the accuracy loss compared to the differentially-private methods using global sensitivity.

Several promising research directions emerge from this study. First, adaptive composition [9] could be leveraged to tighten the computation of the privacy budget of our proposed algorithms. Second, it would be insightful to integrate our closed formula for the smooth sensitivity of the Gini impurity within different mechanisms. For instance, the inverse sensitivity mechanism [34] consists in an Exponential Mechanism scaled with the inverse sensitivity (or path length): rather than measuring the variations of a function between two adjacent databases, the authors compute the minimum distance from a database to reach another one achieving a chosen target value. While the exact path length introduced in their paper is often intractable, they derived a method using smooth sensitivity to approximate the path length. This method provides a better alternative to classic noisy mechanisms using smooth sensitivity for pure-DP mechanisms since they do not have to use heavy-tailed distributions such as Cauchy. Third, the smooth sensitivity of the Gini impurity could be used to train DP decision trees, random forests, or other types of interpretable models. As was the case for rule lists, one can expect an improvement of the resulting accuracy-privacy trade-offs. Furthermore, the supports of the leaves of a decision tree are disjoint, which is not necessarily the case for the rules within a rule list. Then, parallel composition can be better leveraged, resulting in tighter privacy guarantees than for rule list models. This could lead to even greater improvements of the accuracy-privacy trade-offs. Finally, integrating DP within certifiably optimal learning algorithms such as `CORELS` is another promising research avenue. Indeed, this tree-based algorithm could naturally be leveraged to implement the exponential mechanism. However, several technical aspects should be carefully considered. In particular, `CORELS` relies on optimality-based bounds to efficiently prune out solutions, but these bounds impede DP. For instance, for all permutations of a given set of rules, `CORELS` only considers the permutation yielding to the best accuracy for further exploration of the space, which breaks the DP guarantees as the probability of outputting a sub-optimal rule list becomes exactly 0. A possible approach to address this is to deactivate all bounds but then `CORELS` essentially breaks down to a complete exploration of the search space, which highly impacts its performance.

References

[1] H. Aziz, Á. Cseh, J. P. Dickerson, and D. C. McElfresh, "Optimal kidney exchange with immunosuppressants," in *Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021, pp. 21–29.

[2] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. propublica (2016)," *ProPublica, May*, vol. 23, 2016.

[3] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, May 2019.

[4] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 3–18.

[5] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," 2022.

[6] J. Ferry, U. Aïvodji, S. Gambs, M.-J. Huguet, and M. Siala, "Probabilistic Dataset Reconstruction from Interpretable Models," in *2nd IEEE Conference on Secure and Trustworthy Machine Learning*, Toronto, Canada, Apr. 2024.

[7] 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*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds., vol.

235. PMLR, 21–27 Jul 2024, pp. 13 545–13 569. [Online]. Available: https://proceedings.mlr.press/v235/ferry24a.html

[8] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in *Proceedings of the Third Theory of Cryptography Conference, TCC, New York, NY, USA, March 4-7*, vol. 3876, 2006, pp. 265–284.

[9] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, p. 211–407, 2014.

[10] M. Gong, Y. Xie, K. Pan, K. Feng, and A. K. Qin, "A Survey on Differentially Private Machine Learning [Review Article]," *IEEE Comput. Intell. Mag.*, vol. 15, no. 2, pp. 49–64, 2020.

[11] Z. Ji, Z. C. Lipton, and C. Elkan, "Differential Privacy and Machine Learning: a Survey and Review," *CoRR*, vol. abs/1412.7584, 2014.

[12] K. Chaudhuri, A. D. Sarwate, and K. Sinha, "A near-optimal algorithm for differentially-private principal components," *J. Mach. Learn. Res.*, vol. 14, no. 1, p. 2905–2943, 2013.

[13] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep Learning with Differential Privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, oct 2016.

[14] S. Fletcher and M. Z. Islam, "Decision Tree Classification with Differential Privacy: A Survey," *ACM Comput. Surv.*, vol. 52, no. 4, 2019.

[15] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and Regression Trees," *Biometrics*, vol. 40, p. 874, 1984.

[16] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proceedings of the Annual ACM Symposium on Theory of Computing*, June 2007, pp. 75–84.

[17] R. L. Rivest, "Learning Decision Lists," *Machine Learning*, vol. 2, pp. 229–246, 1987.

[18] E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, and C. Rudin, "Learning Certifiably Optimal Rule Lists," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, p. 35–44.

[19] E. L. Merrer and G. Trédan, "The bouncer problem: Challenges to remote explainability," *CoRR*, vol. abs/1910.01432, 2019.

[20] J. Thaler, J. Ullman, and S. Vadhan, "Faster algorithms for privately releasing marginals," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2012, pp. 810–821.

[21] A. Daniely and V. Feldman, "Locally private learning without interaction requires separation," *Advances in neural information processing systems*, vol. 32, 2019.

[22] E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, and C. Rudin, "Learning Certifiably Optimal Rule Lists for Categorical Data," *Journal of Machine Learning Research*, vol. 18, no. 234, pp. 1–78, 2018.

[23] C. Singh, K. Nasseri, Y. S. Tan, T. Tang, and B. Yu, "imodels: a python package for fitting interpretable models," p. 3192, 2021.

