Multinomial Random Forest

Article *in* Pattern Recognition · September 2021

DOI: 10.1016/j.patcog.2021.108331

Jiawang Bai
Tsinghua University
22 PUBLICATIONS **292** CITATIONS
SEE PROFILE

Yiming Li
Nanyang Technological University
79 PUBLICATIONS **1,522** CITATIONS
SEE PROFILE

Jiawei Li
Huawei Technologies
21 PUBLICATIONS **141** CITATIONS
SEE PROFILE

Xue Yang
University of Southampton
25 PUBLICATIONS **417** CITATIONS
SEE PROFILE

Pattern Recognition 122 (2022) 108331

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Multinomial random forest

Jiawang Bai[a,b,1], Yiming Li[a,1], Jiawei Li[a], Xue Yang[a,b,*], Yong Jiang[a,b], Shu-Tao Xia[a,b]

[a] *Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China*
[b] *PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen, China*

ARTICLE INFO

Article history:
Received 16 December 2020
Revised 18 August 2021
Accepted 17 September 2021
Available online 20 September 2021

Keywords:
Random forest
Consistency
Differential privacy
Classification

ABSTRACT

Despite the impressive performance of random forests (RF), its theoretical properties have not been thoroughly understood. In this paper, we propose a novel RF framework, dubbed multinomial random forest (MRF), to analyze its *consistency* and *privacy-preservation*. Instead of deterministic greedy split rule or with simple randomness, the MRF adopts two impurity-based multinomial distributions to randomly select a splitting feature and a splitting value, respectively. Theoretically, we prove the consistency of MRF and analyze its privacy-preservation within the framework of differential privacy. We also demonstrate with multiple datasets that its performance is on par with the standard RF. To the best of our knowledge, MRF is the first consistent RF variant that has comparable performance to the standard RF. The code is available at https://github.com/jiawangbai/Multinomial-Random-Forest.

1. Introduction

Random forest (RF) [1] is a popular type of ensemble learning method. Because of its excellent performance and fast yet efficient training process, RF and other tree-based methods have been widely used in many fields, such as computer vision [2–4] and data mining [5–7]. However, due to the inherent bootstrap randomization and the highly greedy data-dependent construction process, it is very difficult to analyze the theoretical properties of RFs [8], especially for the *consistency*. Since consistency ensures that the model goes to optimal under a sufficient amount of data, it is critical in this big data era.

To address this issue, several RF variants [8–13] were proposed. Unfortunately, all existing consistent RF variants suffer from relatively poor performance compared with the standard RF due to two mechanisms introduced for ensuring consistency. On the one hand, the data partition process allows only half of the training samples to be used for constructing the tree structure, which significantly reduces the performance of consistent RF variants. On the other hand, extra randomness (*e.g.*, Poisson or Bernoulli distribution) is introduced, which further hinders the performance. Accordingly, those mechanisms introduced for theoretical analysis make them difficult to eliminate the performance gap between consistent RF and standard RF.

Is this gap really impossible to fill? In this paper, we propose a novel consistent RF framework[2], dubbed multinomial random forest (MRF), by introducing the randomness more reasonably. In the MRF, two impurity-based multinomial distributions are used as the basis for randomly selecting a splitting feature and a specific splitting value respectively. Accordingly, the “best” split point has the highest probability to be chosen, while other candidate split points that are nearly as good as the “best” one will also have a good chance to be selected, as shown in Fig. 1. This randomized splitting process is more reasonable and makes up the accuracy drop with almost no extra computational costs. Besides, privacy-preservation is very important in the big data era, especially for machine learning, due to the continued emergence of privacy breaches and data abuse. More specifically, data is the huge digital wealth for organizations in machine learning, and attackers may infer or reconstruct the sensitive training data as much as possible from the public model. Therefore, protecting data privacy (*e.g.*, prevent unauthorized access to training data) is becoming an important aspect in the development of machine learning. The introduced impurity-based randomness is essentially an exponential mechanism satisfying differential privacy, therefore we can also analyze the privacy-preservation of MRF under the differential privacy framework. To the best of our knowledge, there is no RF framework could be adopted to analyze the consistency and privacy-preservation simultaneously.

The main contributions of this work are three-fold: **(1)** we propose a novel multinomial-based method to improve the greedy

[*] Corresponding author.
E-mail addresses: yang.xue@sz.tsinghua.edu.cn (X. Yang), xiast@sz.tsinghua.edu.cn (S.-T. Xia).

[1] indicates equal contribution

[2] In this paper, we focus on the consistency and privacy-preservation in the classification problem. We will explore the regression task in our future work.

https://doi.org/10.1016/j.patcog.2021.108331
0031-3203/© 2021 Published by Elsevier Ltd.

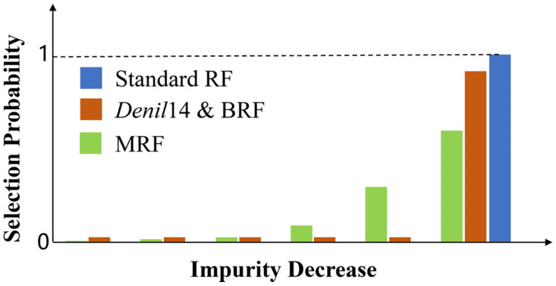


Fig. 1. Splitting criteria of different RFs. The standard RF always chooses the split point with highest impurity decrease. *Denil14* and BRF choose the split point mostlyin a greedy way, while holding a small or even negligible probability in selecting other points randomly. The selection probability in MRF is positively associated with the impurity decrease. All three RF variants introduce randomness to fulfill the consistency, where MRF is the most reasonable method.

split process of decision trees; **(2)** we propose a new RF variant (*i.e.*, MRF), based on which we analyze its consistency and privacy-preservation; **(3)** extensive experiments demonstrate that the performance of MRF is on par with standard RF and is better than all existing consistent RF variants. To the best of our knowledge, MRF is the first consistent RF variant that simultaneously has performance comparable to the standard RF.

2. Related work

2.1. Consistent random forests

Random forest (RF) [1] is a distinguished ensemble algorithm, inspired by the random subspace [14] and random split selection [15]. The standard RF is built upon bootstrap datasets and splitting with the CART methodology [16]. Its various variants, such as quantile regression forests [17], rotation random forests [18], and deep forest [19], were proposed for effectiveness, efficiency, and great interpretability. Especially, the oblique random forests [20–22], where trees employ a linear combination of features to generate an oblique hyperplane at each node, can achieve remarkable performance. RFs were also used in a wide range of applications, including time series forecasting [23] and visual tracking [24]. Despite the success of RFs in practice, their theoretical analysis has yet been fully established. Breiman [1] showed the first theoretical result indicating that the generalization error is bounded by the performance of individual tree and the diversity of the whole forest. Moreover, the relationship between RFs and the nearest neighbor-based estimator was also studied [25].

One of the important properties, consistency, has yet to be established for RFs. Consistency ensures that the result of RF converges to the optimum as the sample size increases, which was first discussed by Breiman [9]. As an important milestone, Biau [10] proved the consistency of two directly simplified RFs. Subsequently, several consistent RF variants were proposed for various purposes, for example, random survival forests [26], an online version of RF variant [27] and generalized regression forests [28]. Recently, Haghiri [29] proposed CompRF, whose split process is relied on triplet comparisons rather than information gain. To ensure consistency, [8] suggested that an independent dataset is needed to fit in the leaf. This approach is called the data partition. Under this framework, [12] developed a consistent RF variant (called *Denil14* in this paper) to narrow the gap between theory and practice. Following *Denil14*, [13] introduced the Bernoulli random forests (BRF), which reached the state-of-the-art performance. Besides, Gao *et al*. [30] discussed the convergence rate of a type of consistent RF variant most recently.

Although several consistent RF variants were proposed, due to the relatively poor performance compared with RF, how to fulfill the gap between theoretical consistency and the performance in practice is still an important open question.

2.2. Privacy-preservation

With the growing concerns about privacy, many strategies for protecting privacy in RFs have been explored in recent years, like k-anonymity [31] and l-diversity [32]. However, these strategies do not provide privacy in a mathematically rigorous way. In order to improve the privacy guarantee, differential privacy (DP) [33], as a new and promising privacy-preservation model, has been widely adopted recently, especially for RFs [34–36]. Specifically, due to the trade-off between privacy and learning accuracy in the DP-based RFs, most researches considered improving the learning accuracy by designing allocation strategies of the privacy budget [37,38] or decreasing the sensitivity [35]. However, in most cases, the learning accuracy of these schemes is still not satisfactory in practice.

Since we use DP technique to guarantee the privacy of sensitive data, we first outline the basic content of differential privacy here. Let $\mathcal{D} = \{(\boldsymbol{X}_i, Y_i)\}_{i=1}^{n}$ denote a dataset consisting of n *i.i.d.* observations, where $\boldsymbol{X}_i \in \mathbb{R}^D$ indicates D-dimensional features and $Y_i \in \{1, \ldots, K\}$ indicates the label. Suppose $\mathcal{A} = \{A_1, \ldots, A_D\}$ represents the feature set. The formal definition of differential privacy is given as follow:

Definition 1 . (**ϵ-Differential Privacy**) A randomized mechanism $\mathcal{M}$ gives ϵ-differential privacy for every set of outputs O and any neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ differing in one record, if $\mathcal{M}$ satisfies:

$$\Pr[\mathcal{M}(\mathcal{D}) \in O] \leqslant \exp(\epsilon) \cdot \Pr[\mathcal{M}(\mathcal{D}') \in O], \tag{1}$$

where $\mathcal{M}(\mathcal{D})$ and $\mathcal{M}(\mathcal{D}')$ are the outputs of the mechanism for input databases $\mathcal{D}$ and $\mathcal{D}'$, respectively, Pr is the randomness of the noise in the mechanism, and ϵ denotes the privacy budget that restricts the privacy guarantee level of $\mathcal{M}$.

The aim of differential privacy is to mask the differences in query between neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$. Specifically, from Eq. (1), we can see that a small ϵ ($\leqslant 1$) means that the difference of mechanism's output probabilities using $\mathcal{D}$ and $\mathcal{D}'$ is small, which indicates high perturbations of ground truth outputs and hence high privacy, and vice versa. That is a smaller ϵ represents a stronger privacy level. The non-private case is given by $\epsilon = \infty$.

Besides, according to Definition 1 and the intuition above, the noise protects the membership of a data point in the dataset. For example, when conducting a clinical experiment, sometimes a person does not want the observer to know that he or she is involved in the experiment. This is due to the fact that the observer may link the test result to the appearance/disappearance of a certain person and harm the interest of that person. Proper membership protection would ensure that replacing this person with another one will not affect the result too much. This property holds only if the algorithm itself is randomized, *i.e.*, the output is associated with a distribution. And this distribution will not change too much if a certain data point is perturbed or even removed. This is exactly what differential privacy tries to achieve.

