Noise-efficient Learning of Differentially Private Partitioning Machine Ensembles

Zhanliang Huang[1]✉, Yunwen Lei[1], and Ata Kabán[1]

[1]School of Computer Science, University of Birmingham, United Kingdom
ZXH898@cs.bham.ac.uk Y.Lei@bham.ac.uk A.Kaban@bham.ac.uk

Abstract. Differentially private decision tree algorithms have been popular since the introduction of differential privacy. While many private tree-based algorithms have been proposed for supervised learning tasks, such as classification, very few extend naturally to the semi-supervised setting. In this paper, we present a framework that takes advantage of unlabelled data to reduce the noise requirement in differentially private decision forests and improves their predictive performance. The main ingredients in our approach consist of a median splitting criterion that creates balanced leaves, a geometric privacy budget allocation technique, and a random sampling technique to compute the private splitting-point accurately. While similar ideas existed in isolation, their combination is new, and has several advantages: (1) The semi-supervised mode of operation comes for free. (2) Our framework is applicable in two different privacy settings: when label-privacy is required, and when privacy of the features is also required. (3) Empirical evidence on 18 UCI data sets and 3 synthetic data sets demonstrate that our algorithm achieves high utility performance compared to the current state of the art in both supervised and semi-supervised classification problems.

Keywords: Differential privacy · Noise reduction · Ensembles.

1 Introduction

Differential privacy (DP) [9] is a notion of privacy that provides a rigorous information-theoretic privacy guarantee. DP algorithms allow their outputs to be shared across multiple parties and used for analysis by introducing randomization in critical steps of the learning algorithm. However, DP algorithms often require a larger training set to achieve good performance due to the added noise. Moreover, large labelled training sets are expensive and sometimes impractical to obtain, especially for a sensitive data set (e.g. HIV positive data). On the other hand, unlabelled data can be less privacy-sensitive and in many cases much more cost-effective to obtain at large scale in comparison to accurately labelled data. Due to these reasons, it is extremely valuable if we can make use of unlabelled data to improve the performance of a learning algorithm in the private setting. In this paper, we present a framework that builds machine ensembles in both supervised and semi-supervised settings. The framework takes advantage of unlabelled data to reduce the noise requirements and hence it only requires a small number of labelled samples to achieve high performance. Our framework invokes constructing a tree structure that uses a density-informed splitting criterion to create balanced leaves and naturally extends to semi-supervised learning with different privacy settings. Current private tree-based algorithms in the literature either use a greedy-decision approach, or a random-tree approach. Methods with greedy approaches take the route of classical decision tree construction [5], and compute the optimal splits at each node privately. Popular splitting techniques such as the Gini index [27, 11] or the information gain criterion [13] are applied in conjunction with a privatized algorithm. The drawback for this approach is that it cannot be naturally extended to semi-supervised learning as they greedily estimate the optimal split using the labels. On the other hand, random tree approaches construct the tree by randomized splits at each node [17, 12, 14]. Randomization is beneficial from the privacy perspective as it is data-independent and leaks zero information about individuals in the data set. However, a fully randomized split creates high variance and requires a large ensemble of trees to perform well. We cannot afford a large ensemble due to the privacy constraint. Furthermore, random-tree approaches do not take advantage of the unlabelled data as the splits are chosen fully randomly. Since these approaches do not naturally extend to the semi-supervised setting, we need to assign labels to the unlabelled set using a trained model if we want to make any use of the unlabelled data [16, 20]. While this method can help to improve accuracy in some cases, it requires the predicted labels to be accurate for the output data to be useful, which cannot be guaranteed in general. Furthermore, since the output data contains the original features of the unlabelled set, it can only be applied where we do not need the privacy of the features at all.

Instead of the previous approaches, our approach proposes a semi-greedy median splitting criterion that uses the features to make formative splits. Median splitting has been used to build trees mostly in spatial decomposition where we partition data sets to allow quick access to different parts of the data [2, 7]. However it also can be used for classification and regression problems with good utility as shown in [4, 18] – even though classical decision tree methods are better in general without privacy. A main intuition of median split is that it creates density-balanced nodes, the concept matches with the density-based dissimilarities in the work of Aryal *et al* [1]. That is, two points are more similar if they lie in a sparse region than two other points in a dense region with equal geometric distance. Each leaf comes with a similar amount of sample points. Hence we avoid empty leaves and the noise level of each leaf is balanced. To achieve high utility, we also employ several techniques to optimize each step of the privatized model as follows. 1) We use a geometric-scaling privacy budget allocation strategy to ensure accurate splits at each level. 2) We use a random sampling technique to compute the private median effectively. 3) We use disjoint subsets to create ensembles for both labelled and unlabelled sets to reduce noise effects. While these techniques have pre-existed in isolation as parts of other algorithms, the combination appears novel and it leads to a novel framework that achieves high performance in both the supervised and semi-supervised setting.

The remainder of the paper is organized as follows. We discuss the related works in Section 2, and introduce some background in Section 3. In Section 4 we present key steps of our strategy and the construction of a supervised algorithm. Section 5 demonstrates our framework of creating ensembles in semi-supervised learning. Finally, we present our experimental analysis of our method for both supervised and unsupervised learning in Section 6.

