Differentially-private supervised learning with random decision trees

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider new supervised learning method that involves decision trees. The construction of the decision tree is completely random and very fast since the choice of nodes' attributes does not depend on the data at all. We show a simple algorithm that uses only logarithmic number of independently selected random decision trees and correctly classifies labels of most of the data points. The random structure of the tree enables us to adapt our setting to the differentially-private scenario. Thus we also propose a differentially-private version of the algorithm. In both settings we give detailed theoretical analysis. To the best of our knowledge, such an analysis has never been done before. We present and compare three different versions of the algorithm: majority voting, threshold averaging and probabilistic averaging. We show that first two versions give the best accuracy, also for conservative users requiring high privacy guarantees. In particular, a simple majority voting rule, that was not considered in the previous works regarding this topic, is much less sensitive to the choice of forest parameters than previously considered methods. This makes it an especially good candidate for the differentially-private classifier. We relate the generalization error of the classifier to the average tree accuracy and explain how to choose the number of random decision trees to avoid overfitting and loss of accuracy due to the perturbation error added to preserve differential-privacy. Thus we obtain a simple differentially-private scalable learning method using random decision trees and with minimal memory requirements, that correctly classifies big fraction of all data points. We show that sampling from the set of all decision trees suffices, even in the setting when differential-privacy guarantees must be maintained.

1 Introduction

Decision tree is one of the fundamental structures used in machine learning. Constructing a tree of good quality is a hard computational problem though. Needless to say, the choice of the optimal attribute according to which data partitioning should be performed in any given node of the tree requires nontrivial calculations involving data points located in that node. Nowadays, with an increasing importance of mechanisms preserving privacy of the data handled by machine learning algorithms, the need arises to construct these algorithms with privacy guarantees (see: [1], [2], [3], [4]). One of the strongest currently used notions of privacy is the so-called *differential privacy* that was introduced by [5] in a quest to achieve the dual goal of maximizing data utility and preserving data confidentiality. A differentially-private database access mechanism preserves the privacy of any individual in the database, irrespectively of the amount of auxiliary information available to an adversarial database client. Differential-privacy techniques add Laplacian noise to perturb data. The magnitude of the added noise depends on the sensitivity of the statistics that are being output. Even though the overall scheme looks simple, in practice it is usually very difficult to obtain a reasonable level of differential privacy and at the same time maintain good accuracy. This is the case since usually too big perturbation error needs to be added. In particular, this happens when machine learning computations access data frequently during the entire execution of the algorithm and output structures that are very sensitive to the data. This is also an obstacle for proposing a scheme that computes an optimal decision tree in the differentially-private way. In such a scenario the attribute chosen in every node as well as every additional information stored there depends on the data and that is why it must be perturbed in order to keep the desired level of differential privacy. Big perturbation added in this setting leads to the substantially smaller quality of the constructed tree.

In this paper we consider a different approach, where instead of constructing one differentially-private decision tree we construct $O(\log(n))$ random decision trees (n is the size of the dataset). An attribute according to which selection is performed in any given node is chosen uniformly at random from all the attributes, independently from the dataset in that node. In the continuous case, the threshold value for the chosen attribute is then also chosen uniformly at random from the range of all possible values. That simple rule enables us to construct each decision tree very fast. After a sufficient number of random decision trees is constructed, the classification of every point from a dataset takes place. Classification is done according to majority voting, threshold averaging or probabilistic averaging. Voting/averaging instead of just taking the best tree for a given dataset is important - it prevents overfitting and enables to add smaller perturbation error to obtain the same level of differential privacy (see: [6] for applications of the voting methods). For each scheme we analyze both: non-differentially-private and differentially-private setting. We compare all three versions and show that first two are good candidates for the differentially-private classifiers. Surprisingly, a simple majority voting rule, that was not considered before in this context, is competitive with the threshold averaging rule that has been empirically studied. It has also one crucial advantage over it - is much less sensitive to the choice of the parameters of the random forest (such as the number of the trees and the height of the tree). We are the first to give a comprehensive theoretical analysis of all three models. We relate the empirical error and the generalization error of the classifier to the average tree accuracy and explain quantitatively how the quality of the system depends on the number of chosen trees. Our experiments show that in practice we can choose far fewer trees and still get good-quality guarantees. To sum it up, we obtain fast and scalable algorithm with minimal memory requirements which, contrary to most learning algorithms dealing with random decision trees, takes only one pass over the data to construct the classifier. Since most of the structure of each random decision tree is constructed without examining the data, the algorithm suits the differentially-private scenario very well.

In Section 2 we describe previous work regarding random decision trees. In Section 3 we introduce our model and the notion of differential privacy. In Section 4 we present a differentially-private supervised algorithm that uses random decision trees. In Section 5 we state theoretical results regarding quality and privacy guarantees of the presented approach. In Section 6 we show results of the experiments we conducted on several real datasets, In the Appendix we prove all theoretical statements from the main body of the paper and add some additional experiments.

2 Previous work versus our approach

This paper is inspired by two articles: [7] and [8]. In [7] the authors introduced the random decision tree approach. They considered a special setting of the non-differentially-private scenario and proved that by increasing the number of random decision trees the accuracy of prediction is increasing too. Besides they presented experimental results showing that the approach using random decision trees is surprisingly effective in practice. However they did not derive any upper bounds on the generalization or empirical error. Thus, even though the experimental results showed that the algorithm works, it was not known how the error depends on data. Selecting too many random decision trees may lead to overfitting. That problem also was not addressed in that paper. In our paper we give upper bounds on errors and prove that in practice the logarithmic number of random decision trees suffices to get good quality guarantees. We also consider differentially private setting since the random decision trees, due to their structure, seem to be perfect candidates to use by the differentially-private algorithm involving decision trees. In comparison to [7], we consider a broader range of possible algorithm settings, also in the non-differentially-private scenario. We show that a simple majority voting scheme significantly outperforms the probabilistic averaging rule studied before and is competitive with the majority voting rule. The dependence of the quality of all the methods on the chosen level of differential privacy, the height of the tree and the number of trees in the forest, which is in the focus of our analysis and is crucial for applying those methods in practice, was not considered before. W. Fan and others analyzed the random decision tree approach also in [9] and [10]. To the best of our knowledge they did not obtain results discussed in this paper. It is worth to mention that several machine learning random structures were considered way before. For instance, in [11] a random forest approach was presented. However, in that method a computation that involves dataset is performed to choose an attribute for every node. The algorithm is not as simple and fast as the one considered in this paper that uses random decision trees. Results regarding random trees and their potential application can be found also here: [12], [13].

It was first noticed in [8] that random decision trees may turn out to be an effective tool for constructing a differentially-private decision tree classifier. The authors showed that an approach that averages over random decision trees gives good results in practice. However, in that paper the approach was presented just as a very efficient heuristic. No theoretical results regarding the quality of the differentially-private algorithm using random decision trees were given. In particular, the authors did not calculate how many random decision trees should be chosen to get reasonable accuracy. This is a fundamental problem in the differentially-private setting. As mentioned before, more random decision trees require adding bigger perturbation error that may decrease the efficiency of the learning algorithm. Besides they may cause overfitting. In this paper we thoroughly investigate this phenomenon.

3 Preliminaries

3.1 Differential privacy

Differential privacy is a model of privacy for database access mechanism. It guarantees that small changes in a database (removal or addition of the element) does not change substantially the output of the mechanism.

Definition
3.1 ([5]). *A randomized algorithm $\mathcal{M}$ satisfies ϵ-differential-privacy if for all databases $\mathcal{D}_1$ and $\mathcal{D}_2$ differing on at most one element, and all $S \in Range(\mathcal{M})$,*

$$\mathbb{P}(\mathcal{M}(\mathcal{D}_1) = S) \leq \exp(\epsilon) \cdot \mathbb{P}(\mathcal{M}(\mathcal{D}_2) = S). \tag{1}$$

The probability is taken over the coin tosses of $\mathcal{M}$.

The smaller ϵ, the stronger level of differential privacy is obtained. Assume that the non-perturbed output of the mechanism can be encoded by the function f. A mechanism $\mathcal{M}$ can compute a differentially-private noisy version of f over a database $\mathcal{D}$ by adding noise with magnitude calibrated to the sensitivity of f.

Definition
3.2 ([5]). *The global sensitivity $S(f)$ of a function f is the smallest number s such that for all $\mathcal{D}_1$ and $\mathcal{D}_2$ which differ on at most one element,*

$$|f(\mathcal{D}_1) - f(\mathcal{D}_2)| \leq s. \tag{2}$$

Let $Lap(0, \lambda)$ denote the Laplace distribution with mean 0 and standard deviation λ.

Theorem
3.1 ([5]). *Let f be a function on databases with range R^m, where m is the number of rows of databases. Then, the mechanism that outputs $f(\mathcal{D}) + (Y_1, \ldots, Y_m)$, where Y_i are drawn i.i.d from $Lap(0, S(f)/\epsilon)$, satisfies ϵ-differential-privacy.*

Stronger privacy guarantees and more sensitive functions need bigger variance of the Laplacian noise being added. Differential privacy is preserved under composition, but with an extra loss of privacy for each conducted query.

Theorem
3.2 ((Dwork et al., 2006)). ***(Composition Theorem)*** *The sequential application of mechanisms $\mathcal{M}_i$, each giving ϵ_i-differential privacy, satisfies $\sum_i \epsilon_i$-differential-privacy.*

More information about differential privacy can be found in [14] and [15]. Machine learning algorithms in context of the differential privacy have been considered before (see: [16], [17]).

3.2 The model

All data points are taken from $\mathcal{F}^m$, where m is the number of the attributes and $\mathcal{F}$ is either a discrete set or the set of real numbers. We assume that for every attribute $attr$ its smallest and largest possible value ($\min(attr)$ and $\max(attr)$ respectively) are publicly available and that the labels are binary. We will consider only binary decision trees. All our results can be easily translated to the setting where inner nodes of the tree have more than two children. However all important details of the algorithm and its analysis are covered by the binary tree setting. Therefore, if $\mathcal{F}$ is discrete then we will assume that $\mathcal{F} = \{0, 1\}$, i.e. each attribute is binary. In the continuous setting for each inner node of the tree we store the attribute according to which the selection is done and the threshold value of this attribute. All decision trees considered in this paper are complete and of a fixed height h that does not depend on the data.

```
Input: train and test sets: $Train, Test$, height $h$ of the decision tree
Random forest construction phase:
    construct $k = \theta(\log(n))$ random decision trees by chossing for each inner node of the tree
    independently at random its attribute (uniformly from the set of all the attributes);
    in the continuous case for each chosen attribute $attr$ choose independently at random
    a threshold value uniformly from the interval $[\min(attr), \max(attr)]$
Training phase:
    For $d \in Train$  {add $d$ to the forest by updating $\theta_l$ for every leaf corresponding to $d$}
Testing phase:
    For $d \in Test$  {
        if (majority voting)  {
            compute $num^d$ - the number of the trees classifying $d$ as $+$;
            classify $d$ as $+$ iff $num^d > \frac{k}{2}$; }
        if (threshold averaging)  {
            compute $\theta^d = \frac{1}{k}\sum_{l \in \mathcal{L}} \theta_l$, where $\mathcal{L}$ is a set of all leaves of the forest that correspond to $d$;
            classify $d$ as $+$ iff $\theta^d > \frac{1}{2}$; }
        if (probabilistic averaging)  {
            compute $\theta^d = \frac{1}{k}\sum_{l \in \mathcal{L}} \theta_l$, where $\mathcal{L}$ is a set of all leaves of the forest that correspond to $d$;
            classify $d$ as $+$ with probability $\theta^d$ ;
            /*random tosses here are done independently from all other previously conducted*/  }}
Output: Classification of all $d \in Test$
```

Algorithm 1: Non-differentially-private RTD classifier

Let T be a random decision tree and let l be one of its leaves. We denote by θ_l the fraction of all training points in l with label $+$. If l does not contain any of the training points we choose the value of θ_l uniformly at random from $[0, 1]$. The set M of all possible decision trees is of size $|M| = m^{2^{h+1}-1}$ in the binary setting. In the continuous case M is infinite but in practice we can also assume that it is finite (in practice the continuous model is always discretized, the real values are given with some fixed precision, etc). Thus from now on, without loss of generality, we will assume that M is finite. It can be very large but we do not care about it since we will never need the actual size of M in our analysis. For a given tree T and given data point d denote by w^T_d the fraction of points (from the training set id d is from it and the test set otherwise) with the same label as d that end up in the same leaf of T as d. We call it the *weight of* d *in* T. Notice that a training point d is classified correctly by T in the single-tree setting iff its weight in T is larger than $\frac{1}{2}$ (for a single decision tree we consider majority voting model for points classification). The average value of w^T_d over all trees of M will be denoted as w_d and called the *weight of* d *in* M. We denote by $\sigma(d)$ the fraction of trees from M with the property that most of the points of the leaf of the tree containing d have the same label as d (again, the points are taken from the training set if d is from it and from the test set otherwise). We call $\sigma(d)$ *the goodness of* d *in* M.