[24] C. Dwork, N. Kohli, and D. Mulligan, "Differential Privacy in Practice: Expose your Epsilons!" *Journal of Privacy and Confidentiality*, vol. 9, no. 2, 2019.

[25] S. Fletcher and M. Z. Islam, "Differentially private random decision forests using smooth sensitivity," *Expert Systems with Applications*, vol. 78, p. 16–31, 2017.

[26] F. Zafarani and C. Clifton, "Differentially private naïve bayes classifier using smooth sensitivity," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, pp. 406 – 419, 2020.

[27] L. Sun, Y. Zhou, P. S. Yu, and C. Xiong, "Differentially private deep learning with smooth sensitivity," *ArXiv*, vol. abs/2003.00505, 2020.

[28] S. Fletcher and M. Islam, "A differentially private decision forest," in *Proceedings of the Thirteenth Australasian Data Mining Conference (AusDM 15)*, 2015, pp. 99–108.

[29] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, p. 78–87, 2012.

[30] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[31] M. Yaghini, B. Kulynych, and C. Troncoso, "Disparate Vulnerability: on the Unfairness of Privacy Attacks Against Machine Learning," *CoRR*, vol. abs/1906.00389v2, 2019.

[32] J. Dai, S. Upadhyay, U. Aivodji, S. H. Bach, and H. Lakkaraju, "Fairness via explanation quality: Evaluating disparities in the quality of post hoc explanations," in *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, ser. AIES '22. ACM, 2022.

[33] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[34] H. Asi and J. C. Duchi, "Instance-optimality in differential privacy via approximate inverse sensitivity mechanisms," *Advances in neural information processing systems*, vol. 33, pp. 14 106–14 117, 2020.

[35] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. Molloy, and B. Edwards, "Adversarial robustness toolbox v1.2.0," *CoRR*, vol. 1807.01069, 2018.

[36] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," 2021.

APPENDIX A
KEY RESULTS FOR DIFFERENTIAL PRIVACY FROM THE LITERATURE

A. Distance between Databases

We consider tabular datasets, with features being $0-1$ encoded and binary labels. In particular, we assume that a sample from the dataset is made of m features and one label. The universe $\mathscr{X}$ of all possible samples is therefore finite, with cardinality 2^{m+1}. An element a of $\mathscr{X}$ can be expanded to its tuple form as $(a_1, \ldots, a_m, a_{m+1})$ in which a_{m+1} is the label. We define the order relation $\preceq$ on $\mathscr{X}$ such that for $(a, b) \in \mathscr{X}$,

$$a \preceq b \iff \begin{cases} \exists i \in [\![1, m+1]\!], \forall k \in [\![1, i-1]\!],\ a_k \leq b_k \text{ and } a_i < b_i \\ or \\ \forall i \in [\![1, m+1]\!], a_i = b_i \end{cases}$$

$\preceq$ yields the symmetric, reflexive and transitive properties and all elements can be compared within $\mathscr{X}$ so this is a total order relation. As such, $(\mathscr{X}, \preceq)$ is a totally ordered set. We can now introduce the expanded notation for datasets. A dataset x is a collection of elements of $\mathscr{X}$ that we write as a tuple $x = (x_0, \ldots, x_{|\mathscr{X}|}) \in \mathbb{N}^{|\mathscr{X}|}$ such that x_i denotes the number of elements of $\mathscr{X}$ of type i stored in the database x. The number of elements in a dataset x is given by the formula: $\|x\|_1 \coloneqq \sum_{i=0}^{|\mathscr{X}|} x_i$.

With this notation, it is easy to interpret the notion of distances between dataset as the $L1$-norm of their difference. We say that two datasets x, y are adjacent if they vary only by one element (*i.e.* $||x - y||_1 = 1$).

B. Composition and Post-Processing Properties

The *Post-processing theorem* guarantees that one cannot make a differentially private algorithm less private due to post-processing (unless this post-processing itself accesses the data).

Theorem A.1 (Post-processing theorem)**.** *Let $\mathcal{M} : \mathbb{N}^{|\mathscr{X}|} \to \mathscr{Y}$ be an (ε, δ)- differentially private algorithm. For any function $f : \mathscr{Y} \to \mathscr{Z}$, the composition $f \circ \mathcal{M} : \mathbb{N}^{|\mathscr{X}|} \to \mathscr{Z}$ is (ε, δ)-DP.*

The differentially private mechanisms considered in this paper all apply on a $\mathbb{R}^k$ valued function. The *composition of differentially private mechanisms* enables us to scale up from functions to algorithms. More precisely, composition tends to deteriorate the privacy guarantees but to a measurable extent. *Sequential composition* occurs when when several differentially private mechanisms, denoted $m_1, \ldots, m_p$ with respective DP-coefficients $(\varepsilon_1, \ldots, \varepsilon_p)$ are applied onto the same dataset x. Then the generated output : $(m_1(x), \ldots, m_p(x))$ satisfies $(\sum_{i=1}^{p} \varepsilon_1)$-DP. For *parallel composition*, the differentially private mechanisms denoted $m_1, \ldots, m_p$ are applied into disjoints subsets of a given dataset $x = \amalg_{i=1}^{p} x_i$ then the generated output : $(m_1(x), \ldots, m_p(x))$ satisfies $(\max_{i=1}^{p} \varepsilon_i)$-DP.