2 Related Work

There is a vast amount of research in machine learning on differential privacy since its introduction ([10, 15]). Tree-based methods are certainly one of the most popular research topics, with early works on private random trees [17] and greedy trees [13]. In [11] the authors proposed the use of local sensitivity [24] to reduce the randomness in greedy trees. The idea has been extended in [12] to improve private random trees using smooth sensitivity, which utilize an upper bound on the local sensitivity. More recently, [27] proposed a greedy approach that takes advantage of a notion of smooth sensitivity with the exponential mechanism for both Gini index and label output. Another DP forest algorithm by [26] considers (ϵ, δ)-differential privacy which is a weaker notion of the pure ϵ-differential privacy that we are concerned with here. As discussed in [12], it is possible to obtain high utility while guaranteeing pure differential privacy. Other differentially private algorithms such as [23] consider private data release – a different problem setting from what we study here.
The construction of a tree structure using a median split has long lasted in spatial trees [2] and private spatial decomposition [7]. For classification problems it was initially used by [4], it then gradually gained attention and was analyzed theoretically by [3]. Similar idea is also used in spatial decomposition such as kd-trees [7]. More recently, [18] extended the idea to a median splitting random forest in the non-private setting. The authors demonstrated a random forest using a median-based splitting along with its theoretical analysis. Furthermore, a recent work by [6] proposed a private random forest using median splits however their method uses a greedy approach to compute each split-attribute and does not extend to semi-supervised learning. For private semi-supervised classifiers, the random forest by [17] can be extended to take advantage of unlabelled data [16]. Furthermore, work by [20] took advantage of unlabelled samples by providing predicted labels using a kNN classifier and then training a linear predictor using the predicted set. However both methods apply only when the privacy of the features is not a concern. Other semi-supervised methods [25] make extra assumptions on the data set and differ from our setting here.

3 Preliminaries: Differential Privacy

In this section, we use $\mathcal{X}$ to denote a universal set that contains all possible data points. We denote by S a set of observations from $\mathcal{X}$. The privacy budget will be denoted by $\epsilon > 0$. Furthermore, we define the distance between two sample sets S,S' to be the Hamming distance denoted as $\|S - S\|_H$, which equals the number of points to be added and removed from S until $S = S'$.

Definition 1 (Differential privacy [9]). *A randomized algorithm $\mathcal{A}$ is said to satisfy ϵ-differential privacy if for any $S, S' \in \mathcal{X}^n$ with $\|S - S'\|_H \leq 1$, we have*

$$\sup_{B \in \mathcal{B}} \frac{\mathbb{P}[\mathcal{A}(S) \in B]}{\mathbb{P}[\mathcal{A}(S') \in B]} \leq \exp(\epsilon), \tag{1}$$

where $\mathcal{B}$ is the collection of all measurable sets in Range$(\mathcal{A})$.

An immediate consequence of the definition is that DP algorithms are immune to *post-processing*. That is, if $\mathcal{A}$ is ϵ-DP, then the composition $f \circ \mathcal{A}$ is also ϵ-DP for an arbitrary mapping f [10]. We note that some other literature on differential privacy considers the case where S and S' (of the same size) differ by at most one sample point. In contrast, we consider adding/removing a point, which is the setting considered in most of the related works. The two settings only differ by a constant factor on the sensitivity analysis.

Definition 2 (Global ℓ_1-sensitivity [9]). *The global ℓ_1-sensitivity* GS(f) *of a function $f : \mathcal{X}^n \to \mathbb{R}^m$ is defined as*

$$\mathrm{GS}(f) = \max_{S,S' \subset \mathcal{X} : \|S - S\|_H = 1} \|f(S) - f(S')\|_1. \tag{2}$$

The global sensitivity captures the maximum difference in the output when swapping a data set with a neighbouring one that differs by at most one point. There are other notions of sensitivity, such as the local sensitivity that considers the particular data set S. Local sensitivity can be significantly smaller than global sensitivity in many cases [11], however local sensitivity in itself does not guarantee differential privacy. Next, we present two well-known mechanisms for private algorithm design – the Laplace and the Exponential mechanisms.

Definition 3 (Laplace mechanism of [10]). *Given any function $f : \mathcal{X}^n \to \mathbb{R}^m$, the Laplace mechanism is defined as*

$$\mathcal{M}(S, f, \epsilon) = f(S) + (Y_1, \ldots, Y_m), \tag{3}$$

where Y_i are i.i.d. random variables drawn from Lap$(\mathrm{GS}(f)/\epsilon)$.

Definition 4 (Exponential mechanism of [21]). *Let $\mathcal{R}$ be an arbitrary set of output candidates. Given a utility function $u : \mathcal{X}^n \times \mathcal{R} \to \mathbb{R}$ that computes the quality of a candidate $r \in \mathcal{R}$, the exponential mechanism $\mathcal{M}(S, u, \mathcal{R})$ selects and outputs one of these with probability proportional to the following*

$$\mathbb{P}[\mathcal{M}(S, u, \mathcal{R}) = r] \propto \exp\left(\frac{\epsilon u(S, r)}{2\mathrm{GS}(u)}\right). \tag{4}$$

It is well-known that the Laplace and the Exponential mechanisms both satisfy ϵ-DP. The reader can refer to [10] for detailed proofs of privacy guarantees. The following composition theorems allow us to combine multiple private mechanisms.

Theorem 1 (Sequential composition [21]). *Let $\{f_i\}_{i=1}^N$ be a sequence of queries on a data set S each satisfying $\{\epsilon_i\}_{i=1}^N$ differential privacy. Then the output sequence $\{f_i(S)\}_{i=1}^N$ of all queries satisfies $\sum_{i=1}^N \epsilon_i$ differential privacy.*

Theorem 2 (Parallel composition [22]). *Let $\{S_i\}_{i=1}^N$ be disjoint subsets of S, and f a query applied on each of the subsets S_i while satisfying ϵ-DP. Then the output sequence $\{f(S_i)\}_{i=1}^N$ satisfies ϵ-DP.*

Sequential composition states that the more queries we send to the original data set, the less privacy guarantee we have. Parallel composition says that we do not lose the independent privacy guarantees if we query disjoint subsets independently.