Currently, two basic mechanisms, *i.e.*, *Laplace mechanism* [39] and *exponential mechanism* [40], are widely used to realize differential privacy. The first one is suitable for numeric queries and the second one is suitable for non-numeric queries. As presented in Section 3, in the multinomial random forest (MRF), we need to choose the splitting feature and splitting value, which belongs to the non-numeric query. Thus, we adopt the exponential mechanism to preserve privacy. More specifically, suppose that one wants to publish $f(\mathcal{D})$, and let O denote the set of possible outputs. To satisfy ϵ-DP, the exponential mechanism should output values in O following some probability distribution. Naturally, some values in O are more desirable than others. For example, the most desirable output is the true value $f(\mathcal{D})$, and one has natural preferences among other values as well. Specifically, consider a transactional dataset $\mathcal{D}$, and one wants to output the item that appears most frequently in $\mathcal{D}$. Then O is the set of all items, and between two items, we prefer to output the one that appears more often. This preference is encoded using a quality function $q : (\mathcal{D}, o) \to \mathbb{R}$, and without loss of generality, we assume that a higher quality value indicates better utility. For example, in the most frequent item case, a natural choice is to define $q(\mathcal{D}, o)$ to be the number of times the item o appears in $\mathcal{D}$.

Definition 2 (Exponential Mechanism)**.** Let $q : (\mathcal{D}, o) \to \mathbb{R}$ be a score function of dataset $\mathcal{D}$ that measures the quality of output $o \in O$. The exponential mechanism $\mathcal{M}(\mathcal{D})$ satisfies ϵ-DP, if it outputs o with probability proportional to $\exp\left(\frac{\epsilon q(\mathcal{D},o)}{2\triangle q}\right)$, *i.e.*,

$$\Pr[\mathcal{M}(\mathcal{D}) = o] = \frac{\exp\left(\frac{\epsilon q(\mathcal{D},o)}{2\triangle q}\right)}{\sum_{o'\in O}\exp\left(\frac{\epsilon q(\mathcal{D},o')}{2\triangle q}\right)}, \tag{2}$$

where $\triangle q$ is the sensitivity of the quality function, as follows:

$$\triangle q = \max_{\forall o,\mathcal{D},\mathcal{D}'} \left| q(\mathcal{D}, o) - q(\mathcal{D}', o)\right|. \tag{3}$$

According to Definition 2, we can obtain that the smaller the privacy budget ϵ, the closer the probability of each output, and thus the attacker cannot judge the true result. When $\epsilon = 0$, the privacy protection level is the highest, and all results have the same probability to be outputted. In addition, given the privacy budget ϵ and the sensitivity $\triangle q$, the higher the value of $q(\mathcal{D}, o)$, the higher the probability $\Pr[\mathcal{M}(\mathcal{D}) = o]$ of outputting o. That is, when applying the exponential mechanism, the probability that a low-quality output is selected is exponentially smaller than that of high-quality output.

3. Multinomial random forest

3.1. Training set partition

In the MRF, we also replace the bootstrap used in standard RF with the training set partition, as suggested in [8]. This is necessary for ensuring consistency. Specifically, to build a tree, the training set $\mathcal{D}$ is divided randomly into two non-overlapping subsets $\mathcal{D}^S$ and $\mathcal{D}^E$, which play different roles (as shown in Fig. 2). $\mathcal{D}^S$ will be used to build the tree's structure, and we call the observations in this subset the **structure points**. Once a tree is built, the labels of its leaves will be re-determined on the basis of another subset $\mathcal{D}^E$, where the corresponding observations are called **estimation points**. The ratio of two subsets is parameterized by **partition rate** $= |\text{Structure points}|/|\text{Estimation points}|$. To build another tree, the training set is re-partitioned randomly and independently.

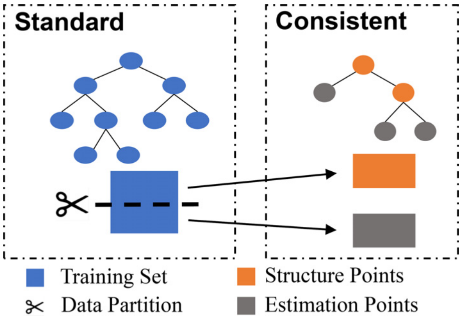


Fig. 2. An illustration of data partition.

3.2. Tree construction

The construction of a tree relies on a recursive partitioning algorithm. Specifically, to split a node, we introduce two impurity-based multinomial distributions: one for splitting feature selection and another for splitting value selection. The specific split point consists of a pair of a splitting feature and a splitting value. Besides, the impurity decrease at a node u caused by a split point v is defined as

$$I(\mathcal{D}_u^S, v) = T(\mathcal{D}_u^S) - \frac{|\mathcal{D}_u^{S_l}|}{|\mathcal{D}_u^S|}T(\mathcal{D}_u^{S_l}) - \frac{|\mathcal{D}_u^{S_r}|}{|\mathcal{D}_u^S|}T(\mathcal{D}_u^{S_r}), \tag{4}$$

where $\mathcal{D}_u^S$ is the subset of $\mathcal{D}^S$ at a node u, $\mathcal{D}_u^{S_l}$ and $\mathcal{D}_u^{S_r}$ generated by splitting $\mathcal{D}_u^S$ with v, are two subsets in the left child and right child of the node u, respectively, and $T(\cdot)$ is the impurity criterion (*e.g.*, Shannon entropy or Gini index). Unless other specification, we ignore the subscript u of each symbol, and use I to denote $I(\mathcal{D}_u^S, v)$ for shorthand in the rest of this paper.

Let $V = \{v_{ij}\}$ denote the set of all possible split points for the node and $I_{i,j}$ is the corresponding impurity decrease, where v_{ij} is i-th value on the j-th feature. In what follows, we first introduce the feature selection mechanism for a node, and then describe the splitting value selection mechanism corresponding to the selected feature. **$M(\boldsymbol{\phi})$-based splitting feature selection.** We obtain a vector $\boldsymbol{I} = (I_1, \cdots, I_D) = \left(\max_i\{I_{i,1}\}, \cdots, \max_i\{I_{i,D}\}\right)$ based on each $I_{i,j}$, where $\max_i\{I_{i,j}\}$ is the largest possible impurity decrease of the feature A_j. Then, the following three steps need to be performed:

- Normalize $\boldsymbol{I}$: $\hat{\boldsymbol{I}} = \left(\frac{I_1 - \min \boldsymbol{I}}{\max \boldsymbol{I} - \min \boldsymbol{I}}, \cdots, \frac{I_D - \min \boldsymbol{I}}{\max \boldsymbol{I} - \min \boldsymbol{I}}\right)$;
- Compute the probabilities $\boldsymbol{\phi} = (\phi_1, \cdots, \phi_D) = \mathrm{softmax}(\frac{B_1}{2}\hat{\boldsymbol{I}})$, where $B_1 \geq 0$ is a hyper-parameter related to privacy budget;
- Randomly select a feature according to the multinomial distribution $M(\boldsymbol{\phi})$.

$M(\boldsymbol{\varphi})$ -based splitting value selection. After selecting the feature A_j for a node, we need to determine the corresponding splitting value to construct two children. Suppose A_j has m possible splitting values, we need to perform the following steps:

- Normalize $\boldsymbol{I}^{(j)} = (I_{1,j}, \cdots, I_{m,j})$ as $\hat{\boldsymbol{I}}^{(j)}$, where j identifies the feature A_j and $\hat{\boldsymbol{I}}^{(j)} = \left(\frac{I_{1,j} - \min \boldsymbol{I}^{(j)}}{\max \boldsymbol{I}^{(j)} - \min \boldsymbol{I}^{(j)}}, \cdots, \frac{I_{m,j} - \min \boldsymbol{I}^{(j)}}{\max \boldsymbol{I}^{(j)} - \min \boldsymbol{I}^{(j)}}\right)$;
- Compute the probabilities $\boldsymbol{\varphi} = (\varphi_1, \cdots, \varphi_m) = \mathrm{softmax}(\frac{B_2}{2}\hat{\boldsymbol{I}}^{(j)})$, where $B_2 \geq 0$ is another hyper-parameter related to privacy budget;
- Randomly select a splitting value based on the multinomial distribution $M(\boldsymbol{\varphi})$.

We repeat the above processes to split nodes until the stopping criterion is met. The stopping criterion relates to the minimum leaf size k. Specifically, the number of estimation points is required to be at least k for every leaf.

$M(\boldsymbol{\psi})$ -based label selection. Once a tree is grown based on $\mathcal{D}^S$, we determine the label of its leafs based on estimation points $\mathcal{D}^E$, as follows:

- Re-determine the predicted probability vector $\boldsymbol{P} = (p_1, \cdots, p_K)$ according to $\mathcal{D}^E$ for each leaf $\mathcal{L}$, *i.e.*, $p_i = \frac{1}{|\mathcal{L}^E|}\sum_{(\boldsymbol{X},Y)\in\mathcal{L}^E}\mathbb{I}\{Y = i\}, (i = 1, \cdots, K)$, where $\mathcal{L}^E$ is the set of estimation points in the leaf $\mathcal{L}$;
- Compute the probabilities $\boldsymbol{\psi} = (\psi_1, \cdots, \psi_K) = \mathrm{softmax}(\frac{B_3}{2}\boldsymbol{P})$ for each leaf $\mathcal{L}$, where $B_3 \geq 0$ is also a hyper-parameter related to privacy budget;

Algorithm 1 Decision Tree Training in MRF: *MTree*() .

```
Input: Structure points $\mathcal{D}^S$, estimation points $\mathcal{D}^E$ and hyper-parameters $k$, $B_1$, $B_2$, and $B_3$.
Output: A decision tree $T$ in MRF.
if $|\mathcal{D}^E| > k$ then
    Calculate the impurity decrease of all possible split points $v_{ij}$.
    Select the largest impurity decrease of each feature to createa vector $\boldsymbol{I}$, calculate the normalized vector $\hat{\boldsymbol{I}}$, and compute the probabilities $\boldsymbol{\phi} = \mathrm{softmax}(\frac{B_1}{2}\hat{\boldsymbol{I}})$.
    Select a splitting feature randomly according to the multinomial distribution $M(\boldsymbol{\phi})$.
    Calculate the normalized vector $\hat{\boldsymbol{I}}^{(j)}$ for the selected splitting feature $A_j$, and compute the probabilities $\boldsymbol{\varphi} = \mathrm{softmax}(\frac{B_2}{2}\hat{\boldsymbol{I}}^{(j)})$.
    Select a splitting value randomly according to the multinomial distribution $M(\boldsymbol{\varphi})$. $\mathcal{D}^S$ and $\mathcal{D}^E$ are correspondingly split into two disjoint subsets $\mathcal{D}^{S_l}, \mathcal{D}^{S_r}$ and $\mathcal{D}^{E_l}, \mathcal{D}^{E_r}$, respectively.
    $T.leftchild \leftarrow MTree(\mathcal{D}^{S_l}, \mathcal{D}^{E_l}, k, B_1, B_2)$
    $T.rightchild \leftarrow MTree(\mathcal{D}^{S_r}, \mathcal{D}^{E_r}, k, B_1, B_2)$
end if
Conduct $M(\boldsymbol{\psi})$-based label selection for each leaf.
Return: A decision tree $T$ in MRF
```

- Randomly select a label for each leaf $\mathcal{L}$ according to the multinomial distribution $M(\boldsymbol{\psi})$ as its representative.

