Federated Extra-Trees with Privacy Preserving

Anonymous Author(s)

Abstract

It is commonly observed that the data are scattered everywhere and difficult to be centralized. The data privacy has become a sensitive topic due to the frequent misusage of personal data as well as the occasionally happened data breaches. The laws and regulations such as the European Union's General Data Protection Regulation (GDPR) are designed to protect the public's data privacy. However, machine learning requires a large amount of data for better performance, and the current circumstances put deploying real-life AI applications in an extremely difficult situation. To tackle challenges, in this paper we propose a novel privacy-preserving federated machine learning model, named *Federated Extra-Trees*, which combines the extremely randomized trees and local differential privacy. A secure multi-institutional machine learning system was also developed to provide a lossless performance by processing the modeling jointly on different clients without exchanging any of their raw data. We have validated the accuracy of our work by performing experiments on the UCI datasets and the efficiency and robustness were also verified by simulating the real-world scenarios. Overall, we provided an extensible, scalable and practical solution to handle the *data island* problem.

1 Introduction

Although we are living in the era of *Big Data*, we often have to face the facts that there are not enough data for modeling, not to mention building state-of-the-art AI models. In the work of BigGAN (Brock, Donahue, and Simonyan 2019), they trained the Generative Adversarial Networks (GAN) with over 300 million images, and 3.3 billion word corpus was used for training the Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al. 2018). But in most cases, a single platform does not have such a large amount of data, and the data are scattered across platforms or organizations. For example, medical data are often isolated in different medical centers or hospitals, not centralized mostly due to privacy policies. This has become a serious problem for medical study because the researchers can only access data from one or two institutions, which is simply not enough for modeling. Similar issues are also often observed in the industries. Finance for example, the national or international financial institutions usually serve at least tens of millions customers, and they don't need to worry too much about the data quantity. But for small size companies like regional banks, they only serve the local residents and the data they collected from customers may not be enough for building effective financial models, such as for tasks of risk management or loan application approval. It is a paradox when we always talk about how big data would improve our lives but in reality we don't have enough data to develop meaningful machine learning models.

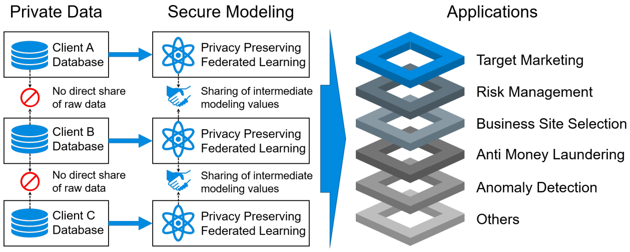

Private Data
Secure Modeling
Applications
Client A Database
Privacy Preserving Federated Learning
No direct share of raw data
Sharing of intermediate modeling values
Client B Database
Privacy Preserving Federated Learning
No direct share of raw data
Sharing of intermediate modeling values
Client C Database
Privacy Preserving Federated Learning
Target Marketing
Risk Management
Business Site Selection
Anti Money Laundering
Anomaly Detection
Others

Figure 1: Federated Machine Learning Applications

Multi-institutional collaborative modeling has a big potential in both academia and industries, and many advanced applications can be developed, as presented in Figure 1. As we know, the data often exist in the form of *data island* and are isolated in different companies, organizations, or government agencies. The direct consequence is lots of valuable information cannot be mined, discovered or studied. In the work of (Sheller et al. 2018), the researchers built a semantic segmentation model on multimodal brain scans. The entire modeling was conducted on a multi-institutional collaboration and no raw patient data were shared. However, not every machine learning engineer has the chance to unite multiple institutions modeling together. One of the biggest challenges is data privacy. Recently, the Federal Trade Commission (FTC) of the United States imposed a record-breaking $5 billion penalty to Facebook, due to its violation on an FTC's 2012 order about user data privacy. This is not the first time that Facebook has violated the regulations. In 2018, the Information Commissioners Office (ICO) of United Kingdom (UK) also fined this company £500,000 over the scandal with the Cambridge Analytica, and this penalty is the maximum possible amount by UK's laws. Many other companies also face similar legal sanctions. In 2018, due to separate data breaches, Marriott paid £99,200,396 to the ICO and British Airways was fined £183.39 million. Recently, Google, Amazon and Apple were all reported to have been hiring people to review the recorded audios from their voice assistant products. Although they stated the data were not linked to the original users and only used to improve their products, under the European Union's (EU) General Data Protection Regulation (GDPR), this kind of processing on users' data could have already violated the regulations.

To meet the regulation requirements and protect data privacy, Google proposed the federated machine learning (FML) (McMahan et al. 2017; Konecný et al. 2016b; 2016a). The key concept of their work is to train models without integrating the data together in one place, and no raw data would be exposed to other parties but fully secured and under users' own control. This approach has been later defined as horizontal FML (Yang et al. 2019), where data are distributed in different places with the same attribute set but different user samples. In Google's case, the data are more extremely scattered across personal devices, each with a limited data volume. In contrast, for a general business scenario, the situation is that several similar and small size companies such as regional banks want to build joint models together for solving a common business problem, e.g. intelligent loan application approval. Inspired by their work, we proposed a novel privacy-preserving federated machine learning model, entitled *Federated Extra-Trees (FET)*. Based on it, a secure multi-institutional machine learning system was developed to support the real-world applications accurately, robustly and safely. We have four major contributions:

- **Lossless-ness** (accuracy) was guaranteed under the horizontal federated scenarios. Although the randomness was introduced in several stages, our model was proved to have the same level of accuracy as the non-federated approach that brings the data into one place.
- **Data privacy was secured** by embedding local differential privacy (LDP) into the Federated Extra-Trees, as well as establishing a third-party trusty server to coordinate and monitor the entire modeling process.
- **High efficiency** was achieved by introducing more random process into the tree building and our model is robust to the complicated network environments. Only necessary and privacy-free modeling information was exchanged and the message size was reduced to a minimum.
- The total solution is **practical, extensible, scalable and explainable** to handle the *data island* problem and can be easily deployed for real-life applications.