4 A Density-based Decision Tree

In this section, we describe the procedures and the key steps of our tree construction. Overall, the algorithm is broken down into the following steps, where the details are outlined in Alg. 1.

1. Decide the parameters (number of trees and maximum depth).
2. At each tree node, uniformly randomly select an attribute to split on.
3. Calculate a private median for the selected attribute using the exponential mechanism.
4. When reaching a leaf node, use the Laplace mechanism to store the privatized counts for each class.
5. For a test point, collect together all the label-counts from all trees and output the label that has the majority count.

Note that, at each split we randomly select an attribute from the whole set of attributes, which avoids the label-dependent greedy computation of an optimal attribute as done in [6, 27], and will allow us to construct the tree using only an unlabelled sample. Furthermore, random selection of the splitting feature can improve the diversity of trees while protecting privacy. After selection of a splitting attribute, we compute privately the median of the values for the selected attribute. This splitting method allows the feature space to be partitioned into even density regions. A key property of median splits is that it only depends on the features of the data, not the labels. Hence it does not overfit the data set easily, which is a concern with classic decision trees. Due to this property, we do not require any pruning process that label-dependent tree construction requires to avoid over-fitting. Other similar techniques involve the centred random forest described in [18] and mean-based rather than median-based splitting. However in the private setting, private mean estimation is usually more expensive than median estimation, as a single out-liar can largely affect the mean value – hence more noise is required to guarantee privacy. We terminate the splitting process of a branch either when we have reached our maximum depth or we only have a few points left (by default we set 10 as the minimum). The setting of a default minimum value prevents further splitting a node that has very few points.

Algorithm 1 BuildTree

```
Inputs: Sample set $S$, maximum depth $k$, privacy budget $\epsilon$.
procedure BuildTree($S, k, \epsilon$)
    if k $\leq 0$ or $|S| \leq 10$ then
        Return a Leaf node
    end if
    LB, RB = $PrivateSplit(S, \epsilon)$
    BuildTree(LB,k-1, $\epsilon$), BuildTree(RB,k-1, $\epsilon$) # build subtrees
    Return a decision node that holds the split criteria and left/right branch.
end procedure
procedure PrivateMedian($V, \epsilon$)
    $a = \min V, b = \max V$
    $\mathcal{R}$ = { set of random i.i.d. draws from $Uniform(a, b)$};
    for each $r$ in $\mathcal{R}$ do
        computes the quality score $u(V, r)$
    end for
    Return $\tilde{r} \in \mathcal{R}$ with exponential mechanism with budget $\epsilon$.
end procedure
procedure PrivateSplit($S, \epsilon$)
    Choose a splitting dimension $i$ uniformly from data dimension $[d]$
    $V$ = sorted{set of values in dimension $i$}
    Choose private median $p$ by $PrivateMedian(V, \epsilon)$
    for each sample $X$ in $S$ do
        if $X_i \leq p$ then
            add to left branch
        else
            add to right branch
        end if
    end for
    Return LeftBranch, RightBranch
end procedure
```

One of the key steps in our private tree construction is being able to compute the median accurately while preserving privacy. This is crucial for the final performance of the tree. A demonstration of the effect of the median estimation on accuracy is shown in Fig. 4. There are different methods of private median computation as discussed in [7], the most common approach is using the exponential mechanism. However, a direct application of the exponential mechanism considers all the feature values occurring in the data as candidates, some of which will be far away from the median. Each of these have low utility, but they accumulate a large fraction of the selection probability. We instead sample the candidates $\mathcal{R}$ uniformly from the interval defined by the minimum and maximum of the feature values occurring in the data – this significantly reduces the computations, and in our experience it also results in a more accurate estimate.

We employ the exponential mechanism with a rank-based utility function as follows. Let ϵ_s denote the desired privacy parameter for tree construction. Denote a subset of the data as $S_i \subset S$ and let V denote the set of all values of the j-th

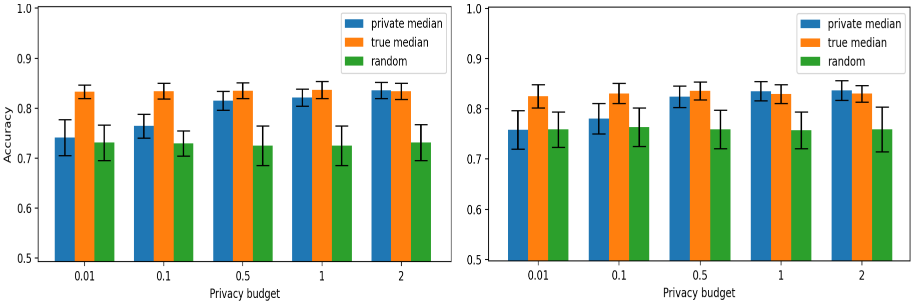

Fig. 1. Effect of median estimation on accuracy when we vary the privacy budget. Synthetic data sets of dimension 5 (left) and 10 (right) are used to compare the accuracy of Alg. 1 with its variation that uses the true median, and random splitting. Observe, the accuracy increases as we make better estimates of the true median due to larger privacy budget.

attribute for points in S_i. Then the utility of a candidate $r \in \mathcal{R}$ for attribute j is defined as $u(V, r) = -|rank_j(r) - |V|/2|$, where $rank_j(r)$ denotes the number of points in V that are no larger than r.