In summary, the training process is summarized in Algorithm 1.

3.3. Prediction

Similar to [1], given an unlabeled sample $\boldsymbol{X}$, we can easily know which leaf of a tree h it falls. The prediction of $\boldsymbol{X}$, *i.e.*, the $h(\boldsymbol{X})$, is the representative label in that leafdetermined in the training process.

Suppose MRF contains t trees, the final prediction $\hat{Y}$ of $\boldsymbol{X}$ by MRF is the majority vote over all trees, which is the same as the one used in [1]:

$$\hat{Y} = \arg \max_{c \in \{1, \cdots, K\}} \sum_{i=1}^{t} \mathbb{I}\left\{h^{(i)}(\boldsymbol{X}) = c\right\}, \tag{5}$$

where $h^{(i)}(\boldsymbol{X})$ is the prediction of $\boldsymbol{X}$ by the tree $h^{(i)}$. Note that if multiple labels achieve the same votes, we break ties by choosing one of them randomly.

4. Consistency and privacy analysis of MRF

In this section, we analyze the consistency and privacy-preservation of the proposed MRF. Note that all proofs are shown in Section Appendix A.

4.1. Consistency

In this section, we first describe the definition of consistency and two previously proven necessary lemmas, then state two new lemmas and the consistency theorem.

Definition 3. Let $\mathcal{D}$ denotes the training set consisting n *i.i.d.* observations, the classifier h is consistent if its probability of error L satisfies

$$\mathbb{E}(L) = \Pr(h(\boldsymbol{X}, Z, \mathcal{D}) \neq Y) \rightarrow L^*, \text{ as } n \rightarrow \infty,$$

where $(\boldsymbol{X}, Y)$ is a random test point, L^* denotes the Bayes risk, Z denotes the randomness involved in the construction of the tree, such as the selection of candidate features.

Lemma 1. *The voting classifier $\overline{h^{(t)}}$ which takes the majority vote over t individually trained classifiers $\{h^{(i)}\}_{i=1}^{t}$ (with the same structure h and different randomizing variables) has consistency if the classifier h is consistent.*

Lemma 2. *Consider a partitioning classification rule building a prediction by a majority vote method in each leaf node. If the labels of the voting data have no effect on the structure of the classification rule, then $\mathbb{E}[L] \rightarrow L^*$ as $n \rightarrow \infty$, when*

1. *The diameter of $\mathcal{N}(\boldsymbol{X}) \rightarrow 0$ as $n \rightarrow \infty$ in probability,*
2. *$|\mathcal{N}^E(\boldsymbol{X})| \rightarrow \infty$ as $n \rightarrow \infty$ in probability,*

where $\mathcal{N}(\boldsymbol{X})$ is the leaf containing $\boldsymbol{X}$, $|\mathcal{N}^E(\boldsymbol{X})|$ is the number of estimation points in $\mathcal{N}(\boldsymbol{X})$.

Lemma 1 [10] states that the consistency of individual tree leads to the consistency of athe forest. Lemma 2 [41] implies that the consistency of a tree can be ensured that as every hypercube at a leaf is sufficiently small while contains infinite number of estimation points $n \rightarrow \infty$.

To prove the consistency based on Lemmas 1–2, there are three main steps, including ensuring that **(1)** each feature has a non-zero probability to be selected, **(2)** each split reduces the expected size of the splitting feature, and **(3)** split process can go on indefinitely. In the following part, we first propose two lemmas for steps **(1)** and **(2)**, respectively, and then describe the consistency theorem of the MRF.

Lemma 3. *In the MRF, the probability that any given feature A is selected to split at each node has lower bound $P_1 > 0$ if the introduced hyper-parameter B_1 for splitting feature selection is upper-bounded.*

Lemma 4. *Suppose that features are all supported on [0,1]. In the MRF, once a splitting feature A is selected, if this feature is divided into $N (N \geq 3)$ equal partitions $A^{(1)}, \cdots, A^{(N)}$ from small to large (i.e., $A^{(i)} = \left[\frac{i-1}{N}, \frac{i}{N}\right]$) and the introduced hyper-parameter B_2 for splitting value selection is upper-bounded, for any split point v,*

$$\exists P_2\ (P_2 > 0), \text{s.t. } \Pr\left(v \in \bigcup_{i=2}^{N-1} A^{(i)} | A\right) \geq P_2.$$

Lemma 3 states that the MRF fulfills the first aforementioned requirement. Lemma 4 states that second condition is also met by showing that the specific splitting value has a large probability that it is not near the two endpoints of the feature interval.

Theorem 1. *Suppose that $\boldsymbol{X}$ is supported on $[0, 1]^D$ and has non-zero density almost everywhere, the cumulative distribution function of the split points is right-continuous at 0 and left-continuous at 1. If $B_3 \rightarrow \infty$ while B_1 and B_2 are upper-bounded, where B_1, B_2, and B_3 are introduced hyper-parameters for the splitting feature selection, splitting value selection, and label selection, respectively, MRF is consistent when $k \rightarrow \infty$ and $k/n \rightarrow 0$ as $n \rightarrow \infty$.*

4.2. Privacy-preservation

In this part, we prove that the MRF satisfies ϵ-differential privacy based on two composition properties [42]. Suppose we have a set of privacy mechanisms $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_p\}$ and each $\mathcal{M}_i$ provides ϵ_i privacy guarantee, then the sequential composition and parallel composition are described as follows:

Property 1 (Sequential Composition). *Suppose $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_p\}$ are sequentially performed on a dataset $\mathcal{D}$, then $\mathcal{M}$ will provide $(\sum_{i=1}^{p} \epsilon_i)$-differential privacy.*

Property 2 (Parallel Composition). *Suppose $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_p\}$ are performed on a disjointed subsets of the entire dataset,*

Table 1
The description of UCI benchmark datasets.

Dataset	Samples	Features	Classes	Dataset	Samples	Features	Classes
Zoo	101	17	7	Banknote	1,372	4	2
Hayes	132	5	3	Cmc	1,473	9	3
Echo	132	12	2	Yeast	1,484	8	10
Hepatitis	155	19	2	Car	1,728	6	4
Wine	178	13	3	Image	2,310	19	7
Wdbc	569	39	2	Chess	3,196	36	2
Transfusion	748	5	2	Ads	3,729	1,558	2
Vehicle	946	18	4	Wilt	4,839	5	2
tic-tac-toe	958	9	2	Wine-Quality	4,898	11	7
Mammo	961	6	2	Phishing	11,055	31	2
Messidor	1,151	19	2	Nursery	12,960	9	5
Website	1,353	9	3	Connect-4	67,557	42	3

i.e., $\{\mathcal{D}_1, \ldots, \mathcal{D}_p\}$, respectively, then $\mathcal{M}$ will provide $(\max\{\epsilon_i\}_{i=1}^{p})$-differential privacy.

Lemma 5. *The impurity-based multinomial distribution $\mathcal{M}(\phi)$ of feature selection is essentially the exponential mechanism of differential privacy, and satisfies B_1-differential privacy.*

Lemma 6. *The impurity-based multinomial distribution $\mathcal{M}(\varphi)$ of splitting value selection is essentially the exponential mechanism of differential privacy, and satisfies B_2-differential privacy.*

Lemma 7. *The label selection of each leaf in a tree satisfies B_3-differential privacy.*

Based on the aforementioned properties and lemmas, we can obtain the following theorem:

Theorem 2. *The proposed MRF satisfies ϵ-differential privacy when the hyper-parameters B_1, B_2 and B_3 satisfy $B_1 + B_2 = \epsilon/(d \cdot t)$ and $B_3 = \epsilon/t$, where t is the number of trees, d is the depth of a tree such that $d \leqslant \mathcal{O}(\frac{|\mathcal{D}^E|}{k})$, and k is the minimum leaf size.*

5. Experiments

In this section, we empirically analyze the performance of our MRF. Specifically, we compare the proposed MRF with four consistent RF variants and Breiman's RF [1] on the UCI datasets and in the semantic segmentation task in Section 5.1 and Section 5.2, respectively. In Section 5.3, we present the performance of MRF with different privacy budgets to show the trade-off between the learning accuracy and the level of privacy preservation. In Section 5.4, we further compare the performance of MRF with some advanced RF variants, which have no properties of consistency or privacy preservation. In Section 5.5, we provide more discussions of the proposed MRF, including the effect of the proposed splitting rule, the effect of hyper-parameters, and computational complexity.

5.1. Performance on UCI datasets

Dataset Selection. Similar to previous consistent RF works [12,13,29], we conduct experiments on twenty-four UCI datasets, which cover a wide range of sample size and feature dimensions to be representative for evaluating the performance of different algorithms. Besides, we have no additional preprocessing for the datasets except for replacing the missing values with '-1'. The description of used datasets is shown in Table 1.

Baseline Selection. We compare MRF with *Denil14* [12], BRF [13], CompRF [29], and the standard RF (*Breiman*) [1] in the following evaluations[12,13,29]. are recent works about the consistent RF. Note that except for the consistent variant (CompRF-C) in [29], we also evaluate its inconsistent version, denoted as CompRF-I. Although the Breiman's RF [1] cannot meet the consistency, we also provide its results as an important baseline for comparison.

Specifically, there are two main differences between MRF and Breiman's RF: **(1)** MRF uses the structure/estimation points partition for the training set while Breiman's RF adopts bootstrap strategy and **(2)** MRF uses the proposed Multinomial distributions splitting rule while Breiman's RF adopts deterministic greedy splitting rule. To analyze which component contributes more to the good performance of our MRF, we also compare Breiman's RF with the structure/estimation points partition (dubbed *Breiman*+SE) and with the multinomial distributions based splitting rule (dubbed *Breiman*+M).