C. Proof of the iterative Computation Lemma of Smooth Sensitivity (Lemma III.1, from [16])

Let $\mathcal{D}$ and $\mathcal{D}'$ denote two datasets. Note that since : $\{\mathcal{D}' \in \mathbb{N}^{|\mathscr{X}|} : \|\mathcal{D}' - \mathcal{D}\|_1 \leq k\} \subset \{\mathcal{D}' \in \mathbb{N}^{|\mathscr{X}|} : \|\mathcal{D}' - \mathcal{D}\|_1 \leq k+1\}$ we have that $\forall k \in \mathbb{N}, \mathcal{T}_{k+1}(\mathcal{D}) \geq \mathcal{T}_k(\mathcal{D})$.

$$\begin{aligned} S^*_{f,\beta}(\mathcal{D}) &= \max_{\mathcal{D}' \in \mathbb{N}^{|\mathscr{X}|}} LS_f(\mathcal{D}') e^{-\beta ||\mathcal{D} - \mathcal{D}'||_1} \\ &= \max_{k \in \{0,\ldots,n\}} \max_{\substack{\mathcal{D}' \in \mathbb{N}^{|\mathscr{X}|} \\ ||\mathcal{D} - \mathcal{D}'||_1 = k}} LS_f(\mathcal{D}') e^{-\beta ||\mathcal{D} - \mathcal{D}'||_1} \\ &= \max_{k \in \{0,\ldots,n\}} e^{-\beta k} \max_{\substack{\mathcal{D}' \in \mathbb{N}^{|\mathscr{X}|} \\ ||\mathcal{D} - \mathcal{D}'||_1 = k}} LS_f(\mathcal{D}') \\ &= \max_{k \in \{0,\ldots,n\}} e^{-\beta k} \mathcal{T}_k(\mathcal{D}) \end{aligned}$$

The transition from the penultimate to the final line is tricky. $\mathcal{T}_k(\mathcal{D})$ is a max over the closed ball of elements at distance at most k of $\mathcal{D}$, not the sphere of elements at distance k exactly from $\mathcal{D}$. Note that since we consider datasets, the distance can only be an integer.

$$\begin{aligned}\mathcal{T}_{k+1}(\mathcal{D}) &= \max\Big(\max_{\substack{\mathcal{D}'\in\mathbb{N}^{|\mathscr{X}|}\\ ||\mathcal{D}'-\mathcal{D}||_1<k+1}} LS_f(\mathcal{D}'), \max_{\substack{\mathcal{D}'\in\mathbb{N}^{|\mathscr{X}|}\\ ||\mathcal{D}'-\mathcal{D}||_1=k+1}} LS_f(\mathcal{D}')\Big)\\ &= \max\Big(\max_{\substack{\mathcal{D}'\in\mathbb{N}^{|\mathscr{X}|}\\ ||\mathcal{D}'-\mathcal{D}||_1\leq k}} LS_f(\mathcal{D}'), \max_{\substack{\mathcal{D}'\in\mathbb{N}^{|\mathscr{X}|}\\ ||\mathcal{D}'-\mathcal{D}||_1=k+1}} LS_f(\mathcal{D}')\Big)\\ &= \max\Big(\mathcal{T}_k(\mathcal{D}), \max_{\substack{\mathcal{D}'\in\mathbb{N}^{|\mathscr{X}|}\\ ||\mathcal{D}'-\mathcal{D}||_1=k+1}} LS_f(\mathcal{D}')\Big)\end{aligned}$$

Since $\beta > 0$, $e^{-\beta k} > e^{-\beta(k+1)}$ therefore $e^{-\beta k}\mathcal{T}_k(\mathcal{D}) > \mathcal{T}_k(\mathcal{D})e^{-\beta(k+1)}$. However, the quantity $e^{-\beta k}\mathcal{T}_k(\mathcal{D})$ appears in the computation of $S^*_{f,\beta}(\mathcal{D})$ and since it is strictly greater than the left term of $\mathcal{T}_{k+1}(\mathcal{D})$ we can ignore this term and it is equivalent to compute $LS_f(\mathcal{D}')$ either on the ball or on the sphere of radius k in that case.

APPENDIX B
PROOF OF THE SMOOTH SENSITIVITY OF THE GINI IMPURITY (THEOREM IV.1)

A. Case 1: For a minimum support of 1

Assume first that the minimum support Λ is 1. We will then generalize the result. To match with the notations used so far, we will consider a dataset $x \in \mathbb{N}^{|\mathscr{X}|}$ and suppose we take interest at the first node splitting this dataset (it is only a matter of notation), we can therefore rewrite the local sensitivity of the Gini impurity at x as:

$$LS_{\mathcal{G}}(x) = 1 - \Big(\frac{\|x\|_1}{\|x\|_1 + 1}\Big)^2 - \Big(\frac{1}{\|x\|_1 + 1}\Big)^2$$

Consider the function

$$g : \left| \begin{array}{ccc} \mathbb{R}^+ & \longrightarrow & [0,1] \\ x & \longmapsto & 1 - \Big(\frac{x}{x+1}\Big)^2 - \Big(\frac{1}{x+1}\Big)^2 \end{array} \right.$$