This utility function assigns a negative quality score to all values except for the median of the sample, which will have quality score zero. Values $r \in \mathcal{R}$ will have decreasing utilities the further away they are from the median. For categorical variables we use the same utility function, except that we let $\mathcal{R}$ to range over all categories for attribute j, and we define $rank_j(r)$ to be the number of points in V that are equal to r. Note that the sensitivity of u is $1/2$. Indeed, adding a data point into V increases $|V|/2$ by $1/2$ and $rank_j(r)$ either increases by 1 or remains the same; removing a data point has a similar effect. Now by the exponential mechanism, for any given ϵ we can guarantee ϵ-DP by outputting $r \in \mathcal{R}$ with probability

$$\mathbb{P}[\mathcal{M}(V, u, \mathcal{R}) = r] \propto \exp\left(\epsilon u(V, r)\right), \tag{5}$$

where the actual probability will be obtained through dividing the sum of proportional probabilities over all $r \in \mathcal{R}$.

Note that we have not queried the sample labels in our construction of the tree. By partitioning the training set into N disjoint subsets, and distributing the labels to the leaves of the trees privately (line 10 in Alg. 2), we obtain a private supervised ensemble model for classification and regression tasks. The full algorithm is given in Alg. 2.

4.1 Privacy analysis

There are two steps in Alg. 1 and Alg. 2 where we have used a privacy mechanism: (i) *PrivateSplit*, and (ii) *DistributeLabels*. In this section, we analyze the privacy guarantee of each mechanism. For the *PrivateSplit* procedure we note

Algorithm 2 Supervised Private Ensemble

```
Inputs: labelled set $S$, size of ensemble $N$, maximum depth $k$, privacy budget $\epsilon$
procedure SupervisedEnsemble($S, N, k, \epsilon$)
    Split the total budget $\epsilon$ into $\epsilon_s, \epsilon_l$. # even split by default
    Randomly partition $S$ into $N$ disjoint subsets $\{S_i\}_{i=1}^{N}$.
    for $i$ in $1, \dots, N$ do
        Tree $i = BuildTree(S_i, k, \epsilon_s)$
        $DistributeLabels$(Tree $i$, $S_i$, $\epsilon_l$) and add Tree $i$ to ensemble
    end for
end procedure
procedure DistributeLabels(Tree, $S, \epsilon$)
    for each sample $X$ in $S$ do
        find the corresponding leaf of $X$ and record the label of $X$
    end for
    for each leaf in the Tree do
        for each label class do
            add $noise \sim Lap(1/\epsilon)$ to the label count.
        end for
    end for
end procedure
```

that the only computation required to query the data set is private median estimation. The splitting process afterwards only partitions the data set by the split condition, which guarantees the same privacy by the post-processing property of DP [10]. To guarantee ϵ_s-DP over the whole sequence of splits along the tree construction, we need to split ϵ_s into a sequence of privacy budgets

$$\epsilon_0 + \epsilon_1 + \cdots + \epsilon_{k-1} = \sum_{i=0}^{k-1} \epsilon_i = \epsilon_s, \tag{6}$$

where ϵ_i is the privacy budget for splits at depth i, and k is the maximum depth. A node with depth k corresponds to a leaf, and hence no split is required. For splits at the same depth (say depth i), we can assign ϵ_i budget to each split because the data set held at the nodes of the same depth are disjoint. Therefore we have ϵ_i-DP guaranteed simultaneously by the parallel composition theorem. Furthermore, by the sequential composition theorem we sum all splits at different depths and obtain $(\epsilon_0 + \cdots + \epsilon_{k-1}) = \epsilon_s$-DP.

Now, for *DistributeLabels* we will use the Laplace mechanism to output a private count of the classes while guaranteeing ϵ_l-DP, where ϵ_l is the desired privacy parameter for leaf construction. We note that the sensitivity of the class count is 1 as adding or removing a point changes the count by at most 1. Hence, by the Laplace mechanism, it suffices to add random noise drawn from $Lap(1/\epsilon_l)$ to achieve ϵ_l-DP. Moreover, since the data set in each leaf is disjoint from each other, by guaranteeing ϵ_l-DP for each leave we can obtain ϵ_l-DP for all leaves simultaneously by parallel composition. Thus, we have shown that, for any given ϵ_s and ϵ_l, the ensemble construction in Alg. 2 achieves $(\epsilon_s + \epsilon_l)$-DP.

4.2 Privacy budget allocation

In this section we discuss our strategy of privacy budget allocation for the construction of ensembles in Alg. 2. For a total privacy budget ϵ, since we have partitioned the sample set S into N disjoint subsets $\{S_i\}_{i=1}^{N}$, we can allocate the whole privacy budget ϵ to every tree by the parallel composition theorem. We further split ϵ to $\epsilon = \epsilon_s + \epsilon_l$ for privacy budget used in node splits and label predictions, respectively. We will use an equal share between the two as default, since both procedures are important to the final performance, i.e. $\epsilon_s = \epsilon_l = \epsilon/2$ (except in semi-supervised setting which we will discuss in Section 5).

We now discuss the budget allocation of ϵ_s along the nodes as follows. As a general intuition, the optimal budget allocation should depend on the difficulty of performing the private median computation. Based on this idea, we propose that the privacy budget allocation follows a geometrically-scaling sequence along the depths of the nodes.