Training Setup. We use the random generator implemented by NumPy [44] to ensure the diversity of each tree in all forests. Similar to the settings in [13], we carry out 10 times 10-fold cross validation to generate 100 forests for each method to alleviate the influence of randomness. All forests have $t = 100$ trees, minimum leaf size $k = 5$. Gini index is used as the impurity measure except for CompRF. In *Denil14*, BRF, CompRF, and RF, we set the size of the set of candidate features to $\sqrt{D}$. The partition rate of all consistent RF variants is set to 1. All settings stated above are based on [12,13]. In MRF, we set $B_1 = B_2 = 10$ and $B_3 \to \infty$ in all datasets, and other hyper-parameters of baseline methods are set according to their paper.

Results. As shown in Table 2, MRF significantly exceeds all existing consistent RF variants, including *Denil14*, BRF, and CompRF-C. For example, MRF achieves more than 2% improvement in most cases, compared with the state-of-the-art consistent method BRF. Compared to the CompRF-I which is the inconsistent version of CompRF-C, our method achieves higher accuracy on twenty-two datasets. Besides, the performance of the MRF even surpasses Breiman's original RF on fourteen datasets, where the advantage of the MRF is statistically significant on eleven. To the best of our knowledge, this has never been achieved by any other consistent RF variants. Moreover, our method achieves a high average ranking, which further verifies that the performance of MRF is comparable with the Breiman's RF. Note that we have not fine-tuned the hyper-parameters such as B_1, B_2, B_3 and t. The performance of the MRF might be further improved with the tuning of these parameters, which would bring additional computational complexity. Moreover, the performance of the *Breiman*+M is better than that of the *Breiman*+SE in most cases. It shows that our new splitting rule contributes more than the structure/estimation partition in improving the performance of MRF.

5.2. Performance in semantic segmentation

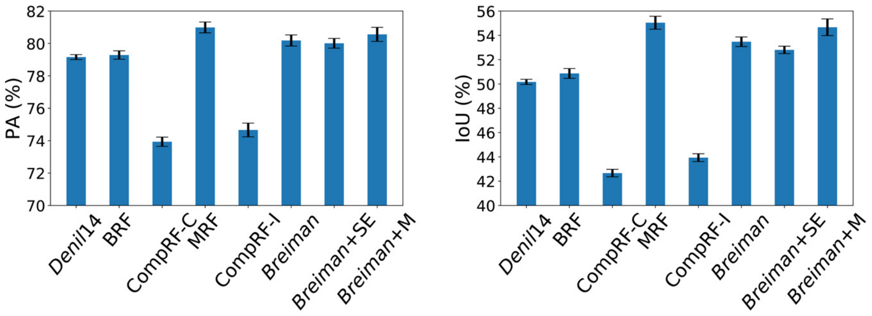
Task Description. We treat the segmentation as a pixel-wise classification and build the dataset based on aerial images[3]. Each pixel of these images are labeled for one of two semantic classes:

[3] https://github.com/dgriffiths3/ml_segmentation

Table 2
Accuracy (%) of different RFs on benchmark UCI datasets.

Dataset	*Denil14*	BRF	CompRF-C	MRF	CompRF-I	*Breiman*	*Breiman*+SE	*Breiman*+M
Zoo	80.00	85.00	87.69	**90.64**†	93.39	87.38	90.95	96.64
Hayes	50.93	45.35	45.82	**79.46**†	46.04	77.58	79.04	84.09
Echo	78.46	88.46	89.63	**91.72**†	88.09	90.64	91.97	90.52
Hepatitis	62.17	63.46	62.50	**64.32**	58.33	64.05	65.60	60.74
Wine	96.80	96.78	71.38	**97.58**	74.37	97.51	97.81	97.92
Wdbc	92.86	95.36	92.39	**95.78**	94.26	96.01	95.29	96.52
Transfusion	72.97	77.70	76.53	**78.53**	75.28	79.52•	78.41	74.51
Vehicle	68.81	71.67	59.68	**73.54**	64.86	74.70•	73.16	75.47
Tic-tac-toe	84.07	79.64	74.53	**98.01**†	77.81	87.97	97.76	98.92
Mammo	79.17	81.25	76.57	**81.86**	78.72	82.31•	82.06	80.02
Messidor	65.65	65.21	65.62	**67.14**	66.14	68.35•	66.90	68.46
Website	85.29	85.58	85.98	**89.80**†	88.34	88.12	90.03	90.04
Banknote	98.29	98.32	99.36	**99.49**†	99.02	99.12	99.02	99.67
Cmc	53.60	54.63	53.93	**56.12**†	54.61	55.11	55.97	53.42
Yeast	58.80	58.38	14.15	**61.03**	10.66	61.71	61.06	60.96
Car	88.02	93.43	79.07	**96.30**	92.17	97.42•	96.42	97.47
Image	95.45	96.06	93.99	**97.47**	96.16	97.71	97.07	98.31
Chess	61.32	97.12	94.77	**99.25**†	97.49	98.72	99.04	99.49
Ads	85.99	94.43	96.05	**96.76**	96.44	97.59•	96.84	97.94
Wilt	97.16	97.25	97.23	**98.56**	98.27	98.10	97.83	98.57
Wine-Quality	57.31	56.68	53.22	**60.56**	55.06	64.78•	61.15	69.81
Phishing	94.35	94.47	95.44	**96.07**†	96.45	95.56	96.21	97.19
Nursery	93.42	93.52	91.01	**99.28**†	95.67	96.89	98.61	99.78
Connect-4	66.19	76.75	72.82	**81.46**†	76.27	80.05	81.54	84.02
Average Rank	6.83	5.83	6.71	2.75	5.58	3.08	3.00	**2.21**

1. We carry out Wilcoxon's signed-rank test [43] to test for the difference between the results from the MRF and the standard RF at significance level 0.05. 2. Among the four consistent RF variants, the best result is indicated in boldface. 3. ”†” indicates MRF is significantly better than the standard RF. 4. ”•” indicates the standard RF is significantly better than MRF. 5. The last line shows the average rank of different methods across all datasets.



(a) PA (b) IOU

Fig. 3. Comparison of different RFs on the pixel-wise classification task with two metrics: the pixel-wise accuracy (PA) and the intersection over union (IOU). The higher the PA and IOU, the better the performance. The standard deviation is indicated by the error bar.

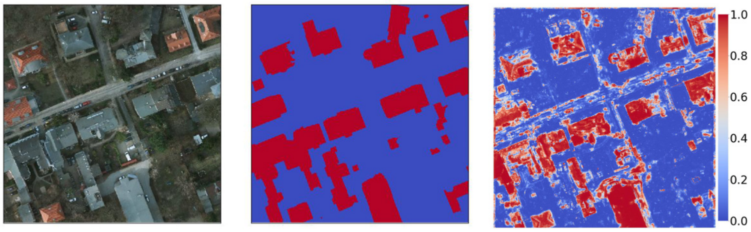
building or *not building*. Except for the RGB values of each pixel, we also construct some other widely used features. Specifically, we adopt local binary pattern [45] with radius 24 to characterize texture, and calculate eight Haralick features [46] (including angular second moment, contrast, correlation, entropy, homogeneity, mean, variance, and standard deviation). We sample 10,000 pixels without replacement for training, and test the performance on the test image. To reduce the effect of randomness, we repeat the experiments 5 times with different training set assignments. To evaluate the performance, we adopt two classical metrics, including the pixel-wise accuracy (PA) and intersection over union (IoU). Besides, all settings are the same as the descriptions in Section 5.1.

Results. As shown in Fig. 3, the performance of MRF is better than that of RF. Compare with existing consistent RF, the improvement of MRF is more significant. Besides, *Breiman*+M is better than *Breiman*+SE in terms of both PA and IOU, which further verify the effectiveness of the proposed splitting rule. We also visualize the segmentation results of MRF, as shown in Fig. 4. Although the performance of MRF may not be as good as some state-of-the-art deep learning based methods, it still achieves plausible results.

5.3. Performance of differential privacy

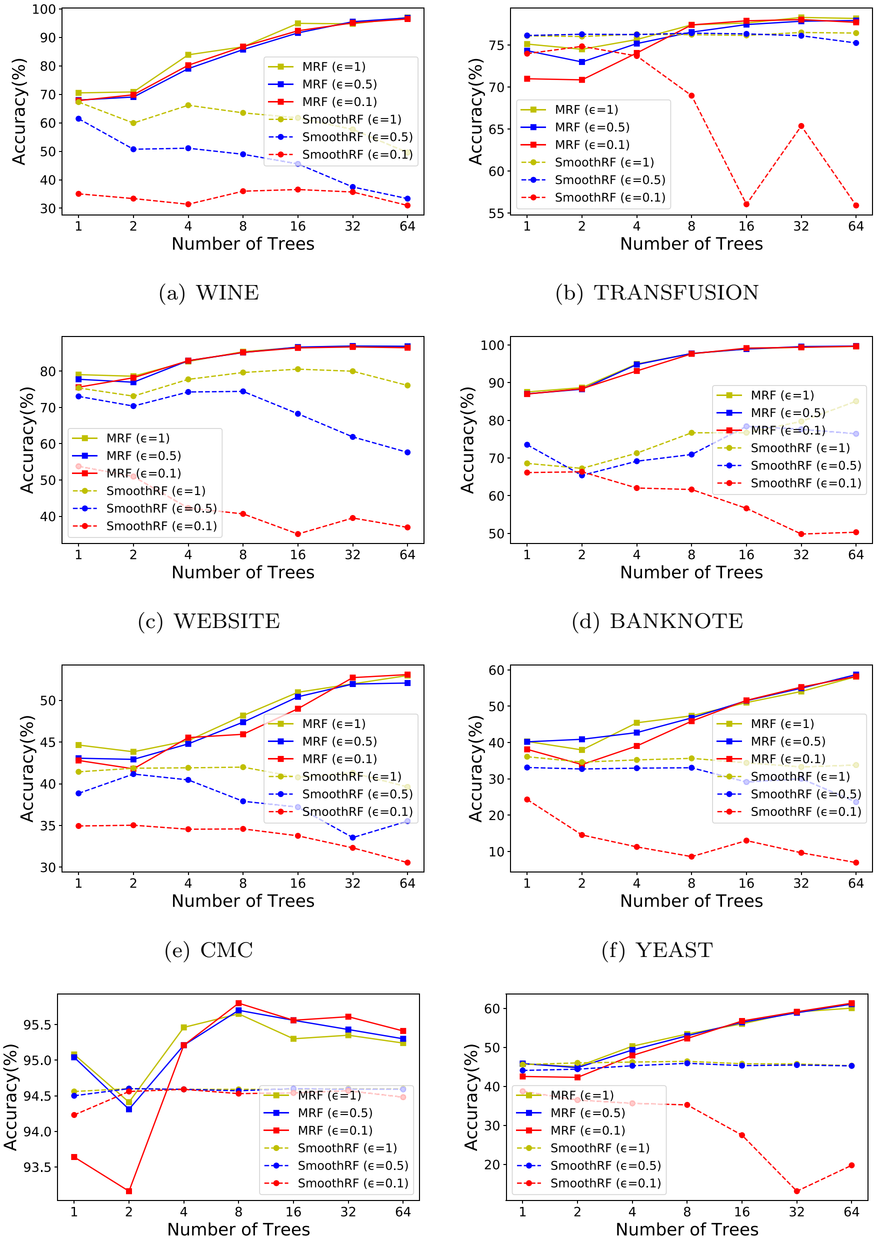
In this section, we investigate the performance of privacy-preservation for the proposed MRF. We conduct experiments to compare MRF with a recent RF variant [35] (denoted as SmoothRF), in terms of the privacy budget ϵ and the number of trees t, respectively. Specifically, for our MRF scheme, based on Theorem 2, given the fixed privacy budget ϵ, we set $B_1 = B_2 = \epsilon/2(d \cdot t)$ and $B_3 = \epsilon/t$, where t is the number of trees and d is the depth of a tree. Besides, we observe that the value of d constructed based on selected datasets is no more than 10, therefore we directly set $d = 10$ for simplicity. We implement the SmoothRF [35] based on the open-source code[4] with the default setting.

