软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007332] [CSTR: 32375.14.jos.007332]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

eDPRF: 高效的差分隐私随机森林训练算法*

王树兰[1], 邱 瑶[1], 赵陈斌[2], 邹家须[1], 王彩芬[1]

[1](深圳技术大学 大数据与互联网学院, 广东 深圳 518118)

[2](空天信息安全与可信计算教育部重点实验室 (武汉大学 国家网络安全学院), 湖北 武汉 430072)

通信作者: 赵陈斌, E-mail: chenbinzhao96@whu.edu.cn

摘 要: 差分隐私凭借其强大的隐私保护能力被应用在随机森林算法解决其中的隐私泄露问题, 然而, 直接将差分隐私应用在随机森林算法会使模型的分类准确率严重下降. 为了平衡隐私保护和模型准确性之间的矛盾, 提出了一种高效的差分隐私随机森林训练算法 eDPRF (efficient differential privacy random forest). 具体而言, 该算法设计了决策树构建方法, 通过引入重排翻转机制高效地查询输出优势, 进一步设计相应的效用函数实现分裂特征以及标签的精准输出, 有效改善树模型在扰动情况下对于数据信息的学习能力. 同时基于组合定理设计了隐私预算分配的策略, 通过不放回抽样获得训练子集以及差异化调整内部预算的方式提高树节点的查询预算. 最后, 通过理论分析以及实验评估, 表明算法在给定相同隐私预算的情况下, 模型的分类准确度优于同类算法.

关键词: 随机森林; 差分隐私; 隐私预算; 重排翻转; 扰动方式

中图法分类号: TP18

中文引用格式: 王树兰, 邱瑶, 赵陈斌, 邹家须, 王彩芬. eDPRF: 高效的差分隐私随机森林训练算法. 软件学报. http://www.jos.org.cn/1000-9825/7332.htm

英文引用格式: Wang SL, Qiu Y, Zhao CB, Zou JX, Wang CF. eDPRF: Efficient Differential Privacy Random Forest Training Algorithm. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7332.htm

eDPRF: Efficient Differential Privacy Random Forest Training Algorithm

WANG Shu-Lan[1], QIU Yao[1], ZHAO Chen-Bin[2], ZOU Jia-Xu[1], WANG Cai-Fen[1]

[1](College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China)

[2](Key Laboratory of Aerospace Information Security and Trusted Computing of Ministry of Education (School of Cyber Science and Engineering, Wuhan University), Wuhan 430072, China)

Abstract: Differential privacy, owing to its strong privacy protection capacity, is applied to the random forest algorithm to address the privacy leakage problem. However, the direct application of differential privacy to the random forest algorithm leads to a significant decline in the model's classification accuracy. To balance the contradiction between privacy protection and model accuracy, this study proposes an efficient differential privacy random forest training algorithm, efficient differential privacy random forest (eDPRF). Specifically, the study designs a decision tree construction method based on the permute-and-flip mechanism. By introducing the efficient query output advantage of the permute and flip mechanism, the corresponding utility functions are further designed to achieve the precise output of split features and labels, effectively enhancing the learning ability of the tree model for data information under perturbation circumstances. At the same time, the study designs a privacy budget allocation strategy based on the composition theorem, which improves the privacy budget utilization rate of nodes by obtaining training subsets without replacement sampling and adjusting internal budgets through differentiation. Finally, through theoretical analysis and experimental evaluation, it is demonstrated that the proposed algorithm outperforms similar algorithms in terms of the model's classification accuracy when given the same privacy budget.

Key words: random forest; differential privacy; privacy budget; permute and flip; perturbation method

---

* 基金项目: 国家自然科学基金 (61702341); 深圳技术大学深圳市高等院校稳定支持项目 (SZWD2021012); 深圳技术大学研究生校企合作研究基金 (20223108010009)
本文由"新兴软件与系统的可信赖性与安全"专题特约编辑向剑文教授、陈厅教授、杨珉教授、周俊伟教授推荐.
收稿时间: 2024-07-10; 修改时间: 2024-10-15; 采用时间: 2024-11-25; jos 在线出版时间: 2024-12-10

随机森林[1]是由多棵决策树集成而来的模型, 其中, 每棵决策树使用不同的样本子集和特征子集构建. 在面对大规模数据时, 随机森林通过并行计算和随机抽样技术实现快速训练和预测, 具有较低的计算复杂度. 然而, 随机森林算法存在的隐私泄露风险却容易被人忽略. 例如, 在经典的成员推理攻击机制下[2−4], 攻击者可以访问随机森林输出的置信概率以推测输入样本是否参与过随机森林训练过程. 尽管一些研究通过修改置信度降低攻击成功率[5], 但随着攻击手段的升级, 攻击者可通过访问标签实现与访问置信概率同等有效的攻击效果[6]. 为了避免随机森林存在的隐私泄露风险, 一些学者将差分隐私应用在随机森林的训练过程中[7,8].

差分隐私是一种具有严格数学证明的隐私保护方法[9,10], 满足差分隐私的计算过程会返回合理的扰动输出, 无论数据集中是否存在攻击者想要获知的敏感信息, 计算过程的输出都不会产生统计差异. 即使攻击者具有很强的背景知识也很难通过分析输出结果对数据集中的隐私做出置信度高的推断, 极大地降低了攻击者预测个人敏感信息的可能性[11]. 因此, 差分隐私为解决随机森林中的数据隐私泄露这一科学难题提供了新的机遇. 然而, 差分隐私在为随机森林提供隐私保护的同时, 所带来的随机扰动不可避免地会对模型性能产生极大影响. 为此, 一些学者通过优化决策树内部的隐私预算分配方法改善随机森林模型的可用性[7,8]. 然而上述研究中随机森林模型在扰动情况下学习能力弱, 同时所提出的隐私预算分配方法无法适用于模型结构不平衡的实际场景, 仅仅采用逐层分配的方式使训练算法无法充分利用宝贵的隐私预算资源.

为克服上述不足, 本文提出一种高效的差分隐私随机森林训练算法 eDPRF (efficient differential privacy random forest), 有效缓解扰动情况下传统模型学习能力弱、解决隐私预算分配方法不均衡等现实问题, 增强模型可用性. 本文主要贡献总结如下.

(1) 提出一种新的决策树创建方法, 通过引入重排翻转机制[12]替代传统的指数机制和拉普拉斯机制辅助决策树的建立过程, 基于该机制优势为叶子节点和分支节点分别设计效用函数, 实现分裂特征和标签的精准输出, 有效改善树模型在扰动情况下对于数据信息的学习能力.

(2) 提出一种新的隐私预算分配策略, 基于并行组合定理[13]调整训练子集的划分方式, 有效提高单棵树的查询预算, 并结合决策树结构的差异性执行预算对齐操作, 不仅有效解决按层分配方式下浪费隐私预算的问题还提升了叶子节点的查询预算.

(3) 通过理论分析证明 eDPRF 算法满足 ε-差分隐私, 并与其他同类方法进行详细的实验分析对比, 证实 eDPRF 算法相比于同类算法, 可以在给定相同的隐私预算时有效提升随机森林模型的分类准确性.

1 相关工作

文献 [14,15] 最先引入差分隐私应用在随机森林中, 通过在随机森林分裂点的选择过程以及叶子节点的计数查询过程中应用指数机制与拉普拉斯机制避免隐私泄露, 然而当设定的隐私预算较小时, 算法中会引入过量的随机扰动, 极其影响模型性能. Zhang 等人[16]设计了一种新的特征度量标准, 在将 CART 与指数机制结合后对具有连续特征的数据集进行离散化提升随机森林模型可用性, 然而该研究并未给出隐私保护需要的敏感度, 并且在实验中只用到两个特征, 无法充分说明其改进效果. Wang 等人[17]根据输入特征与模型输出之间的相关性, 将输入特征划分为具有不同相关性的不同区域实现差异化的噪声输入, 然而忽略了预处理阶段的隐私泄露问题. Guan 等人[18]提出选择性聚合方法对随机森林进行优化, 但选择学习器的过程同样存在泄露隐私的风险. 并且, 上述学者在研究中使用的局部敏感度[19]甚至不满足差分隐私定义. Niu 等人[20]提出多群体量子遗传算法为决策树赋予权重以提升决策树的投票性能, 该方案同样忽略了权重本身会导致隐私泄露问题.

此外, 有学者通过优化隐私预算的分配方法来提升随机森林的性能. 例如, Wang 等人[21]利用随机森林 Bootstrap 抽样后的袋外数据权重为不同决策树分配差异化的隐私预算, 但未考虑权重本身导致的隐私泄露风险. Li 等人[22]在错误分类的样本数目上施加拉普拉斯噪声应对权重的隐私泄露问题, 然而, 该方案中正例或负例样本的计数信息同样泄露个人隐私, 并没有真正意义上实现差分隐私保护. 此外, 现有的大多数随机森林对不同层节点的预算分配策略采用均分的方式, 文献 [7,8,23] 指出, 这种方式会导致深层结点的信噪比不均衡. 为确保随机森林

模型效用性, 上述文献建议在按层分配的基础上为深层节点分配更多的隐私预算. 然而却忽略了模型结构对隐私预算分配的影响, 导致部分预算被闲置. 与此同时, 上述方案在决策树间采用的等分预算方式在森林规模较大时容易引入过多随机性影响模型性能.