Note that : $LS_{\mathcal{G}} \equiv g \circ \|\cdot\|_1$. g is differentiable on $\mathbb{R}^+$ and $\forall x \in \mathbb{R}^+, g'(x) = \dfrac{2(1-x)}{(x+1)^3}$

x	0		1		$+\infty$
$g'(x)$		$+$	0	$-$	
g	0	↗	$\frac{1}{2}$	↘	0

As a reminder, we are trying to determine the smooth sensitivity of the Gini impurity:

$$S^*_{\mathcal{G},\beta}(x) = \max_{k\in\mathbb{N}} \; e^{-k\beta}\mathcal{T}_k(x)$$

where

$$\mathcal{T}_k(x) = \max_{\substack{y\in\mathbb{N}^{|\mathscr{X}|}\\ ||y-x||_1\leq k}} LS_{\mathcal{G}}(y) = \max_{\substack{y\in\mathbb{N}^{|\mathscr{X}|}\\ ||y-x||_1\leq k}} g \circ \|y\|_1 = \max_{\substack{y\in\mathbb{N}\\ y\in[\|x\|_1-k,\|x\|_1+k]}} g(y)$$

We consider that $\|x\|_1 \geq 1$ as we do not build nodes when there are no samples to classify. $[\|x\|_1 - k, \|x\|_1 + k]$ is an interval with integer bounds. With the previous study of g monotonicity, this maximum is reached in $y = \max(1, \|x\|_1 - k)$.

Explanation:

- if $k \geq \|x\|_1 \geq 1$ then $1 \in [\|x\|_1 - k, \|x\|_1 + k]$ so the maximum is the global maximum of g : $1 = \max(1, \|x\|_1 - k)$.
- if $k < \|x\|_1$ then $[\|x\|_1 - k, \|x\|_1 + k] \subset [1, +\infty]$ and g is monotonously decreasing on $[1, +\infty[$ so the maximum is the leftmost bound of the interval : $\|x\|_1 - k = \max(1, \|x\|_1 - k)$.

$$\mathcal{T}_k(x) = g\big[\max(1, \|x\|_1 - k)\big]$$

Now that we obtained a close formula for $\mathcal{T}_k(x)$, we can determine :

$$S^*_{\mathcal{G},\beta}(x) = \max_{k\in\mathbb{N}} \; e^{-k\beta}\mathcal{T}_k(x) = \max_{k\in\mathbb{N}} \; e^{-k\beta}\cdot g\big[\max(1, \|x\|_1 - k)\big]$$

Let

$$\xi_{x,\beta}(t) : \left| \begin{array}{ccc} \mathbb{R}^+ & \longrightarrow & \mathbb{R}^+ \\ t & \longmapsto & e^{-t\beta}\cdot g\big[\max(1, \|x\|_1 - t)\big] \end{array} \right.$$

$$\xi_{x,\beta}(t) = \begin{cases} e^{-t\beta}\cdot g(1) & \text{if } t \geq \|x\|_1 - 1 \\ e^{-t\beta}\cdot g(\|x\|_1 - t) & \text{if } t \leq \|x\|_1 - 1 \end{cases}$$

$\xi_{x,\beta}$ is continuous on $\mathbb{R}^+$ and differentiable on $[0, \|x\|_1 - 1[$ and $]\,\|x\|_1 - 1, +\infty[$. The monotonicity of $\xi_{x,\beta}$ is trivial for high values of t:

$$\forall t \in]\,\|x\|_1 - 1, +\infty[, \xi'_{x,\beta}(t) = -\beta \times e^{-t\beta} g(1) < 0$$

$\forall t \in [0, \|x\|_1 - 1[$,

$$\begin{aligned} \xi'_{x,\beta}(t) &= -\beta \times e^{-t\beta} g(\|x\|_1 - t) + e^{-t\beta} \times (-1) \times g'(\|x\|_1 - t) \\ &= -e^{-t\beta}\big[\beta g(y) + g'(y)\big] \\ &= -e^{-t\beta}\Big[\beta\Big(1 - \frac{y^2}{(y+1)^2} - \frac{1}{(y+1)^2}\Big) + \frac{2(1-y)}{(y+1)^3}\Big] \\ &= -e^{-t\beta} \times \frac{\beta\cdot(y+1)^3 - \beta\cdot y^2(y+1) - \beta\cdot(y+1) + 2(1-y)}{(y+1)^3} \\ &= \frac{e^{-t\beta}}{(1+y)^3} \times \Big[-\beta\cdot(y+1)^3 + \beta\cdot y^2(y+1) + \beta\cdot(y+1) - 2(1-y)\Big] \end{aligned} \qquad \Big\} y \coloneqq \|x\|_1 - t$$

Since $\dfrac{e^{-t\beta}}{(1+y)^3} > 0$ the sign of $\xi'_{x,\beta}(t)$ on $[0, \|x\|_1 - 1[$ is given by the polynomial $P(Y) = -\beta\cdot(Y+1)^3 + \beta\cdot Y^2(Y+1) + \beta\cdot(Y+1) - 2(1-Y) = -2\beta Y^2 + (2-2\beta)Y - 2$.

