FedXGBoost: Privacy-Preserving XGBoost for Federated Learning

Nhan Khanh Le[1]* , **Yang Liu**[2] , **Quang Minh Nguyen**[4] , **Qingchen Liu**[1] , **Fangzhou Liu**[3] , **Quanwei Cai**[2] and **Sandra Hirche**[1]
[1]Chair of Information-Oriented Control, Technical University of Munich
[2]Security Research, Bytedance Inc.
[3]Chair of Automatic Control Engineering, Technical University of Munich
[4] Department of EECS, Massachusetts Institute of Technology
liuyang.fromthu@bytedance.com, caiquanwei@bytedance.com, {nhankhanh.le, qingchen.liu, fangzhou.liu, hirche}@tum.de, nmquang@mit.edu

Abstract

Federated learning is the distributed machine learning framework that enables collaborative training across multiple parties while ensuring data privacy. Practical adaptation of XGBoost, the state-of-the-art tree boosting framework, to federated learning remains limited due to high cost incurred by conventional privacy-preserving methods. To address the problem, we propose two variants of federated XGBoost with privacy guarantee: *FedXGBoost-SMM* and *FedXGBoost-LDP*. Our first protocol *FedXGBoost-SMM* deploys enhanced secure matrix multiplication method to preserve privacy with lossless accuracy and lower overhead than encryption-based techniques. Developed independently, the second protocol *FedXGBoost-LDP* is heuristically designed with noise perturbation for local differential privacy, and empirically evaluated on real-world and synthetic datasets.

1 Introduction

As a distributed machine learning (ML) model, FL benefits from the variety of multiple data holders and facilitates collaborative training. Its widespread usage can be found in credit card fraud detection [Yang *et al.*, 2019b], banking prediction [Shingi, 2020], and health care application by [Xu and Wang, 2019]. Nevertheless, participants of a FL model are required to share the knowledge of their data, leading to the threat of privacy leakage. *Privacy-preserving* in *Federated Learning* (FL) is thus one of the key challenges. Several anonymization approaches that cover users' identification are shown to be insufficient [Narayanan and Shmatikov, 2006]. Furthermore, the European Union recently imposed General Data Protection Regulation (GDPR) to increase the privacy protection of user's private data. Therefore, any FL framework must satisfy privacy-preserving criteria while offering high-quality ML service. An informative overview of privacy-preserving under FL is provided in [Yang *et al.*, 2019a].

XGBoost - Gradient boosted decision trees by [Chen and Guestrin, 2016] is a novel tree ensemble model that achieves state-of-the-art results on a variety of machine learning problems. In this paper, we aim to bring the benefits of XGBoost to the FL settings with privacy-preserving guarantees. Many existing work of FL-based gradient boosting methods require different types of encryption-based protocols: homomorphic encryption [Liu *et al.*, 2020] [Aono *et al.*, 2016] [Fang *et al.*, 2020a], secret-sharing [Fang *et al.*, 2020b] and locality sensitivity hashing [Li *et al.*, 2019], all of which result in significant communication/computation overhead. Applying *Homomorphic Encryption (HE)* , [Cheng *et al.*, 2019] provides SecureBoost that offers high degree of privacy-preserving but requires high communication cost. Other approaches utilize *Differential Privacy (DP)* and perform the analysis directly with the perturbed data as studied in [Li *et al.*, 2021] , [Shi *et al.*, 2021]. Despite the reduction in training time, the model suffers accuracy loss by the injected noise. Our approach deviates from all the previous work. We study a protocol that has lossless accuracy and achieves a compromise between model complexity and privacy-preserving. We first formulate the evaluation of splitting score, the major step requiring privacy guarantee in XGBoost, as the multiplication of a categorical matrix and a vector, and then deploy a modified version of secure matrix multiplication (SMM) introduced in [Karr *et al.*, 2007]. We show that if SMM is naively applied, due to its categorical entries the privacy guarantee of the matrix can be violated. We further point out the analogous scenario of privacy leakage that could be inherent in HE, yet neglected by the literature. To address the challenge, we enhance the SMM protocol to provide additional deniability and propose FedXGBoost-SMM, XGBoost for FL over vertically partitioned data with privacy-preserving guarantee. In addition, utilizing noise perturbation with local differential privacy (LDP), we introduce FedXGBoost-LDP, which is a heuristic for practical perspective. Our contributions can be summarized as follows:

- **FedXGBoost-SMM:** a linear algebra based approach designed to achieve lossless model accuracy and more efficient in comparison to HE. We vertically encode the categorical matrix and show that the extremely curious party can infer the true value of the matrix with low

*Contact Author

probability, while the overhead is negligible. We provide modifications to enhance the privacy-preserving.