综上所述, 上述研究并不能很好地平衡隐私保护性与模型可用性之间的关系. 为此, 本文提出一种高效且满足差分隐私的随机森林训练算法, 设计新的决策树创建方法以及隐私预算分配策略, 在提升节点的查询预算时进一步改善其在扰动下对数据信息的学习能力, 并通过隐私性分析证明本算法可以提供 ε-差分隐私保护, 实现了隐私保护与模型可用性之间的有效平衡.

2 基础知识

2.1 差分隐私

定义 1 (相邻数据集)[13]. 如果两个数据集 D_1 和 D_2 之间只相差一条数据, 则这两个数据集被称为相邻数据集.

定义 2 ((ε,δ) -差分隐私)[13]. 假设 D_1 和 D_2 是任意相邻数据集, S 是数据集上的一个随机函数. 如果函数 S 满足公式 (1), 则称函数 S 满足 (ε,δ) -差分隐私.

$$\Pr[S(D_1)=O] \leqslant e^{\varepsilon} \times \Pr[S(D_2)=O] + \delta \tag{1}$$

其中, O 是 S 在数据集上的任意可能的输出结果, ε 是用来量化隐私损失的隐私预算, 隐私预算 ε 越小则对隐私的保护程度越高. 隐私参数 δ 是一个松弛控制概率, 即存在大小为 δ 的概率, 数据会在没有任何保护措施的情况下被释放. 通常 δ 会设置成一个很小的数, 当 $\delta>0$ 时被称为松弛差分隐私, 当 $\delta=0$ 时, 被称为纯差分隐私, 纯差分隐私的保护强度要高于松弛差分隐私的保护强度. 如果函数 S 满足公式 (2), 则称函数 S 满足 ε-差分隐私.

$$\Pr[S(D_1)=O] \leqslant e^{\varepsilon} \times \Pr[S(D_2)=O] \tag{2}$$

定义 3 (全局敏感度)[13]. 给定函数 $f: D \to R^d$, f 的全局敏感度如公式 (3) 所示.

$$\Delta f = \max_{D_1,D_2} \| f(D_1) - f(D_2) \|_1 \tag{3}$$

其中, D_1 和 D_2 为任意的相邻数据集.

定义 4 (隐私保护机制). 拉普拉斯机制、指数机制以及重排翻转机制是实现差分隐私的 3 种机制, 下面给出相关概念.

(1) 拉普拉斯机制[24]: 拉普拉斯机制通过向原始数据的查询结果添加一些噪声以保护数据隐私.

对于任意数据集 D 以及查询算法 f, 若算法 S 的输出结果满足公式 (4), 则认为算法 S 满足 ε-差分隐私.

$$S(D) = f(D) + Lap\left(\frac{\Delta f}{\varepsilon}\right) \tag{4}$$

其中, $Lap(\Delta f/\varepsilon)$ 表示服从拉普拉斯分布的随机噪声, Δf 是算法 f 的全局敏感度, ε 是隐私预算.

拉普拉斯噪声的概率密度函数定义见公式 (5).

$$Lap(b) = \frac{1}{2b}\exp\left(-\frac{|x|}{b}\right) = \begin{cases} \dfrac{1}{2b}\exp\left(-\dfrac{x}{b}\right), x \geqslant 0 \\ \dfrac{1}{2b}\exp\left(\dfrac{x}{b}\right), x < 0 \end{cases} \tag{5}$$

(2) 指数机制[25]: 给定数据集 D, 对于范围 R 的任意对象 r, 效用函数 $u(D,r): D \times R \to R$ 都会输出对象 r 的效用分数. 若随机算法 M 以公式 (6) 的概率从 R 中选择并输出 r, 则算法 M 满足 ε-差分隐私.

$$\Pr[M=r] = \frac{\exp\left(\dfrac{\varepsilon u(D,r)}{2\Delta u}\right)}{\Omega} \tag{6}$$

其中, Ω 为指数机制的归一化因子, Δu 是函数 $u(D,r)$ 的全局敏感度.

(3) 重排翻转机制[12]: 假设数据集为 D, 由一组选项构成的集合为 Q, D 与 Q 的效用函数为 $u: D \times Q \to Q$. 对

于集合 Q 的任意选项 q, 函数 u 都会输出选项 q 的效用分数. 若算法 F 以算法 1 所示的形式获得集合 Q 中的选项 q, 则算法 F 满足 ε-差分隐私.

算法 1. 重排翻转算法.

输入: 数据集 D, 隐私预算 ε, 效用函数 u, 选项集合 Q;
输出: 选项.

函数: $PermuteAndFlip(D,\varepsilon,u,Q)$.
① $s_* = \max_{q\in Q} u(D,q)$;
② for q in $RandomPermutation(Q)$ do
③ $p_q \leftarrow \exp\left(\frac{\varepsilon}{2\Delta u}(u(D,q)-s_*)\right)$ // Δu 是 u 的全局敏感度;
④ if $Bernoulli(p_q)$ then
⑤ return q;
⑥ end if
⑦ end for

定理 1 (**串行组合定理**)[13]. 给定一个数据集 D, 如果 n 个随机函数 S_i 对于数据集 D 的访问都满足 ε_i-差分隐私, 其中 $1 \leqslant i \leqslant n$, 则 n 个函数 S_i 构成的组合函数对于数据集 D 的访问满足 $\sum \varepsilon_i$-差分隐私.

定理 2 (**并行组合定理**)[13]. 假如 n 个两两互不相交的数据集 D_i 共同构成数据集 D. 如果存在随机函数 S_i 对于数据集 D_i 的访问满足 ε_i-差分隐私, 其中 $1 \leqslant i \leqslant n$, 则 n 个函数 S_i 构成的组合函数对于数据集 D 的访问满足 $max(\varepsilon_i)$-差分隐私.

性质 1 (**后处理性**)[13]. 假设作用在数据集 D 上的随机机制 $G: D \to M$ 满足 (ε,δ)-差分隐私, 则对于随机机制 $f: M \to M'$, 有 $A(\circ) = f(G(\circ)): D \to M'$ 满足 (ε,δ)-差分隐私.

2.2 随机森林

随机森林采用 Bagging 集成策略以及自助采样方法 (Bootstrap) 对多棵决策树进行组合. 在差分隐私化的随机森林[7,8]中, 基尼指数是最常用的划分标准, 对应的数学定义如下所示.

定义 5 (**数据集的基尼指数**). 假设数据集 D 存在 c 个类别特征值, 则数据集 D 的基尼指数定义如公式 (7) 所示.

$$Gini(D) = \sum_{i=1}^{c}\sum_{i'\neq i} p_i p_{i'} = 1 - \sum_{i=1}^{c} p_i^2 \tag{7}$$

其中 p_i 代表每个类别特征值的概率. 基尼指数越小, 表明数据集的纯度越高.

定义 6 (**特征划分子集的基尼指数**). 对于特征 F_i, 当 F_i 有 v 种取值时, 经过 F_i 划分后的基尼指数定义如公式 (8) 所示.

$$Gini\_index(D) = \sum_{k=1}^{v} \frac{|D_k|}{|D|} Gini(D_k) \tag{8}$$

3 高效的差分隐私随机森林训练算法 eDPRF

本文提出一种高效的差分隐私随机森林训练算法 eDPRF, 其中包括: 隐私预算分配环节、决策树创建环节以及随机森林生成环节. 在设计过程中, 主要考虑以下原则: 1) 需要对训练数据隐私提供保护; 2) 尽可能提高随机森林模型的可用性. 因此, 在隐私预算分配环节中, 本文引入并行组合定理进行预算对齐操作, 提高每个树节点获得的查询预算. 在决策树的创建环节中, 本文提出 pfDPDT(differential privacy decision tree with permute and flip) 算法, 通过重排翻转机制优化节点查询过程, 增强每个树节点获取数据信息的能力. 在随机森林模型的生成环节中, 本文将隐私预算分配环节以及决策树的创建环节结合后, 获得差分隐私随机森林模型. 算法框架见图 1 所示.表 1 所示的是本算法使用的主要符号及其含义.