Let $r, r' \in \mathcal{R}$ be any two potential outputs drawn uniformly from $[a, b], a, b \in \mathbb{R}$. The private estimation of the median is easier if we can distinguish the utility of r, r' and output the better option with higher probability. This means that we want the utility difference $|u(r) - u(r')|$ to be large, as calculated as the number of values between r and r'. We observe that the expected difference between two randomly chosen points from a uniform distribution is equal to $|a-b|/3$ (we cannot make any assumption on the values of the input samples). This observation implies that for every point added or removed, the probability that $|u(r)-u(r')|$ will change due to the added/removed point equals to $1/3$. Since any parent node is expected to receive 2 times the number of samples compared to its child nodes, the median estimation problem will be $2 \times (1/3)$ times easier comparing to its child node. Hence, for any node at depth i that received ϵ_i budget, we assign $\epsilon_{i+1} = (3/2) * \epsilon_i$ privacy budget to its child nodes at depth $i + 1$. Furthermore, we must full-fill the condition that the sum of privacy budgets over all depths equals ϵ_s. Hence we scale each ϵ_i by a constant C so that we have $\sum_{i=0}^{k-1} \epsilon_i = \epsilon_s$, WLOG we assume $\epsilon_0 = C\epsilon_s$.

$$\begin{aligned}\epsilon_s = \sum_{i=0}^{k-1} \epsilon_i &= \epsilon_0 + (3/2)\epsilon_0 + \cdots + (3/2)^{k-1}\epsilon_0 \\ &= C\epsilon_s(1 + (3/2) + \cdots + (3/2)^{k-1}) = C\epsilon_s \left(\frac{(3/2)^k - 1}{(3/2) - 1} \right), \qquad (7)\end{aligned}$$

which implies $\epsilon_i = C\epsilon_s \left(\frac{3}{2}\right)^i$ and $C = \frac{1}{2(3/2)^k - 2}$.

To illustrate the effectiveness of our privacy budget allocation strategy, we run simulations of Alg. 1 to analyse the quality of the estimated median by comparing our allocation strategy with the uniform allocation baseline. We run our experiments on a synthetic data set generated from a normal distribution and we kept all other parameters fixed except the allocation strategy.

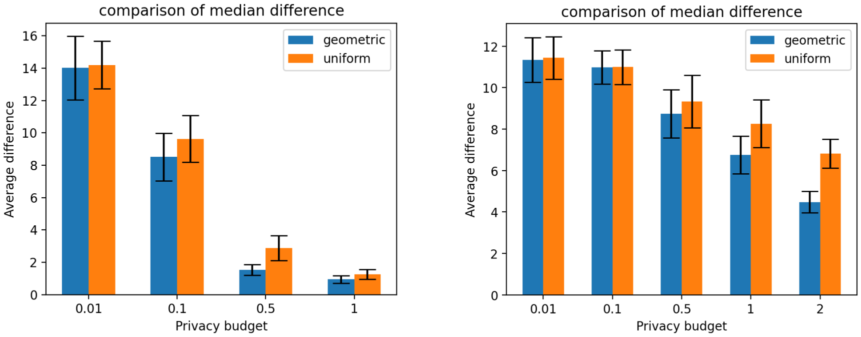

Fig. 2. Comparison between geometric and uniform allocation at different privacy levels with the maximum depth 5 (left) and 10 (right). Error bars indicate (±1) standard deviation from the mean, and each experiment is repeated 50 times. We observe that our strategy achieves smaller average distance to the true median over all sets of experiments. This shows that the proposed allocation strategy has a significant effect on the median estimation while guaranteeing the same level of privacy.

5 Differentially Private Semi-Supervised Ensembles

In this section we present our framework of private semi-supervised learning that creates private ensembles in the following private settings: (I) privacy of both the features and labels are required; (II) only privacy of labels is required. For the first case, existing work can only train on a labelled set, and cannot take advantage of a separate unlabelled set. In contrast, the method of our framework can build the tree using the unlabelled set only; as a result we can assign all privacy budget to the label predictions and reduce the noise. Moreover, we significantly reduce the number of labelled samples needed while achieving a good accuracy level. We present the procedures of the private ensembles in Alg. 3 and 4, Alg. 3 applied to both private settings (I) and (II) where Alg. 4 is applicable only in setting (II). For setting (II) in Alg. 3 we use ∞-privacy to build trees, meaning that we can compute the true median for each split, and no partitioning of the unlabelled set is required.

In setting (II) we can perform computations with the features as many times as needed and release the output without privacy concern. A transductive approach can be applied in this case to take further advantage of the unlabelled data [20]. The transductive approach trains a small ensemble using a labelled set and then predicts labels for each sample in the unlabelled set using the trained ensemble. The newly-labelled set can then be used to train a larger ensemble. Our framework also takes advantage of this approach in the label-only privacy setting. A draw-back of this technique is that the newly-labelled set can have noisy labels due to inaccurate predictions which can decrease the accuracy of the final model.