2 Related Work and Preliminaries

Federated Learning

In the work of (Yang et al. 2019), they have provided a clear definition for the federated machine learning and how it distinguishes from other research subjects, such as distributed machine learning, secure multi-party computation, etc. Generally, we can categorize the FML methods into three types, horizontal federated learning, vertical federated learning and federated transfer learning. The horizontal FML (McMahan et al. 2017; Konecný et al. 2016b; 2016a; Chen et al. 2018; Yao et al. 2019) is focused on solving problems with data from different sample space but same feature space. The vertical FML (Hardy et al. 2017; Cheng et al. 2019; Liu et al. 2019) is the opposite, which works on problems with the same sample space but different feature space. The federated transfer learning (Liu, Chen, and Yang 2018) is mainly about tasks that data from different organizations are overlapped in both sample and feature space, but still mostly different from each other. Currently, most FML methods were developed for solving data problems under horizontal scenarios. Google applied FML in applications like on-device item ranking and next word prediction (Bonawitz et al. 2019). In the work of (Smith et al. 2017) the researchers applied FML to solve multi-task problems and a novel federated recommender system was proposed in the work of (Chen et al. 2018). An agnostic federated learning model was presented in the work of (Mohri, Sivek, and Suresh 2019) and it focused on the natural target distribution formed by the mixed clients.

Privacy Preserving

Homomorphic Encryption (HE) (Rivest, Adleman, and Dertouzos 1978; Acar et al. 2018) and Differential Privacy (DP) (Dwork 2008) are two mostly applied privacy-preserving methods in the federated learning. The HE differs from other encryption methods because it supports secure multiplication and addition on encrypted data. Once the computation result is decrypted, it matches the output of the same operations on the original raw data. Examples of applications include (Boemer et al. 2019; Hardy et al. 2017; Phong et al. 2018; Kim et al. 2018). However, the recognized weakness of HE is a slow speed, so it lacks practicality in many scenarios.

Differential privacy (DP) aims to minimize the possibility of individual identification to ensure user-level privacy (Kairouz, Oh, and Viswanath 2014). DP has been used extensively in machine learning tasks against privacy inference attacks. Existing work mainly focuses on adding perturbations on parameters in the gradient descent algorithms (Song, Chaudhuri, and Sarwate 2013; Abadi et al. 2016; Geyer, Klein, and Nabi 2017). DP methods typically take a balance between privacy and utility. Recent work (Jayaraman and Evans 2019) has investigated into the trade-offs for learning tasks. Global differential privacy (GDP) and local differential privacy (LDP) are the two main classes of DP. The existing DP approaches in federated learning are mostly GDP, where a trusted curator will apply calibrated noise on the aggregated data to provide differential privacy. Conversely, LDP mechanisms, where owners will perturb their data before aggregation, provide better privacy without trusting any third party as a curator. LDP applications is on the rise in federated learning due to its higher privacy and simpler implementation (Bhowmick et al. 2018;

Zhao 2018; Chamikara et al. 2019). In this paper, we adopt a LDP method for privacy issues in the federated trees model.

3 Methodology

Learning Scenario

In this work, we focused on applying Federated Extremely Randomized Trees, abbreviated to Federated Extra-Trees, to solve horizontal distributed data problems, that all data providers have the same attribute set $\mathcal{F}$ but different sample space. Each data provider was considered as one institutional data domain and denoted as $\mathcal{D}_i$. The overall data domain is $\mathcal{D} = \{\mathcal{D}_1; \mathcal{D}_2; \cdots; \mathcal{D}_M\}$, where $1 \leq i \leq M$ and M is the number of institutional domains. On each data domain, we have $\mathcal{D}_i = \left(\left(x_i^1, y_i^1\right), \left(x_i^2, y_i^2\right), ..., \left(x_i^{n_i}, y_i^{n_i}\right)\right)$. Here x is the input sample and y is the corresponding label, $(x, y) \in (\mathcal{X}, \mathcal{Y})$ and n_i is the total number of samples in $\mathcal{D}_i$. We have deployed a master machine as the parameter server to coordinate the entire modeling process and assigned each institutional domain one client machine. Since we are trying to build FML models jointly on different organizations, M is usually small and in our work, we only consider situations when $M <= 10$. For more parties involved, the algorithm design could be much more different. The notations appeared in this paper are also shown in Table 1.