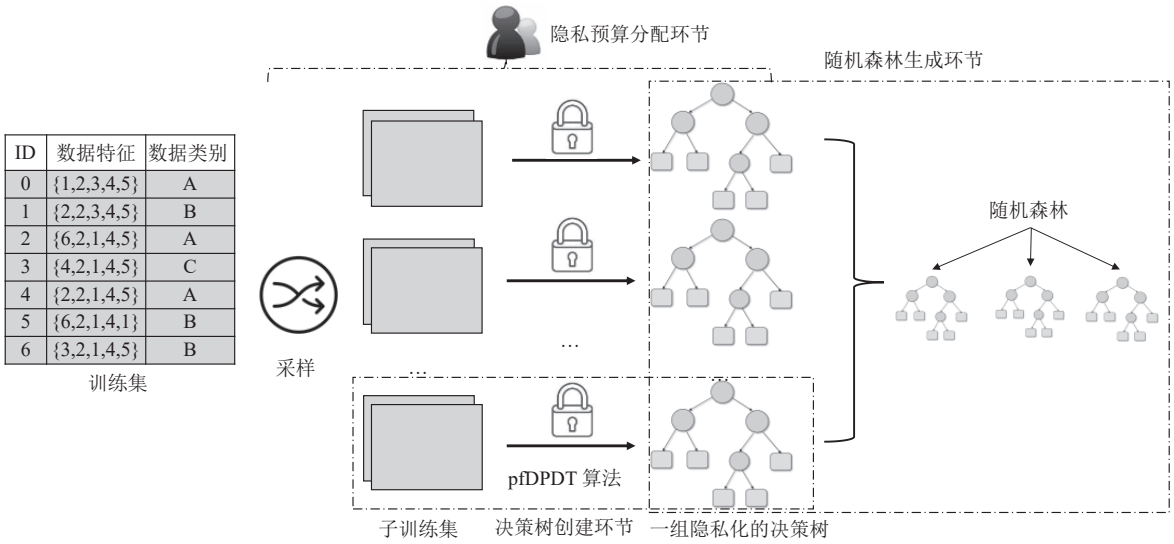


图 1 eDPRF 算法框架示意图

表 1 主要符号及其含义

符号	说明	符号	说明
ε	隐私预算	F	D的特征集
$treeDepth$	决策树的最大深度 (简称深度)	L_k	第 k 种标签
Δ	全局敏感度	N_k	D中标签为 L_k 的样本数量
T	随机森林的规模	F_a	第 a 种特征
D	训练集	$v_{F_a}^{w}$	F_a的第 w 种取值情况
L	D的标签集	$n_{L,L_k}^{F_a,w}$	D中 F_a 的取值为 $v_{F_a}^{w}$ 且标签为 L_k 的样本数量

3.1 隐私预算分配环节

如图 2 所示, 隐私预算分配环节分为两个部分: 1) 决策树之间的隐私预算分配, 2) 决策树内部的隐私预算分配, 具体过程描述如下:

1) 决策树之间的隐私预算分配. 首先, 定义如下 $Tree_1,Tree_2,\ldots,Tree_n$ 表示随机森林的不同决策树, $S_1,S_2,\ldots,S_n$ 表示每棵决策树的构建算法, $\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_n$ 表示分配给每棵决策树的隐私预算. 本文采用不放回抽样方式将数据集划分为 n 个训练子集 $D_1,D_2,\ldots,D_n$, 其中每个训练子集被单独分配给每棵决策树, 确保任意两棵决策树的训练集无交集.

本文进一步引入差分隐私的并行组合定理将随机森林中每棵决策树构建算法的隐私预算放大至整个随机森林算法的预算水平. 假设训练算法被分配的隐私预算为 ε, 随机森林的规模为 n, 完整的训练集为 D. 在获得每棵树的训练集 D_i ($i\in[1,n]$) 时, 采用不放回抽样的方式令 $D=D_1\cup\ldots D_i\cup D_n$, 则第 i 棵树的构建算法 S_i 获得的查询预算 $\varepsilon_i=\varepsilon$.

2) 决策树内部的隐私预算分配. $S_{i,1},S_{i,2},\ldots$ 表示第 i 棵决策树中每层节点的构建算法, $\varepsilon_{i,1},\varepsilon_{i,2},\ldots$ 表示分配给每层节点的隐私预算. 在为决策树内部节点分配隐私预算时, 初始阶段中同一层节点将会被分配相同的隐私预算. 随着节点层级增加, 预算呈现递增效果[8], 有助于缓解差分隐私扰动少数样本节点导致性能被严重破坏的问题. 同时为了解决该方法存在的隐私预算浪费问题, 在建树的过程中, 本文根据节点类型的差别, 执行 $S'_{i,1},S'_{i,2},S'_{i,3}$ 所示的预算对齐过程.

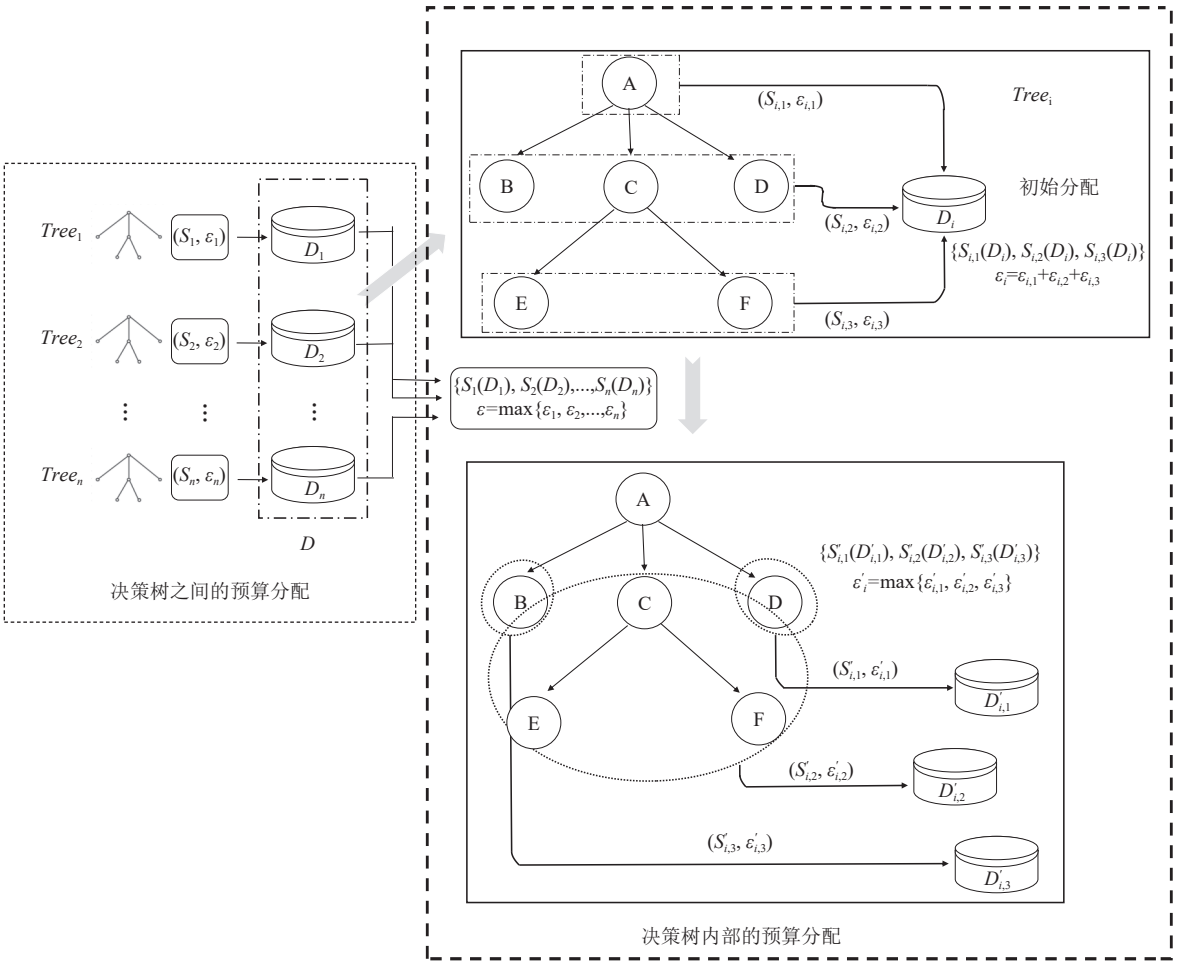


图 2 隐私预算分配策略示意图

假设第 i 棵决策树对应的构建算法 S_i 共获得隐私预算 ε_{tree}，且该树的最大深度为 $treeDepth$，则第 k 层节点分配的初始预算 $\varepsilon_{i,k}$ 可以表示为:

$$\varepsilon_{i,k}=\frac{\varepsilon_{tree}}{s_t}\times\frac{1}{treeDepth-k+1} \tag{9}$$

其中 $s_t=\frac{1}{treeDepth}+\frac{1}{treeDepth-1}+...+\frac{1}{2}+1$.

在建树过程中, 假设当前所遍历的节点处于决策树的第 l 层第 j 列, 则该节点实际获得的隐私预算 $\varphi(l,j)$ 为:

$$\varphi(l,j)=\begin{cases}\varepsilon_{i,l}, & node(l,j)\text{ 是分支}\\ \varepsilon_{tree}-\sum_{m=1}^{l-1}\varepsilon_{i,m}, & node(l,j)\text{ 是叶子}\end{cases} \tag{10}$$

如图 3 所示, 假设给予建树算法的隐私预算为 1, 模型的最大深度为 3, 则经过上述隐私预算分配过程后, 任何支路上获得的查询预算都为 1. 同时, 对齐操作使得叶子节点获得更多的查询预算, 降低噪声, 从而提高模型的决策能力.

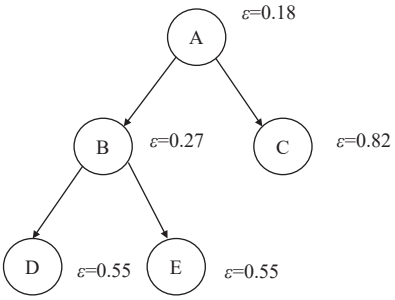


图 3　决策树内部隐私预算分配示例

3.2　决策树创建环节

如图 4 所示, 决策树创建环节主要包括两部分: 创建分支节点和创建叶子节点. 核心思路在于提出一种基于重排翻转机制[12]的差分隐私决策树算法 pfDPDT, 进一步优化了树节点的特征计算以及计数查询过程, 在提供隐私保护的同时增强算法的抗干扰能力, 改善了模型对数据的学习能力.