Let $Q \coloneqq -\beta Y^2 + (1-\beta)Y - 1 = P/2$. P and Q share the same roots, we will therefore study Q. Let Δ the discriminant of polynomial Q. We associate it to the function $\Delta(\beta)$ since its value depends on β. The value of the discriminant gives whether or not the underlying function is monotonous. $\Delta(\beta) = (1-\beta)^2 - 4\beta = (\beta - 3 - 2\sqrt{2})(\beta - 3 + 2\sqrt{2})$

β	0		$\beta_1 \coloneqq 3 - 2\sqrt{2}$		$\beta_2 \coloneqq 3 + 2\sqrt{2}$		$+\infty$
$\Delta(\beta)$		$+$	0	$-$	0	$+$	

- For $\beta \in]3 - 2\sqrt{2}, 3 + 2\sqrt{2}[$, $\Delta(\beta) < 0$ so Q has no roots in $\mathbb{R}$ so it is negative on $\mathbb{R}$.

t	0		$\|x\|_1 - 1$		$+\infty$
$\xi'_{x,\beta}(t)$		$-$	$-$	$-$	
$\xi_{x,\beta}$	$g(\|x\|_1)$	$\searrow$	$\frac{\exp(-(\|x\|_1 - 1)\beta)}{2}$	$\searrow$	0

  In that scenario, $\boxed{S_{\mathcal{G},\beta}(x) = \xi_{x,\beta}(0) = g(\|x\|_1) = LS_{\mathcal{G}}(x)}$
- For $\beta = 3 - 2\sqrt{2}$ or $\beta = 3 + 2\sqrt{2}$, $\Delta(\beta) = 0$ so Q admits a unique root $y_0 = \frac{1-\beta}{2\beta}$ (we will ignore these two values of β as there are enough β that we can choose).

- For $\beta \in]0, 3-2\sqrt{2}[\cup]3+2\sqrt{2}, +\infty[$, $\Delta(\beta) > 0$ so Q admits two distinct roots :

$$y_1 = \frac{1-\beta+\sqrt{(1-\beta)^2-4\beta}}{2\beta} \qquad \text{and} \qquad y_2 = \frac{1-\beta-\sqrt{(1-\beta)^2-4\beta}}{2\beta}$$

y	-∞		y_2		y_1		$+\infty$
$Q(y)$		$-$	0	$+$	0	$-$	

The problem is that the roots $t_1 \coloneqq \|x\|_1 - y_1$ and $t_2 \coloneqq \|x\|_1 - y_2$ might overflow the interval $[0, \|x\|_1 - 1[$. Since $y \mapsto \|x\|_1 - y =: t$ is a strictly decreasing function (it is a bijection from $\mathbb{R}$ to $\mathbb{R}$) we have that $y_2 < y_1 \implies t_2 > t_1$. What we want to study is the mapping from $[y_2, y_1]$ to $[t_1, t_2]$ with respect to the domain of validity for the studied form of $\xi_{x,\beta}$. A case per case analysis (detailed below) shows that the smooth sensitivity of the Gini for these values of β is given by the formula:

$$\begin{aligned} S^*_{\mathcal{G},\beta}(x) &= \max\left[\xi_{x,\beta}(0), \xi_{x,\beta}(\lfloor t_2 \rfloor), \xi_{x,\beta}(\lceil t_2 \rceil)\right] \\ &= \max\left[g(\|x\|_1), e^{-\lfloor t_2 \rfloor \beta} g(\|x\|_1 - \lfloor t_2 \rfloor), e^{-\lceil t_2 \rceil \beta} g(\|x\|_1 - \lceil t_2 \rceil)\right] \end{aligned}$$

We first compute the values of the roots y_1 and y_2 according to β (in particular the asymptotic values).

$$y_1 \underset{\beta\to 0}{\sim} \frac{1}{\beta} \xrightarrow[\beta\to 0]{} +\infty \qquad \text{and} \qquad y_1 \underset{\beta\to+\infty}{\sim} \frac{1}{1-\beta} \xrightarrow[\beta\to+\infty]{} 0^-$$

$$y_2 \underset{\beta\to 0}{\sim} \frac{1}{(1-\beta)^2} \xrightarrow[\beta\to 0]{} 1 \qquad \text{and} \qquad y_2 \underset{\beta\to+\infty}{\sim} -1$$

β	0		$3-2\sqrt{2}$		$3+2\sqrt{2}$		$+\infty$
$y_1(\beta)$	$+\infty$	$\searrow$	$3 > \cdot > 2$		$-1 < \cdot < 0$	$\nearrow$	1^-
$y_2(\beta)$	1	$\nearrow$	$3 > \cdot > 2$		$-1 < \cdot < 0$	$\searrow$	-1

That gives us two cases to treat:
If $\beta \in]0, \beta_1[$. In the case that $\|x\|_1 \geq 5$ (which is a reasonable assumption) $\exists \beta^* \in]0, \beta_1[, \forall \beta \geq \beta^*, 0 < t_1(\beta) < \|x\|_1 - 1$ and $0 < t_2 < \|x\|_1 - 1$ (for all β in the considered interval) which gives : $0 < t_1 < t_2 < \|x\|_1 - 1$