For a given dataset $\mathcal{D}$ the average tree accuracy $e(\mathcal{D})$ of a random decision tree model is an average accuracy of the random decision tree from M, where the accuracy is the fraction of data points that a given tree classifies correctly (accuracy is computed under assumption that the same distribution $\mathcal{D}$ was used in both: the training phase and test phase).

4 Supervised learning with random decision trees - algorithms

We will propose now differentially-private algorithms (see: Algorithm 1) for supervised learning with random decision trees for the model described by us above. For completeness we will first present analogous results for the non-differentially-private scenario (see: Algorithm 2). All algorithms are very simple to implement.

As mentioned previously, we give three versions for each setting (differentially- and non-differentially-private): majority voting, threshold averaging and probabilistic averaging. Only variables n_l^+, n_l^- stored in leaves depend on data. This fact will play crucial role in the analysis of the differentially-private version of the algorithm, where laplacian error is added to the point counters at every leaf with variance calibrated to the number of all trees used by the algorithm. The non-differentially-private setting is presented in Algorithm 1 and the differentially-private in Algorithm 2. We denote by $g(\lambda)$ an independent copy of the Laplacian with pdf: $\frac{\lambda}{2}e^{-|x|\lambda}$.

```
Input: train and test sets: $Train$, $Test$, height $h$ of the decision tree, privacy parameter $\eta$
Random forest construction phase:
    the same as in Algorithm 1
Training phase:
    For $d \in Train$ {
        find the leaf $l$ for $d$ in every tree and update $n_l^+$, $n_l^-$, where:
            $n_l^+$ - the number of training points with label $+$ belonging to that leaf;
            $n_l^-$ - the number of training points with label $-$ belonging to that leaf; }
    For every leaf $l$ {
        calculate $n_l^{p,+} = n_l^+ + g(\frac{\eta}{k})$;   calculate $n_l^{p,-} = n_l^- + g(\frac{\eta}{k})$;
        if ($n_l^{p,+} < 0$ or $n_l^{p,-} < 0$ or ($n_l^{p,+} = 0$ and $n_l^{p,-} = 0$))
            choose $\theta_l^p$ uniformly at random from $[0, 1]$;
        else   let $\theta_l^p = \frac{n_l^{p,+}}{n_l^{p,+}+n_l^{p,-}}$;
        publish $\theta_l^p$ for every leaf; }
Testing phase: as in Algorithm 1 but replace : $\theta_l$ by $\theta_l^p$
Output: Classification of all $d \in Test$
```

Algorithm 2: η-Differentially-private RTD classifier

5 Theoretical results

In this section we derive upper bounds on the empirical and generalization error for all the versions of the algorithm described earlier. We also show how to find the number of random decision trees in the differentially-private setting to obtain both: good accuracy and privacy guarantees.

We start with two results that may seem to be a little bit technical but, as we will see later, give an intuition why the random decision tree approach works very well in practice.

Theorem
5.1. *Assume that the average tree accuracy of the set M of all decision trees of height h on the training/test set $\mathcal{D}$ is $e = 1 - \epsilon$ for some $0 < \epsilon \leq \frac{1}{2}$. Then the average goodness $\sigma(d)$ of a training/test point d in M is at least $e \geq \frac{1}{2}$.*

Theorem
5.2. *Assume that the average tree accuracy of the set M of all decision trees of height h on the training/test set $\mathcal{D}$ is $e = 1 - \epsilon$ for some $0 < \epsilon \leq \frac{1}{2}$. Then the average weight w_d of a training/test point d in M is at least $e^2 + (1-e)^2 \geq \frac{1}{2}$.*

The theorems above imply that if the average accuracy of the tree is better than random, then this is also reflected by the average values of w_d and σ_d. The latter fact is crucial for the theoretical analysis since we will show that if the average values of w_d and σ_d are slightly better than random then this implies very small empirical and generalization error. Furthermore, for most of the training/test points d their values of σ_d and w_d are well concentrated around those average values. That is in a nutshell, why the random decision trees approach works. Notice that Theorem 5.1 gives better quality guarantees than Theorem 5.2.

5.1 Non-differentially-private setting

We consider here majority voting and threshold averaging. All the results for the probabilistic averaging as well as all the proofs regarding all versions will be given in the Appendix. Let us first take the non-differentially-private setting.

Theorem
5.3. *Let $K > 0$. Assume that the average tree accuracy of the set M of all decision trees of height h on the training/test set $\mathcal{D}$ is $e = 1 - \epsilon$ for some $0 < \epsilon \leq \frac{1}{2}$. Let μ be: the fraction of training/test points with goodness in M at least $\sigma = \frac{1}{2} + \delta$ / $\sigma = \frac{1}{2} + \delta + \frac{1}{K}$ for $0 < \delta < \frac{1}{2}$ (in the majority version) or: the fraction of training/test points with weight in M at least $w = \frac{1}{2} + \delta$ / $w = \frac{1}{2} + \delta + \frac{1}{K}$ for $0 < \delta < \frac{1}{2}$ (in the threshold averaging version). Then Algorithm 1 for every $C > 0$ and $k = \frac{(1+C)\log(n)}{2\delta^2}$ selected random decision trees gives empirical error $err_1 \leq 1 - \mu$ with probability $p_1 \geq 1 - \frac{1}{n^C}$. The generalization error $err_2 \leq 1 - \mu$ will be achieved for $k = \frac{(1+C)\log(n)}{2(\frac{\delta}{2})^2}$ trees with probability $p_2 \geq p_1 - 2^{h+3}ke^{-2n\phi^2}$, where $\phi = \frac{\delta}{2(4+\delta)2^h K}$. Probabilities p_1 and p_2 are under random coin tosses used to construct the forest.*

Let us comment on this result. Note that parameter e is always in the range $[\frac{1}{2}, 1]$. The more decision trees that classify data in the nontrivial way (i.e. with accuracy greater than $\frac{1}{2}$), the larger the value of e is. The result above in particular implies that if most of the points have goodness/weight in M a little bit larger than $\frac{1}{2}$ then both errors are very close to 0. This is indeed the case - the average point's goodness/weight in M, as Theorem 5.1 and Theorem 5.2 say, is at least e / $e^2 + (1-e)^2$. The latter expression is greater than $\frac{1}{2}$ if the average tree accuracy is just slightly bigger than the worst possible. Besides goodness/weight of most of the points, as was tested experimentally, is well concentrated around that average goodness/weight. We conclude that if the average accuracy of the decision tree is separated from $\frac{1}{2}$ (but not necessarily very close to 1) then it suffices to classify most of the datapoints correctly. The intuition behind this result is as follows: if the constructed forest of the decision trees contains at least few "nontrivial trees" giving better accuracy than random then they guarantee correct classification of most of the points.

If we know that the average tree accuracy is itself big then techniques used to prove Theorem 5.3 give us more direct bounds on the empirical and generalization errors. The following is true:

Theorem
5.4. *Let $K > 0$. Assume that the average tree accuracy of the set M of all decision trees of height h on the training/test set $\mathcal{D}$ is $e = 1 - \epsilon$ for some $0 < \epsilon \leq \frac{1}{2}$. Then Algorithm 1 for every $C > 0$, $0 < \delta < \frac{1}{2}$ and $k = \frac{(1+C)\log(n)}{2\delta^2}$ selected random decision trees gives empirical error: $err_1 \leq \frac{\epsilon}{\frac{1}{2}-\delta}$ (in the majority version) or: $err_1 \leq \frac{2\epsilon-2\epsilon^2}{0.5-\delta}$ (in the threshold averaging version) with probability $p_1 \geq 1 - \frac{1}{n^C}$. The generalization error: $err_2 \leq \frac{\epsilon+\frac{1}{K}}{\frac{1}{2}-\delta}$ (in the majority version) or: $err_2 \leq \frac{2(\epsilon+\frac{1}{K})-2(\epsilon+\frac{1}{K})^2}{0.5-\delta}$ (in the threshold averaging version) will be achieved for $k = \frac{(1+C)\log(n)}{2(\frac{\delta}{2})^2}$ trees with probability $p_2 \geq p_1 - 2^{h+3}ke^{-2n\phi^2}$, where $\phi = \frac{\delta}{2(4+\delta)2^h K}$. Probabilities p_1 and p_2 are under random coin tosses used to construct the forest.*

5.2 Differentially-private setting

As in the previous section, we consider here majority voting and threshold averaging setting. We have the following:

Theorem
5.5. *Assume that we are given a parameter $\eta > 0$. Let $K > 0$. Assume that the average tree accuracy of the set M of all decision trees of height h on the training/test set $\mathcal{D}$ is $e = 1 - \epsilon$ for some $0 < \epsilon \leq \frac{1}{2}$. Let μ be the fraction of training/test points with: goodness in M at least $\sigma = \frac{1}{2} + \delta + \frac{1}{K}$ / $\sigma = \frac{1}{2} + \delta + \frac{2}{K}$ (in the majority version) or: weight in M at least $w = \frac{1}{2} + \delta + \frac{1}{K}$ / $w = \frac{1}{2} + \delta + \frac{2}{K}$ (in the threshold averaging version) for $0 < \delta < \frac{1}{2}$. Then Algorithm 2 for k selected random decision trees and differential privacy parameter η gives empirical error $err_1 \leq 1 - \mu$ with probability $p_1 \geq 1 - n(e^{-\frac{k\delta^2}{2}} + e^{-\frac{k}{2}} + ke^{-\frac{\lambda n \eta}{k}})$ and generalization error $err_2 \leq 1 - \mu$ with*

probability $p_2 \geq p_1 - 2^{h+3}ke^{-2n\phi^2}$*, where:* $\lambda = \frac{\delta}{24K \cdot 2^h}$ *and* $\phi = \frac{\delta}{2(4+\delta)2^h K}$*. Probabilities* p_1 *and* p_2 *are under random coin tosses used to construct the forest. Furthermore, we always have:* $\mu \geq 1 - \frac{\epsilon}{\frac{1}{2} - \delta - \frac{1}{K}}$ */* $\mu \geq 1 - \frac{\epsilon}{\frac{1}{2} - \delta - \frac{2}{K}}$ *in the majority version and:* $\mu \geq 1 - \frac{2\epsilon - 2\epsilon^2}{\frac{1}{2} - \delta - \frac{1}{K}}$ */* $\mu \geq 1 - \frac{2\epsilon - 2\epsilon^2}{\frac{1}{2} - \delta - \frac{2}{K}}$ *in the threshold averaging version.*

Again, as in the non-differentially-private case, we see that if there are many points of goodness/weight in M close to the average goodness/weight then empirical and generalization error are small. Notice however that right now increasing the number of the trees too much may not only lead to overfitting but has also an impact on the empirical error (term $ke^{-\frac{\lambda n \eta}{k}}$ in the lower bound on p_1). More trees means bigger variance of the single Laplacian used in the leaf of the tree. This affects tree quality. The theorem above describes this phenomenon quantitatively.

If the average tree accuracy is big enough then the following result becomes of its own interest.