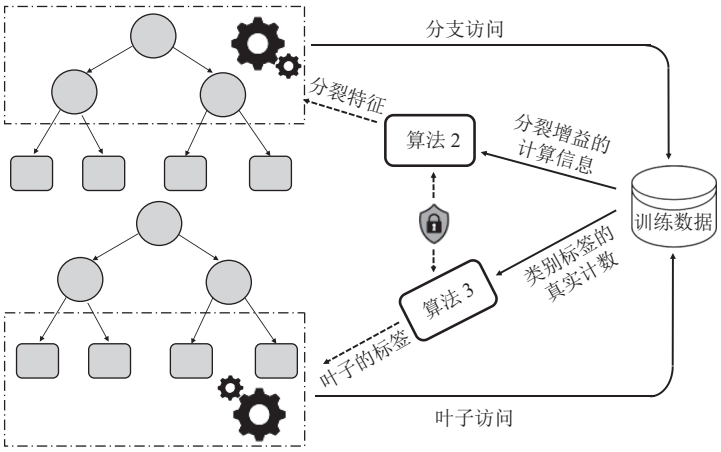


图 4　pfDPDT 算法流程

根据不同节点对于训练数据的访问形式不同, 本文分别设计了算法 2: 分裂特征的输出过程和算法 3: 标签的输出过程. 对于创建分支节点而言, 在分裂增益的相关信息流向待创建的分支节点之前, pfDPDT 算法利用算法 2 对该信息进行脱敏, 输出脱敏后的分裂特征. 这一分裂特征可以被用于构建分支节点, 将训练数据集划分为不同的子集. 对于创建叶子节点而言, 在类别标签的计数信息流向待创建的叶子节点之前, pfDPDT 算法利用算法 3 对信息进行脱敏, 输出脱敏后的标签, 该标签代表了决策树的最终预测结果, 并用于对新的未知数据进行分类.

3.2.1　创建分支节点

本文将基尼指数作为分支节点的特征选择依据, 使用基尼指数衡量节点纯度, 进而辅助分支节点确定最佳的划分方式, 提高子节点纯度, 从而获得表现效果好的决策树模型. 然而, 在确定分支节点的分裂特征时, 涉及对训练数据的查询操作, 存在个人隐私的泄露风险. 尽管传统算法基于指数机制[15]防止隐私泄露, 但是指数机制在分裂点的选择上会引入一些随机扰动, 导致构建出来的树模型精准度低. 相比之下, 本文采用的重排翻转机制, 具有比指数机制具有更小的预期误差, 从而为分支节点提供更准确的分裂点信息.

在重排翻转过程中, 首先为查询特征的操作设置一个恰当的效用函数, 该效用函数被用于后续的分数概率计算过程中. 基于基尼指数以及重排翻转机制的实施要求, 该效用函数的定义如下.

假设当前访问的训练集为 D, 则对特征集 F 执行查询的效用函数 $u_{spl}(D,F_a)$ 定义如公式 (11) 所示.

$$u_{spl}(D,F_a)=-\sum_{i=1}^{C}\left|v_{F_a}^{i}\right|\left(1-\sum_{k=1}^{K}\left(\frac{n_{L,L_k}^{F_a,i}}{\left|v_{F_a}^{i}\right|}\right)^2\right) \tag{11}$$

其中, F_a 表示 F 中任一特征, $\left|v_{F_a}^{i}\right|$ 表示 F_a 取值为 $v_{F_a}^{i}$ 的样本数量, C 表示 F_a 的所有取值情况数, K 表示 L 的所有标签种数.

在分支节点的特征选择过程中, 针对待选的特征集合, 根据公式 (11) 为集合中每个特征计算效用分数. 同时, 为每个特征计算其分数概率, 计算过程如公式 (12) 所示. 对于特征集合中的任一特征 F_a, 其分数概率 $p_{spl\_F_a}$ 定义如下:

$$p_{spl\_F_a}=\exp\left(\frac{\varepsilon_{spl}\left(u_{spl}^{F_a}-u_{spl}^{*}\right)}{2\Delta u_{spl}}\right) \tag{12}$$

其中, ε_{spl} 表示分支节点获得的隐私预算, $u_{spl}^{F_a}$ 表示 F_a 的效用分数, 通过公式 (11) 计算获得. u_{spl}^{*} 表示待选特征集合中最大的效用分数. Δu_{spl} 为公式 (11) 的全局敏感度. 为简化计算, 假设对于任意数据集 D_2, 数据集 D_1 是通过在 D_2 上增加 1 条样本后形成的, 新增样本的特征 F_a 取值为 $v_{F_a}^{i'}$, 标签为 $L_{k'}$. 全局敏感度 Δu_{spl} 的计算过程如公式 (13).

$$\begin{aligned}\Delta u_{spl}&=\max_{D_1,D_2}\left|u_{spl}(D_1,F_a)-u_{spl}(D_2,F_a)\right|\\&=\left|\left(-\sum_{i=1}^{C}\left|v_{F_a}^{i}\right|\left(1-\sum_{k=1}^{K}\left(\frac{n_{L,L_k}^{F_a,i}}{\left|v_{F_a}^{i}\right|}\right)^2\right)\right)_{D_1}-\left(-\sum_{i=1}^{C}\left|v_{F_a}^{i}\right|\left(1-\sum_{k=1}^{K}\left(\frac{n_{L,L_k}^{F_a,i}}{\left|v_{F_a}^{i}\right|}\right)^2\right)\right)_{D_2}\right|\end{aligned} \tag{13}$$

其中, Δu_{spl} 等价于公式 (14).

$$\begin{aligned}\Delta u_{spl}&=\left|1-\frac{\left(n_{L,L_{k'}}^{F_a,i'}+1\right)^2+\sum_{k=1,k\neq k'}^{K}\left(n_{L,L_k}^{F_a,i'}\right)^2}{\left(\left|v_{F_a}^{i'}\right|+1\right)}+\frac{\sum_{k=1}^{K}\left(n_{L,L_k}^{F_a,i'}\right)^2}{\left|v_{F_a}^{i'}\right|}\right|\\&=\left|1+\frac{\left(\left|v_{F_a}^{i'}\right|+1\right)\sum_{k=1}^{K}\left(n_{L,L_k}^{F_a,i'}\right)^2-\left|v_{F_a}^{i'}\right|\left(\left(n_{L,L_{k'}}^{F_a,i'}+1\right)^2+\sum_{k=1,k\neq k'}^{K}\left(n_{L,L_k}^{F_a,i'}\right)^2\right)}{\left(\left|v_{F_a}^{i'}\right|+1\right)\left|v_{F_a}^{i'}\right|}\right|\\&=\left|1+\frac{\sum_{k=1}^{K}\left(n_{L,L_k}^{F_a,i'}\right)^2-2\left|v_{F_a}^{i'}\right|n_{L,L_{k'}}^{F_a,i'}-\left|v_{F_a}^{i'}\right|}{\left|v_{F_a}^{i'}\right|\left(\left|v_{F_a}^{i'}\right|+1\right)}\right|\\&=\left|1+\frac{\sum_{k=1}^{K}\left(n_{L,L_k}^{F_a,i'}\right)^2}{\left|v_{F_a}^{i'}\right|\left(\left|v_{F_a}^{i'}\right|+1\right)}-\frac{2n_{L,L_{k'}}^{F_a,i'}+1}{\left|v_{F_a}^{i'}\right|+1}\right|\end{aligned} \tag{14}$$

根据 $0\leqslant\left|\frac{\sum_{k=1}^{K}\left(n_{L,L_k}^{F_a,i'}\right)}{\left|v_{F_a}^{i'}\right|\left(\left|v_{F_a}^{i'}\right|+1\right)}\right|\leqslant\left|\frac{\left(\left|v_{F_a}^{i'}\right|\right)^2}{\left|v_{F_a}^{i'}\right|\left(\left|v_{F_a}^{i'}\right|+1\right)}\right|\leqslant 1$, $0\leqslant\left|\frac{2n_{L,L_{k'}}^{F_a,i'}+1}{\left|v_{F_a}^{i'}\right|+1}\right|\leqslant\left|\frac{2\left|v_{F_a}^{i'}\right|+1}{\left|v_{F_a}^{i'}\right|+1}\right|\leqslant 2$. 可以确定 Δu_{spl} 的取值范围位于 0 到 2 之间, 由于当 Δu_{spl} 取值过小时, 差分隐私的保护效果将受到影响, 因此本文将 Δu_{spl} 设定为 2.

此外, 在上述计算过程中, D_1 是通过在 D_2 上增加 1 条样本后形成的数据集, 同理可得, 如果假设 D_2 是通过在 D_1 增加 1 条样本后形成的数据集, Δu_{spl} 的范围仍然位于 0 到 2 之间.

算法 2. 分裂特征的输出过程.