Notation	Description
M	number of institutional domains
$\mathcal{D}_i$	dataset held by client i
$\mathcal{D}$	entire dataset $\mathcal{D} = \{\mathcal{D}_1; \mathcal{D}_2; \cdots; \mathcal{D}_M\}$
$\mathcal{S}_i$	the random sampled subset of $\mathcal{D}_i$
i	the index of client
j	the index of feature
n_i	total number of samples on client i
f	sample feature/attribute
v	value of feature f
$\mathcal{F}$	entire attribute set $\mathcal{F} = \{f_1; f_2; \cdots; f_J\}$
L	size of the label set $\mathcal{F}$
(x_i^k, y_i^k)	k-th sample of client i
$\mathcal{X}$	input sample space
$\mathcal{Y}$	output labels
$\mathcal{Y}^e$	encoded labels $\mathcal{Y}$
B_i^k	Bloom Filter of k-th label on client i
$R_{i,t}^k$	t-th item of the instant random response string for k-th label on client i
$\mathcal{C}$	count of encoded labels $\mathcal{Y}^e$

Table 1: Notations

Problem Statement

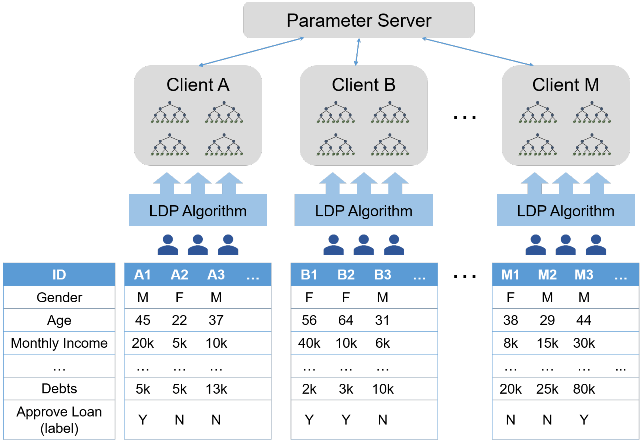
The formal statement of the problem is given as below:

- **Given:** Institutional data domain D_i on each client i, $1 <= i <= M$.
- **Learn:** Privacy-preserved Federated Extra-Trees.
- **Constraint:** The performance (accuracy, f1-score, etc) of the Federated Extra-Trees must be comparable to the non-federated approach.

Framework Overview

The framework of the Federated Extra-Trees is based on the previous work of Extra-Trees (Geurts, Ernst, and Wehenkel 2006), bagging (Breiman 1996) and local differential privacy (LDP) (Duchi, Wainwright, and Jordan 2013). The Extra-Trees model is naturally suitable for distributed machine learning because of its concise algorithm design and limited computation complexity and is an ideal solution for the federated scenarios. Our FET model is able to solve classification problems and without applying LDP it also supports regression tasks.

Let's look at the example in Figure 2, assume we have M clients and each of them provides information on their own loan application records, and we want to build an intelligent system to automatically decide if we should approve or reject the loan application. Before federated modeling, we applied LDP to transform the clients' labels into encoded binary strings, then all clients work together to build a complete forest that is available for subsequent use on every client. During the training, no raw data such as Gender, Age or others would be exposed. When modeling is finished, the model will be saved locally for subsequent use and no communication is necessary.



ID	A1	A2	A3	…	B1	B2	B3	…	M1	M2	M3	…
Gender	M	F	M		F	F	M		F	M	M	
Age	45	22	37		56	64	31		38	29	44	
Monthly Income	20k	5k	10k		40k	10k	6k		8k	15k	30k	
…	…	…	…		…	…	…		…	…	…	...
Debts	5k	5k	13k		2k	3k	10k		20k	25k	80k	
Approve Loan (label)	Y	N	N		Y	Y	N		N	N	Y	

Figure 2: Framework of the Federated Extra-Trees

Algorithms

We carefully extended the Extra-Trees to a distributed version with full consideration of privacy issues. In Extra-Trees, the optimal splitting under a certain feature is randomly selected instead of being calculated. With the idea of bagging, a forest can accommodate the errors caused by randomness in the single trees. Previous work has proved the scheme to be as accurate as of the ordinary tree models, and the training speed is greatly improved (Geurts, Ernst, and Wehenkel 2006). Therefore, this method is naturally suitable for federated scenarios. The training process of clients and master are described in Algorithm 1 and 2. All participants, including the master and clients, share the same feature set. Initially, each client samples data from its own training set $\mathcal{D}_i$. We constructed the trees in the depth-first search sequence. The key steps of building a tree are as follows.

Stopping criterion: Before creating a new tree node, participants will check if the stop conditions have been satisfied. Here we adopted a CART-tree (Breiman et al. 1984) like design. The stopping conditions are set by a maximum threshold for the depth of trees, a limit on the number of remaining samples in leaf nodes as well as other corner conditions.