Theorem 5.6. *Assume that we are given a parameter* $\eta > 0$*. Assume besides that the average tree accuracy of the set* M *of all decision trees of height* h *on the training set* $\mathcal{D}$ *is is* $e = 1 - \epsilon$ *for some* $0 < \epsilon \leq \frac{1}{2}$*. Let* $0 < \delta < \frac{1}{2}$*. Let* $\gamma = \frac{1}{2^h \cdot 9600}$ *and let* k_{opt} *be the integer value for which the value of the function* $f(k) = e^{-\frac{k}{200}} + 2ke^{-\frac{\gamma\sqrt{n}\eta}{k}}$ *is smallest possible. Then with probability at least* $p = 1 - n(e^{-\frac{k_{opt}}{200}} + 2k_{opt}e^{-\frac{\gamma\sqrt{n}\eta}{k_{opt}}} + e^{-\frac{n}{2}})$ *the* η*-differentially-private threshold averaging version of Algorithm 2 gives empirical error at most* $\frac{1}{8} + \frac{9}{2}\epsilon - 5\epsilon^2$ *for the forest with* k_{opt} *randomly chosen decision trees. Probability* p *is under random coin tosses used to construct the forest.*

Both theorems show that logarithmic number of random decision trees in practices suffices to obtain good accuracy and level of differential privacy. As in the previous cases, we also have the version of the last result for the generalization error and the majority voting version of the algorithm. Since it can be derived using very similar techniques to those in the proofs of Theorem 5.5 and Theorem 5.6, this time we omit it and leave to the reader.

6 Experiments

The experiments were performed on the benchmark datasets[1]: *banknote authentication*, *Blood Transfusion Service Center*, *Congressional Voting Records*, *Mammographic Mass*, *Mushroom*, *adult*, *covertype* and *quantum*. 90% of each dataset was used for training and the remaining part for testing. All codes that we use to perform our experiments are publicly released.

Table 1: Comparison of the performance of CART with random forests.

			Method												
Dataset	n	m	**rpart**	**n-dpRFMV**			**n-dpRFTA**			**dpRFMV**			**dpRFTA**		
			Error	Error	k	h	Error	k	h	Error	k	h	Error	k	h
ban_aut	1372	5	3.65	3.09	21	15	3.46	17	9	5.44	21	11	5.22	7	12
BTSC	748	5	18.92	22.19	1	14	22.47	1	14	23.42	1	14	23.42	1	13
CVR	435	16	9.30	9.05	19	6	5.95	13	9	8.10	15	9	6.90	15	9
Mam_M	961	6	21.88	16.95	9	12	16.21	19	15	16.95	5	12	17.37	9	8
Mush	8124	22	3.33	0.83	21	15	0.26	13	14	4.69	3	13	4.16	3	15
adult	32561	123	17.75	21.70	3	14	21.58	3	14	22.18	3	11	21.72	7	11
covertype	581012	54	26.90	33.39	21	15	30.80	21	15	38.75	3	13	37.82	3	13
quantum	50000	78	32.08	34.81	21	15	33.06	19	14	39.91	21	13	39.01	13	9

We first compare the test error (%) obtained using five different methods: open-source implementation of CART called *rpart* [18], non-differentially-private (*n-dp*) and differentially-private (*dp*) random forest with majority voting (*RFMV*) and with threshold averaging (*RFTA*). For all methods except *rpart* we also report the number of trees in the forest (k) and the height of the tree (h) for which the smallest error was obtained, where we explored: $h \in \{1, 2, 3, \ldots, 15\}$ and $k \in \{1, 3, 5, \ldots, 21\}$.

[1] downloaded from http://archive.ics.uci.edu/ml/datasets.html, http://osmot.cs.cornell.edu/kddcup/ and http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets

In all the experiments the differential privacy parameter η was set to $\eta = 1000/n_{tr}$, where n_{tr} is the number of training examples. Table 1 captures the results (for each experiment we report average test error over 10 runs). The performance of random forest with probabilistic averaging (*RFPA*) was significantly worse than the competitive methods (*RFMV*, *RFTA*, *rpart*) and is not reported in the table. The performance of *RFPA* is however shown in the following figures.

Next set of results[2] (Figure 1 and 2) are reported for an exemplary dataset (*Mushrooms*) and for the following methods: *dpRFMV*, *dpRFTA* and *dpRFPA*. Note that similar results were obtained for other datasets and they are deferred to the Supplementary material. In Figure 1a we report the test error vs. h for selected settings of k. In Figure 1b we also show minimal, average and maximal test error vs. h for dpRFMV, whose performance is overall the best. Similarly, in Figure 1c we report the test error vs. k for two selected settings of h. In Figure 1d we also show minimal, average and maximal test error vs. k for dpRFMV.

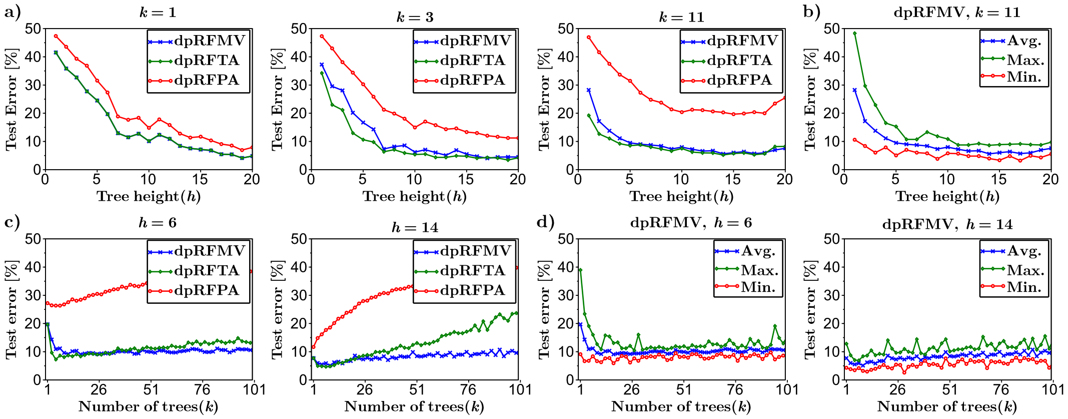

Figure 1: Comparison of dpRFMV, dpRFTA and dpRFPA. $\eta = 1000/n_{tr} = 0.137$. Test error resp. vs. **a)** h across various settings of k and vs. **c)** k across various settings of h; Minimal, average and maximal test error resp. vs. h (**b)**) and vs. k (**d)**) for dpRFMV.

Finally, in Figure 2a we report test error for various settings of η and two selected settings of h. For each experiment k was chosen from the set $\{1, 2, \ldots, 101\}$ to give the smallest error. Additionally, in Figure 2b we show how the test error changes with k for a fixed h and different levels of η.

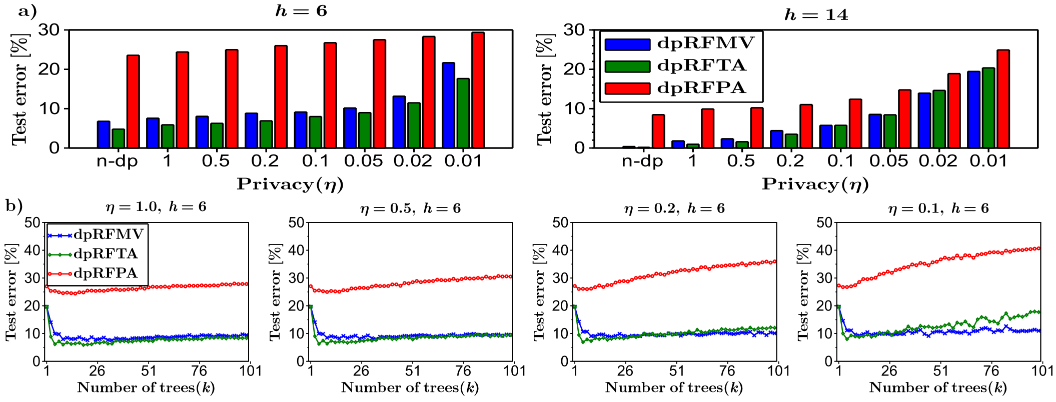

Figure 2: Comparison of dpRFMV, dpRFTA and dpRFPA. **a)** Test error vs. η for two settings of h. **b)** Test error vs. k for fixed h and across different settings of η.

Figure 2a shows that in most cases *dpRFTA* outperforms remaining differentially-private classifiers, however it requires careful selection of the forest parameters (h and k) in order to obtain the optimal performance as is illustrated on Figure 1c and 2b. This problem can be overcome by using *dpRFMV* which has comparable performance to *dpRFTA* but is much less sensitive to the setting of the forest parameters. Therefore *dpRFMV* is much easier to use in the differentially-private setting.

[2] All figures in this section and in the Supplementary material should be read in color.

References

[1] R. Agrawal and R. Srikant. Privacy-preserving data mining. *ACM SIGMOD*, pages 439–450, 2000.

[2] K. Chaudhuri and C. Monteleoni. Privacy-preserving logistic regression. In *NIPS*, pages 289–296, 2008.

[3] W. Du and J. Zhan. Using randomized response techniques for privacy-preserving data mining. In *KDD*, pages 505–510, 2003.

[4] Krzysztof Choromanski, Tony Jebara, and Kui Tang. Adaptive anonymity via *b*-matching. In *NIPS*, pages 3192–3200, 2013.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *TCC*, pages 265–284, 2006.

[6] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26:1651–1686, 1998.

[7] W. Fan, H. Wang, P.S Yu, and S. Ma. Is random model better? on its accuracy and efficiency. *ICDM 2003*, pages 51–58, 2003.

[8] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright. A practical differentially private random decision tree classifier. *Transactions on data privacy*, pages 273–295, 2012.

[9] W. Fan. On the optimality of probability estimation by random decision trees. *AAAI*, pages 336–341, 2004.

[10] W. Fan, , E. Greengrass, J. McCloskey, P. Yu, and K. Drummey. Effective estimation of posterior probabilities: Explaining the accuracy of randomized decision tree approaches. *ICDM*, pages 154–161, 2005.

[11] L. Breiman. Random forests. *Machine Learning*, 45, 2001.

[12] T. Ho. Random decision forest. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pages 278–282, 1995.

[13] A. Choromanska, K. Choromanski, G. Jagannathan, and C. Monteleoni. Differentially-private learning of low dimensional manifolds. In *ALT*, pages 249–263, 2013.

[14] Rob Hall, Alessandro Rinaldo, and Larry A. Wasserman. Differential privacy for functions and functional data. *Journal of Machine Learning Research*, 14(1):703–727, 2013.

[15] F. McSherry and K. Talwar. Mechanism design via differential privacy. *FOCS'07*, pages 94–103, 2007.

[16] D. Mir. Differentially-private learning and information theory. In *EDBT/ICDT Workshops*, pages 206–210, 2012.

[17] P. Jain, P. Kothari, and A. Thakurta. Differentially private online learning. In *COLT*, pages 24.1–24.34, 2012.

[18] Terry M. Therneau, Beth Atkinson, and Brian Ripley. rpart: Recursive Partitioning. *http://CRAN.R-project.org/package=rpart*, 2011.

Differentially-private supervised learning with random decision trees (Supplementary Material)

7 Differential privacy guarantees

First we will prove that versions of the algorithm with Laplacian noise added, presented in the main body of the paper, are indeed η-differentially private.

Proof. Notice that in every version of the algorithm to obtain the forest of random decision trees with perturbed counters in leaves we need k queries to the private data (this is true since the structure of the inner nodes of the trees does not depend at all on the data and data subsets corresponding to leaves are pairwise disjoint). Furthermore, the values that are being perturbed by the Laplacian noise are simple counts of global sensitivity 1. Thus we can use use Theorem 3.1 and Theorem 3.2 to conclude that in order to obtain η-differential privacy of the entire system we need to add a $Lap(0, \frac{k}{\eta})$ to every count in the leaf. This proves that our algorithms are indeed η-differentially private. □

Above, we have just proven the ”differentially-private” parts of all the statements from the theoretical section.

8 Empirical and generalization errors

8.1 Preliminaries

We will now prove results regarding empirical and generalization errors of all the variants of the algorithm mentioned in the paper as well as Theorem 5.1 and Theorem 5.2. Without loss of generality we will assume that all attributes are binary (taken from the set $\{0, 1\}$). It can be easily noticed that the proofs can be directly translated to the continuous case. We leave this simple exercise to the reader.