[4] https://github.com/sam-fletcher/Smooth_Random_Trees



(a) (b) (c)

Fig. 4. Visualization result of the proposed MRF. **(a)**: Aerial image; **(b)**: Groud-truth; **(c)**: The heat map of the prediction. The pixel is predicted as within the building area if and only if its color is red in the heat map. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) WINE (b) TRANSFUSION

(c) WEBSITE (d) BANKNOTE

(e) CMC (f) YEAST

(g) WILT (h) WINE-QUALITY

Fig. 5. Comparison between MRF and SmoothRF with different privacy budgets ϵ and the number of trees on eight datasets.

Table 3
Accuracy (%) of compared advanced RFs and our MRF on UCI datasets.

Dataset	RoF-PCA	RoF-LDA	MPSVM-T	MPSVM-P	MPSVM-N	HobRaF	MRF
Zoo	89.58	$88.10^{\bullet}$	$82.38^{\bullet}$	$88.69^{\bullet}$	$82.61^{\bullet}$	$95.35^{\dagger}$	90.64
Hayes	$70.79^{\bullet}$	$72.29^{\bullet}$	$65.64^{\bullet}$	$65.02^{\bullet}$	$60.77^{\bullet}$	79.27	79.46
Echo	92.15	91.16	$84.72^{\bullet}$	91.46	$85.71^{\bullet}$	$90.45^{\bullet}$	91.72
Hepatitis	$65.23^{\dagger}$	65.83	63.40	63.87	64.53	63.95	64.32
Wine	$96.21^{\bullet}$	$98.15^{\dagger}$	97.47	97.58	$96.23^{\bullet}$	$98.41^{\dagger}$	97.58
Wdbc	95.51	$97.20^{\dagger}$	$96.87^{\dagger}$	96.59	$97.14^{\dagger}$	$96.99^{\dagger}$	95.78
Transfusion	77.95	78.39	78.64	78.88	$77.22^{\bullet}$	$76.70^{\bullet}$	78.53
Vehicle	$76.04^{\dagger}$	$78.77^{\dagger}$	$75.76^{\dagger}$	$75.10^{\dagger}$	72.69	$79.15^{\dagger}$	73.54
Tic-tac-toe	$86.35^{\bullet}$	$85.34^{\bullet}$	$83.49^{\bullet}$	$84.81^{\bullet}$	$81.34^{\bullet}$	$92.01^{\bullet}$	98.01
Mammo	$82.80^{\dagger}$	$82.56^{\dagger}$	$82.48^{\dagger}$	$82.52^{\dagger}$	$82.65^{\dagger}$	81.48	81.86
Messidor	$74.95^{\dagger}$	$74.74^{\dagger}$	$73.70^{\dagger}$	$72.36^{\dagger}$	$73.26^{\dagger}$	$74.50^{\dagger}$	67.14
Website	$87.69^{\bullet}$	$88.18^{\bullet}$	$86.01^{\bullet}$	$87.23^{\bullet}$	$85.22^{\bullet}$	89.49	89.80
Banknote	99.97	99.84	99.91	99.90	99.93	99.91	99.49
Cmc	$55.14^{\bullet}$	$54.73^{\bullet}$	$54.52^{\bullet}$	$54.49^{\bullet}$	$51.97^{\bullet}$	$53.72^{\bullet}$	56.12
Yeast	61.20	61.32	$55.35^{\bullet}$	61.43	$41.88^{\bullet}$	$62.05^{\dagger}$	61.03
Car	$95.69^{\bullet}$	96.00	$91.37^{\bullet}$	$95.35^{\bullet}$	$86.42^{\bullet}$	$97.86^{\dagger}$	96.30
Image	97.26	97.34	$96.46^{\bullet}$	97.10	$93.68^{\bullet}$	$98.12^{\dagger}$	97.47
Chess	$98.10^{\bullet}$	$98.22^{\bullet}$	$93.82^{\bullet}$	$97.75^{\bullet}$	$82.38^{\bullet}$	98.94	99.25
Ads	97.13	96.74	$89.13^{\bullet}$	$97.33^{\dagger}$	$86.60^{\bullet}$	$97.45^{\dagger}$	96.76
Wilt	98.56	98.61	98.21	$98.11^{\bullet}$	98.24	98.17	98.56
Wine-Quality	$64.19^{\dagger}$	$64.26^{\dagger}$	59.30	$63.82^{\dagger}$	$44.88^{\bullet}$	$71.27^{\dagger}$	60.56
Phishing	95.81	95.60	$95.04^{\bullet}$	95.42	$94.54^{\bullet}$	$96.74^{\dagger}$	96.07
Nursery	$97.13^{\bullet}$	$96.35^{\bullet}$	$94.26^{\bullet}$	$97.01^{\bullet}$	$92.09^{\bullet}$	98.50	99.28
Connect-4	$77.56^{\bullet}$	$76.83^{\bullet}$	$65.83^{\bullet}$	$76.73^{\bullet}$	$65.83^{\bullet}$	$87.40^{\dagger}$	81.46
Average Rank	3.21	3.17	5.33	4.46	5.83	2.75	3.25

1. We carry out Wilcoxon's signed-rank test [43] to test for the difference between the results from the MRF and other RF variants at significance level 0.05. 2. "•" indicates MRF is significantly better than this method. 3. "†" indicates this method is significantly better than MRF. 4. The last line shows the average rank of different methods across all datasets.

To show the performance of both methods, we set $\epsilon \in \{0.1, 0.5, 1\}$ and $t \in \{1, 2, 4, 8, 16, 32, 64\}$, respectively. As shown in Fig. 5, the MRF can achieve better performance compared with the SmoothRF in terms of different ϵ and t. Specifically, when the privacy budget ϵ is relatively small, the added noise is relatively high which results in performance degradation. In contrast, when ϵ is relatively big, the added noise is relatively low, and thus theperformance will increase. Besides, when the number of trees t increases, the performance of MRF increases significantly, while the performance of SmoothRF decreases significantly.

5.4. Comparison with advanced random forest variants

Besides the comparison with the consistent and privacy-preservation RFs, we also conduct experiments to compare the MRF with some advanced RFs, including rotation random forests [18] and oblique random forests [21,22]. Specifically, there are two rotation RFs (*i.e.*, RoF-PCA and RoF-LDA) with different transformations, including principal component analysis (PCA) and linear discriminate analysis (LDA). We present the results of three multi-surface proximal support vector machine (MPSVM) based oblique random forests[21], *i.e.*, MPSVM-based RFs with Tikhonov regularization (MPSVM-T), MPSVM-based RFs with axis-parallel regularization (MPSVM-P), and MPSVM-based RFs with NULL space regularization (MPSVM-N). We implement the above methods using the open-source code[5]. We also compare with a recent method that employs linear classifiers at each non-leaf node, namely, heterogeneous oblique random forest (HobRaF) [22]. HobRaF is implemented based on its open-sourced codes[6]. Other settings are the same as those used in Section 5.1.

As shown in Table 3, HobRaF achieves remarkable performance with the best average rank. Nevertheless, the performance of MRF is still on par with that of the HobRaF. Besides, MRF has theoretical consistency and privacy-preservation property, which is the main merit of the proposed method.

5.5. Discussions

5.5.1. Results of RFs under optimal hyper-parameters

To keep in line with previous works [1,12,13], we compare MRF with all baseline methods under their default settings. However, these parameters may not be optimal for each dataset. To provide a more comprehensive comparison, we select some representative datasets and tune the parameters of all methods to achieve their optimal performance in each dataset based on the grid-search.

As shown in Table 4–5, all methods achieve better performance than training without tuning the hyper-parameters, especially advanced RFs evaluated in Table 5. In particular, MRF is still better than all other consistent RF variants and its performance is on par with Breiman's RF and other advanced RF variants under the use of optimal hyper-parameters. These results verify the effectiveness of MRF again.

5.5.2. The effect of B_1 and B_2

We evaluate the performance of the consistent MRF under different hyper-parameters B_1 and B_2. Specifically, we consider a range of [0,20] for both B_1 and B_2, and other hyper-parameters are the same as those stated in Section 5.1.

Fig. 6 displays the results for six datasets representing small, medium and large datasets. It shows that the performance of MRF is significantly improved as B_2 increases from zero, and it further becomes relatively stable when $B_2 \geq 10$. Similarly, the performance also improves as B_1 increases from zero, but the effect is not obvious. When B_2 is too small, the resulting multinomial distributions would allow too much randomness, leading to the poor performance of the MRF. Besides, as shown in the figure, although the optimal values of B_1 and B_2 may depend on the specific characteristics of a dataset, such as the outcome scale and the dimension of

[5] https://drive.google.com/file/d/0B9nwWnaaZcNWZGFuRnZYTlR2LVk/view?usp=sharing

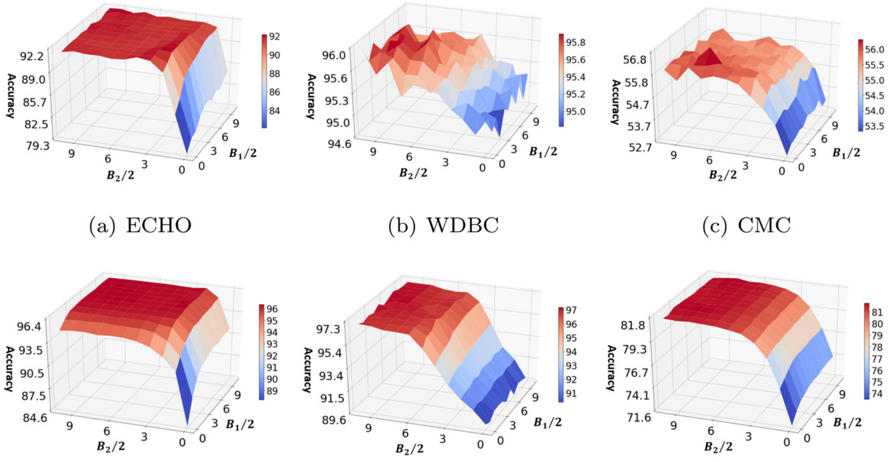
[6] https://github.com/P-N-Suganthan/CODES/blob/master/2020-PRJ-Het-ob-RaF.zip

Table 4
Accuracy (%) of consistent RFs and Breiman's RF with optimal hyper-parameters.