ALGORITHM 1: Federated Extra-Tree – Client

```
Input  : Training set $\mathcal{D}_i$ of client $i$, feature set $\mathcal{F}$
Output: A Federated Extra-Tree
$\mathcal{S}_i \leftarrow$ subsample of $\mathcal{D}_i$ on client $i$;
Function build_tree ($\mathcal{S}_i, \mathcal{F}$)
    if $stopping\_condition$ is true then
        Mark current node as leaf node;
        Send $Sum_i$ ←label aggregation of $\mathcal{S}_i$ to master;
        Receive leaf labels from master;
        return leaf node;
    Receive feature candidate set $\mathcal{F}^*$ from master;
    for each feature $f_{i,j} \in \mathcal{F}^*$ do
        $v_{i,j}^{min}, v_{i,j}^{max} \leftarrow$ local min & max value of $f_{i,j}$;
        Send $v_{i,j}^{min}, v_{i,j}^{max}$ to master;
        Receive random split threshold $v_j^*$ from master;
        $\mathcal{S}_{i_L,j}, \mathcal{S}_{i_R,j} \leftarrow$ Split $\mathcal{S}_i$ by $v_j^*$ of feature $f_j$;
        $Sum_{i_L,j}, Sum_{i_R,j}$ ←label aggregation;
        Send $Sum_{i_L,j}, Sum_{i_R,j}$ to master;
    Receive global best split feature $f^*$ and value $v^*$;
    $\mathcal{S}_{i_L,j_*}, \mathcal{S}_{i_R,j_*} \leftarrow$ Split $\mathcal{S}_i$ by $v^*$ of feature $f^*$;
    left_subtree ← build_tree ($\mathcal{S}_{i_L,j_*}, \mathcal{F}$) ;
    right_subtree ← build_tree ($\mathcal{S}_{i_R,j_*}, \mathcal{F}$) ;
    return tree node
Append current tree to forest;
```

Random feature selection: Master is responsible for coordinating the collection of information from clients and decide which feature to use on a node. We inherited the randomness solution in Extra-Trees and extended it to the entire process of feature selection. Experiments in Section 4 have shown that the randomization does not necessarily lead to loss of precision. In order to create a new tree node, the master would randomly extract a candidate feature set $\mathcal{F}^* \subset \mathcal{F}$, and send it to all clients. Then the master collects each feature's maximum and minimum values from all clients and determines the overall range for it. After that, master arbitrarily selects a value in the range as the threshold for each feature and broadcasts the values to all clients.

Clients would split local data temporarily into left and right subtrees according to the received feature threshold, then sending perturbed information of the data labels to master. This process also increases the randomness. Receiving all the information of local subsets, the master would aggregate the data to calculate a $Gini\_Gain$ value for the feature. Feature f^* with the maximum $Gini\_Gain$ will be chosen as the best split feature for the current node. Clients should record the subsets for each feature. When the split feature f^* is finally determined, they could use the corresponding subsets directly to avoid repeated calculations.

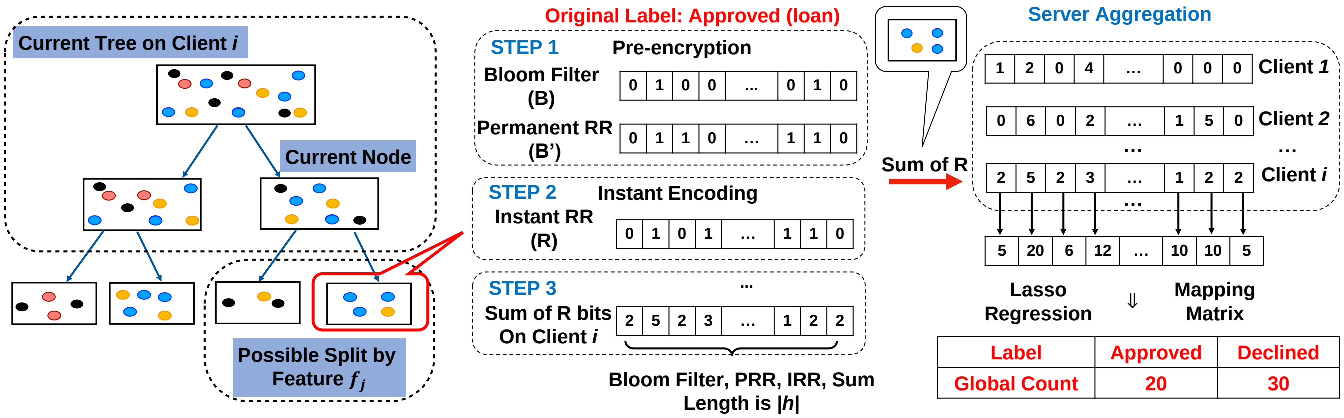
ALGORITHM 2: Federated Extra-Tree – Master

```
Input  : Feature set $\mathcal{F}$
Output: A Federated Extra-Tree
Function build_tree ($\mathcal{F}$)
    if $stopping\_condition$ is true then
        Mark current node as leaf node;
        Receive $Sum_i$ from client $i = 1, ..., M$;
        Send global $Sum$ to clients as leaf labels;
        return leaf node;
    $\mathcal{F}^* \subset \mathcal{F} \leftarrow$ Randomly-chosen feature subset;
    Send feature candidate set $\mathcal{F}^*$ to clients;
    for each feature $f_j \in \mathcal{F}^*$ do
        Gather $v_{i,j}^{min}, v_{i,j}^{max}$ from client $i = 1, ..., M$;
        $v_j^{min}, v_j^{max}$ ←global min & max value of $f_j$;
        Pick a random value $v_j^* \in [f_j^{min}, f_j^{max}]$;
        Broadcast $v_j^*$ to all clients;
        Gather $Sum_{i_L,j}, Sum_{i_R,j},\ i = 1, ..., M$;
        $\mathcal{C}_{L,j}, \mathcal{C}_{R,j} \leftarrow$ estimated global label counts;
        Calculate $Gini\_Gain(f_j)$ with ($\mathcal{C}_{L,j}, \mathcal{C}_{R,j}$);
    $f^* = \arg\max_{f_j} Gini\_Gain(\mathcal{F}^*)$ ;
    Broadcast the global best split feature $f^*$ and the corresponding split threshold $v^*$;
    left_subtree ← build_tree ($\mathcal{F}$) ;
    right_subtree ← build_tree ($\mathcal{F}$) ;
    return tree node
Append current tree to forest;
```

During the process, there are two kinds of data exchange between participants—the min-max values of f_j, and the information of labels $\mathcal{Y}$ in subsets. The former does not involve privacy issues since in horizontal settings, all participants are aware of the features by default. Therefore, we designed a privacy-preserving method for the sharing of labeling information.