输入: 训练数据 D, 训练数据的特征集合 F, 训练数据的标签集合 L, 隐私预算 ε_{spl};
输出: 节点的分裂特征.

函数: $BuildSplit(D,F,L,\varepsilon_{spl})$.
① $QS = \{\}$ /*设置一个空集合*/;
② for f in F do /*遍历集合 F 中每一个特征*/
③ 根据公式 (11) 计算特征 f 的效用分数 u_{spl}^{f}
④ end for
⑤ 将 QS 集合中最大的效用分数记为 u_{spl}^{*};
⑥ 对特征集合 F 执行随机排列操作, 获得新的特征集合 B;
⑦ for b in B do /*遍历集合 B 中每一个特征*/
⑧ 设置 $\Delta u_{spl} = 2$
⑨ 根据公式 (12) 计算特征 b 的概率 p_{spl_b}:
⑩ if $Bernoulli(p_{spl\_b})$ then /*根据 p_{spl_b} 生成一个服从伯努利分布的随机数 r */
⑪ return b /*当随机数 r 取值为 1 时, 将 b 作为分支节点的分裂特征*/
⑫ end if
⑬ end for

在创建分支节点过程中, 本文基于算法 2 访问训练数据并输出分裂特征, 并将输出的特征作为分支节点的分裂点.

3.2.2 创建叶子节点

本文进一步调整训练数据信息的返回形式, 仅将标签传递给叶子节点, 以避免拉普拉斯机制下错误的噪声计数结果. 与此同时, 仅使用叶子节点的标签即可完成对新样本的预测, 因此对叶子节点传递标签的操作并不会对决策树模型的预测功能造成影响. 上述过程中, 同样采用重排翻转机制实现访问数据过程的隐私保护. 其中叶子节点对训练数据查询操作的效用函数设置如下.

假设当前访问的训练集为 D, 则对标签集 L 执行查询的效用函数 $u_{leaf}(D,L_i)$ 定义如公式 (15) 所示:

$$u_{leaf}(D,L_i) = N_i \tag{15}$$

其中, L_i 表示标签集合中任意标签, N_i 表示 D 中标签为 L_i 的样本数量. 根据全局敏感度的定义, 可以计算出公式 (15) 的全局敏感度为 1.

通过公式 (15) 计算每种标签的效用分数, 同时为每种标签计算分数概率, 计算过程如下.

假设 L_i 表示标签集合中任意标签, 则 L_i 的分数概率 p_{leaf_i} 如公式 (16) 所示:

$$p_{leaf\_i} = \exp\left(\varepsilon_{leaf}\left(u_{leaf}^{i} - u_{leaf}^{*}\right)/2\right) \tag{16}$$

其中, ε_{leaf} 表示叶子节点获得的隐私预算, u_{leaf}^{i} 表示根据公式 (15) 计算得到的标签 L_i 的效用分数, u_{leaf}^{*} 表示待选标签集合中的最大效用分数.

算法 3. 标签的输出过程.

输入: 训练数据 D, 标签集合 L, 隐私预算 ε_{leaf};
输出: 节点的标签.

函数: $BuildLeaf(D,L,\varepsilon_{leaf})$.

① $QS = \{\}$ /*设置一个空集合*/;
② for i in L do /*遍历集合 L 中每一种标签*/
③ 根据公式 (15) 计算标签 i 的效用分数 u_{leaf}^{i}
④ $QS = u_{leaf}^{i}$
⑤ end for
⑥ 将 QS 集合中最大的效用分数记为 u_{leaf}^{*};
⑦ 对标签集合 L 执行随机排列操作, 获得新的标签集合 U;
⑧ for u in U do /*遍历集合 U 中每一种标签*/
⑨ 根据公式 (16) 计算标签 u 的概率 p_{leaf_u}
⑩ if $Bernoulli(p_{leaf\_u})$ then /*根据 p_{leaf_u} 生成一个服从伯努利分布的随机数 r */
⑪ return u /*当随机数 r 取值为 1 时, 将 u 作为叶子节点的标签*/
⑫ end if
⑬ end for

每次创建叶子节点时, 本文通过算法 3 的流程对训练数据进行访问并将输出的标签作为叶子节点的标签.

3.3 随机森林生成环节

基于第 3.1 节以及第 3.2 节, 进一步生成差分隐私随机森林模型. 首先, 从原始数据集中采用不放回抽样的方式抽取若干样本, 基于这些样本构建多个训练子集, 将第 3.1 节划分好的隐私预算分配给每个训练子集. 其次, 每个训练子集基于第 3.2 节的 pfDPDT 算法构建相应的差分隐私决策树模型. 随后, 将多个差分隐私决策树模型集成为差分隐私随机森林模型, 决策树模型之间会采用投票方式选举出整个森林模型的预测结果.

至此, 整个 eDPRF 算法的主要设计思路表述完毕, 为了更清晰描述算法的具体实施步骤, 整个 eDPRF 算法执行流程如算法 4 所示.

算法 4. eDPRF 算法.

输入: 训练数据 D, 训练数据的特征集合 F, 训练数据的标签集合 L, 随机森林的隐私预算 ε, 随机森林的深度 $treeDepth$, 随机森林的规模 T;
输出: 随机森林模型.

函数: $BuildForest(D,F,L,\varepsilon,depth,treeDepth)$.
① $Forest = \{\}$;
② for $i = 1$ to T do
③ $depth \leftarrow 1$;
④ 设置每个决策树的隐私预算为 $\varepsilon_{tree} = \varepsilon$;
⑤ 从训练集 D 使用不放回抽样的方式抽取大小为 $\frac{|D|}{T}$ 的训练子集 D_i;
⑥ $D = D - D_i$;
⑦ $tree_i \leftarrow BuildTree(D_i,F,L,\varepsilon_{tree},depth,treeDepth)$ /*调用算法 5 构建决策树 */
⑧ $Forest = Forest \cup tree_i$
⑨ end for
⑩ return 随机森林模型

其中, 算法 4 中调用的算法 5 的伪代码如下.

算法 5. pfDPDT.

输入: 训练数据 D, 训练数据的特征集合 F, 训练数据的标签集合 L, 决策树的隐私预算 ε_{tree}, 决策树的深度 $treeDepth$, 当前节点所在的深度 $depth$;
输出: 决策树模型.

函数: $BuildTree(D,F,L,\varepsilon_{tree},depth,d_{\max})$.
① if 特征集合为空 $F=\varnothing$ or 当前深度达到最大深度 $depth == treeDepth$ or D 只含一类标签 then
② 设置叶子节点的隐私预算为

$$\varepsilon_{leaf}=\varepsilon_{tree}-\sum_{m=1}^{depth-1}\frac{\varepsilon_{tree}}{s_t(treeDepth-m+1)};$$

③ $leaf\_label \leftarrow BuildLeaf(D_i,L,\varepsilon_{leaf})$ /*调用算法 3 返回叶子节点标签*/;
④ 设置叶子节点的标签为 $leaf\_label$;
⑤ return 叶子节点
⑥ else
⑦ for f in F do
⑧ if f 是连续型特征 then
⑨ 对 f 通过分箱操作完成离散化
⑩ end if
⑪ end for
⑫ 设置分支结点的隐私预算为

$$\varepsilon_{spl}=\frac{\varepsilon_{tree}}{s_t(treeDepth-depth+1)};$$

⑬ $split\_value \leftarrow BuildSplit(D,F,L,\varepsilon_{spl})$ /*调用算法 2 返回分裂特征*/;
⑭ 根据选出的特征 $split\_value$, 对训练集 D 执行划分操作;
⑮ 从 F 中删除使用过的特征, $F \leftarrow F-split\_value$;
⑯ for D' in $\{D[1],...,D[k]\}$ do /*递归地执行建树操作*/

$$BuildTree(D',F,L,\varepsilon_{tree},depth+1,treeDepth)$$

⑰ end for
⑱ end if
⑲ return 决策树模型

至此, 满足差分隐私的随机森林模型算法 5 构建完成. 用户可以基于该模型进一步实现新样本预测过程, 具体流程见算法 6 所示.

算法 6. 基于随机森林实现预测过程.

输入: 未知样本集 D_{test}, 随机森林 $Forest$;
输出: 预测结果.

函数: $Predict(D_{test},Forest)$.
① $Result=\{\}$;
② for d in D_{test} do
③ for $tree$ in $Forest$ do
④ 遍历当前决策树, 到达叶子节点, 得到预测值 $predict\_value$;

⑤ $Result = Result \cup predict\_value$;
⑥ end for
⑦ end for
⑧ return $max\{Result\}$

4 隐私性分析

本节对 pfDPDT 算法、eDPRF 算法以及基于随机森林实现预测过程 (算法 6) 的隐私性进行分析.

定理 3. 假设分配给 pfDPDT 算法的隐私预算为 ε_{tree}, 则 pfDPDT 算法满足 ε_{tree}-差分隐私.