Dataset	*Denil14*	BRF	CompRF-C	MRF	CompRF-I	*Breiman*	*Breiman*+SE	*Breiman*+M
Hayes	66.62	62.52	45.82	82.28	46.86	80.53	80.12	85.70
Echo	91.66	91.75	89.93	92.11	90.17	92.14	92.00	91.90
Hepatitis	64.03	65.66	62.50	66.72	60.26	65.06	67.77	64.77
Vehicle	72.96	72.34	62.57	73.75	64.94	74.97	73.45	75.93
Yeast	59.84	59.09	16.17	61.79	11.16	62.03	61.81	62.60
Wilt	97.72	97.66	97.76	98.56	98.43	98.30	97.90	98.83

Table 5
Accuracy (%) of advanced RFs and our MRF with optimal hyper-parameters.

Dataset	RoF-PCA	RoF-LDA	MPSVM-T	MPSVM-P	MPSVM-N	HobRaF	MRF
Hayes	80.92	80.62	74.46	79.38	70.77	80.31	82.28
Echo	92.92	92.00	91.08	92.31	90.77	91.38	92.11
Hepatitis	68.27	67.47	65.87	66.67	65.60	64.00	66.72
Vehicle	81.08	84.92	81.88	78.70	74.60	83.47	73.75
Yeast	64.19	65.47	60.59	64.64	57.88	62.39	61.79
Wilt	98.76	98.80	98.83	98.80	98.69	98.71	98.56



(a) ECHO (b) WDBC (c) CMC

(d) CAR (e) ADS (f) CONNECT-4

Fig. 6. Accuracy (%) of the MRF under different hyper-parameter values.

the impurity decrease vector, at our default setting ($B_1 = B_2 = 10$), the MRF achieves competitive performance in all datasets.

5.5.3. Computational complexity analysis

For a given dataset, we suppose that its feature dimension is D and it consists of n samples. In the best case the depth would be $\mathcal{O}(\log n)$ for a balanced binary tree, the theoretical computational complexity of constructing a tree would be $\mathcal{O}(D \cdot n \log n)$. Because the complexity of the RF is the summation of the complexities of constructing individual trees, the complexity of MRF with t trees should be $\mathcal{O}(t \cdot D \cdot n \log n)$. Note that this analysis ignores the computational time involved in selecting the splitting feature and value for each node.

6. Conclusion and future work

In this paper, we proposed a novel random forest framework, dubbed multinomial random forest (MRF), based on which we analyzed its consistency and privacy-preservation. In the MRF, we proposed two impurity-based multinomial distributions for the selection of splitting feature and splitting value, respectively. Accordingly, the best split point has the highest probability to be chosen, while other candidate split points that are nearly as good as the best one will also have a good chance to be selected. This split process is more reasonable compared with the greedy splitting criterion used in existing methods. Besides, we also introduced another exponential mechanism of differential privacy for selecting the label of a leaf to discuss the privacy-preservation of MRF. Experiments and comparisons demonstrated that the MRF surpassed existing consistent random forest variants, and its performance is on par with advanced random forests. It is by far the first random forest variant that is consistent and has comparable performance to the standard random forest simultaneously.

For future work, although our MRF achieves the best performance among all consistent RFs, it can not always surpass Breiman's RF. Therefore, how to obtain a significantly better RF in theory and practice is still an open problem. Besides, in this paper, the MRF is designed only for classification tasks. We will further explore how to extend the proposed MRF in regression tasks in our future work. Moreover, the proposed multinomial distribution based randomization strategy is only explored in RFs. We will explore whether it is also effective in improving other algorithms with the deterministic selection process.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The publication of this work go through a very long journey. This work is supported in part by the National Natural Science Foundation of China under Grant 62171248, the China Postdoctoral Science Foundation under Grant 2020M670374, the Guangdong Province Key Area R&D Program under grant 2018B010113001, the Guangdong Basic and Applied Basic Research Foundation under Grant 2019A1515110644, the Natural Science Foundation of Zhejiang Province under Grant LSY19A010002, the R&D Program of Shenzhen under Grant JCYJ20180508152204044. We are also grateful to professor Chun Li, from the Case Western Reserve University, for his helpful comments on an early draft of this paper.

Appendix A. The Proof of Lemma 3

Proof. Recall that the normalized impurity decrease vector $\hat{\boldsymbol{I}} \in [0,1]^D$. When $\hat{\boldsymbol{I}} = (1, 0, \cdots, 0)$, the probability that the first feature is selected for splitting is the largest, and when $\hat{\boldsymbol{I}} = (0, 1, \cdots, 1)$, the probability reaches smallest. Therefore

$$P_1 \triangleq \frac{1}{1+(D-1)e^{B_1}} \le \Pr(v \in A) \le \frac{e^{B_1}}{e^{B_1}+(D-1)}.$$

□

Appendix B. The Proof of Lemma 4

Proof. Suppose m is the number of possible splitting values of feature A, similar to Lemma 3, the probability that a value is selected for splitting satisfies the following restriction:

$$\frac{1}{1+(m-1)e^{B_2}} \le \Pr(v) \le \frac{e^{B_2}}{e^{B_2}+(m-1)}. \tag{B.1}$$

In this case,

$$\begin{aligned} &\Pr\left(v \in \textstyle\bigcup_{i=2}^{N-1} A^{(i)} | A\right) = \frac{\int_{\bigcup_{i=2}^{N-1} A^{(i)}} f(v)dv}{\int_A f(v)dv} \\ &\ge \lim_{m\to+\infty} \left(\frac{\int_{\bigcup_{i=2}^{N-1} A^{(i)}} \frac{1}{1+(m-1)e^{B_2}} dv}{\int_A \frac{e^{B_2}}{e^{B_2}+(m-1)} dv} \right) \\ &= \lim_{m\to+\infty} \frac{N-2}{N} \cdot \frac{e^{B_2}+(m-1)}{e^{B_2}+(m-1)e^{2B_2}} \\ &= \tfrac{N-2}{N} e^{-2B_2} \triangleq P_2. \end{aligned} \tag{B.2}$$

□

Appendix C. The Proof of Theorem 1

Proof. When $B_3 \to \infty$, the prediction in each node is based on majority vote, therefore it meets the prerequisite of Lemma 2. Accordingly, we can prove the consistency of MRF by showing that it meets two requirements in Lemma 2.

Firstly, since MRF requires $|\mathcal{N}^E(\boldsymbol{X})| \ge k$ where $k \to \infty$ as $n \to \infty$, $|\mathcal{N}^E(\boldsymbol{X})| \to \infty$ when $n \to \infty$ is trivial.

Let $V_m(a)$ denote the size of the a-th feature of $\mathcal{N}_m(\boldsymbol{X})$, where $\boldsymbol{X}$ falls into the node $\mathcal{N}_m(\boldsymbol{X})$ at m-th layer. To prove $diam(\mathcal{N}(\boldsymbol{X})) \to 0$ in probability, we only need to show that $\mathbb{E}(V_m(a)) \to 0$ for all $A_a \in \mathcal{A}$. For a given feature A_a, let $V_m^*(a)$ denote the largest size of this feature among all children of node $\mathcal{N}_{m-1}(\boldsymbol{X})$. By Lemma 4, we can obtain

$$\mathbb{E}(V_m^*(a)) \le (1-P_2)V_{m-1}(a) + P_2 \frac{N-1}{N} V_{m-1}(a) = \left(1 - \frac{1}{N}P_2\right) V_{m-1}(a). \tag{C.1}$$

By Lemma 3, we can know

$$\mathbb{E}(V_m(a)) \le (1-P_1)V_{m-1}(a) + P_1 \mathbb{E}(V_m^*(a)) = \left(1 - \frac{1}{N}P_1P_2\right) V_{m-1}(a). \tag{C.2}$$

Since $V_0(a) = 1$,

$$\mathbb{E}(V_m(a)) \le \left(1 - \frac{1}{N}P_1P_2\right)^m. \tag{C.3}$$

Unlike the deterministic rule in the *Breiman*, the splitting point rule in our proposed MRF has randomness, therefore the final selected splitting point can be regarded as a random variable $W_i (i \in \{1, \cdots, m\})$, whose cumulative distribution function is denoted by F_{W_i}.

Let $M_1 = \min(W_1, 1-W_1)$ denotes the size of the root smallest child, we have

$$\Pr(M_1 \ge \sigma^{1/m}) = \Pr(\sigma^{1/m} \le W_1 \le 1 - \sigma^{1/m}) = F_{W_1}(1-\sigma^{1/m}) - F_{W_1}(\sigma^{1/m}). \tag{C.4}$$

Without loss of generality, we normalize the values of all attributes to the range $[0,1]$ for each node, then after m splits, the smallest child at the m-th layer has the size at least σ with the probability at least

$$\prod_{i=1}^{m} \left(F_{W_i}(1-\sigma^{1/m}) - F_{W_i}(\sigma^{1/m}) \right). \tag{C.5}$$

Since F_{Wi} is right-continuous at 0 and left-continuous at 1, $\forall \alpha_1 > 0, \exists \sigma, \alpha > 0$ s.t.

$$\prod_{i=1}^{m} \left(F_{W_i}(1-\sigma^{1/m}) - F_{W_i}(\sigma^{1/m}) \right) > (1-\alpha_1)^m > 1-\alpha.$$

Since the distribution of $\boldsymbol{X}$ has a non-zero density, each node has a positive measure with respect to $\mu_{\boldsymbol{X}}$. Defining

$$p = \min_{\mathcal{N}:\text{a node at } m-\text{th level}} \mu_X(\mathcal{N}),$$

we know $p > 0$ since the minimum is over finitely many nodes and each node contains a set of positive measure.