Privacy Preserving Methods

In 3, the colors of dots represent labels. when the tree building process proceeds to a new tree node, the master sends a random-picked feature set to the clients and assigns corresponding threshold values according to their response. Mas-



Label	Approved	Declined
Global Count	20	30

Figure 3: Privacy-Preserving Methodology in Federated Extra-Trees

ter then needs to know the global label distribution of split data by those feature thresholds, so that it can calculate the $Gini\_Gain$ value for each feature f_j.

On the one hand, we hope that the algorithm will not reveal the category of any single user. On the other hand, we do not want the specific amount of each category on each client to be compromised. Here we modify an aggregation algorithm to protect privacy, which was first proposed by Google (Erlingsson, Pihur, and Korolova 2014) in the crowd-sourcing business and proven to be locally differential private. We implemented a multi-layer mechanism, including one Bloom Filter layer and two separate random-response based layers. Bloom Filter (Broder and Mitzenmacher 2004) is a randomized structure for representing a set in a space-efficient way. It adds extra uncertainty for user identification and compacts large data to reduce the communication traffic in federated scenarios.

STEP 1: Before the tree is created, the corresponding labels $\mathcal{Y}_i$ for samples in $\mathcal{D}_i$ will be pre-encoded into binary Bloom Filter strings. For k-th sample in $\mathcal{D}_i$, the label y_i^k maps to Bloom Filter B_i^k of size h using several hash factions. We set $h = 32$ and the bits corresponding to four hash function results of y_i^k are set to 1.

STEP 2: By adding random response perturbations, these binary strings are encrypted as permanent random responses (Permanent RR). Each bit in B_i^k would maintain the original value with probability p; otherwise it will be replaced by 0 or 1, as shown in Equation 1,

$$B_{i,t}^{k'} = \begin{cases} B_{i,t}^k, & P = p \\ 1, & P = 1/2(1-p) \\ 0, & P = 1/2(1-p) \end{cases} \qquad (1)$$

where $t = 1, 2, \ldots, h$. We set p to 0.5 in our mechanism, while the value could be tuned to control the perturbation performance. In the whole process of building trees, the original labels will never be used or transmitted. Instead, the permanent binary string $B^{'}$ is reused as a substitute.

STEP 3: For feature selection, another layer of temporary perturbation shall be added on $B_i^{k'}$ to get an instant random response value (Instant RR), denoted as R_k. Each bit $R_{i,t}^k$ is set to 1 with a certain probability, as is shown in Equation 2.

$$P(R_{i,t}^k = 1) = \begin{cases} \xi, & if\ B_{i,t}^{k'} = 1 \\ \zeta, & if\ B_{i,t}^{k'} = 0 \end{cases}, t = 1, 2, \ldots, h \qquad (2)$$

We set ξ to 0.8 and ζ to 0.2, so as to add a reasonable interference. Here ξ and ζ are independent of each other. Client i adds each bit of the local R values for users in $\mathcal{D}_i$,

$$Sum_{i,t} = \sum_{k=1}^{n_i} R_{i,t}^k, t = 1, 2, \ldots, h \qquad (3)$$

and sends the result array Sum_i to master.

Aggregation: As is shown in Figure 3, the master will aggregate the received results into Sum, with the count of each bit being:

$$Sum_t = \sum_{i=1}^{M} Sum_{i,t}, t = 1, \ldots, h \qquad (4)$$

With the label space mapped into $B_1, B_2, \ldots, B_L$, master estimates the overall label counts using linear estimation methods such as Lasso Regression.

In this scheme, the Permanent RR is already fixed, and the instant perturbation is calculated at an individual level without trusting any third party as a curator. It has been proven to be effective in protecting privacy (Erlingsson, Pihur, and Korolova 2014). Our LDP method also applies to other models that use statistics as an intermediate value.

If we adopt GDP in Federated Extra-Trees, the client adds a disturbance to the local labeling statistics and transmits it to master. Master adds the received statistics directly for further calculation. In this case, the clients must be fully trusted to be responsible for ensuring data privacy of end-users. Moreover, GDP causes larger fluctuations in the accuracy of results. In the experimental part, we also carried out a GDP-based method using the Laplace mechanism for comparison.