- **FedXGBoost-LDP**: a heuristically designed protocol with LDP that yields acceptable practical results.
- **Practicality:** We experiment on real-world data and evaluate the utility of the heuristic FedXGBoost-LDP.

The remainder of the paper is organized as follows. Section 2 represents the preliminaries and the problem statement. Section 3 introduces the applied privacy-preserving techniques and analyzes the potential information leakage of the proposed algorithms. Section 4 describes the procedures of FedXGBoost. The experiments and the evaluation of the protocols are provided in section 5. Section 6 concludes the study.

2 Preliminaries

2.1 XGBoost - Gradient Tree Boosting

2.1.1 Regularized Learning Objective of Tree Boosting

Given a dataset $D = \{(x_i, y_i), x_i \in \mathbb{X}, y_i \in \mathbb{R}\}$, x_i denotes feature vectors in feature space $\mathbb{X}$ and y_i is the label of the i^{th} instance. Let $n = |D|$ be the total amount of instances and K be the amount of constructed regression trees. We have a regression model $\phi(.)$ for an instance $x_j \in \mathbb{X}$ from multiple regression trees as follows

$$\hat{y}_j = \phi(x_j) = \sum_{k=1}^{K} f_k(x_j),\ f_k \in \mathcal{F}, \tag{1}$$

The set of regression trees $\mathcal{F}$ is defined as

$$\mathcal{F} = \{f(x) = w_{q(x)}\},\ q : \mathbb{R}^m \to T, w \in \mathbb{R}$$

where q denotes the tree structure that maps the instance to an unique leaf, w is a weight of leaf, and T is the amount of leaves of one tree. For any differentiable convex loss function $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, the objective function $\mathcal{L}(\phi)$ for the model training process is defined as

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k} \Omega(f_k) \tag{2}$$

in which the term $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \left\| w \right\|^2$ is the regularization term to avoid over-fitting.

2.1.2 XGBoost: Regression Tree Boosting

[Chen and Guestrin, 2016] applied iterative optimization procedure to minimize the objective function (2). At the t^{th} iteration, new tree f_t is constructed and contributes to the regression model. Therefore, the objective function at the t^{th} iteration is formulated as

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} (l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{3}$$

The learning process aims to minimize the second order approximation of the objective function

$$\min_{f_t} \sum_{i=1}^{n} (l(y_i, \hat{y}^{(t-1)}) + g_i f_t(x_i) + \frac{h_i f_t^2(x_i)}{2}) + \Omega(f_t) \tag{4}$$

where $g_i = \partial_{\hat{y}_{(t-1)}} l$ and $h_i = \partial^2_{\hat{y}_{(t-1)}} l$ are the first and second derivative of the loss function at $\hat{y}_{(t-1)}$, respectively. Then the optimal candidate is selected to split the instances into left and right nodes. [Chen and Guestrin, 2016] used the following score to evaluate the splitting candidates

$$\mathcal{L}_{split} = -\gamma + \frac{1}{2}\left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda}\right] \tag{5}$$

where I_L, I_R indicate the obtained left and right nodes by splitting the node I. Then, it continues to split from the new constructed nodes until reaching the maximum depth of the tree. The node at the last layer is the tree leaf that represents the weight for the common instance in this node. The leaf weight w is equivalent to the prediction of a new tree and contributes to the minimization of (4) via $f_t = w$.

2.2 Federated Learning (FL) over vertically partitioned data

Our proposed protocols focus on FL over vertically partitioned data, in which multiple databases own different features of the same sample instances. We introduce the concept of Active Party and Passive Party proposed by [Cheng *et al.*, 2019] through the following definition

Definition 1. *Active Party and Passive Party*

- ***Active Party (AP)**: The party that holds both feature data and the class label.*
- ***Passive Party (PP)**: The data provider party, which has only the feature data.*

The protocol studied by [Liang and S.Chawathe, 2004] is applied to determine the common database intersection between participants and align the features and class label with the corresponding sample ID securely. The main concern of XGBoost under this configuration is how to conduct the training process jointly between participants with the aligned database.