Algorithm 3 Semi-Supervised Private Ensemble

```
Inputs: labelled set $S$, unlabelled set $D$, maximum depth $k$, size of ensemble $N$, privacy budget $\epsilon$
procedure SS-ENSEMBLE($S, D, k, N, \epsilon$)
    if need privacy for features and labels then
        Partition $S, D$ into $N$ disjoint portions: $\{S_i\}_{i=1}^N, \{D_i\}_{i=1}^N$
        for $i$ in range $1, \dots, N$ do
            Tree $i$ = BuildTree($D_i$, $k$, $\epsilon$)
            $DistributeLabels(Tree\ i, S_i, \epsilon)$ and add Tree $i$ to ensemble
        end for
    else #privacy for labels only
        $S' = S$ with labels removed
        Partition $S$ into $N$ disjoint portions: $\{S_i\}_{i=1}^N$
        for $i$ in range $1, \dots, N$ do
            Tree $i$ = BuildTree($S' \cup D$, $k$, $\infty$)
            $DistributeLabels(Tree\ i, S_i, \epsilon)$ and add Tree $i$ to ensemble
        end for
    end if
    Return ensemble
end procedure
```

Algorithm 4 Private Transductive Ensemble

```
Inputs: labelled set $S$, unlabelled set $D$, maximum depth $k$, size of first ensemble $N1$, size of second ensemble $N2$, privacy budget $\epsilon$
procedure TRANSDUCTIVEENSEMBLE($S, D, k, N1, N2, \epsilon$)
    Partition $S$ into $N1$ disjoint portions: $\{S_i\}_{i=1}^N$
    for $i$ in range $1, \dots, N1$ do
        Tree $i$ = BuildTree($S' \cup D$, $k$, $\epsilon$)
        $DistributeLabels(Tree\ i, S_i, \epsilon)$ and add Tree $i$ to ensemble1
    end for
    Assign labels to samples in $D$ using ensemble1 and denote by $D_l$
    for $i$ in range $1, \dots, N2$ do
        Tree $i$ = BuildTree($S' \cup D$, $k$, $\epsilon$)
        $DistributeLabels(Tree\ i, D_l, \epsilon)$ and add Tree $i$ to ensemble2
    end for
    Return $ensemble1 \cup ensemble2$
end procedure
```

6 Experimental Analysis

To illustrate the performance of the proposed algorithm, we perform a series of experiments using synthetic data sets as well as real data sets from the UCI [8]. The synthetic data sets are generated by forming normally distributed clusters with random centers using the python package *sklearn.make_classification*. We generate three synthetic data sets each with 3000 samples with $5, 10$ and 15 attributes, each data set contains 2 classes. The UCI data sets cover a wide range of real data with size ranging from 150 to 32561 and dimensions ranging from

4 to 33. We use 90% of the data for training and the remaining 10% for testing in all of our experiments. Each data set is randomly shuffled before training. Each experiment is repeated on the same data set 50 times, and the average and standard deviation of the prediction accuracy is reported.

6.1 Varying the parameters

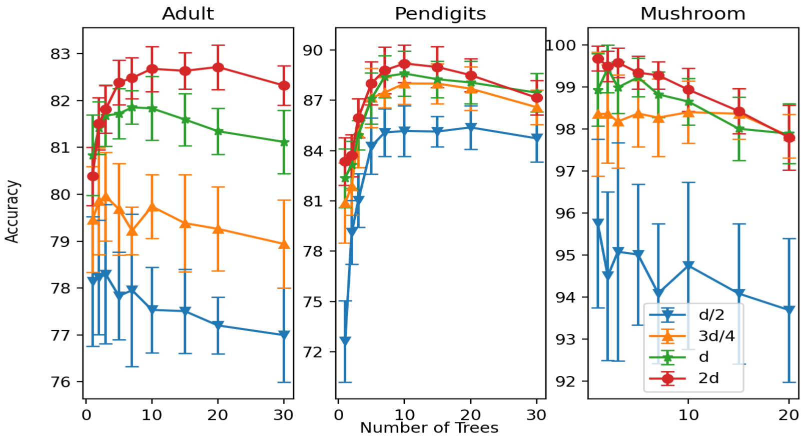

Fig. 3. Effect of number of trees (x-axis) and maximum depth (legend) on accuracy, where d denotes the dimension of the data. Error bars indicate ±1 standard deviation from the mean, and ϵ is fixed to 2.

We demonstrate the effect of parameters (N trees and max-depth) on the accuracy in Fig. 3 with three UCI data sets using the supervised ensemble (Alg. 2). We see the accuracy is high with small N and decreases as we add more trees into the forest, as we expected, since under privacy constraints each tree works on a disjoint subset of the data, leading to weak learning of the individual trees. From the plots we see that most of the accuracy curves decline after 10 trees, hence we have set $N = 10$ as our default. Furthermore, we see that the choice of max-depth=d is a reasonable default as it reaches high accuracy across data sets. We also experimented with deeper ($2d$) trees (features to split on are sampled with replacement) and see in Fig. 1 that this may win in some cases. However, we must be cautious in general, as this increases the complexity of the function class and we run the risk of over-fitting as the leaf nodes become too small.

6.2 Comparison with other supervised algorithms

We analyse the prediction accuracy in the supervised setting (no unlabelled set) by comparing our Supervised Private Ensemble (SPE) with three state-of-the-art differentially private tree-based algorithms: Smooth random trees (SRT) by

[12], Random Decision Trees (RDT) by [17] and a version of greedy decision trees (MaxTree) by [19]. In the experiments, we build 10 trees with maximum depth d as a default value where d is the dimension. For competitors we used parameters recommended by the authors. For a non-private reference we use a random forest classifier with 100 trees as a benchmark for the best performance achievable on the particular data set without privacy constraints. The data sets used are described in Table 1.