		Binary Classification			
Metric	Dataset	ET	FET	FET-LDP	FET-GDP
Accuracy	agaricus-lepiota	0.9750 ± 1.29E-02	**0.9806 ± 8.72E-05**	0.9804 ± 1.54E-05	0.9665 ± 1.61E-04
	spambase	0.9307 ± 4.06E-03	**0.9348 ± 1.50E-05**	0.9259 ± 1.03E-05	0.9339 ± 1.72E-05
	audit-data-2018	0.9833 ± 2.80E-03	**0.9859 ± 6.62E-06**	0.9794 ± 4.38E-05	0.9849 ± 6.31E-06
	credit-card	0.8509 ± 1.62E-03	0.8523 ± 1.07E-06	**0.8451 ± 3.32E-06**	0.8511 ± 7.28E-06
	bank-additional	0.9423 ± 1.03E-03	0.9414 ± 1.76E-06	**0.9334 ± 1.46E-06**	0.9220 ± 4.24E-04
F1 Score	agaricus-lepiota	0.9748 ± 1.38E-02	0.9804 ± 8.96E-05	**0.9804 ± 1.54E-05**	0.9661 ± 1.66E-04
	spambase	0.9316 ± 3.74E-03	**0.9355 ± 1.34E-05**	0.9259 ± 1.03E-05	0.9346 ± 1.55E-05
	audit-data-2018	0.9834 ± 2.75E-03	**0.9858 ± 6.40E-06**	0.9795 ± 4.38E-05	0.9850 ± 6.06E-06
	credit-card	0.8424 ± 1.64E-03	**0.8440 ± 1.53E-06**	0.8362 ± 5.13E-06	0.8426 ± 9.71E-06
	bank-additional	**0.9427 ± 1.07E-03**	0.9418 ± 1.84E-06	0.9335 ± 1.46E-06	0.9219 ± 4.46E-04
		Multiclass Classification			
Metric	Dataset	ET	FET	FET-LDP	FET-GDP
Accuracy	Waveform	0.8342 ± 4.71E-03	**0.8361 ± 1.88E-05**	0.8343 ± 2.26E-05	0.8361 ± 6.53E-05
	Letter	0.9527 ± 1.96E-03	**0.9553 ± 5.71E-06**	0.9240 ± 6.55E-06	0.9458 ± 7.55E-06
	KDD Cup 99	**0.9999 ± 2.57E-05**	0.9998 ± 3.31E-07	0.9974 ± 6.25E-07	0.9994 ± 2.35E-07
F1 Score	Waveform	0.8342 ± 4.71E-03	**0.8361 ± 1.88E-05**	0.8343 ± 2.26E-05	0.8361 ± 6.53E-05
	Letter	0.9527 ± 1.96E-03	**0.9554 ± 5.71E-06**	0.9240 ± 6.55E-06	0.9460 ± 7.55E-06
	KDD Cup 99	**0.9999 ± 2.57E-05**	0.9998 ± 3.31E-07	0.9974 ± 6.25E-07	0.9994 ± 2.35E-07

Table 2: Experimental Results

4 Experimental Studies

Experimental Setup

To verify the effectiveness of our algorithm and the utility of privacy-preserving methods, we designed comparative experiments of the following four algorithms:

- **Extra-Trees (ET)**: The non-federated implementation of the extremely randomized trees.
- **Federated Extra-Trees (FET)**: Our federated extremely randomized trees without perturbations on the input data.
- **FET-LDP**: Our Federated Extra-Trees with local differential privacy (random response mechanisms).
- **FET-GDP**: Our Federated Extra-Trees with global differential privacy (Laplace mechanisms).

We have carried out tests on various UCI datasets (Dua and Graff 2017) with a different number of samples and attributes for classification tasks. Numerical and categorical data were all considered. The datasets are shown in Table 3.

Dataset	Size	Feature Size	Labels
Audit-data-2018	777	18	2
Spambase	4601	57	2
Agaricus-lepiota	8124	22	2
Credit-card	30000	23	2
Bank-additional	45211	17	2
Waveform	5000	21	3
Letter-recognition	20000	16	26
Kdd-cup99	4000000	42	23

Table 3: Dataset Details

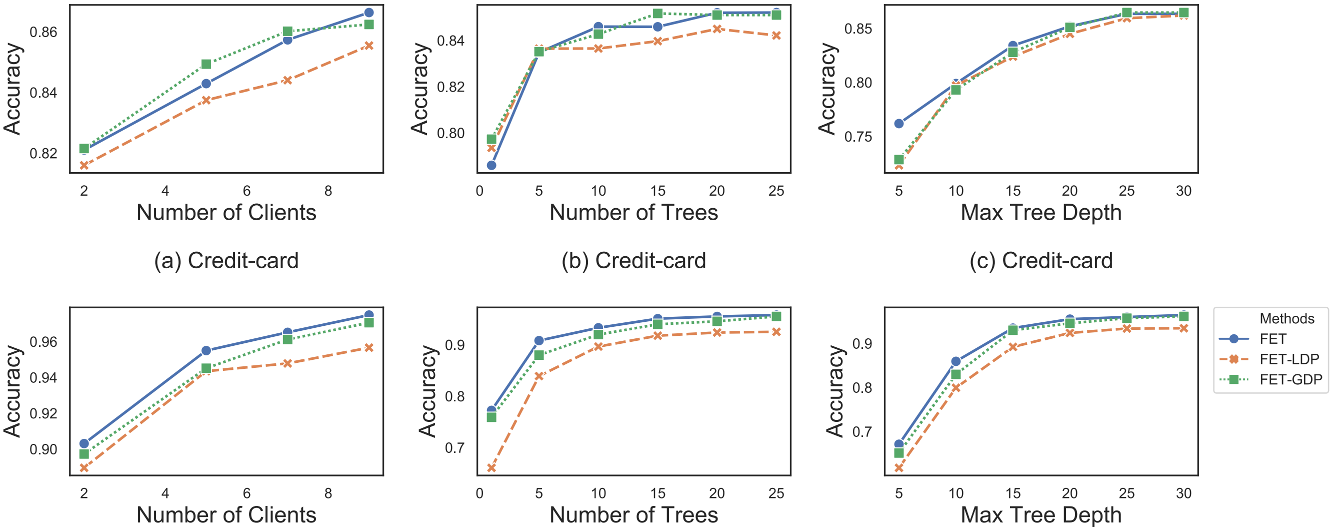
Each dataset was divided into a training set and a test set. The division ratio of the training set and the test set is 8:2, and for KDD Cup 99 dataset with a large amount of data, the ratio is 99:1. The evaluation criteria are the accuracy and F1 score. For multi-classification, the F1 score refers to the Micro F1 value.