2.3 Secure Multi-party Computation (SMC)

[Cramer *et al.*, 2015] defines SMC as techniques that allow multiple participants to compute accurately the final output without revealing private information. SMC protocols require the participants to follow a particular procedure that leads to the final result and guarantee under some assumptions that the private data can not be reconstructed. The SMC protocol applied in this paper is the Secure Matrix Multiplication (SMM) protocol motivated by [Karr *et al.*, 2007] as follows

Protocol 1. ***SMM by [Karr et al., 2007]***
Let the parties A and B possess the private data matrix $D^A \in \mathbb{R}^{n \times m}$ and $D^B \in \mathbb{R}^{n \times l}$, respectively. They want to obtain the result $S = (D^A)^T D^B \in \mathbb{R}^{m \times l}$ without knowing the private information of the other participant.

1. *Party A finds the set $\mathcal{U} = \{u_i \in \mathbb{R}^n | (D^A)^T u_i = 0\}$, which contains orthonormal null space vectors of $(D^A)^T$. Then it selects r kernel vectors to construct the matrix $Z = [u_1 \cdots u_r] \in \mathbb{R}^{n \times r}$. We have*

$$(D^A)^T Z = \boldsymbol{0}^{m \times r}$$

2. *Party A sends the matrix Z to party B, then party B computes the matrix $W \in \mathbb{R}^{n \times l}$ as*
$$W = (I^{n\times n} - ZZ^T)D^B$$
3. *Party B sends W back to party A to compute the true result of the multiplication by*
$$(D^A)^T W = (D^A)^T(I^{n\times n} - ZZ^T)D^B = (D^A)^T D^B$$

The designed protocol assumes that all participants are semi-honest, i.e., they honestly follow the protocol but they are curious about the private data of other participants. The private data can not be uniquely reconstructed from the data exchange of $Z,\ W$ in step 2 and 3 thanks to the rank deficiency of the linear equation systems.

2.4 Local Differential Privacy (LDP)

Differential Privacy by [Dwork and Roth, 2014] guarantees privacy-preserving by introducing deniability in the private data. Informally, DP mechanisms inject calibrated noise into the query of the private dataset to make individual's data indistinguishable. This allows statistical analysis to be conducted while individual data is protected. A stronger DP approach is LDP, which perturbs the private data directly.

Definition 2. *LDP, [Úlfar Erlingsson* et al.*, 2014]*
The perturbation mechanism $\pi(.)$ satisfies ϵ- local differential privacy if for any two input $t,\ t'$ in the domain of $\pi(.)$ and any output t^* in the range of $\pi(.)$, there is an $\epsilon > 0$ that
$$Pr[\pi(t) = t^*] \leq exp(\epsilon)Pr[\pi(t') = t^*]$$

LDP perturbation mechanisms are studied in [Duchi *et al.*, 2013], [Wang *et al.*, 2019]. In comparison to SMC, LDP methods offer efficient computational costs with trade-off in model accuracy due to injected noise. This motivates our design of FedXGBoost-LDP, a heuristic approach that finds the compromise between model complexity and accuracy.

3 Main Results

To construct the new tree nodes, passive parties first (PP) analyze their user's distribution according to their feature data. They they propose splitting candidates that separate the current node (the set of users being analyzed) into left and right nodes. The distribution analysis over a large data set is executed by the efficient Approximate Quantile algorithms as studied in [Karnin *et al.*, 2016], [Li *et al.*, 2008], [Tyree *et al.*, 2011]. Afterward, the optimal splitting candidate is determined by comparing the loss reduction between splitting candidates as shown in (5). For brevity, in XGBoost under FL settings, PP plays a role as a splitting candidate owner. On the other side, the active party with the private class label owns the confidential gradient and hessian values of users. During the regression learning, AP and PP desire to compute the optimal splitting candidate securely.