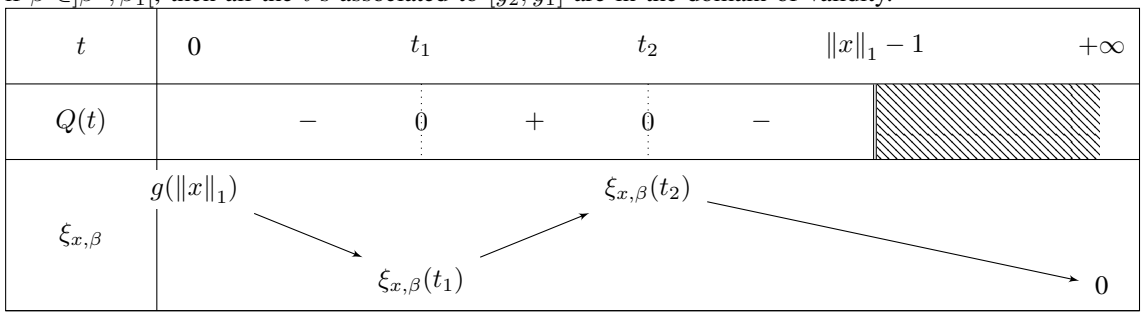
- So if β is too small, then the t's associated to the interval $[y_2, y_1]$ are (< 0) partly outside the domain of validity which yields

t	t_1		0		t_2		$\|x\|_1 - 1$		$+\infty$
$Q(t)$	0	$+$			0	$-$			
$\xi_{x,\beta}$		$\nearrow$	$g(\|x\|_1)$		$\xi_{x,\beta}(t_2)$	$\searrow$			0

Hence :

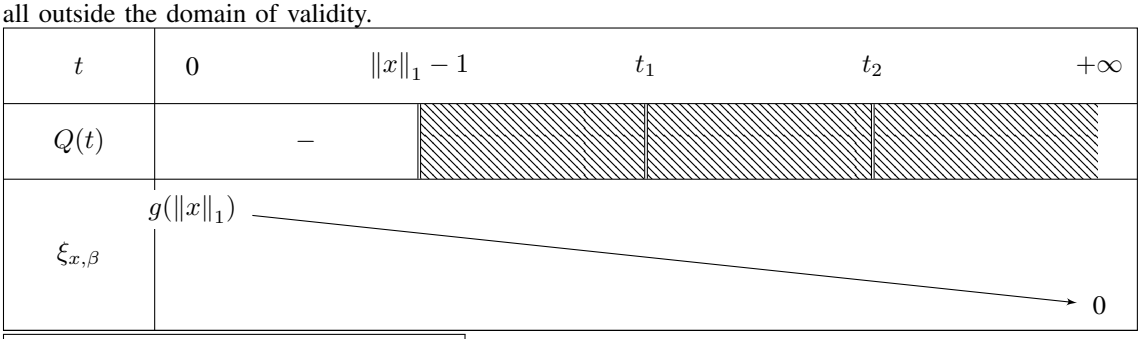
$$\boxed{\begin{array}{l} S^*_{\mathcal{G},\beta}(x) = \max\Big[\xi_{x,\beta}(\lfloor t_2 \rfloor), \xi_{x,\beta}(\lceil t_2 \rceil)\Big] \\ S^*_{\mathcal{G},\beta}(x) = \max\Big[e^{-\lfloor t_2 \rfloor \beta} g(\|x\|_1 - \lfloor t_2 \rfloor), e^{-\lceil t_2 \rceil \beta} g(\|x\|_1 - \lceil t_2 \rceil)\Big] \end{array}}$$

– if $\beta \in]\beta^*, \beta_1[$, then all the t's associated to $[y_2, y_1]$ are in the domain of validity.



$$\boxed{\begin{array}{l} S^*_{\mathcal{G},\beta}(x) = \max\Big[\xi_{x,\beta}(0), \xi_{x,\beta}(\lfloor t_2 \rfloor), \xi_{x,\beta}(\lceil t_2 \rceil)\Big] \\ S^*_{\mathcal{G},\beta}(x) = \max\Big[g(\|x\|_1), e^{-\lfloor t_2 \rfloor \beta} g(\|x\|_1 - \lfloor t_2 \rfloor), e^{-\lceil t_2 \rceil \beta} g(\|x\|_1 - \lceil t_2 \rceil)\Big] \end{array}}$$

If $\beta \in]\beta_2, +\infty[$. $t_2 > \|x\|_1 - 1$ and $t_1 > \|x\|_1 - 1$ which means that the t's associated to the $[y_2, y_1]$ are ($> \|x\|_1 - 1$) all outside the domain of validity.



$$\boxed{S^*_{\mathcal{G},\beta}(x) = \xi_{x,\beta}(0) = g(\|x\|_1) = LS_{\mathcal{G}}(x)}$$

B. Case 2: Generalization. Assume now that $\Lambda \in \mathbb{N}^*$

It is easy to prove that: $\mathcal{T}_k(x) = g\big[\max(\Lambda, \|x\|_1 - k)\big]$. Therefore, we define the function:

$$\xi_{x,\beta}(t) : \left| \begin{array}{ccc} \mathbb{R}^+ & \longrightarrow & \mathbb{R}^+ \\ t & \longmapsto & e^{-t\beta} \cdot g\big[\max(\Lambda, \|x\|_1 - t)\big] \end{array} \right.$$

$$\xi_{x,\beta}(t) = \begin{cases} e^{-t\beta} \cdot g(\Lambda) & \text{if } t \geq \|x\|_1 - \Lambda \\ e^{-t\beta} \cdot g(\|x\|_1 - t) & \text{if } t \leq \|x\|_1 - \Lambda \end{cases}$$

$\xi_{x,\beta}$ is continuous on $\mathbb{R}^+$ and differentiable on $[0, \|x\|_1 - \Lambda[$ and $]\,\|x\|_1 - \Lambda, +\infty[$. $\xi_{x,\beta}$ derivatives remain unchanged but the bounds are shifted (from 1 to Λ). The roots y_1 and y_2 are unchanged (they solely depend on β). Instead of re-doing the case per case analysis, we will propose the following exact method.