Suppose the data set with size n, the number of data points falling in the node A, where A denotes the m-th level node with measure p, follows $\text{Binomial}(n, p)$. Note that this node A is the one containing the smallest expected number of samples. WLOG, considering the partition rate $= 1$, the expectation number of estimation points in A is $np/2$. From Chebyshevs inequality, we have

$$\begin{aligned} &\Pr\left(|\mathcal{N}^E(\boldsymbol{X})| < k\right) = \Pr\left(|\mathcal{N}^E(\boldsymbol{X})| - \tfrac{np}{2} < k - \tfrac{np}{2}\right) \\ &\le \Pr\left(\left||\mathcal{N}^E(\boldsymbol{X})| - \tfrac{np}{2}\right| > \left|k - \tfrac{np}{2}\right|\right) \le \tfrac{np(1-p)}{2|k - \frac{np}{2}|^2} \\ &= \tfrac{p(1-p)}{2n|\frac{k}{n} - \frac{p}{2}|^2}, \end{aligned} \tag{C.6}$$

where the first inequality holds since $k - \frac{np}{2}$ is negative as $n \to \infty$ and the second one is by Chebyshev's inequality.

Since the right hand side goes to zero as $n \to \infty$, the node contains at least k estimation points in probability. By the stopping condition, the tree will grow infinitely often in probability, *i.e.*,

$$m \to \infty. \tag{C.7}$$

By (C.3) and (C.7), the theorem is proved. □

Appendix D. The Proof of Lemma 5

Proof. As we all know, the softmax function is

$$f(\mathbf{x})_j = \frac{\exp\left(z_j\right)}{\sum_{i=1}^{D} \exp\left(z_i\right)}.$$

Obviously, the above formula is the same as the exponential mechanism (see the formula (2) in Definition 2). In what follows, we prove $\mathcal{M}(\phi)$ satisfies B_1-differential privacy.

For any two neighboring datasets $\mathcal{D}^S$ and $\mathcal{D}'^S$, and any selected feature $A \in \mathcal{A}$, we can obtain

$$\frac{\exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}^S, A)}{2}\right)}{\exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}'^S, A)}{2}\right)} = \exp\left(\frac{B_1\left(\hat{\boldsymbol{I}}(\mathcal{D}^S, A) - \hat{\boldsymbol{I}}(\mathcal{D}'^S, A)\right)}{2}\right) \leqslant \exp\left(\frac{B_1}{2}\right), \tag{D.1}$$

where the quality function $\hat{\boldsymbol{I}}(\mathcal{D}^S, A)$ represents the a-th item of the normalized feature vector $\hat{\boldsymbol{I}}$ based on the structure points dataset $\mathcal{D}^S$, and through the normalized operation (*i.e.*, $\triangle\hat{\boldsymbol{I}} = \max_{\forall A, \mathcal{D}^S, \mathcal{D}'^S} |\hat{\boldsymbol{I}}(\mathcal{D}^S, A) - \hat{\boldsymbol{I}}(\mathcal{D}'^S, A)| = 1$) the corresponding sensitivity is 1. Accordingly, the privacy of the split feature mechanism satisfies that for any output A of $\mathcal{M}(\phi)$, we can obtain

$$\begin{aligned}
\frac{\Pr[\mathcal{M}(\phi, \mathcal{D}^S) = A]}{\Pr[\mathcal{M}(\phi, \mathcal{D}'^S) = A]} &= \frac{\dfrac{\exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}^S, A)}{2}\right)}{\sum_{A' \in \mathcal{A}} \exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}^S, A')}{2}\right)}}{\dfrac{\exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}'^S, A)}{2}\right)}{\sum_{A' \in \mathcal{A}} \exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}'^S, A')}{2}\right)}} \\
&= \frac{\exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}^S, A)}{2}\right)}{\exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}'^S, A)}{2}\right)} \cdot \frac{\sum_{A' \in \mathcal{A}} \exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}'^S, A')}{2}\right)}{\sum_{A' \in \mathcal{A}} \exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}^S, A')}{2}\right)} \\
&\leqslant \exp\left(\frac{B_1}{2}\right) \cdot \left(\frac{\sum_{A' \in \mathcal{A}} \exp\left(\frac{B_1}{2}\right) \exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}^S, A')}{2}\right)}{\sum_{A' \in \mathcal{A}} \exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}^S, A')}{2}\right)}\right) \\
&\leqslant \exp\left(\frac{B_1}{2}\right) \cdot \exp\left(\frac{B_1}{2}\right) \left(\frac{\sum_{A' \in \mathcal{A}} \exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}^S, A')}{2}\right)}{\sum_{A' \in \mathcal{A}} \exp\left(\frac{B_1 \hat{\boldsymbol{I}}(\mathcal{D}^S, A')}{2}\right)}\right) \\
&= \exp\left(B_1\right).
\end{aligned}$$

Therefore, for each layer of a tree, the privacy budget consumed by the split mechanism of features is B_1. That is, $\mathcal{M}(\phi)$ satisfies B_1-differential privacy. □

Appendix E. The Proof of Lemma 6

Proof. Similar to the proof of Lemma 5, the split value selection $\mathcal{M}(\varphi)$ is essentially the exponential mechanism of differential privacy. For any two neighboring datasets $\mathcal{D}^S$ and $\mathcal{D}'^S$, and any selected split value $a_j[i] \in a_j = \{a_j[1], \ldots, a_j[m]\}$ of the feature A_j, we can obtain

$$\frac{\exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}^S, a_j[i])}{2}\right)}{\exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}'^S, a_j[i])}{2}\right)} = \exp\left(\frac{B_2\left(\hat{\boldsymbol{I}}^{(j)}(\mathcal{D}^S, a_j[i]) - \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}'^S, a_j[i])\right)}{2}\right) \leqslant \exp\left(\frac{B_2}{2}\right),$$

where the quality function $\hat{\boldsymbol{I}}^{(j)}(\mathcal{D}^S, a_j[i])$ represents the i-th item of the normalized feature vector $\hat{\boldsymbol{I}}^{(j)}$ based on the structure points dataset $\mathcal{D}^S$, and the corresponding sensitivity is 1 through the normalized operation.

Accordingly, for any output $a_j[i]$ of $\mathcal{M}(\varphi)$, we can obtain

$$\begin{aligned}
\frac{\Pr\left[\mathcal{M}(\varphi, \mathcal{D}^S) = a_j[i]\right]}{\Pr\left[\mathcal{M}(\varphi, \mathcal{D}'^S) = a_j[i]\right]} &= \frac{\dfrac{\exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}^S, a_j[i])}{2}\right)}{\sum_{a_j[k] \in A_j} \exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}^S, a_j[k])}{2}\right)}}{\dfrac{\exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}'^S, a_j[i])}{2}\right)}{\sum_{a_j[k] \in A_j} \exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}'^S, a_j[k])}{2}\right)}} \\
&= \frac{\exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}^S, a_j[i])}{2}\right)}{\exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}'^S, a_j[i])}{2}\right)} \cdot \frac{\sum_{a_j[k] \in A_j} \exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}'^S, a_j[k])}{2}\right)}{\sum_{a_j[k] \in A_j} \exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}^S, a_j[k])}{2}\right)} \\
&\leqslant \exp\left(\frac{B_2}{2}\right) \cdot \left(\frac{\sum_{a_j[k] \in A_j} \exp\left(\frac{B_2}{2}\right) \exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}^S, a_j[k])}{2}\right)}{\sum_{a_j[k] \in A_j} \exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}^S, a_j[k])}{2}\right)}\right) \\
&\leqslant \exp\left(\frac{B_2}{2}\right) \exp\left(\frac{B_2}{2}\right) \left(\frac{\sum_{a_j[k] \in A_j} \exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}^S, a_j[k])}{2}\right)}{\sum_{a_j[k] \in A_j} \exp\left(\frac{B_2 \hat{\boldsymbol{I}}^{(j)}(\mathcal{D}^S, a_j[k])}{2}\right)}\right) \\
&= \exp\left(B_2\right).
\end{aligned}$$

Thus, the selection mechanism of split value for a specific feature satisfies B_2-differential privacy. □

Appendix F. The Proof of Lemma 7

Proof. For any two neighboring datasets $\mathcal{D}^E$ and $\mathcal{D}'^E$, and any label $c \in \mathcal{K} = \{1, 2, \ldots, K\}$ of a specific leaf, we can obtain

$$\frac{\exp\left(\frac{B_3 \eta(\mathcal{D}^E, c)}{2}\right)}{\exp\left(\frac{B_3 \eta(\mathcal{D}'^E, c)}{2}\right)} = \exp\left(\frac{B_3\left(\eta(\mathcal{D}^E, c) - \eta(\mathcal{D}'^E, c)\right)}{2}\right) \leqslant \exp\left(\frac{B_3}{2}\right), \tag{F.1}$$

where the quality function $\eta(\mathcal{D}^E, c)$ represents the empirical probability that the leaf has the label c, and thus the corresponding sensitive is 1. Then, for any output $c \in \{1, 2, \ldots, K\}$ of this leaf, we can obtain

$$\begin{aligned}
\frac{\Pr[h(\boldsymbol{x}, \mathcal{D}^E) = c]}{\Pr[h(\boldsymbol{x}, \mathcal{D}'^E) = c]} &= \frac{\dfrac{\exp\left(\frac{B_3 \eta(\mathcal{D}^E, c)}{2}\right)}{\sum_{c' \in \mathcal{K}} \exp\left(\frac{B_3 \eta(\mathcal{D}^E, c')}{2}\right)}}{\dfrac{\exp\left(\frac{B_3 \eta(\mathcal{D}'^E, c)}{2}\right)}{\sum_{c' \in \mathcal{K}} \exp\left(\frac{B_3 \eta(\mathcal{D}'^E, c')}{2}\right)}} \\
&= \frac{\exp\left(\frac{B_3 \eta(\mathcal{D}^E, c)}{2}\right)}{\exp\left(\frac{B_3 \eta(\mathcal{D}'^E, c)}{2}\right)} \cdot \frac{\sum_{c' \in \mathcal{K}} \exp\left(\frac{B_3 \eta(\mathcal{D}'^E, c')}{2}\right)}{\sum_{c' \in \mathcal{K}} \exp\left(\frac{B_3 \eta(\mathcal{D}^E, c')}{2}\right)} \\
&\leqslant \exp\left(\frac{B_3}{2}\right) \cdot \left(\frac{\sum_{c' \in \mathcal{K}} \exp\left(\frac{B_3}{2}\right) \exp\left(\frac{B_3 \eta(\mathcal{D}^E, c')}{2}\right)}{\sum_{c' \in \mathcal{K}} \exp\left(\frac{B_3 \eta(\mathcal{D}^E, c')}{2}\right)}\right) \\
&\leqslant \exp\left(\frac{B_3}{2}\right) \cdot \exp\left(\frac{B_3}{2}\right) \left(\frac{\sum_{c' \in \mathcal{K}} \exp\left(\frac{B_3 \eta(\mathcal{D}^E, c')}{2}\right)}{\sum_{c' \in \mathcal{K}} \exp\left(\frac{B_3 \eta(\mathcal{D}^E, c')}{2}\right)}\right) \\
&= \exp\left(B_3\right).
\end{aligned}$$

Since each leaf divides the dataset $\mathcal{D}^E$ into disjoint subsets, according to Property 2, the label selection mechanism for the leaf in each tree satisfies B_3-differential privacy. □

Appendix G. The Proof of Theorem 2

Proof. Based on Property 1 together with Lemma 5 and Lemma 6, the privacy budget consumed for each layer of a tree is $B_1 + B_2 = \epsilon/(d \cdot t)$. Since the depth of a tree is d, the total privacy budget consumed by the generation of tree structure is $d(B_1 + B_2) = \epsilon/t$. Since the datasets $\mathcal{D}^S$ and $\mathcal{D}^E$ are disjoint, according to Property 2, the total privacy budget of a tree is $\max\{d(B_1 + B_2), B_3\} = \epsilon/t$.