The experiments of the classical ET are performed on a single node with the entire datasets. For the overall-performance evaluation, the distributed experiments are conducted on two client nodes and one master node, which is the minimum size of a parameter-server-based federated learning system. A dataset is randomly divided into several subsets and distributed to the clients to simulate the horizontal scenario. The sample space between the clients does not intersect. We also provide supplementary experiments to analyze the influence of the number of clients, the number of trees, and the maximum tree depth on the performance.

Overall Performance

The overall performance is shown in Table 2. Each experiment was repeated for 30 times, and the mean and variance of the accuracy and F1 score are given. We can see these points from the table 2:

- **Lossless-ness (accuracy):** On different scales of datasets, our algorithms have achieved comparable results to the non-federated Extra-Trees. For both binary and multiclass classification tasks, our FET method performs even better than the basic ET model on some datasets.
- **Utility of privacy-preserving methods:** As observed in most tests, methods with perturbations (FET-LDP & FET-GDP) have brought a loss to the accuracy, but overall this loss is acceptable. The performances of both FET-LDP and FET-GDP are close to that of FET. This is because the framework of Extra-Trees is inclusive and can accommodate the fluctuations caused by the perturbations. The other reason is that the perturbed data is used to select features, not directly involved in numerical modeling.



(a) Credit-card

(b) Credit-card

(c) Credit-card

(d) Letter-recognition

(e) Letter-recognition

(f) Letter-recognition

Figure 4: Experimental Results on Parameter Impacts

- **Stability:** The variance values show the stability of FET-series algorithms compared with ET. Our LDP-based approach presented relatively smaller variances than the GDP method and maintained almost the same stability as the privacy-free FET. The datasets we used have covered a wide range of data volume and feature types, which in addition shows the adaptability of our algorithm.

To analyze the impacts of the number of clients and tree parameters on the models, we experimented the algorithmic efficiency of the three federated algorithms with different parameters on two datasets: the binary-class Credit-card dataset and the multi-class Letter-recognition dataset.

Effect of Client Numbers

In this set of experiments, the number of trees and the maximum tree depth were fixed to 20. We have randomly divided both datasets into 9 folds, and each was placed on one client. As is shown in Figure 4 (a,d), when we have more clients modeling together, there is a constant increase in the accuracy. This has supported our vision that by uniting more institutions a better modeling performance could be achieved. And because the LDP method was introduced and a parameter server was deployed, each client was blind to others and the data privacy was not compromised.

Effect of Tree Settings

We tested the effects of the number of trees and the maximum tree depth respectively. When experimenting with the effect on the number of trees, the maximum tree depth is set to 20, and vice versa. By observing the change of the fold lines in Figure 4, we could find:

- **The number of trees:** The accuracy rate has been greatly improved from a single tree to multiple trees, which demonstrates the advantages of forest structure as we mentioned in Section 3. However, in the comparison of multiple trees, the increase in the tree numbers has minimal impact on the results.
- **The maximum tree depth:** The maximum tree depth has a greater influence on the results. The accuracy of Federated Extra-Trees models is rising continuously as the tree depth threshold grows. When the maximum depth reaches 20 or so, the model converges.

5 Conclusions

In this paper, we proposed a novel privacy-preserving federated machine learning method, called Federated Extra-Trees, which achieves a lossless performance on the modeling accuracy and protects the data privacy. We also developed a secure multi-institutional federated learning system which allows the modeling task can be jointly processed across different clients with the same attribute sets but different user samples. The raw data on each client will never be exposed, and only limited amount of intermediate modeling values were exchanged to reduce the communication and secure the data privacy. The introduction of local differential privacy and a third-party trusted server strengthens privacy protection and makes it impossible to backdoor the actual statistical information from the clients. We set up multiple clients to simulate the real-world situations and performed experiments on the public UCI datasets. The experimental results presented a superior performance for the classification tasks, and the lossless criterion was satisfied by comparing to the non-federated approach that requires data gathered in one place. We also proved that the introduction of local differential privacy does not affect the overall performance. The efficiency and robustness of our proposed system were also verified. To summarize, the Federated Extra-Trees successfully solved the *data island* problem and provided a brand new approach to protect the data privacy while realizing the cross-institutional collaborative machine learning, and it is strong practical for real-world applications.

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, 308–318. New York, NY, USA: ACM.