Let us first state our results for the probabilistic averaging setting since we have not done it in the main body of the paper.

Theorem
8.1. *Let $K > 0$. Assume that the average tree accuracy of the set M of all decision trees of height h on the training/test set $\mathcal{D}$ is $e = 1 - \epsilon$ for some $0 < \epsilon \leq \frac{1}{2}$. Let $C > 0$ be a constant. Let $0 < \delta, c < 1$. Then with probability at least $p_1 = (1 - \frac{1}{n^C})(1 - e^{-2nc^2})$ the probabilistic averaging version of Algorithm 1 gives empirical error $err_1 \leq 2\epsilon - 2\epsilon^2 + \delta + c$ and with probability $p_2 \geq p_1 - 2^{h+3}ke^{-2n\phi^2}$, where $\phi = \frac{\delta}{2(4+\delta)2^h K}$, it gives generalization error $err_2 \leq 2(\epsilon + \frac{1}{K}) - 2(\epsilon + \frac{1}{K})^2 + \delta + c$. Probabilities p_1, p_2 are under random tosses used to construct the forest.*

Notice that this result is nontrivial for almost the entire range $[0, \frac{1}{2}]$ of ϵ for δ, c close to 0 and large K. This is the case since: $1 - 2\epsilon + 2\epsilon^2 \geq \frac{1}{2}$ and the equality holds only for $\epsilon = \frac{1}{2}$.

Let us assume now the differentially-private setting:

We have the following:

Theorem
8.2. *Assume that we are given a parameter $\eta > 0$. Let $K, c > 0$ and $0 < \delta < 1$. Assume that the average tree accuracy of the set M of all decision trees of height h on the training/test set $\mathcal{D}$ is*

$e = 1 - \epsilon$ for some $0 < \epsilon \leq \frac{1}{2}$. Let $\lambda = \frac{\delta}{24K \cdot 2^h}$. Then for k selected random decision trees the η-differentially-private probabilistic averaging version of Algorithm 2 gives empirical error $err_1 \leq 2(\epsilon + \frac{1}{K}) - 2(\epsilon + \frac{1}{K})^2 + \delta + c$ with probability $p_1 \geq (1 - n(e^{-\frac{k\delta^2}{2}} + e^{-\frac{k}{2}} + ke^{-\frac{\lambda n \eta}{k}}))(1 - e^{-2nc^2})$ and generalization error $err_2 \leq 2(\epsilon + \frac{2}{K}) - 2(\epsilon + \frac{2}{K})^2 + \delta + c$ with probability $p_2 \geq p_1 - 2^{h+3}ke^{-2n\phi^2}$, where: $\phi = \frac{\delta}{2(4+\delta)2^h K}$. Probabilities p_1 and p_2 are under random coin tosses used to construct the forest.

The following is also true:

Theorem
8.3. *Assume that we are given a parameter $\eta > 0$. Assume besides that the average tree accuracy of the set M of all decision trees of height h on the training set $\mathcal{D}$ is $e = 1 - \epsilon$ for some $0 < \epsilon \leq \frac{1}{2}$. Let $\gamma = \frac{1}{2^h \cdot 9600}$ and let k_{opt} be the integer value for which the value of the function $f(k) = e^{-\frac{k}{200}} + 2ke^{-\frac{\gamma\sqrt{n}\eta}{k}}$ is smallest possible. Then with probability at least $p = 1 - n(e^{-\frac{k_{opt}}{200}} + 2k_{opt}e^{-\frac{\gamma\sqrt{n}\eta}{k_{opt}}} + e^{-\frac{n}{2}})(1 - e^{-\frac{n}{200}})$ the η-differentially-private probabilistic averaging version of Algorithm 2 gives empirical error at most $\frac{1}{5} + \frac{19}{10}\epsilon - 2\epsilon^2$ for the forest with k_{opt} randomly chosen decision trees. Probability p is under random coin tosses used to construct the forest.*

Let us introduce now some useful notation that will be very helpful in the proofs we present next.

We denote by n the size of the dataset (training or test) $\mathcal{T}$. Let us remind that m is the number of attributes of any given data point, h is the height of the random decision tree and M is the set of all random decision trees under consideration.

Lets focus first on classifying with just one decision tree. Fix some decision tree T_j and one of its leaves. Assume that it contains a points with label: $-$ and b points with label: $+$. We associate label $-$ with that leaf if $a > b$ and label 1 otherwise. To classify a given point using that tree we feed our tree with that point and assign to a point a label of the corresponding leaf. Denote by m^i the number of data points that were correctly classified by a tree T_i. Denote $e^i = \frac{m^i}{n}$. We call e^i the *quality* (or *accuracy*) of the tree T_i. Note that obviously we always have: $e^i \geq \frac{1}{2}$, since for every leaf of any given tree the majority of the data points from that leaf are classified correctly. Denote: $e = \frac{1}{|M|}\sum_{i=1}^{|M|} e^i$. We call e the average tree accuracy. This parameter measures how well data points are classified on average by a complete decision tree of a given height h. Note that $e \geq \frac{1}{2}$. Denote $t = 2^h$. Parameter t is the number of leaves of the decision tree.

For $i = 1, 2, ..., |M|$ and $j = 1, 2, ..., t$ denote by n^i_j the number of points from the dataset in the j^{th} leaf of a decision tree T_i. Denote by m^i_j the number of points from the dataset in the j^{th} leaf of the decision tree T_i that were classified correctly. Denote $e^i_j = \frac{m^i_j}{n^i_j}$ for $n^i_j > 0$ and $e^i_j = 1$ for $n^i_j = 0$. Note that $e^i_j \geq \frac{1}{2}$ for every i, j. Note also that we have: $n = n^i_1 + ... + n^i_t$ and $m^i = m^i_1 + ... + m^i_t$. Denote by a^i_j the number of data points in the j^{th} leaf of the decision tree T_i that are of label 0. Denote by b^i_j the number of data points in the j^{th} leaf of the decision tree T_i that are of label 1.

We will use frequently the following structure in the proofs. Let $\mathcal{G}$ be a bipartite graph with color classes: $\mathcal{A}$, $\mathcal{B}$ and weighted edges. Color class $\mathcal{A}$ consists of n points from the dataset. Color class $\mathcal{B}$ consists of $2t|M|$ elements of the form $y^{i,b}_j$, where $i \in \{1, 2, ..., |M|\}$, $b \in \{0, 1\}$ and $j \in \{1, 2, ..., t\}$.
Data point $x \in \mathcal{A}$ is adjacent to $y^{i,1}_j$ iff it belongs to larger of the two groups (these with labels: 0 and 1) of the data points that are in the j^{th} leaf of the decision tree T_i. An edge joining x with $y^{i,1}_j$ has weight e^i_j. Data point $x \in \mathcal{A}$ is adjacent to $y^{i,0}_j$ iff it belongs to smaller of the two groups of the data points that are in the j^{th} leaf of the decision tree T_i. An edge joining x with $y^{i,0}_j$ has weight $1 - e^i_j$. Note that the degree of a vertex $y^{i,1}_j$ is m^i_j and the degree of a vertex $y^{i,0}_j$ is $n^i_j - m^i_j$.

In the proofs we will refer to the size of the set of decision trees under consideration as: $|M|$ or k (note that k is used in the main body of the paper).

We are ready to prove Theorem 5.1 and Theorem 5.2.

Proof. We start with the proof of Theorem 5.2. Note that from the definition of w_d we get:

$$\sum_{d \in \mathcal{T}} w_d = \frac{1}{|M|} \sum_{i=1}^{|M|} \sum_{j=1}^{t} (m_j^i e_j^i + (n_j^i - m_j^i)(1 - e_j^i)).$$

Therefore, using formula on m_j^i, we get:

$$\sum_{d \in \mathcal{T}} w_d = \frac{1}{|M|} \sum_{i=1}^{|M|} \sum_{j=1}^{t} (n_j^i (e_j^i)^2 + n_j^i (1 - e_j^i)^2).$$

Note that we have: $\sum_{i=1}^{|M|} \sum_{j=1}^{t} n_j^i = n|M|$. From Jensen's inequality, applied to the function $f(x) = x^2$, we get: $\sum_{i=1}^{|M|} \sum_{j=1}^{t} \frac{n_j^i}{|M|n} (e_j^i)^2 \geq (\sum_{i=1}^{|M|} \sum_{j=1}^{t} \frac{n_j^i e_j^i}{|M|n})^2 = (\frac{\sum_{i=1}^{|M|} \sum_{j=1}^{t} m_j^i}{|M|n})^2 = (\frac{en|M|}{n|M|})^2 = e^2$, where e is the average quality of the system of all complete decision trees of height h (the average tree accuracy). Similarly, $\sum_{i=1}^{|M|} \sum_{j=1}^{t} \frac{n_j^i}{|M|n} (1 - e_j^i)^2 \geq (1 - e)^2$. Thus we get:

$$\sum_{d \in \mathcal{T}} w_d \geq n(e^2 + (1 - e)^2).$$

That completes the proof of Theorem 5.2. The proof of Theorem 5.1 is even simpler. Notice that for any data point d the expression $\sigma(d) \cdot |M|$ counts the number of decision trees from M that classified d correctly (follows directly from the definition of θ). Thus we have: $\sum_{d \in \mathcal{T}} \sigma(d) \cdot |M| = \sum_{i=1}^{|M|} m^i$. Therefore $\frac{1}{n} \sum_{d \in \mathcal{T}} \sigma(d) = \frac{1}{|M|} \sum_{i=1}^{|M|} e^i$ and we are done. □

We need one more technical result, the Azuma's inequality:

Lemma
8.1. *Let* $\{W_n, n \geq 1\}$ *be a martingale with mean* 0 *and suppose that for some non-negative constants:* α_i, β_i *we have:* $-\alpha_i \leq W_i - W_{i-1} \leq \beta_i$ *for* $i = 2, 3, \ldots$. *Then for any* $n \geq 0$, $a > 0$*:*

$$\mathbb{P}(W_n \geq a) \leq e^{-\frac{2a^2}{\sum_{i=1}^{n} (\alpha_i + \beta_i)^2}} \quad \textit{and} \quad \mathbb{P}(W_n \leq -a) \leq e^{-\frac{2a^2}{\sum_{i=1}^{n} (\alpha_i + \beta_i)^2}}.$$

8.2 Majority voting and threshold averaging setting - empirical error

We will now prove parts of theorems: 5.3 and 5.4 regarding empirical errors.

Proof. Again, we start with the analysis of the threshold averaging. Take i^{th} random decision tree T_i^R, where $i \in \{1, 2, \ldots, k\}$. For a given data point d from the training set let X_i^d be a random variable defined as follows. If d does not belong to any leaf of T_i^R then let $X_i^d = 0$. Otherwise let a_i^R be the number of points from the training set with label 0 in that leaf and let b_i^R be the number of points from the training set with label 1 in that leaf. If d has label 0 then we take $X_i^d = \frac{a_i^R}{a_i^R + b_i^R}$. Otherwise we take $X_i^d = \frac{b_i^R}{a_i^R + b_i^R}$. Denote $X^d = \frac{X_1^d + \ldots + X_k^d}{k}$. When from the context it is clear to which data point we refer to we will skip upper index and simply write X or X_i respectively.
Fix some point d from the training set. Note that if $X > \frac{1}{2}$ then point d is correctly classified. Notice that the weight of the point d denoted as w_d is nothing else but the sum of weights of all the edges of $\mathcal{G}$ incident to d divided by the number of all trees (or the average weight of an edge indecent to d if we consider real-valued attributes). Note that we have $EX = w_d$ and that from Theorem 5.2 we get:

$$\sum_{d \in \mathcal{T}} w_d \geq n(e^2 + (1 - e)^2).$$

Take $0 < \delta < \frac{1}{2}$. Denote by μ the fraction of points d from the training data such that $w_d \geq \frac{1}{2} + \delta$. From the lower bound on $\sum_{d \in \mathcal{T}} w_d$, we have just derived, we get: $(\frac{1}{2} + \delta)(1 - \mu)n + \mu n \geq n(e^2 + (1 - e)^2)$, which gives us:

$$\mu \geq 1 - \frac{2\epsilon - 2\epsilon^2}{0.5 - \delta},$$

where $\epsilon = 1 - e$.
Take point d from the training set such that $w_d \geq \frac{1}{2} + \delta$. Denote by p_d the probability that d is misclassified. We have:

$$p_d \leq \mathbb{P}(\frac{X_1 + ... + X_k}{k} \leq w_d - \delta).$$

Denote: $Z_i = X_i - w_d$ for $i = 1, 2, ..., k$. We have:

$$p_d \leq \mathbb{P}(Z_1 + ... + Z_k \leq -k\delta).$$

Note that, since $w_d = EX$ and random variables X_i are independent, we can conclude that $\{Z_1, Z_1 + Z_2, ..., Z_1 + Z_2 + ... + Z_k\}$ is a martingale. Note also that $-\alpha_i \leq Z_i \leq \beta_i$ for some $\alpha_i, \beta_i > 0$ such that $\alpha_i + \beta_i = 1$.