As a result, the consumed privacy budget of the MRF containing t trees is $\frac{\epsilon}{t} \cdot t = \epsilon$, which implies that the MRF satisfies ϵ-differential privacy. □

References

[1] L. Breiman, Random forests, Mach Learn 45 (1) (2001) 5–32.

[2] T.F. Cootes, M.C. Ionita, C. Lindner, P. Sauer, Robust and accurate shape model fitting using random forest regression voting, in: ECCV, Springer, 2012, pp. 278–291.

[3] P. Kontschieder, M. Fiterau, A. Criminisi, S. Rota Bulo, Deep neural decision forests, in: ICCV, 2015, pp. 1467–1475.

[4] J.F. Randrianasoa, P. Cettour-Janet, C. Kurtz, É. Desjardin, P. Gançarski, N. Bednarek, F. Rousseau, N. Passat, Supervised quality evaluation of binary partition trees for object segmentation, Pattern Recognit 111 (2021) 107667.

[5] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, R. Gavaldà, New ensemble methods for evolving data streams, in: ACM SIGKDD, 2009, pp. 139–148.

[6] C. Xiong, D. Johnson, R. Xu, J.J. Corso, Random forests for metric learning with implicit pairwise position dependence, in: ACM SIGKDD, 2012, pp. 958–966.

[7] Y. Li, J. Bai, J. Li, X. Yang, Y. Jiang, S.-T. Xia, Rectified decision trees: exploring the landscape of interpretable and effective machine learning, arXiv preprint arXiv:2008.09413 (2020).

[8] G. Biau, Analysis of a random forests model, Journal of Machine Learning Research 13 (Apr) (2012) 1063–1095.

[9] L. Breiman, Consistency for a simple model of random forests, Technical Report, Statistical Department, University of California at Berkeley, 2004.

[10] G. Biau, L. Devroye, G. Lugosi, Consistency of random forests and other averaging classifiers, Journal of Machine Learning Research 9 (Sep) (2008) 2015–2033.

[11] R. Genuer, Variance reduction in purely random forests, J Nonparametr Stat 24 (3) (2012) 543–562.

[12] M. Denil, D. Matheson, N. De Freitas, Narrowing the gap: Random forests in theory and in practice, in: ICML, 2014, pp. 665–673.

[13] Y. Wang, S.-T. Xia, Q. Tang, J. Wu, X. Zhu, A novel consistent random forest framework: bernoulli random forests, IEEE Trans Neural Netw Learn Syst 29 (8) (2017) 3510–3523.

[14] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans Pattern Anal Mach Intell 20 (8) (1998) 832–844.

[15] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, Mach Learn 40 (2) (2000) 139–157.

[16] L. Breiman, Classification and regression trees, Routledge, 2017.

[17] N. Meinshausen, Quantile regression forests, Journal of Machine Learning Research 7 (Jun) (2006) 983–999.

[18] L. Zhang, P.N. Suganthan, Random forests with ensemble of feature spaces, Pattern Recognit 47 (10) (2014) 3429–3437.

[19] Z.-H. Zhou, J. Feng, Deep forest: towards an alternative to deep neural networks, in: IJCAI, 2017, pp. 3553–3559.

[20] B.H. Menze, B.M. Kelm, D.N. Splitthoff, U. Koethe, F.A. Hamprecht, On oblique random forests, in: ECML-PKDD, 2011, pp. 453–469.

[21] L. Zhang, P.N. Suganthan, Oblique decision tree ensemble via multisurface proximal support vector machine, IEEE Trans Cybern 45 (10) (2014) 2165–2176.

[22] R. Katuwal, P.N. Suganthan, L. Zhang, Heterogeneous oblique random forest, Pattern Recognit 99 (2020) 107078.

[23] X. Qiu, L. Zhang, P.N. Suganthan, G.A. Amaratunga, Oblique random forest ensemble via least square estimation for time series forecasting, Inf Sci (Ny) 420 (2017) 249–262.

[24] L. Zhang, J. Varadarajan, P. Nagaratnam Suganthan, N. Ahuja, P. Moulin, Robust visual tracking using oblique random forests, in: CVPR, 2017, pp. 5589–5598.

[25] Y. Lin, Y. Jeon, Random forests and adaptive nearest neighbors, J Am Stat Assoc 101 (474) (2006) 578–590.

[26] H. Ishwaran, U.B. Kogalur, Consistency of random survival forests, Statistics & probability letters 80 (13–14) (2010) 1056–1064.

[27] M. Denil, D. Matheson, N. Freitas, Consistency of online random forests, in: ICML, 2013, pp. 1256–1264.

[28] S. Athey, J. Tibshirani, S. Wager, et al., Generalized random forests, The Annals of Statistics 47 (2) (2019) 1148–1178.

[29] S. Haghiri, D. Garreau, U. Luxburg, Comparison-based random forests, in: ICML, 2018, pp. 1871–1880.

[30] W. Gao, Z.-H. Zhou, Towards convergence rate analysis of random forests for classification, NeurIPS, 2020.

[31] L. Sweeney, Achieving k-anonymity privacy protection using generalization and suppression, Int. J. Uncertain. Fuzziness Knowl. Based Syst. 10 (5) (2002) 571–588.

[32] A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, *L*-diversity: privacy beyond *k*-anonymity, ACM Trans Knowl Discov Data 1 (1) (2007) 3.

[33] C. Dwork, Differential privacy, in: ICALP, 2006, pp. 1–12.

[34] A. Patil, S. Singh, Differential private random forest, in: ICACCI, 2014, pp. 2623–2630.

[35] S. Fletcher, M.Z. Islam, Differentially private random decision forests using smooth sensitivity, Expert Syst Appl 78 (2017) 16–31.

[36] Z. Guan, X. Sun, L. Shi, L. Wu, X. Du, A differentially private greedy decision forest classification algorithm with high utility, Computers & Security (2020) 101930.

[37] S. Rana, S.K. Gupta, S. Venkatesh, Differentially private random forest with high utility, in: ICDM, 2015, pp. 955–960.

[38] G. Jagannathan, K. Pillaipakkamnatt, R.N. Wright, A practical differentially private random decision tree classifier, Trans Data Priv 5 (1) (2012) 273–295.

[39] C. Dwork, F. McSherry, K. Nissim, A.D. Smith, Calibrating noise to sensitivity in private data analysis, in: TCC, 2006, pp. 265–284.

[40] F. McSherry, K. Talwar, Mechanism design via differential privacy, in: FOCS, 2007, pp. 94–103.

[41] L. Devroye, L. Györfi, G. Lugosi, A probabilistic theory of pattern recognition, volume 31, Springer Science & Business Media, 2013.

[42] F. McSherry, Privacy integrated queries: an extensible platform for privacy-preserving data analysis, Commun ACM 53 (9) (2010) 89–97.

[43] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine learning research 7 (Jan) (2006) 1–30.

[44] T.E. Oliphant, Python for scientific computing, Computing in Science & Engineering 9 (3) (2007) 10–20.

[45] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Trans Pattern Anal Mach Intell 28 (12) (2006) 2037–2041.

[46] R.M. Haralick, I. Dinstein, K. Shanmugam, Textural features for image classification, IEEE Transactions on Systems Man and Cybernetics 3 (6) (1973) 610–621.

Jiawang Bai is currently a Ph.D. student from the Department of Computer Science and Technology, Tsinghua University. Before that, he received his bachelor's degree from Jilin University in 2019. His research interest generally includes ensemble learning, computer vision, and adversarial machine learning.

Yiming Li is currently a Ph.D. candidate from the Tsinghua-Berkeley Shenzhen Institute, Tsinghua Shenzhen International Graduate School, Tsinghua University. Before that, he received his B.S. degree in Mathematics and Applied Mathematics from Ningbo University in 2018. His research interests are in the domain of AI security, especially backdoor learning, adversarial learning, and data privacy. He is the senior program committee member of AAAI'22 and the reviewer of TDSC, TCSVT, TII, etc.

Dr. Jiawei Li received his Ph.D. digree from the Department of Computer Science and Technology, Tsinghua University, in 2021. Before that, he received his bachelor's degree from Chongqing University in 2015. His research interest generally includes unsupervised learning, ensemble deep learning, and the applications in computer vision.

Dr. Xue Yang received the Ph.D. degree in information and communication engineering from Southwest Jiaotong University, Chengdu, China, in 2019. She was a visiting student at the Faculty of Computer Science, University of New Brunswick, Canada, from 2017 to 2018. She is currently a postdoctoral fellow with the Tsinghua Shenzhen International Graduate School, Tsinghua University, China. Her research interests include big data security and privacy, applied cryptography and federated learning

Dr. Yong Jiang received his M.S. and Ph.D. degrees in computer science from Tsinghua University, China, in 1998 and 2002, respectively. Since 2002, he has been with the Graduate School of Shenzhen of Tsinghua University, Guangdong, China, where he is currently a full professor. His research interests include Internet architecture and its protocols, IP routing technology, machine learning, etc.

Dr. Shu-Tao Xia received the B.S. degree in mathematics and the Ph.D. degree in applied mathematics from Nankai University, Tianjin, China, in 1992 and 1997, respectively. Since January 2004, he has been with the Graduate School at Shenzhen of Tsinghua University, Guangdong, China. He is now a full professor there. From March 1997 to April 1999, he was with the research group of information theory, Department of Mathematics, Nankai University, Tianjin, China. From September 1997 to March 1998 and from August to September 1998, he visited the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. His current research interests include coding and information theory, networking, machine learning, and deep learning. His researches have been published in multiple top-tier conferences and journals, including TIP, TNNLS, CVPR, ICCV, ECCV, ICLR, etc.