Acar, A.; Aksu, H.; Uluagac, A. S.; and Conti, M. 2018. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv.* 51(4):79:1–79:35.

Bhowmick, A.; Duchi, J. C.; Freudiger, J.; Kapoor, G.; and Rogers, R. 2018. Protection against reconstruction and its applications in private federated learning. *CoRR* abs/1812.00984.

Boemer, F.; Costache, A.; Cammarota, R.; and Wierzynski, C. 2019. ngraph-he2: A high-throughput framework for neural network inference on encrypted data. *IACR Cryptology ePrint Archive* 2019:947.

Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konecný, J.; Mazzocchi, S.; McMahan, H. B.; Overveldt, T. V.; Petrou, D.; Ramage, D.; and Roselander, J. 2019. Towards federated learning at scale: System design. *CoRR* abs/1902.01046.

Breiman, L.; Friedman, J.; Stone, C.; and Olshen, R. 1984. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis.

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24(2):123–140.

Brock, A.; Donahue, J.; and Simonyan, K. 2019. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Broder, A. Z., and Mitzenmacher, M. 2004. Network applications of bloom filters: A survey. *Internet Mathematics* 1(4):485–509.

Chamikara, M. A. P.; Bertok, P.; Khalil, I.; Liu, D.; and Camtepe, S. 2019. Local differential privacy for deep learning. *CoRR* abs/1908.02997.

Chen, F.; Dong, Z.; Li, Z.; and He, X. 2018. Federated meta-learning for recommendation. *CoRR* abs/1802.07876.

Cheng, K.; Fan, T.; Jin, Y.; Liu, Y.; Chen, T.; and Yang, Q. 2019. Secureboost: A lossless federated learning framework. *CoRR* abs/1901.08755.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805.

Dua, D., and Graff, C. 2017. UCI machine learning repository.

Duchi, J.; Wainwright, M. J.; and Jordan, M. I. 2013. Local privacy and minimax bounds: Sharp rates for probability estimation. In *Advances in Neural Information Processing Systems*, 1529–1537.

Dwork, C. 2008. Differential privacy: a survey of results. In *TAMC'08 Proceedings of the 5th international conference on Theory and applications of models of computation*, volume 4978, 1–19.

Erlingsson, ; Pihur, V.; and Korolova, A. 2014. Rappor. *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*.

Geurts, P.; Ernst, D.; and Wehenkel, L. 2006. Extremely randomized trees. *Machine Learning* 63:3–42.

Geyer, R. C.; Klein, T.; and Nabi, M. 2017. Differentially private federated learning: A client level perspective. *CoRR* abs/1712.07557.

Hardy, S.; Henecka, W.; Ivey-Law, H.; Nock, R.; Patrini, G.; Smith, G.; and Thorne, B. 2017. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *CoRR* abs/1711.10677.

Jayaraman, B., and Evans, D. 2019. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*, 1895–1912. Santa Clara, CA: USENIX Association.

Kairouz, P.; Oh, S.; and Viswanath, P. 2014. Extremal mechanisms for local differential privacy. In *Advances in neural information processing systems*, 2879–2887.

Kim, S.; Omori, M.; Hayashi, T.; Omori, T.; Wang, L.; and Ozawa, S. 2018. Privacy-preserving naive bayes classification using fully homomorphic encryption. In *International Conference on Neural Information Processing*, 349–358. Springer.

Konecný, J.; McMahan, H. B.; Ramage, D.; and Richtárik, P. 2016a. Federated optimization: Distributed machine learning for on-device intelligence. *CoRR* abs/1610.02527.

Konecný, J.; McMahan, H. B.; Yu, F. X.; Richtárik, P.; Suresh, A. T.; and Bacon, D. 2016b. Federated learning: Strategies for improving communication efficiency. *CoRR* abs/1610.05492.

Liu, Y.; Liu, Y.; Liu, Z.; Zhang, J.; Meng, C.; and Zheng, Y. 2019. Federated forest. *CoRR* abs/1905.10053.

Liu, Y.; Chen, T.; and Yang, Q. 2018. Secure federated transfer learning. *CoRR* abs/1812.03337.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, 1273–1282.

Mohri, M.; Sivek, G.; and Suresh, A. T. 2019. Agnostic federated learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, 4615–4625.

Phong, L. T.; Aono, Y.; Hayashi, T.; Wang, L.; and Moriai, S. 2018. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security* 13(5):1333–1345.

Rivest, R. L.; Adleman, L.; and Dertouzos, M. L. 1978. On data banks and privacy homomorphisms. *Foundations of Secure Computation, Academia Press* 169–179.

Sheller, M. J.; Reina, G. A.; Edwards, B.; Martin, J.; and Bakas, S. 2018. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. *International MICCAI Brainlesion Workshop* 92–104.

Smith, V.; Chiang, C.; Sanjabi, M.; and Talwalkar, A. S. 2017. Federated multi-task learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, 4424–4434.

Song, S.; Chaudhuri, K.; and Sarwate, A. D. 2013. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, 245–248.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10(2):12.

Yao, Q.; Guo, X.; Kwok, J. T.; Tu, W.; Chen, Y.; Dai, W.; and Yang, Q. 2019. Privacy-preserving stacking with application to cross-organizational diabetes prediction. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 4114–4120.

Zhao, J. 2018. Distributed deep learning under differential privacy with the teacher-student paradigm. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*.