Using Lemma 8.1, we get:

$$\mathbb{P}(Z_1 + ... + Z_k \leq -k\delta) \leq e^{-\frac{2(k\delta)^2}{k}}.$$

Therefore the probability that at least one of μn points d for which $w_d \geq \frac{1}{2} + \delta$ will be misclassified by the set of k random decision trees is, by union bound, at most: $\mu n e^{-2k\delta^2} \leq n e^{-2k\delta^2}$. That, for $k = \frac{(1+C)\log(n)}{2\delta^2}$, completes the proof of the upper bound on the empirical error from theorems: 5.3 and 5.4 since we have already proved that $\mu \geq 1 - \frac{2\epsilon - 2\epsilon^2}{0.5-\delta}$. The proof of the majority voting version goes along exactly the same lines. This time, instead of Theorem 5.2, we use Theorem 5.1. We know that $\sum_{d \in \mathcal{T}} \sigma(d) \geq ne$, where $e = 1 - \epsilon$. Denote the fraction of points d with $\sigma(d) \geq \frac{1}{2} + x$ for $0 < x < \frac{1}{2}$ by μ^x. Then, by the argument similar to the one presented above,we have:

$$\mu^x \geq 1 - \frac{\epsilon}{0.5 - x}. \tag{3}$$

All other details of the proof for the majority voting are exactly the same as for the threshold averaging scheme. □

Next we prove parts of Theorem 5.5 regarding empirical error and Theorem 5.6.

Proof. Let $K > 0$ be a constant. We first consider the threshold averaging scheme. Take a decision tree T_i. Denote by S_{T_i} the set of points d from the training set with the following property: point d belongs in T_i do the leaf that contains at least $\frac{n}{K2^h}$ points. Note that since each T_i has exactly 2^h leaves, we can conclude that $|S_{T_i}| \geq n(1 - \frac{1}{K})$. In this proof and proof of theorems: 8.2 and 8.2 (presented in the next section) we will consider graph $\mathcal{G}^D$ that is obtained from $\mathcal{G}$ by deleting edges adjacent to those vertices of the color class $\mathcal{B}$ that correspond to leaves containing less than $\frac{n}{K2^h}$ points from the training set. Take point d from the training set with $w_d^t \geq \frac{1}{2} + \delta$, where w_d^t is the average weight of an edge incident to d in $\mathcal{G}^D$. Notice that $w_d \geq \frac{1}{2} + \delta + \frac{1}{K}$ implies: $w_d^t \geq \frac{1}{2} + \delta$. We say that a decision tree T_i is *d-good* if the leaf of T_i to which d belongs contains at least $\frac{n}{K2^h}$ points from the training set. Let us now define X_i^d. If i^{th} chosen random decision tree is d-good then X_i^d is defined as in the proof of Theorem 5.3. Otherwise we put $X_i^d = 0$. Denote $Z_i = X_i^d - w_d^t$. Note that the probability p_d that point d is misclassified by selected random decision trees is $p_d \leq \mathbb{P}(\frac{Z_1 + ... + Z_k}{k} + \frac{\sum_{j \in \mathcal{I}} R_j}{|\mathcal{I}|} \leq -\delta)$, where $\mathcal{I}$ is the set of indices corresponding to those chosen random decision trees that are d-good and random variables R_j are correction terms for d-good random decision trees that must be introduced in order to take into account added laplacians (if $\mathcal{I} = \emptyset$ then we assume that the value of the expression $\frac{\sum_{j \in \mathcal{I}} R_j}{|\mathcal{I}|}$ is 0). Note also that set $\{R_j, Z_j : j = 1, 2, ..., k\}$ is a set of independent random variables. We get:

$$p_d \leq \mathbb{P}(\frac{Z_1 + ... + Z_k}{k} \leq -\frac{\delta}{2}) + \mathbb{P}(\frac{\sum_{j \in \mathcal{I}} R_j}{|\mathcal{I}|} \leq -\frac{\delta}{2}).$$

Since from the Azuma's inequality we get: $\mathbb{P}(\frac{Z_1 + ... + Z_k}{k} \leq -\frac{\delta}{2}) \leq e^{-\frac{k\delta^2}{2}}$, we have:

$$p_d \leq e^{-\frac{k\delta^2}{2}} + \mathbb{P}(\frac{\sum_{j \in \mathcal{I}} R_j}{|\mathcal{I}|} \leq -\frac{\delta}{2}) \tag{4}$$

We will now estimate the expression $p_r = \mathbb{P}(\frac{\sum_{j \in \mathcal{I}} R_j}{|\mathcal{I}|} \leq -\frac{\delta}{2})$.

For $i \in \mathcal{I}$ denote by $\mathcal{A}_i$ an event that each of the two perturbation errors added to the leaf containing point d was of magnitude at most $\frac{\sqrt{n}}{K2^h}\delta_1$, where $\delta_1 = \frac{\delta}{24}$. Denote $\mathcal{A} = \bigcap_{i \in \mathcal{I}} \mathcal{A}_i$. Denote by $\mathcal{A}^c$ the complement of $\mathcal{A}$. We have: $\mathbb{P}(\frac{\sum_{j \in \mathcal{I}} R_j}{|\mathcal{I}|} \leq -\frac{\delta}{2}) = \mathbb{P}(\frac{\sum_{j \in \mathcal{I}} R_j}{|\mathcal{I}|} \leq -\frac{\delta}{2}|\mathcal{A})\mathbb{P}(\mathcal{A}) + \mathbb{P}(\frac{\sum_{j \in \mathcal{I}} R_j}{|\mathcal{I}|} \leq -\frac{\delta}{2}|\mathcal{A}^c)(1 - \mathbb{P}(\mathcal{A}))$. Thus we get:

$$p_r \leq \mathbb{P}(\frac{\sum_{j \in \mathcal{I}} R_j}{|\mathcal{I}|} \leq -\frac{\delta}{2}|\mathcal{A}) + (1 - \mathbb{P}(\mathcal{A})). \tag{5}$$

Now take one of the chosen random decision trees T_i with $i \in \mathcal{I}$. Take its leaf that contains given point d from the training set. Assume that this leaf contains r points from the training set with some fixed label $l \in \{-, +\}$ and that it altogether contains n_a points. Note that from the definition of $\mathcal{I}$ we have: $n_a \geq \frac{n}{K2^h}$. Let g_1, g_2 be two independent laplacian random variables, each of density function $\frac{\eta}{2k}e^{-\frac{|x|\eta}{k}}$. We would like to estimate the following random variable $\Theta = \frac{r+g_1}{n_a+g_1+g_2} - \frac{r}{n_a}$ for an event $\mathcal{A}$. Note that in particular we know that $|g_1|, |g_2| \leq \frac{\delta_1 n_a}{\sqrt{n}}$. Simple calculation gives us:

$$|\Theta| \leq \frac{\delta}{4\sqrt{n}}. \tag{6}$$

Now consider truncated probability space $\Omega|\mathcal{A}$ and truncated random variables $R_i^t = R_i|\mathcal{A}$ for $i \in \mathcal{I}$. We have: $\mathbb{P}(\sum_{i \in \mathcal{I}} R_i \leq -\frac{|\mathcal{I}|\delta}{2}|\mathcal{A}) = \mathbb{P}(\sum_{i \in \mathcal{I}} R_i^t \leq -\frac{|\mathcal{I}|\delta}{2})$. Using inequality 6, we get:

$$|R_i^t| \leq \frac{\delta}{4\sqrt{n}}, E|R_i^t| \leq \frac{\delta}{4\sqrt{n}}. \tag{7}$$

Thus we can use Azuma's inequality once more, this time to find the upper bound on the expression: $\mathbb{P}(\sum_{i \in \mathcal{I}} R_i^t \leq -\frac{|\mathcal{I}|\delta}{2})$ (we assume here that the random decision trees have been selected thus $\mathcal{I}$ is given). Without loss of generality we can assume that $\mathcal{I} \neq \emptyset$. We have: $\mathbb{P}(\sum_{i \in \mathcal{I}} R_i^t \leq -\frac{|\mathcal{I}|\delta}{2}) = \mathbb{P}(\sum_{i \in \mathcal{A}}(R_i^t - ER_i^t) \leq -\frac{\mathcal{I}\delta}{2} - \sum_{i \in \mathcal{I}} ER_i^t) \leq \mathbb{P}(\sum_{i \in \mathcal{I}}(R_i^t - ER_i^t) \leq -\frac{|\mathcal{I}|\delta}{4}) \leq e^{-\frac{2|\mathcal{I}|(\frac{\delta}{4})^2}{(\frac{\delta}{4\sqrt{n}} + \frac{\delta}{4\sqrt{n}})^2}}$. Therefore we get:

$$p_r \leq e^{-\frac{n}{2}} + (1 - \mathbb{P}(\mathcal{A})). \tag{8}$$

It remains to bound the expression: $(1 - \mathbb{P}(\mathcal{A}))$. Let g be a laplacian random variable with density function $\frac{\eta}{2k}e^{-\frac{|x|\eta}{k}}$. Note that from the union bound we get: $1 - \mathbb{P}(\mathcal{A}) \leq 2k\mathbb{P}(|g| > \frac{\sqrt{n}\delta}{24K2^h})$, where factor 2 in the expression $2k\mathbb{P}(g > \frac{\sqrt{n}\delta}{24K2^h})$ comes from the fact that for a given data point d we need to add perturbation error in two places in the leaf of the chosen random decision tree corresponding to d.
Denote $\gamma = \frac{\delta}{24K2^h}$. We have:

$$p_r \leq e^{-\frac{n}{2}} + 4k \int\limits_{\gamma\sqrt{n}}^{\infty} \frac{\eta}{2k}e^{-\frac{x\eta}{k}}dx. \tag{9}$$

Evaluation of the RHS-expression gives us:

$$p_r \leq e^{-\frac{n}{2}} + 2ke^{-\frac{\lambda\sqrt{n}\eta}{k}}, \quad \text{where} \quad \lambda = \frac{\delta}{24K2^h}. \tag{10}$$

Thus we can conclude that the probability p_d that the fixed point d from the training set will be misclassified by the set of k randomly chosen random decision trees satisfies:

$$p_d \leq e^{-\frac{k\delta^2}{2}} + e^{-\frac{n}{2}} + 2ke^{-\frac{\gamma\sqrt{n}\eta}{k}}. \tag{11}$$

Note that by the similar argument to the one presented in the proof of Theorem 5.3 and Theorem 5.4, we can conclude that at least $n(1-\frac{2(\epsilon+\frac{1}{K})-2(\epsilon+\frac{1}{K})^2}{0.5-\delta})$ points d from the training data satisfy: $w^t_d \geq \frac{1}{2}+\delta$. Let μ^t be a fraction of points with this property. As we observed earlier, if the points d satisfies: $w_d \geq \frac{1}{2}+\delta+\frac{1}{K}$ then it also satisfies: $w^t_d \geq \frac{1}{2}+\delta$. Thus $\mu \geq \mu^t$. We also have: $\mu^t \geq 1-\frac{2(\epsilon+\frac{1}{K})-2(\epsilon+\frac{1}{K})^2}{0.5-\delta}$. Thus $\mu \geq 1-\frac{2(\epsilon+\frac{1}{K})-2(\epsilon+\frac{1}{K})^2}{0.5-\delta}$. We replace ϵ by $\epsilon+\frac{1}{K}$ in the formula derived in the proof of Theorem 5.3 since now for any fixed decision tree we do not take into account points that belong to leaves with less that $\frac{n}{K2^h}$ points from the training set. For every given decision tree T_i there are at most $\frac{n}{K}$ points d from the training set such that T_i is not d-good. Note that, by union bound, the probability that at least one from the $n\mu$ points d with $w^t_d \geq \frac{1}{2}+\delta$ is misclassified is at most $n\mu p_d \leq np_d$. To see how Theorem 5.6 and the part of Theorem 5.5 regarding empirical error follow now, take $K=40$ and $\delta=\frac{1}{10}$. The proof of the majority voting version is very similar. We use inequality 3 (that was derived from Theorem 5.1) but all other details are exactly the same. Therefore we will not give it in details here since it would basically mean copying almost exactly the proof that we have just showed. □

8.3 Probabilistic averaging setting - empirical error

Let us switch now to the probabilistic averaging setting. In practice, as was shown in the experimental section, it is the least effective method. However for the completeness of our theoretical analysis and since for very large datasets theoretical guarantees regarding also this setting can be obtained, we focus on it now.