1) Compute t_1 and t_2 if the roots y_1 and y_2 exist
2) Compute the relative positions of t_1 and t_2 with respect to 0 and $\|x\|_1 - \Lambda$
3) We know that $\xi_{x,\beta}$ is increasing between t_1 and t_2 granted that they are in the $[0, \|x\|_1 - \Lambda[$ interval so there is an eventual max in this interval, to compare to $\xi_{x,\beta}(0)$ and $\xi_{x,\beta}(\|x\|_1 - \Lambda)$

This achieves the proof of Theorem IV.1.

APPENDIX C
MEMBERSHIP INFERENCE ATTACKS - DETAILED RESULTS

Figure 5 illustrates how a dataset is split into different subsets to train a Membership Inference Attack (MIA) model. Note that the $\setminus$ symbol represents the minus operation on sets (set difference).

We consider two MIAs from the ART toolkit[4] [35]. The results we obtained for a black-box MIA[5] using Random Forests are presented in Figure 6. The ROC curves are displayed in log scale to highlight the results at low FPR since it is the relevant regime for MIAs [5]. They show that, as mentioned in Section V-E, (even non-DP) rule lists are already resilient to MIAs. On the smallest `German credit` dataset, we observed a slightly higher distributional overfitting, which results in slightly higher TPRs at low FPR.

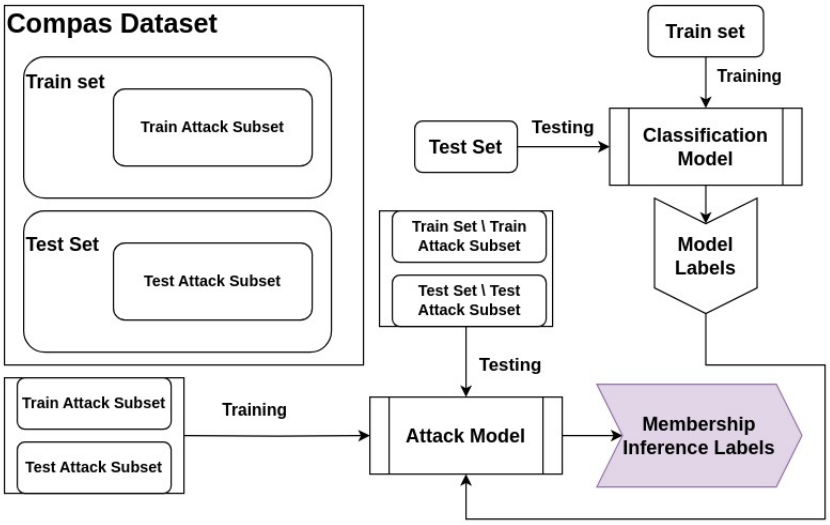
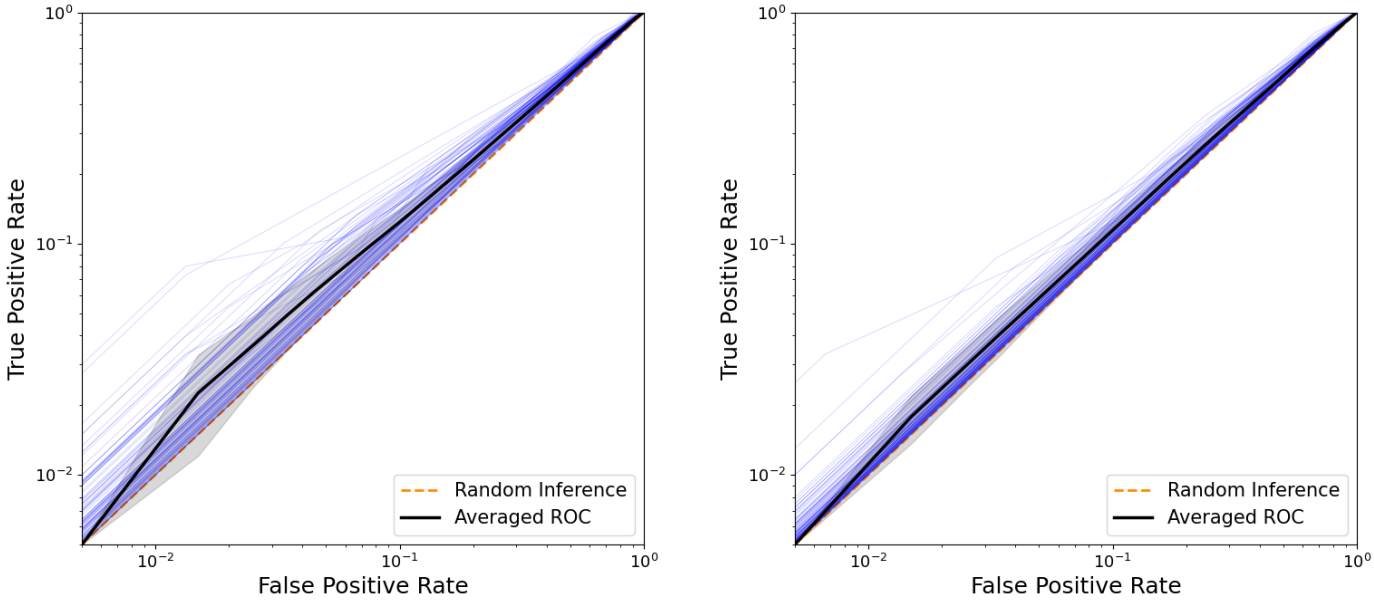