	Size	Attributes	RF	SPE	SRT	RDT	MaxTree
Adults	32561	14	85.47%	**82.05**% ⋆	76.89%	76.16%	81.80%
Bank	4520	16	89.39%	88.24%	**88.73**%	88.21%	88.31%
Banknotes	1372	4	99.38%	**93.54**% ⋆	77.39%	70.59%	86.93%
Blood Transfusion	747	4	74.99%	78.00%	75.71%	**78.05**%	75.68%
Car	1728	7	97.88%	**72.71**% ⋆	71.16%	69.38%	71.42%
Claves	10787	16	80.82%	**73.84**% ⋆	70.80%	73.02%	73.22%
Credit Card	30000	24	82.48%	**80.22**% ⋆	77.80%	78.28%	78.61%
Dry Bean	13611	17	92.61%	**90.74**% ⋆	84.42%	75.28%	89.99%
GammaTele	19020	10	87.99%	**82.34**% ⋆	71.62%	66.84%	78.58%
Iris	150	4	95.33%	**81.87**%	80.53%	81.20%	31.07%
Letter	20000	17	96.13%	**67.86**% ⋆	47.61%	40.25%	62.99%
Mushroom	8124	7	100.00%	**99.15**% ⋆	98.36%	93.89%	97.51%
Nursery	12960	8	98.56%	79.19%	80.64%	65.57%	**88.29**% ⋆
Occupancy	8142	7	99.71%	**98.19**% ⋆	90.26%	82.87%	97.07%
Pendigits	7494	16	99.22%	**91.72**% ⋆	89.03%	81.61%	47.12%
Robot	5456	4	99.46%	**87.43**% ⋆	61.50%	56.86%	47.70%
Student	648	33	78.58%	65.29%	60.77%	64.28%	**65.60**%
Wine	178	4	100.00%	73.00%	72.56%	**74.11**%	36.00%
Syn5d	3000	5	92.49%	**90.41**% ⋆	86.52%	79.78%	88.34%
Syn10d	3000	10	94.52%	**87.64**%	80.55%	78.57%	86.71%
Syn15d	3000	15	94.08%	**87.11**%	80.67%	80.24%	86.83%

Table 1. Comparisons with other private tree ensemble methods in supervised learning. The privacy budget is fixed to 2. The best result is highlighted in **bold**. We use the symbol ⋆ to indicate if the best result is statistically significant compared to others.

To further evaluate if the reported result is statistically significant, we perform a Mann-Whitney U test (Wilcoxon rank sum test) at a 95% confidence level. From Table 1 we see that our method achieves higher accuracy for the majority of data sets with a statistical significance. The largest improvement is more than 25%, obtained on the *Robot* data set. Out of 22 total data sets tested, the only data set where our method is significantly worse is the *Nursery* data set.

6.3 Comparisons in private semi-supervised learning

To assess our framework in the semi-supervised setting, we perform experiments with a reduced number of labelled samples and a separate unlabelled set. The data sets will be the same as our analysis in section 6.2 except *Iris* and *Wine*, which have less than 200 points, hence too small for semi-supervised learning. For each of the remaining data sets we only use 20% of the training set as the labelled training set, the other 80% will be used as an unlabelled set with their labels removed. We fix the privacy budget to 2 as before. We compare the methods of our framework in semi-supervised learning with two state-of-the-art competitors - Semi-supervised RDT (SSRDT) by [16] and Transductive Output Perturbation (TOP) by [20], where SSRDT is a tree-structured non-parametric method and TOP is a combination of kNN and linear predictors. Both competitors apply in the case where feature privacy is not required, hence we compare them with our second setting of Alg. 3 (DPE-2) and our Private Transductive Ensemble (PTE) as they are in the same setting. We also include the first setting of Alg. 3 in our comparison however the result can be worse since it guarantees a stronger privacy (both features and labels). From table 2 we observe that our methods SSPE-2 and PTE achieve a better accuracy over SSRDT and TOP for the majority of data sets tested, and most improvements are statistically significant. Note that the results are in general worse than the figures in the supervised case, since we only have access to 20% of the labels. For SSPE-1, despite it guarantees the same level of privacy for both the features and the labels, the result has shown

	SSPE - 1	SSPE - 2	PTE	SSRDT	TOP
Adults	**80.53** % ⋆	**80.88**% ⋆	**80.29**%⋆	75.97%	76.18%
Bank	87.97%	88.02%	**88.84**%	88.58%	88.44%
Banknotes	53.86%	**89.67**%	**90.41**%⋆	88.52%	55.30%
Blood_transfusion	76.16%	75.84%	76.11%	75.68%	**77.36**%
Car	70.23%	**72.55**% ⋆	**71.11**%	70.40%	33.88%
Claves	67.88%	67.70%	57.68%	**74.16**%⋆	9.49%
Credit Card	**78.79** % ⋆	**78.38**% ⋆	**77.89**%	77.82%	77.80%
Dry Bean	**86.84**% ⋆	**87.76**% ⋆	**88.61**%⋆	81.41%	72.30%
GammaTele	**74.77** % ⋆	**77.02**%⋆	**74.11**% ⋆	65.50%	64.83%
Letter	42.27%	**53.80**%	**56.91**% ⋆	16.72%	53.42%
Mushroom	90.09%	**95.96**% ⋆	**95.46**% ⋆	92.69%	51.65%
Nursery	**74.31** % ⋆	**77.17**% ⋆	**76.37**% ⋆	66.30%	50.66%
Occupancy	**96.14** % ⋆	94.20%	92.77%	**95.16**% ⋆	79.08%
Pendigits	72.04%	**85.22**% ⋆	**87.00**% ⋆	70.10%	84.26%
Robot	**77.30**% ⋆	**87.01**% ⋆	**82.56**% ⋆	64.33%	61.43%
Student	64.77%	**65.14**%	**65.05**%	63.05%	64.98%
Syn5d	86.82%	**91.59**% ⋆	**91.54**% ⋆	90.09%	90.29%
Syn10d	86.30%	90.51%	91.11%	83.93%	**91.31**%
Syn15d	76.40%	85.94%	**86.45**%	79.59%	86.29%