We will first focus on the part of Theorem 8.1 regarding empirical error.

Proof. We already know that: $\sum_{d\in\mathcal{T}} w_d \geq n(e^2+(1-e)^2)$, where e is the average quality. Assume that k random decision trees have been selected. Denote by Y_d the indicator of the event that a fixed data point d from the training set will be correctly classified. We have:

$$Y_d = \begin{cases} 1 & \text{with probability} \quad X^d \\ 0 & \text{with probability} \quad 1-X^d, \end{cases}$$

where X^d is random variable defined in the proof of theorems: 5.3 and 5.4. Note that after random decision trees have been selected, X^d has a deterministic value. Note also that random variables Y_d are independent and $EY_d = X^d$. Thus, we can use Lemma 8.1 in the very similar way as in the proof of theorems: 5.3 and 5.4 to get that for any given $c>0$:

$$\mathbb{P}(\sum_{d\in\mathcal{T}}(Y_d - X^d) \leq -nc) \leq e^{-2nc^2}. \qquad (12)$$

Let us focus now on the process of choosing random decision trees. Fix parameter $\delta>0$. Fix some point d from the training set. Using Lemma 8.1 in exactly the same way as in the proof of theorems: 5.3 and 5.4, we conclude that $\mathbb{P}(X^d < w_d-\delta) \leq e^{-2k\delta^2}$. Therefore, by the union bound, with probability at least $(1-ne^{-2k\delta^2})$ we have: $\sum_{d\in\mathcal{T}} X^d \geq \sum_{d\in\mathcal{T}}(w_d-\delta)$. Thus, according to the lower bound for $\sum_{d\in\mathcal{T}} w_d$ we presented at the beginning of the proof, we get that with probability at least $(1-ne^{-2k\delta^2})$ the following holds: $\sum_{d\in\mathcal{T}} X^d \geq n(1-2\epsilon+2\epsilon^2-\delta)$, where $\epsilon=1-e$. Note that random variables Y_d are independent from random variables X_d. We can conclude, using inequality 12, that with probability at least $(1-ne^{-2k\delta^2})(1-e^{-2nc^2})$ at least $n(1-2\epsilon+2\epsilon^2-\delta-c)$ points will be correctly classified. Now we can take $k=\frac{(1+C)\log(n)}{2\delta^2}$ and that completes the proof. Again, as in the previous proof, the majority voting scheme requires only minor changes in the presented proof so we will leave to the reader. □

Lets focus now on parts of theorems: 8.2 and 8.3 regarding empirical errors.

Proof. Proofs of statements regarding empirical errors go along exactly the same lines as presented proof of the part of Theorem 8.1 (regarding empirical error). The changes in the statement, due to the added perturbation error, follow from the proof of bounds on the empirical error from theorems: 5.5 and 5.6 . Therefore we will not give the entire proof but only mention few things.

In comparison with the statement of Theorem 8.1, in the expression on the upper bound on empirical error the term ϵ is replaced by $\epsilon + \frac{1}{K}$. This is, as explained in the proof of Theorem 5.5 (regarding empirical error), due to the fact that while dealing with weights of edges in graph $\mathcal{G}^D$ we do not take into account points from the training set corresponding to leaves with too few data points. To see how Theorem 8.3 can be derived, take $K = 40$, $\delta = \frac{1}{10}$, $c = \frac{1}{20}$. Again, as for Theorem 5.6, Theorem 8.3 follows now by simple calculations.

□

8.4 Generalization error

We will now prove upper bounds regarding generalization error for all the theorems presented in the previous paragraphs. We do it for all of them in the same section since all the proofs are very similar. Besides, right now, when we have already developed tools for obtaining upper bounds on the empirical error, we can use them to simplify our analysis regarding generalization error. Random decision trees give strong bounds on the generalization error since they do not lead to data overfitting. The internal structure of each constructed tree (i.e. the set of its inner nodes) does not depend at all on the data. This fact is crucial in obtaining strong guarantees on the generalization error. This is also the reason why we did not consider an approach where k random decision trees are selected and the one that gives the smallest empirical error is being returned. Such an approach may lead to overfitting (there are other problems with it too, as we mentioned before - not only generalization error analysis but also differential-privacy analysis is much more complicated and gives much weaker guarantees). As long as the number of the trees in the constructed random forest is not too large the overfitting will not take place. All the experiments presented in the main body of the paper measured generalization error of the random tree approach and stand for the empirical verification that this method is a good learning technique in the setting requiring high privacy guarantees. Below is the proof of the presented upper bounds on the generalization error.

Proof. Consider test set of n points. Whenever we refer to the weight or goodness of the test point d, this is in respect to the test set (see: definition of goodness and other terms in the description of the model). Let $\phi > 0$ be a small constant and denote by $\mathcal{E}_\phi$ an event that for the selected forest $\mathcal{F}$ of random decision trees the non-perturbed counts in all leafs (for each leaf we count points with label $+$ and $-$ in that leaf) for the test set and training set differ by at most $2\phi n$. We start by finding a lower bound on $\mathbb{P}(\mathcal{E}_\phi)$. Let us fix a forest, a particular tree of that forest and a particular leaf of that tree. Denote by X_i a random variable that takes value 1 if i^{th} point of the training set corresponds to that leaf and 0 otherwise. Similarly, denote by Y_i a random variable that takes value 1 if i^{th} point of the test set corresponds to that leaf and 0 otherwise. Denote by X_i^+ a random variable that takes value 1 if i^{th} point of the training set corresponds to that leaf and has label $+$ and is 0 otherwise. Similarly, denote by Y_i^+ a random variable that takes value 1 if i^{th} point of the test set corresponds to that leaf and has label $+$ and is 0 otherwise. Denote by p_1 the probability that i^{th} point of the training/test set corresponds to that leaf and by p_2 the probability that i^{th} point of the training/test set corresponds to that leaf and has label $+$. Notice that p_1, p_2 are the same for the training and test set since we assume that training and test set are taken from the same distribution. Since all the random variables introduced above are independent, we can conclude using Azuma's inequality that: $\mathbb{P}(X_1 + ... + X_n \in [n(p_1 - \phi), n(p_1 + \phi)]) \geq 1 - 2e^{-2n\phi^2}$. Similarly, $\mathbb{P}(Y_1 + ... + Y_n \in [n(p_1 - \phi), n(p_1 + \phi)]) \geq 1 - 2e^{-2n\phi^2}$. Therefore, by the union bound, $\mathbb{P}(|(X_1 + ... + X_n) - (Y_1 + ... + Y_n)| \leq 2\phi n) \geq 1 - 4e^{-2n\phi^2}$. By the same analysis we can show that $\mathbb{P}(X_1^+ + ... + X_n^+ \in [n(p_2 - \phi), n(p_2 + \phi)]) \geq 1 - 2e^{-2n\phi^2}$ and $\mathbb{P}(Y_1^+ + ... + Y_n^+ \in [n(p_2 - \phi), n(p_2 + \phi)]) \geq 1 - 2e^{-2n\phi^2}$. Thus we also have: $\mathbb{P}(|(X_1^+ + ... + X_n^+) - (Y_1^+ + ... + Y_n^+)| \leq 2\phi n) \geq 1 - 4e^{-2n\phi^2}$. We can conclude that the probability of the following event:

$$|(X_1 + ... + X_n) - (Y_1 + ... + Y_n)| \leq 2\phi n \quad \text{and} \quad |(X_1^+ + ... + X_n^+) - (Y_1^+ + ... + Y_n^+)| \leq 2\phi n$$

is at least $1 - 8e^{-2n\phi^2}$. If we now take the union bound over all $2^h k$ leafs of the forest then we obtain: $\mathbb{P}(\mathcal{E}_\phi) \geq 1 - 2^{h+3} k e^{-2n\phi^2}$. We will now consider average weights w_d of the test points. The analysis for the majority voting uses $\sigma(d)$ and is completely analogous. Assume now that all the counts for all the leaves for the test and training set differ by at most 2ϕ. As in the analysis of the empirical error in the differentially-private setting, lets focus on those leaves of the forest that contain at least $\frac{n}{2^h K}$ of the test points each, for a constant $K > 0$. Take a leaf l with this property. Denote by x^1 the number of test points corresponding to that leaf and with label $+$. Denote by x^2 the number of training points corresponding to that leaf and with label $+$. Denote by y^1 the number of all test points corresponding to that leaf and by y^2 the number of all training points corresponding to that leaf. We want to find an upper bound on the expression $q = |\frac{x^1}{y^1} - \frac{x^2}{y^2}|$. Simple algebra gives us: $q \leq \frac{2\phi n(x^1+y^1)}{y^1(y^1-2\phi n)}$. If we now take $\zeta = \frac{2\phi}{\theta}$, where $\theta = \frac{1}{2^h K}$ then we get: $q \leq \frac{2\zeta}{1-\zeta}$. Let us take ζ such that: $\frac{2\zeta}{1-\zeta} \leq \frac{\delta}{2}$, where $\delta > 0$ is a positive constant. Thus we want: $\zeta \leq \frac{\delta}{4+\delta}$, i.e. $\phi \leq \frac{\delta\theta}{2(4+\delta)}$. Take $\phi = \frac{\delta\theta}{2(4+\delta)}$. We can conclude that with probability at least $\mathbb{P}(\mathcal{E}_\phi)$ the difference between ratios of counts in leaves containing at least $\frac{n}{\theta}$ test points for the test and training set is at most $\frac{\delta}{2}$. This in particular implies that if we consider test point d and a truncated bipartite graph G^d (but this time with respect to the test set, not training set) then weights of d in G^d and its corresponding version for the training set differ by at most $\frac{\delta}{2}$.

We are almost done. Consider first majority voting/threshold averaging scheme. The only changes we need to introduce in the statement of Theorem 5.3 for the empirical error is to subtract from p_1 the probability that $\mathcal{E}_\phi$ does not hold to obtain a lower bound on p_2, add factor $\frac{1}{K}$ to the expression on w (since we are using the truncated model) and change δ by $\frac{\delta}{2}$ in the expression on number of random decision trees used. Similarly, in the statement of Theorem 5.4 we need to replace ϵ in the expression on err_1 by $\epsilon + \frac{1}{K}$ to obtain an upper bound on err_2 (again, because we are using truncation argument) and make the same change in the number of decision trees as the one above. To obtain a lower bound on p_2 it suffices to subtract the probability that $\mathcal{E}_\phi$ does not hold. Let us focus now on Theorem 5.5. Again we need to add extra factor $\frac{1}{K}$ to the expression on w and subtract probability that $\mathcal{E}_\phi$ does not hold to obtain a lower bound on p_2.