Fig. 5: Pipeline of Membership Inference Attack

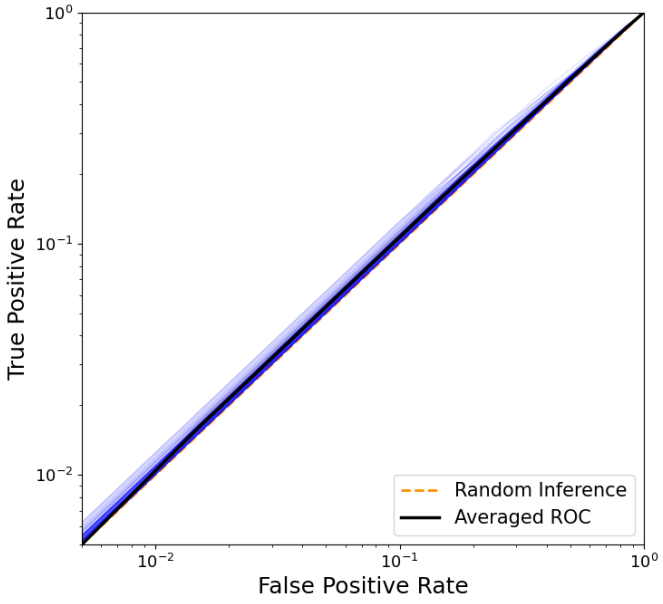
We also considered the Label Only Membership Inference Attack[6] [36] but results were sub-par due to the datasets used. Indeed, the rule lists use as input binarized features whereas the attack explores the latent variables space by studying how the model output varies when the features values are tweaked. The issue here is that the model can only read features that are 0 or 1 and therefore we had to truncate the latent space exploration to the much sparser space of $\{0,1\}^m$, making it inefficient. In addition, since the datasets are binarized, some features are actually a one-hot-encoding of a categorical feature, which means it does not make sense that several of them can be set to 1. An interesting avenue of research would be to use the latent space exploration on the non-binarized features and re-apply the binarization process at each step. This is unfortunately computationally expensive and would likely lead to similar results due to the loss of information incurred by the binarization step preceding inference by the model.

[4]https://github.com/Trusted-AI/adversarial-robustness-toolbox/wiki/ART-Attacks\#4-inference-attacks
[5]https://adversarial-robustness-toolbox.readthedocs.io/en/latest/modules/attacks/inference/membership_inference.html\#membership-inference-black-box
[6]https://adversarial-robustness-toolbox.readthedocs.io/en/latest/modules/attacks/inference/membership_inference.html\#membership-inference-label-only-decision-boundar

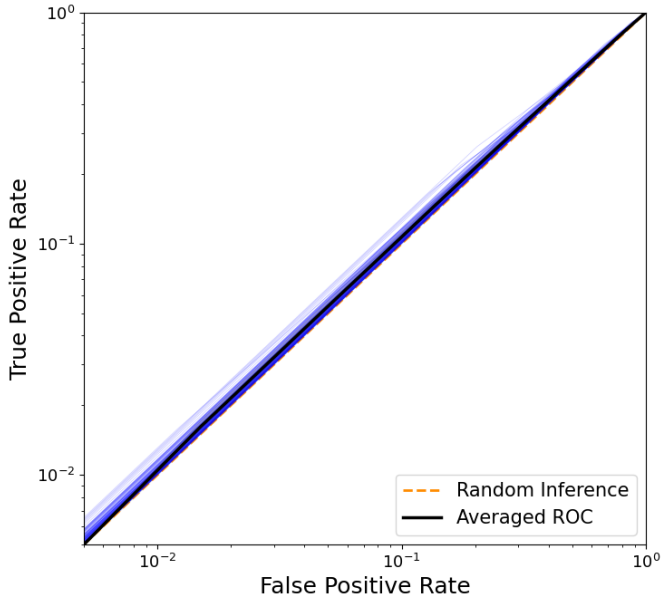


(a) `German credit - GreedyRL`

(b) `German credit - sm-Laplace`



(c) `Compas - GreedyRL`



(d) `Compas - sm-Laplace`

Fig. 6: ROC Curves of Membership Inference Attacks on the DP `sm-Laplace` and on the baseline `GreedyRL` models.