证明: 假设决策树的最大深度为 $treeDepth$. 使用 pfDPDT 算法构建第 i 层的任意节点时将分为两类: 如果该节点是分支节点, pfDPDT 算法消耗的隐私预算如公式 (17) 所示. 由于 pfDPDT 算法对于分支节点的创建过程严格遵从重排翻转机制[12], 因此该分支节点的构建过程满足 $\varepsilon_{i,spl}$-差分隐私.

$$\varepsilon_{i,spl} = \frac{\varepsilon_{tree}}{s_t} \times \frac{1}{treeDepth - i + 1} \tag{17}$$

如果该节点是叶子节点, pfDPDT 算法消耗的隐私预算如公式 (18) 所示. 由于算法 5 对于叶子节点的创建过程严格遵从重排翻转机制, 因此该叶子节点的构建过程满足 $\varepsilon_{i,leaf}$-差分隐私.

$$\varepsilon_{i,leaf} = \varepsilon_{tree} - \sum_{j=1}^{i-1} \varepsilon_{j,spl} \tag{18}$$

根据并行组合定理, 构建决策树所消耗的隐私预算 ε_{spend} 取决于最长支路消耗的总隐私预算 $\varepsilon_{\max}$. 假设 ε_{split} 是最长支路上所有分支节点的预算消耗, ε_{leaf} 是最长支路上叶子节点的预算消耗, 则总隐私预算 $\varepsilon_{\max}$ 的计算过程如下:

$$\begin{aligned}\varepsilon_{\max} &= \varepsilon_{split} + \varepsilon_{leaf} \\ &= \sum_{i=1}^{treeDepth-1} \varepsilon_{i,spl} + \varepsilon_{treeDepth,leaf} \\ &= \varepsilon_{tree}\end{aligned}$$

所以可以得出 $\varepsilon_{spend} = \varepsilon_{\max} = \varepsilon_{tree}$. 由于在使用 pfDPDT 算法构建决策树的过程中, 所消耗的隐私预算没有超过 ε_{tree}, 因此 pfDPDT 算法满足 ε_{tree}-差分隐私.

综上所述, 定理 3 证明完毕.

定理 4. 假设分配给 eDPRF 算法的隐私预算为 ε, 则 eDPRF 算法满足 ε-差分隐私.

证明: 假设随机森林规模为 T, 随机森林的最大深度为 $treeDepth$. 在构建每棵决策树时, eDPRF 算法为 pfDPDT 算法分配的隐私预算均为 ε. 根据定理 3 可得, 对于每棵决策树, pfDPDT 算法满足 ε-差分隐私.

由于 eDPRF 算法中每棵决策树被分配的训练数据互不相交, 即存在 $D_i \cap D_j = \varnothing, i \neq j$ 且 $i, j \in [1, T]$.

根据差分隐私的并行组合定理可得, eDPRF 算法构建随机森林所消耗的隐私预算为 ε, 该过程消耗的隐私预算没有超过 ε, 因此 eDPRF 算法满足 ε-差分隐私.

综上所述, 定理 4 证明完毕.

定理 5. 假设 eDPRF 算法满足 ε-差分隐私, 则使用 eDPRF 算法构建而成的随机森林模型满足 ε-差分隐私, 并且基于随机森林实现预测过程 (算法 6) 同样满足 ε-差分隐私.

证明: 由于 eDPRF 算法满足 ε-差分隐私, 根据差分隐私的后处理性质, 使用 eDPRF 算法构建而成的随机森林模型满足 ε-差分隐私.

由于随机森林模型满足 ε-差分隐私, 根据差分隐私的后处理性质, 基于随机森林实现预测过程 (算法 6) 满足 ε-差分隐私.

综上所述, 定理 5 证明完毕.

5 实验结果与分析

本节首先给出算法的时间复杂度分析, 在第 2、3 节对所使用的实验环境以及实验指标进行介绍, 最后在第 4 节与同类的 DiffPRF_linear 算法[7]、TpDPRF 算法[8]以及 DiffPRF 算法[23]展开实验比较和分析.

5.1 时间复杂度

表 2 给出了本文方案 eDPRF 与 DiffPRF_linear 算法[7]、TpDPRF 算法[8]以及 DiffPRF 算法[23]的时间复杂度分析, 以及性能对比. 其中, n 和 m 分别是特征个数与标签种类数, 可以看出, 分支节点构造阶段, 所有方案的时间复杂度与特征个数 n 成线性相关, 叶子节点构造阶段的时间复杂度与标签种类数 m 呈线性相关, 在此过程中与整个数据集的大小无关, 因此可以适用于大规模数据集. 针对隐私安全和准确度分析, 安全性分析可以看出本文方案达到 ε-差分隐私, 与其他方案对比可以达到同等级别, 但是从准确度评估结果可以明显看出本文方案更优.

表 2 时间复杂度和性能对比

方案	分支节点构造阶段	叶子节点构造阶段	隐私安全	准确度 (%)
DiffPRF_linear算法[7]	O(n)	1+ O(m)	ε-差分隐私	82.95
TpDPRF算法[8]	O(n)	1+ O(m)	ε-差分隐私	81.91
DiffPRF算法[23]	O(n)	1+ O(m)	ε-差分隐私	78.89
eDPRF (本文方案)	O(3n)	O(3m)	ε-差分隐私	85.98

5.2 实验平台介绍

本文实验环境是 64 位 Windows 10 操作系统的台式机, 处理器型号是英特尔酷睿 i9-9900K, 内存大小为 32GB. 本文所有的算法代码基于 Python 语言编写, 使用的代码编辑器是 VSCode, 运行环境是 Python 3.8. 实验数据集来源于 Kaggle 的 diabetes[26]以及 UCI 的 wall-following robot 数据集[27]. 由于原始的 diabetes 数据集不平衡比例达到 92:8, 为了避免数据不平衡给实验带来的误导性判断, 本文对原始的 diabetes 数据集, 使用 imblearn.under_sampling 库的 RandomUnderSampler() 函数执行欠采样操作, 生成一个平衡的数据集, 欠采样的参数为 sampling_strategy={0:7000, 1:7000}和 random_state=42. 表 3 给出实验中所使用的数据集信息.

表 3 实验数据集的统计信息

数据集	数据集大小	离散型特征个数	连续型特征个数	标签类别个数
diabetes	14000	4	4	2
wall-following robot	5456	0	25	2

5.3 实验指标介绍

本文采用预测准确度对训练集上生成的模型进行性能评估. 通过计算模型在测试集上正确预测的样本数量与总测试样本数量之比获得预测准确度. 在训练集和测试集的划分中, 本文设置的比例为 8:2. 具体计算过程如下.

假设测试集中包含 N 个样本, 其中第 i 个样本为 x_i, 对应的真实标签为 y_i, 将第 i 个样本输入到随机森林模型后, 输出对应的预测标签为 $model(x_i)$, 则 N 个测试样本的预测准确度可以通过公式 (19) 计算得出.

$$Acc = \frac{\sum_{i=1}^{N} model(x_i) == y_i \ ? \ 1 : 0}{N} \tag{19}$$

5.4 实验结果分析

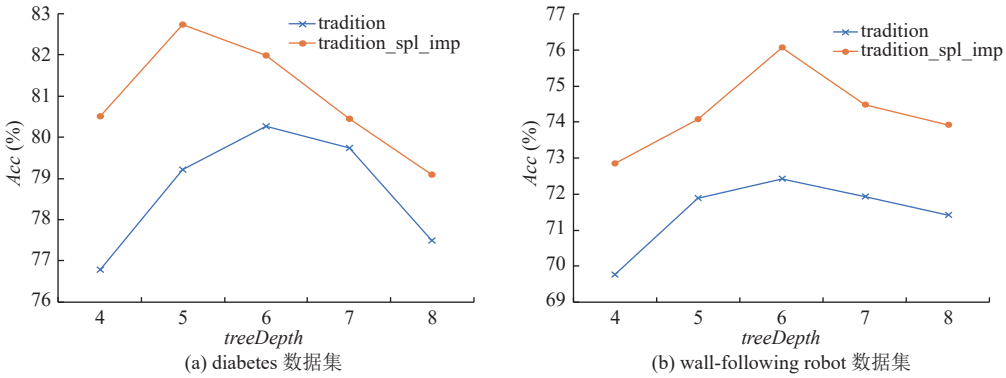
为说明所提算法的有效性, 接下来将本文方案 eDPRF 与其他方案[7,8,23]进行实验比较和分析.

5.4.1 决策树创建方法比较

由于 DiffPRF 算法[7]、TpDPRF 算法[8]以及 DiffPRF_linear 算法[23]在决策树的创建方式上均属于传统方法[15]. 为了便于分析, 本文将传统建树算法记为 tradition, 然后与该算法进行 3 组实验对比. 第 1 组实验中, 在

tradition 上对本文提出的分支节点创建方法进行有效性验证. 第 2 组实验中, 在 tradition 上对本文提出的叶子节点创建方法进行有效性验证. 第 3 组实验中, 与 tradition 进行全面比较, 其中实验的隐私预算以及深度默认为 1 和 6.