Now lets consider probabilistic averaging scheme. Take the statement of Theorem 8.1 first. We make similar correction to those mentioned earlier to get a lower bound on p_2. Besides in the upper bound on err_1 we need to replace ϵ by $\epsilon + \frac{1}{K}$ to obtain an upper bound on err_2. In Theorem 8.2 we need to add one extra term $\frac{1}{K}$ in the upper bound on err_1 to obtain an upper bound on err_2 and again modify p_1 in the same way as before to obtain a lower bound on p_2. □

9 Experiments on all datasets

In this section we enclose the experimental results we obtained for all benchmark datasets. The plots have similar form to the ones shown on *Mushrooms* dataset in the main body of the paper. The presented results further support the claims of the paper.

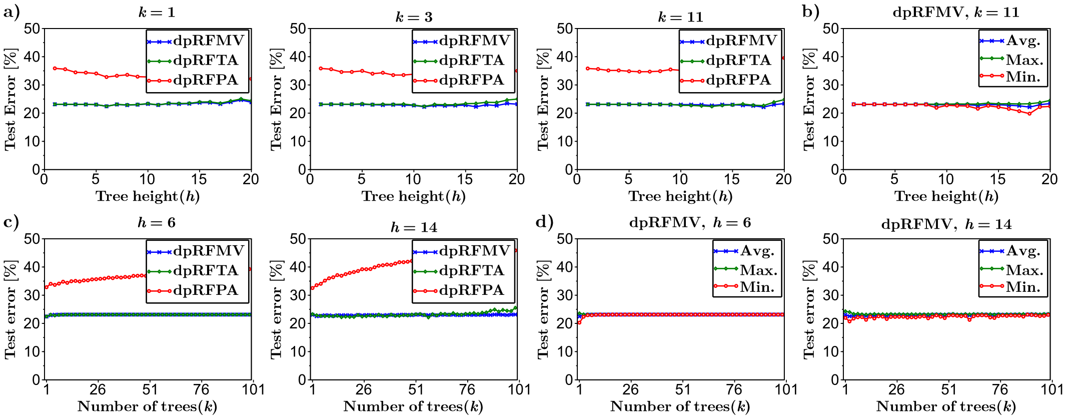

Figure 3: *adult* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. $\eta = 1000/n_{tr}$. Test error resp. vs. **a)** h across various settings of k and vs. **c)** k across various settings of h; Minimal, average and maximal test error resp. vs. h (**b)**) and vs. k (**d)**) for dpRFMV.

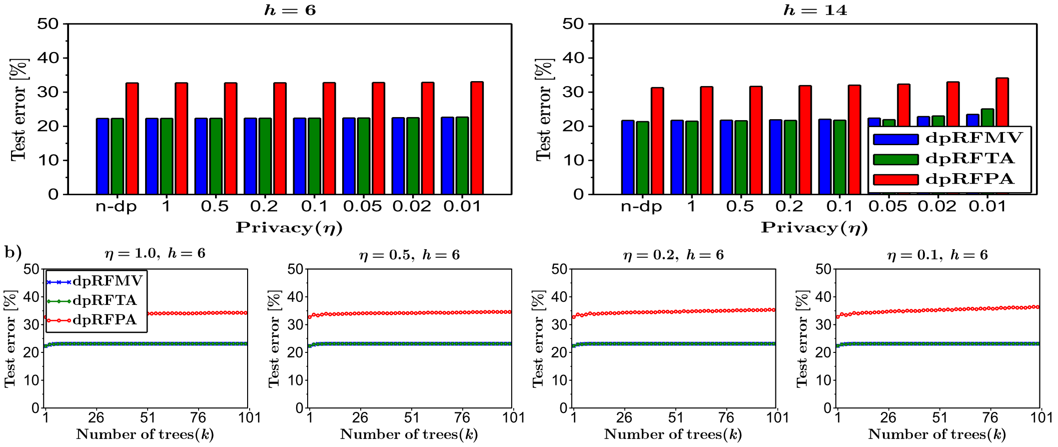

Figure 4: *adult* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. **a)** Test error vs. η for two settings of h. **b)** Test error vs. k for fixed h and across different settings of η.

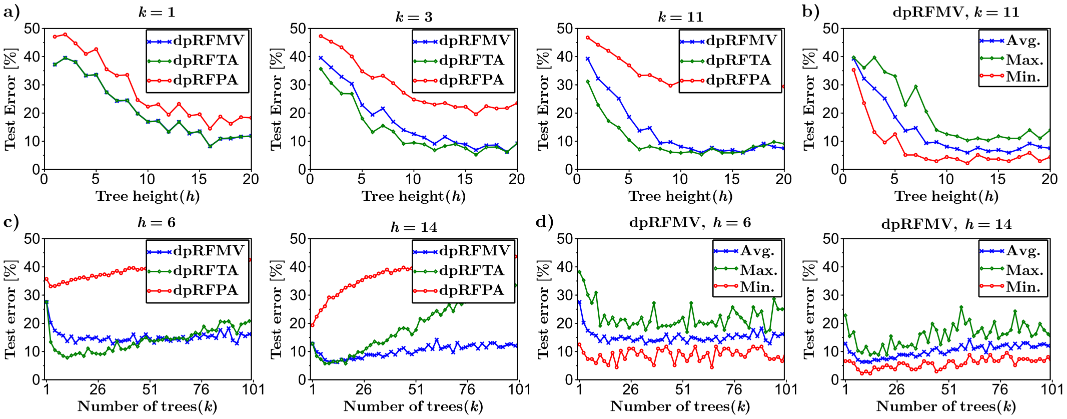

Figure 5: *ban_aut* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. $\eta = 1000/n_{tr}$. Test error resp. vs. **a)** h across various settings of k and vs. **c)** k across various settings of h; Minimal, average and maximal test error resp. vs. h (**b)**) and vs. k (**d)**) for dpRFMV.

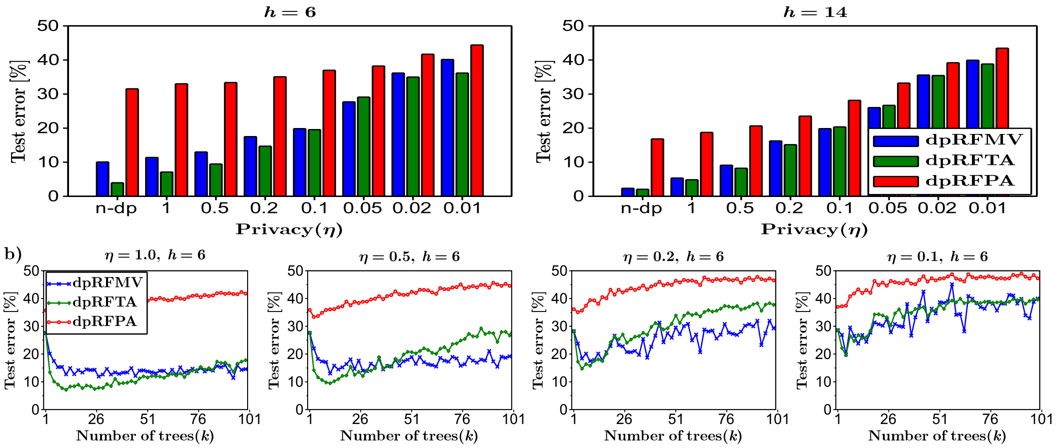

Figure 6: *ban_aut* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. **a)** Test error vs. η for two settings of h. **b)** Test error vs. k for fixed h and across different settings of η.

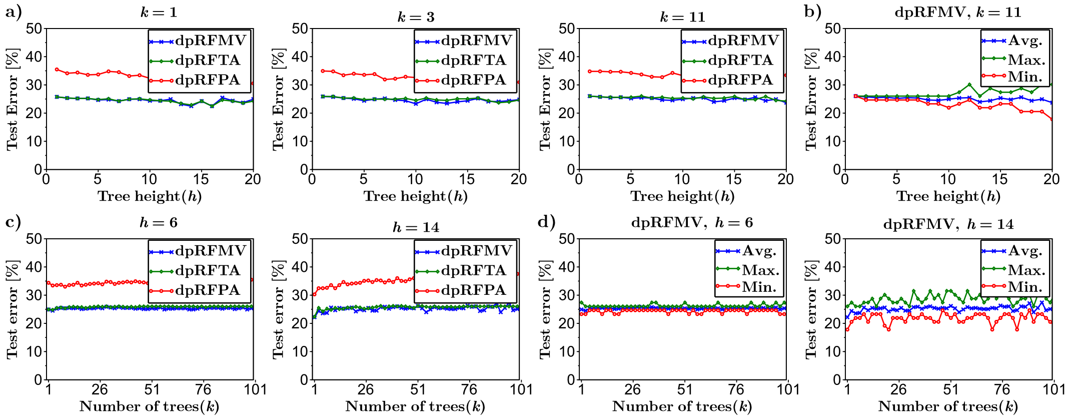

Figure 7: *BTSC* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. $\eta = 1000/n_{tr}$. Test error resp. vs. **a)** h across various settings of k and vs. **c)** k across various settings of h; Minimal, average and maximal test error resp. vs. h (**b)**) and vs. k (**d)**) for dpRFMV.

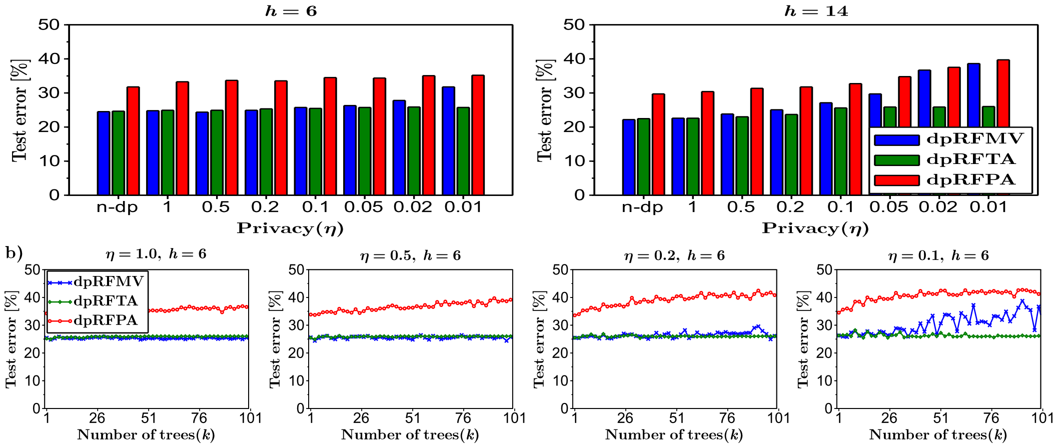

Figure 8: *BTSC* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. **a)** Test error vs. η for two settings of h. **b)** Test error vs. k for fixed h and across different settings of η.

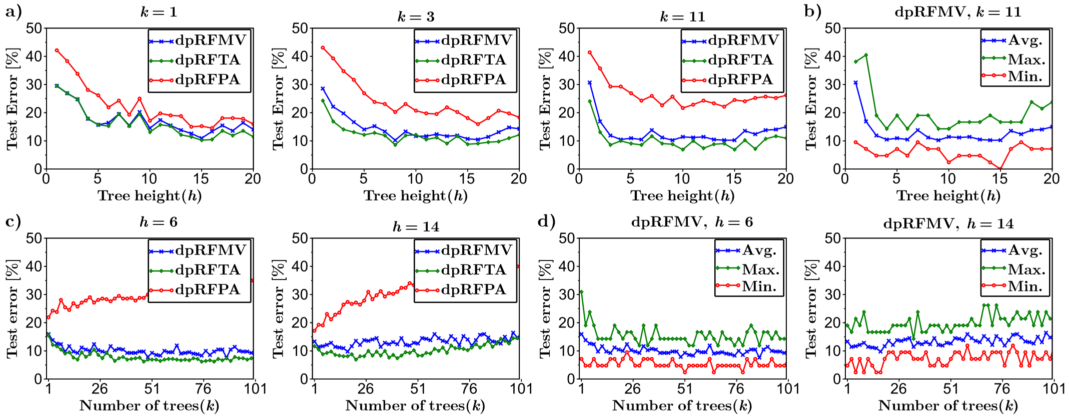

Figure 9: *CVR* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. $\eta = 1000/n_{tr}$. Test error resp. vs. **a)** h across various settings of k and vs. **c)** k across various settings of h; Minimal, average and maximal test error resp. vs. h (**b)**) and vs. k (**d)**) for dpRFMV.