Table 2. Comparison with existing private semi-supervised methods. **Bold** indicates if the result is better than its competitors and ⋆ indicates if the difference is statistically significant. For SSPE-1, we use *underline* to indicate when it performs not significantly worse than SSRDT and TOP.

its performance remains on a same level (no significant difference detected) in comparison with the competitors which only guarantee privacy for the labels. Moreover, it has significant improvements over SSRDT and TOP on some data sets despite the additional added noise as shown in Table 2. Hence we conclude that the methods in our framework achieve a significant improvement over the state of the art in utility performance and privacy guarantee.

7 Conclusions

We have proposed a framework of differentially private classification for supervised and semi-supervised learning with high utility. This is based on a novel combination of techniques to build a new private machine ensemble for supervised learning, which naturally extends to a semi-supervised setting. We proposed a novel privacy budget allocation scheme that increases the usage efficiency of the available privacy budget and improves accuracy. Our experimental analysis over a wide range of data sets demonstrates that our method provides significantly better performance than the state of the art. In the semi-supervised setting, we proposed private ensembles that can be trained efficiently using a small number of labelled samples while achieving high utility, which allows us to reduce labelling efforts in data sets with sensitive information. In particular, we proposed the first semi-supervised private ensemble that is applicable in two privacy settings (feature and label privacy, and just label privacy). Empirically we found that our method provides high performance in both settings.

Acknowledgements

The last author is funded by EPSRC grant EP/P004245/1.

References

1. Aryal, S., Ting, K.M., Washio, T., Haffari, G.: Data-dependent dissimilarity measure: an effective alternative to geometric distance measures. Knowledge and information systems **63**, 479–506 (11 2017)
2. Beniley, J.: Multidimensional binary seareh trees used for assoeiative searehing. ACM Communications **18**(9), 509–517 (1975)
3. Biau, G.: Analysis of a random forests model. The Journal of Machine Learning Research **13**(1), 1063–1095 (2012)
4. Breiman, L.: Consistency for a simple model of random forests (2004)
5. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. CRC press (1984)
6. Consul, S., William, S.A.: Differentially private random forests for regression and classification. Association for the Advancement of Artificial Intelligence (2021)
7. Cormode, G., Procopiuc, C., Srivastava, D., Shen, E., Yu, T.: Differentially private spatial decompositions. In: 2012 IEEE 28th International Conference on Data Engineering. IEEE (2012)
8. Dua, D., Graff, C.: UCI machine learning repository (2017)

9. Dwork, C.: Differential privacy. Automata, Languages and Programming pp. 1–12 (2006)
10. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science **9**(3-4), 211–407 (2014)
11. Fletcher, S., Islam, M.Z.: A differentially private decision forest. In: Proceedings of the 13-th Australasian Data Mining Conference (2015)
12. Fletcher, S., Islam, M.Z.: Differentially private random decision forests using smooth sensitivity. Expert Systems with Applications **78**, 16–31 (2017)
13. Friedman, A., Schuster, A.: Data mining with differential privacy. In: Proceedings of the 16th SIGKDD conference on knowledge discovery and data mining. pp. 493–502 (2010)
14. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning **63**(1), 3–42 (2006)
15. Gong, M., Xie, Y., Pan, K., Feng, K., Qin, A.: A survey on differentially private machine learning. IEEE Computational Intelligence Magazine **15**(2), 49–64 (2020)
16. Jagannathan, G., Monteleoni, C., Pillaipakkamnatt, K.: A semi-supervised learning approach to differential privacy. In: 2013 IEEE 13th International Conference on Data Mining Workshops. pp. 841–848. IEEE (2013)
17. Jagannathan, G., Pillaipakkamnatt, K., Wright, R.N.: A practical differentially private random decision tree classifier. In: 2009 IEEE International Conference on Data Mining Workshops. pp. 114–121. IEEE (2009)
18. Klusowski, J.: Sharp analysis of a simple model for random forests. In: International Conference on Artificial Intelligence and Statistics (2021)
19. Liu, X., Li, Q., Li, T., Chen, D.: Differentially private classification with decision tree ensemble. Applied Soft Computing **62**, 807–816 (2018)
20. Long, X., Sakuma, J.: Differentially private semi-supervised classification. In: 2017 IEEE International Conference on Smart Computing (SMARTCOMP). pp. 1–6. IEEE (2017)
21. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07). pp. 94–103. IEEE (2007)
22. McSherry, F.D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. pp. 19–30 (2009)
23. Mohammed, N., Chen, R., Fung, B.C., Yu, P.S.: Differentially private data release for data mining. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 493–501 (2011)
24. Nissim, K., Raskhodnikova, S., Smith, A.: Smooth sensitivity and sampling in private data analysis. In: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing. pp. 75–84 (2007)
25. Pham, A.T., Xi, J.: Differentially private semi-supervised learning with known class priors. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 801–810. IEEE (2018)
26. Rana, S., Gupta, S.K., Venkatesh, S.: Differentially private random forest with high utility. In: 2015 IEEE International Conference on Data Mining. pp. 955–960. IEEE (2015)
27. Xin, B., Yang, W., Wang, S., Huang, L.: Differentially private greedy decision forest. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2672–2676 (2019)