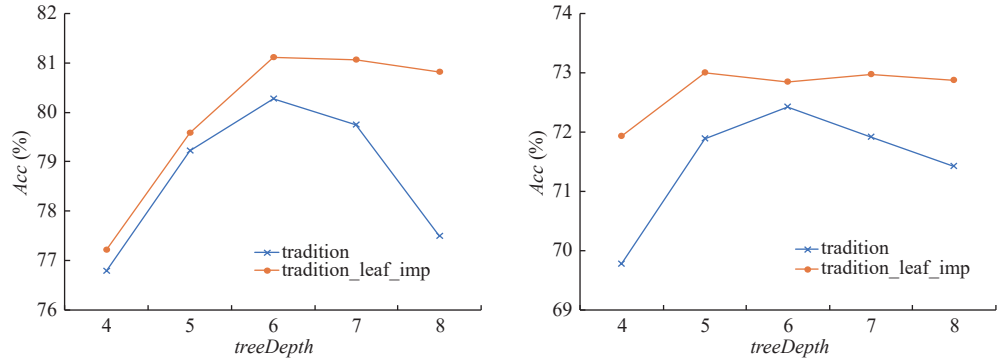
为验证分支节点创建方法有效性, 本文将 tradition 算法的分支节点创建方法替换成所提出的方法, 并将替换后的算法记为 tradition_spl_imp. 本文在深度 *treeDepth* 取值为 4, 5, 6, 7, 8 的情况下进行实验. 从图 5 所示的结果可以看出, 使用 tradition 算法构建出来的模型, 在所有数据集上的准确度始终低于 tradition_spl_imp. 这表明本文提出的分支节点创建方法是有效的. 相比传统方法使用的指数机制, 重排翻转机制将部分概率从得分较低的选项重新分配到得分较高的选项, 从而使得得分较高的选项具有比指数机制更高的输出概率. 本文巧妙地引入重排翻转机制, 在分裂特征的选择上进行有效设计. 在算法设计阶段, 本文可以确保基尼指数最小的特征具有最高的得分, 基尼指数越大的特征具有越低的分数, 从而保证算法在访问数据时有极大的概率获得更优特征. 同时, 从图中也可以看出, 在深度不同时, 本方法始终优于传统方法, 进一步说明本方法的鲁棒性, 即在不同条件下都能保持更好的表现和稳定性.



(a) diabetes 数据集 (b) wall-following robot 数据集

图 5 分支节点有效性评估

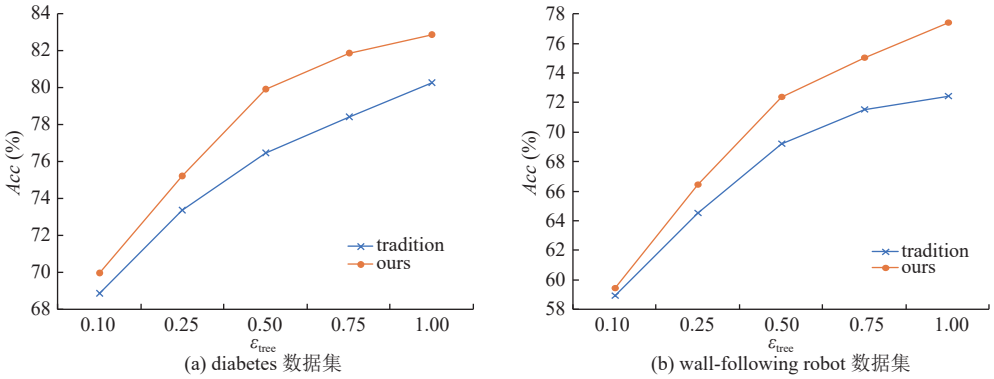
为验证叶子节点创建方法有效性, 本文将 tradition 算法的叶子节点创建方法替换成所提出的方法, 并将替换后的算法记为 tradition_leaf_imp. 与前一个实验相同, 本次实验在深度分别为 4, 5, 6, 7 和 8 的情况下进行. 图 6 可以看出 tradition_leaf_imp 的预测准确度始终高于 tradition, 该结果证实了本方法的有效性. 传统方法直接向计数值添加拉普拉斯噪声容易导致数值的随机增加或减少, 有极大可能会导致标签的类别计数相对大小发生变化, 导致生成的叶子节点标签偏离正确的结果, 进而影响决策树模型的精确预测能力. 相比之下, tradition_leaf_imp 算法修改了叶子节点标签的获取方式, 将其与更为先进的重排翻转机制结合, 相比于传统方法可以实现更精准的标签输出.



(a) diabetes 数据集 (b) wall-following robot 数据集

图 6 叶子节点有效性评估

图 7 描述了本文算法与 tradition 算法生成的决策树模型预测准确度对比结果, 可以看到在所有数据集上, 本文算法的预测准确度均要高于 tradition 算法, 通过前面的分析可知, 本文算法在分支节点的创建方式以及叶子节点的创建方式上均优于 tradition 算法, 因此使用本文算法创建而成的决策树在准确度方面也会优于 tradition 算法. 此外, 当分配给建树算法的隐私预算 ε_{tree} 不同时, 随着 ε_{tree} 的增加, 算法构成的决策树模型准确度不断上升, 其主要原因在于 ε_{tree} 过小会导致算法中引入过多噪声, 影响模型的学习能力. 从图中可以看出, 本文所提建树算法形成的模型始终保持优势, 该结果表明本文算法具有更强的抗干扰能力. 即使算法中存在扰动, 本文算法中模型获取数据信息的能力也优于传统算法.



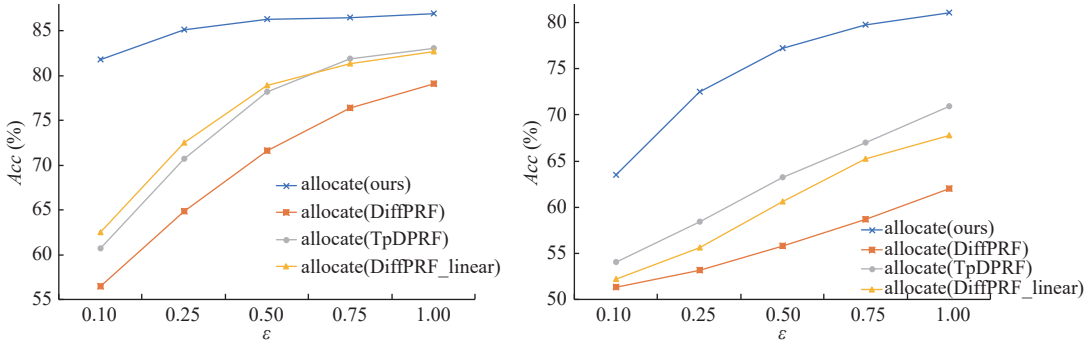
(a) diabetes 数据集 (b) wall-following robot 数据集

图 7 建树方法对比

5.4.2 隐私预算分配方式比较

为验证隐私预算分配方法有效性, 本文将在森林规模 35、深度 6 以及总隐私预算 ε 分别取值为 0.1, 0.25, 0.5, 0.75, 1 的情况下, 与 DiffPRF 算法、TpDPRF 算法以及 DiffPRF_linear 算法所使用的隐私预算分配方法进行比较.

从图 8 可以看出, 本文分配方法更具优势, 其主要原因在于上述研究尽管改进了决策树内部的隐私预算分配方式, 但只是对隐私预算实行按层分配. 这种方式会导致一些隐私预算被闲置, 而本文方法可以对所有隐私预算实现合理利用, 从而降低噪声改善模型性能. 此外, 这些方法在决策树之间的预算分配方式采用传统方式, 当森林规模较大时, 每个决策树获得的预算变得极小, 严重损害决策树性能, 从而破坏集成后的随机森林准确度. 而本文通过并行组合定理改变训练子集的划分方式, 可以使每个决策树获得非常可观的隐私预算, 缓解随机扰动. 同时, 可以看出在不同隐私预算下, 该算法相比于其他算法具有明显的优势, 说明在不同的隐私预算情况下, 该算法可以很好地实现对隐私预算的充分利用, 改善模型性能.



(a) diabetes 数据集 (b) wall-following robot 数据集

图 8 隐私预算分配方法对比

5.4.3 训练算法的比较

最后, 本文与 DiffPRF 算法、TpDPRF 算法以及 DiffPRF_linear 算法进行比较. 其中, 训练算法的隐私预算设置为 1, 深度设置为 6, 森林规模 T 设置为 5, 11, 17, 35. 从图 9 可以看出, eDPRF 算法的预测准确度显著优于其他算法. 通过前面的实验结果可知, eDPRF 算法在建树方法以及隐私预算分配方法上均优于其他算法, 因此从整体上看, eDPRF 算法优于其他算法是有理可循的. 此外, 其他算法在森林规模变化时, 准确度的变化比较剧烈, 容易随着森林规模的增加而减少, 而 eDPRF 算法在图中所示的森林规模下, 准确度保持较好状态, 且变化幅度小, 因此本文算法具有更强的鲁棒性.