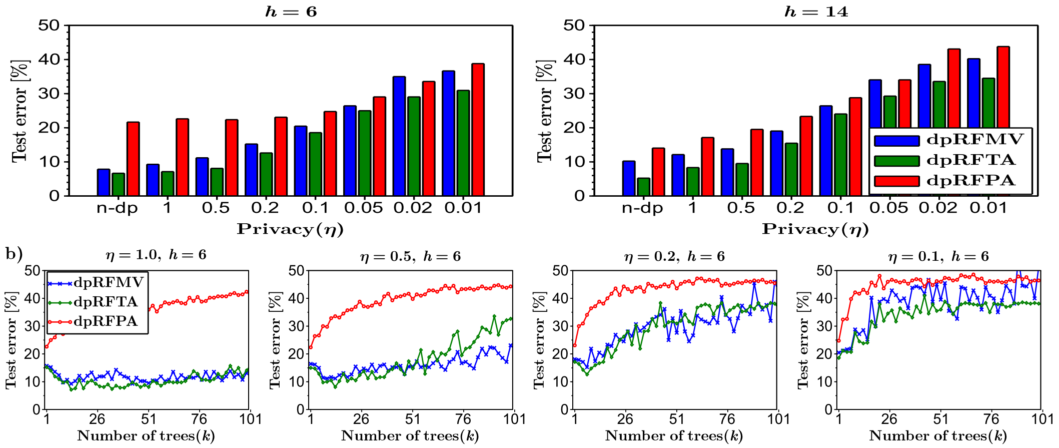

Figure 10: *CVR* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. **a)** Test error vs. η for two settings of h. **b)** Test error vs. k for fixed h and across different settings of η.

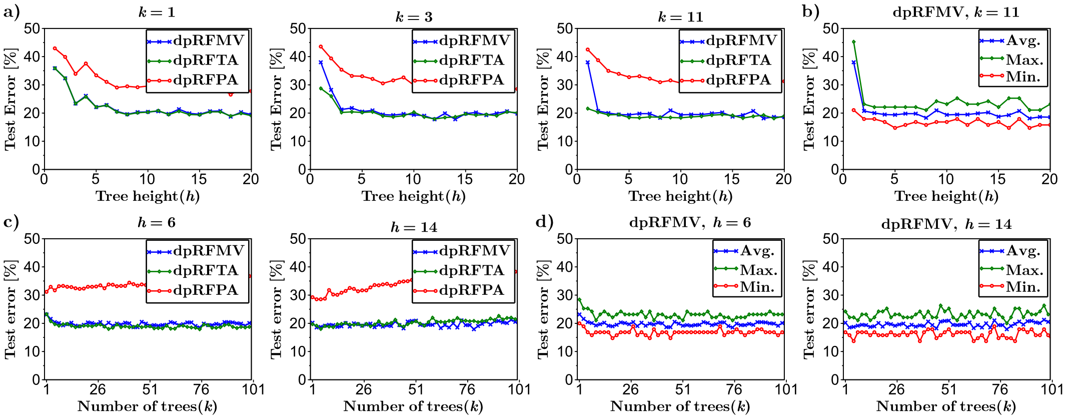

Figure 11: *Mam_M* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. $\eta = 1000/n_{tr}$. Test error resp. vs. **a)** h across various settings of k and vs. **c)** k across various settings of h; Minimal, average and maximal test error resp. vs. h (**b)**) and vs. k (**d)**) for dpRFMV.

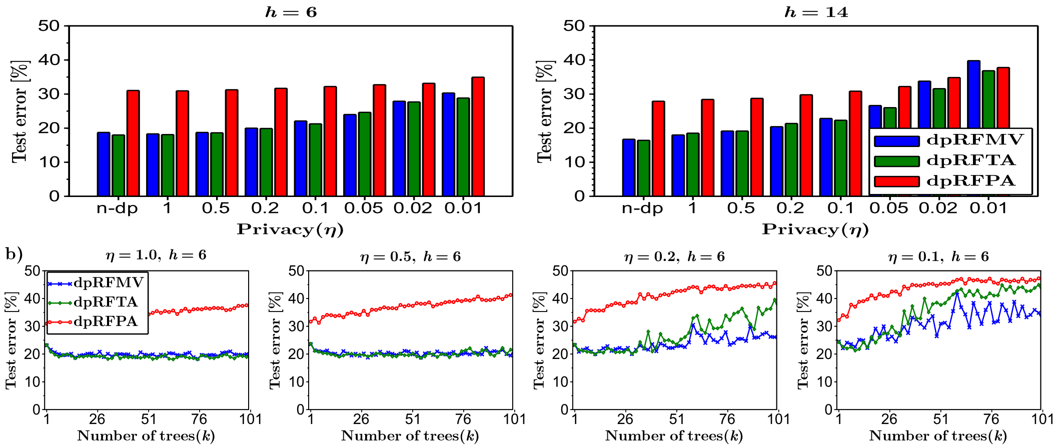

Figure 12: *Mam_M* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. **a)** Test error vs. η for two settings of h. **b)** Test error vs. k for fixed h and across different settings of η.

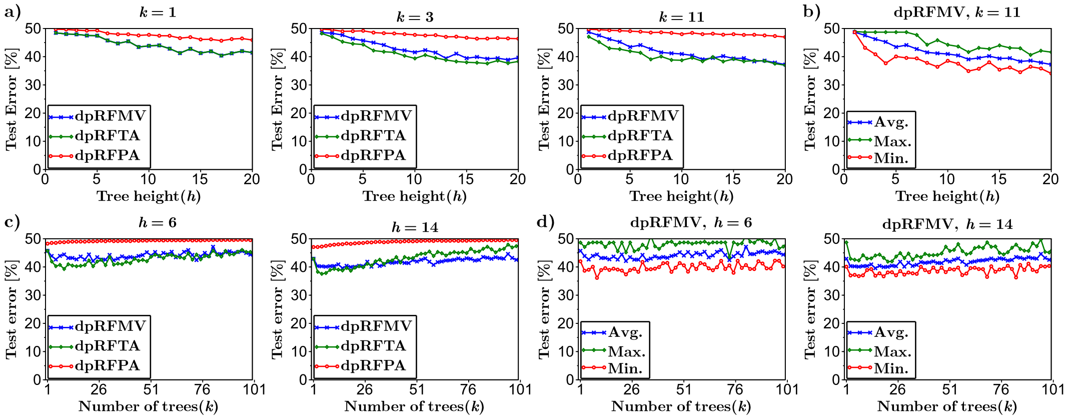

Figure 13: *Covertype* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. $\eta = 1000/n_{tr}$. Test error resp. vs. **a)** h across various settings of k and vs. **c)** k across various settings of h; Minimal, average and maximal test error resp. vs. h (**b**)) and vs. k (**d**)) for dpRFMV.

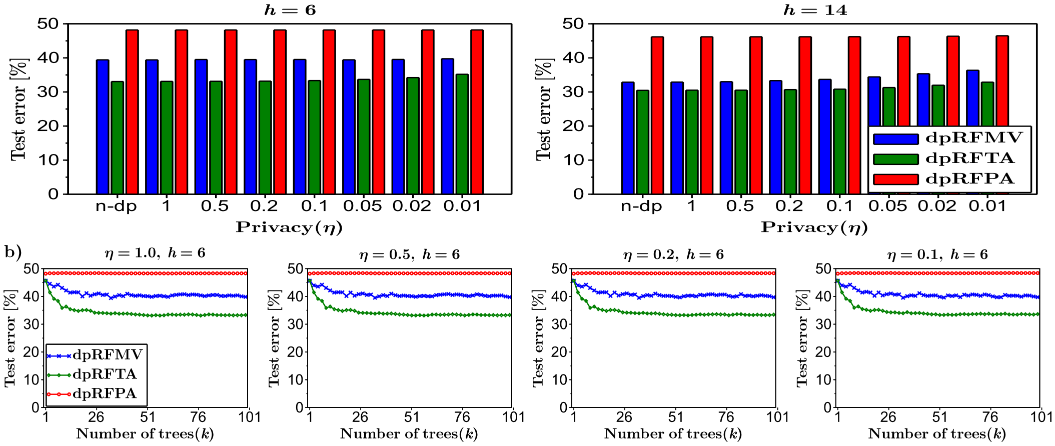

Figure 14: *Covertype* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. **a)** Test error vs. η for two settings of h. **b)** Test error vs. k for fixed h and across different settings of η.

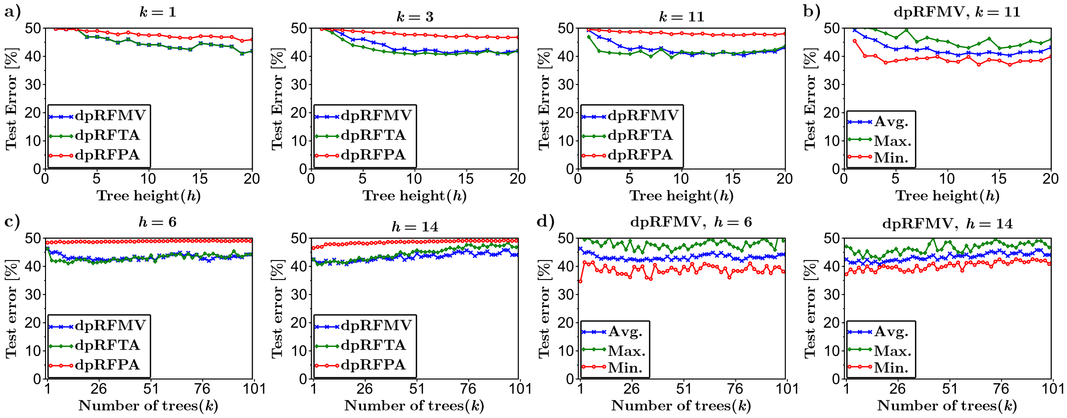

Figure 15: *Quantum* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. $\eta = 1000/n_{tr}$. Test error resp. vs. **a)** h across various settings of k and vs. **c)** k across various settings of h; Minimal, average and maximal test error resp. vs. h (**b**)) and vs. k (**d**)) for dpRFMV.

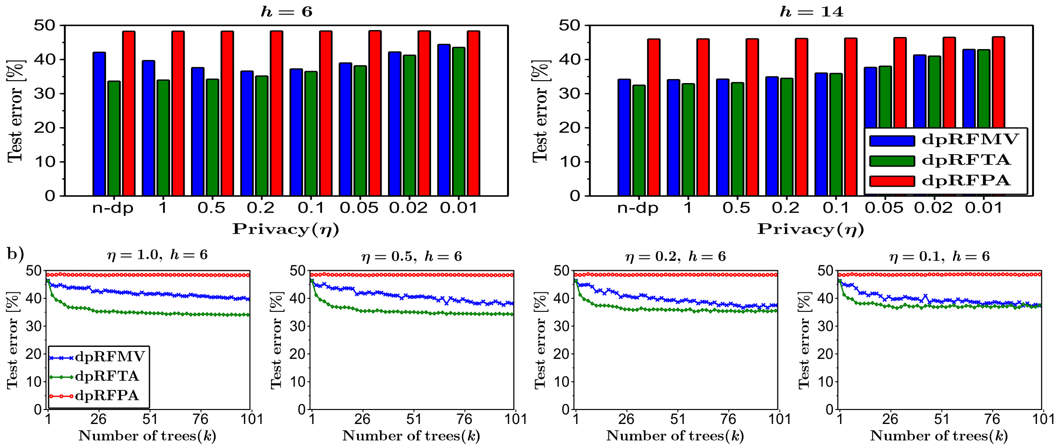

Figure 16: *Quantum* dataset. Comparison of dpRFMV, dpRFTA and dpRFPA. **a)** Test error vs. η for two settings of h. **b)** Test error vs. k for fixed h and across different settings of η.