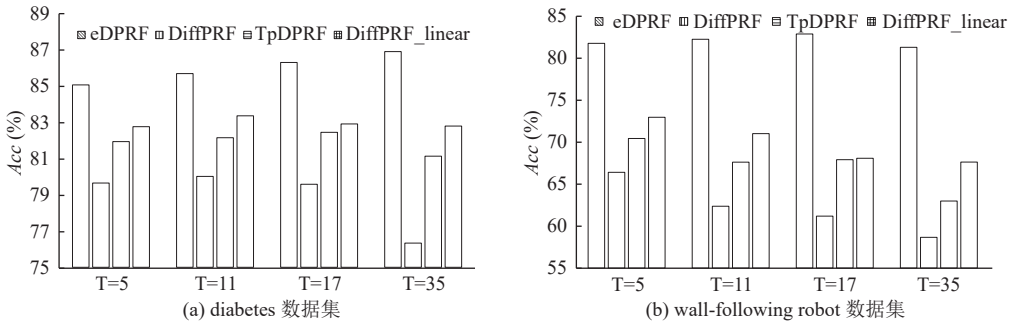


图 9 训练算法对比

6 总 结

本文提出了一种高效的差分隐私随机森林训练算法 eDPRF, 该算法在训练过程中首次引入重排翻转机制, 有效提升决策树模型在扰动情况下的数据学习能力, 同时设计了有效的隐私预算分配方法, 降低训练过程中的随机扰动. 最后, 本文通过隐私性分析证明所提算法满足差分隐私保护, 并通过实验评估表明该算法有效改善随机森林模型的分类性能.

References:

[1] Deng CL, Guan B, Liu DF, Liu LX, Shi QL, Wang HR, Wang YJ. Prediction of the efficacy of radiotherapy and chemotherapy for cervical squamous cell carcinoma based on random forests. Ruan Jian Xue Bao/Journal of Software, 2021, 32(12): 3960–3976 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6136.htm [doi: 10.13328/j.cnki.jos.006136]

[2] Bertran M, Tang S, Kearns M, Morgenstern J, Roth A, Wu ZS. Scalable membership inference attacks via quantile regression. In: Proc. of the 37th Int'l Conf. on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2024. 16. [doi: 10.5555/3666122.3666138]

[3] Liu GY, Xu TL, Zhang R, Wang ZX, Wang C, Liu L. Gradient-leaks: Enabling black-box membership inference attacks against machine learning models. IEEE Trans. on Information Forensics and Security, 2024, 19: 427–440. [doi: 10.1109/TIFS.2023.3324772]

[4] Wang XD, Wu LF, Guan ZT. GradDiff: Gradient-based membership inference attacks against federated distillation with differential comparison. Information Sciences, 2024, 658: 120068. [doi: 10.1016/j.ins.2023.120068]

[5] Lu ZB, Liang H, Zhao MH, Lv QZ, Liang TC, Wang YL. Label-only membership inference attacks on machine unlearning without dependence of posteriors. Int'l Journal of Intelligent Systems, 2022, 37(11): 9424–9441. [doi: 10.1002/int.23000]

[6] Rajabi A, Sahabandu D, Niu LY, Ramasubramanian B, Poovendran R. LDL: A defense for label-based membership inference attacks. In: Proc. of the 2023 ACM Asia Conf. on Computer and Communications Security. Melbourne: ACM, 2023. 95–108. [doi: 10.1145/3579856.3582821]

[7] Dong YL, Zhang SF, Xu JC, Wang HS, Liu JQ. Random forest algorithm based on linear privacy budget allocation. Journal of Database Management, 2022, 33(2): 19. [doi: 10.4018/JDM.309413]

[8] Liu J, Li XX, Wei QM, Liu SF, Liu Z, Wang JY. A two-phase random forest with differential privacy. Applied Intelligence, 2023, 53(10): 13037–13051. [doi: 10.1007/s10489-022-04119-6]

[9] Jain P, Raskhodnikova S, Sivakumar S, Smith A. The price of differential privacy under continual observation. In: Proc. of the 40th Int'l

Conf. on Machine Learning. Honolulu: PMLR, 2023. 14654–14678.

[10] Ha T, Vo T, Dang TK, Trang NTH. Differential privacy under membership inference attacks. In: Proc. of the 10th Int'l Conf. on Future Data and Security Engineering. Da Nang: Springer, 2023. 255–269. [doi: 10.1007/978-981-99-8296-7_18]

[11] Zhang XJ, He FC, Gai JY, Bao JD, Huang HY, Du XG. A differentially private federated learning model for fingerprinting indoor localization in edge computing. Journal of Computer Research and Development, 2022, 59(12): 2667–2688 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.20210270]

[12] McKenna R, Sheldon DR. Permute-and-Flip: A new mechanism for differentially private selection. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc. , 2020. 17. [doi: 10.5555/3495724.3495741]

[13] Dwork C, Roth A. The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science, 2014, 9(3-4): 211–407. [doi: 10.1561/0400000042]

[14] Patil A, Singh S. Differential private random forest. In: Proc. of the 2014 Int'l Conf. on Advances in Computing, Communications and Informatics. Delhi: IEEE, 2014. 2623–2630. [doi: 10.1109/ICACCI.2014.6968348]

[15] Mu HR, Ding LP, Song YN, Lu GQ. DiffPRFs: Random forest under differential privacy. Journal on Communications, 2016, 37(9): 175–182 (in Chinese with English abstract). [doi: 10.11959/j.issn.1000-436x.2016169]

[16] Zhang YL, Peng PF, Ning Y. Random forest algorithm based on differential privacy protection. In: Proc. of the 20th Int'l Conf. on Trust, Security and Privacy in Computing and Communications. Shenyang: IEEE, 2021. 1259–1264. [doi: 10.1109/TrustCom53373.2021.00172]

[17] Wang FW, Xie MY, Tan ZY, Li QR, Wang CG. Preserving differential privacy in deep learning based on feature relevance region segmentation. IEEE Trans. on Emerging Topics in Computing, 2024, 12(1): 307–315. [doi: 10.1109/TETC.2023.3244174]

[18] Guan ZT, Sun XW, Shi LY, Wu LF, Du XJ. A differentially private greedy decision forest classification algorithm with high utility. Computers & Security, 2020, 96: 101930. [doi: 10.1016/j.cose.2020.101930]

[19] Zhao Y, Du JT, Chen JJ. Scenario-based adaptations of differential privacy: A technical survey. ACM Computing Surveys, 2024, 56(8): 199. [doi: 10.1145/3651153]

[20] Niu XF, Ma WP. An ensemble learning model based on differentially private decision tree. Complex & Intelligent Systems, 2023, 9(5): 5267–5280. [doi: 10.1007/s40747-023-01017-3]

[21] Wang CY, Chen SY, Li XC. Adaptive differential privacy budget allocation algorithm based on random forest. In: Proc. of the 16th Int'l Conf. on Bio-inspired Computing: Theories and Applications. Taiyuan: Springer, 2021. 201–216. [doi: 10.1007/978-981-19-1256-6_15]

[22] Li X, Qin BD, Luo YY, Zheng D. A differential privacy budget allocation algorithm based on out-of-bag estimation in random forest. Mathematics, 2022, 10(22): 4338. [doi: 10.3390/math10224338]

[23] Deng W, Chen XT, Zhang QH, Wang GY. Differential privacy protection algorithms based on tree model. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2020, 32(5): 848–856 (in Chinese with English abstract). [doi: 10.3979/j.issn.1673-825X.2020.05.018]

[24] Dwork C, Mcsherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. In: Proc. of the 3rd Theory of Cryptography Conference. New York: Springer, 2006. 265–284. [doi: 10.1007/11681878_14]

[25] McSherry F, Talwar K. Mechanism design via differential privacy. In: Proc. of the 48th Annual IEEE Symp. on Foundations of Computer Science. Providence: IEEE, 2007. 94–103. [doi: 10.1109/FOCS.2007.66]

[26] Mustafa M. Diabetes prediction dataset. 2024. http://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset/data

[27] Freire A, Veloso M, Barreto G. Wall-following robot navigation data. 2024. http://archive.ics.uci.edu/dataset/194/wall+following+robot+navigation+data

附中文参考文献:

[1] 邓成龙, 关贝, 刘德丰, 刘兰祥, 石清磊, 王浩然, 王永吉. 基于随机森林的宫颈鳞癌放化疗疗效预测. 软件学报, 2021, 32(12): 3960–3976. http://www.jos.org.cn/1000-9825/6136.htm [doi: 10.13328/j.cnki.jos.006136]

[11] 张学军, 何福存, 盖继扬, 鲍俊达, 黄海燕, 杜晓刚. 边缘计算下指纹室内定位差分私有联邦学习模型. 计算机研究与发展, 2022, 59(12): 2667–2688. [doi: 10.7544/issn1000-1239.20210270]

[15] 穆海蓉, 丁丽萍, 宋宇宁, 卢国庆. DiffPRFs: 一种面向随机森林的差分隐私保护算法. 通信学报, 2016, 37(9): 175–182. [doi: 10.11959/j.issn.1000-436x.2016169]

[23] 邓蔚, 陈秀婷, 张清华, 王国胤. 基于树模型的差分隐私保护算法. 重庆邮电大学学报 (自然科学版), 2020, 32(5): 848–856. [doi: 10.3979/j.issn.1673-825X.2020.05.018]

王树兰(1987－), 女, 博士, 副教授, CCF 专业会员, 主要研究领域为机器学习, 隐私保护, 密码算法与协议.

邹家须(1985－), 男, 实验师. 主要研究领域为网络与信息安全, 终端设备安全, 数据安全.

邱瑶(1996－), 女, 硕士, 主要研究领域为机器学习, 隐私保护.

王彩芬(1963－), 女, 博士, 教授, 博士生导师, 主要研究领域为密码学, 隐私保护.

赵陈斌(1996－), 男, 博士生, 主要研究领域为数据安全, 隐私保护, 应用密码学.