Remark 1. *The description of the Approximate Quantile algorithm to analyze the users' distribution and propose splitting candidates are introduced in [Chen and Guestrin, 2016]. Note that we apply the trivial Quantile algorithm instead of the Weighted Quantile Sketch.*

We introduce the concept of the splitting operator and splitting matrix to formulate the private information of the passive parties as follows

3.1 Splitting Matrix - Passive Party's Private Data

We simplify the notations during the formulation by considering only one AP that has the true label of users, and one PP that constructs the splitting matrix based from one feature. The complete procedure for multiple PP with many features are described in section 4.

Definition 3. *Splitting Operator and Splitting Matrix*
Let $\mathbb{X}$ be the feature space, $f^k = [x_1\ x_2 \cdots x_n]^T \in \mathbb{X}^n$ be the values of the k^{th} feature of n users, and $\mathcal{S} = \{s_1, s_2, \cdots, s_l\} \subseteq \mathbb{X}^l$ be the set of l splitting candidates. The splitting operator Split$(f^k, \mathcal{S}) : \mathbb{X}^n \times \mathbb{X}^l \longrightarrow \{0,1\}^{n\times l}$ performs the splitting operation by comparing all feature data with all splitting candidates and outputs the splitting matrix $M \in \{0,1\}^{n\times l}$ as

$$M = \textit{Split}(f^k, \mathcal{S}) = \begin{pmatrix} u_{1s_1} & u_{1s_2} & \cdots & u_{1s_l} \\ \vdots & \vdots & \vdots & \vdots \\ u_{ns_1} & u_{ns_2} & \cdots & u_{ns_l} \end{pmatrix}, \textit{where}$$
$$u_{is_j} = \begin{cases} 1 & , x_i \leq s_j \\ 0 & , x_i > sj \end{cases}$$

Each column of the splitting matrix represents one splitting candidate, which labels any user as "1" if they belong to the left node and "0" for the right node. The following example depicts the functionality of the splitting operator and the splitting matrix.

Example 1. *Let the feature vector be $f^k =$ $[1,\ 11,\ 9,\ 4,\ 2,\ 12,\ 17,\ 13,\ 5]^T$. The PP proposes a set of three splitting candidates $\mathcal{S} = \{s_1 = 11,\ s_2 = 6,\ s_3 = 12\}$. The result of the splitting operator applied to the feature vector f^k and the candidate set $\mathcal{S}$ is the following splitting matrix M*

$$M = \textit{Split}(f^k, \mathcal{S}) = \begin{pmatrix} 1\,1\,1\,1\,1\,0\,0\,0\,1 \\ 1\,0\,0\,1\,1\,0\,0\,0\,1 \\ 1\,1\,1\,1\,1\,1\,0\,0\,1 \end{pmatrix}^T$$

According to (5), the evaluation of each splitting candidates require the aggregated gradients and hessians of instances in the left and right nodes. These are computed through the multiplication of the splitting matrix and the data vector. Particularly, let $i \in \{1 \cdots n\}$ denote the user index, $\mathcal{N}$ indicate the set of all users being analyzed and $n = |\mathcal{N}|$. The two subsets $\mathcal{N}_L,\ \mathcal{N}_R$ are obtained by the splitting operator, such that $\mathcal{N}_L \cap \mathcal{N}_R = \varnothing,\ \mathcal{N}_L \cup \mathcal{N}_R = \mathcal{N}$. Let the gradient and hessian vector of n users with numerical values be $g = (g_1 \cdots g_n)^T, h = (h_1 \cdots h_n)^T \in \mathbb{R}^n$. From the set $\mathcal{S} = \{s_1, \cdots, s_l\}$ that represents l splitting candidates, we use the index s_i to indicate one particular candidate. The aggregated gradients and hessians by **each splitting candidate** s_i are $G^L = (G^L_{s_1} \cdots G^L_{s_l})^T$, $H^L = (H^L_{s_1} \cdots H^L_{s_l})^T$ and $G^R = (G^R_{s_1} \cdots G^R_{s_l})^T$, $H^R = (H^R_{s_1} \cdots H^R_{s_l})^T$. They are computed from M, g, h as follows

$$\begin{aligned} G &= \sum_{i\in\mathcal{N}} g_i \in \mathbb{R},\ G^L = M^T g,\ G^R = G.\mathbf{1}^l - G^L \\ H &= \sum_{i\in\mathcal{N}} h_i \in \mathbb{R},\ H^L = M^T h,\ H^R = H.\mathbf{1}^l - H^L \end{aligned} \tag{6}$$

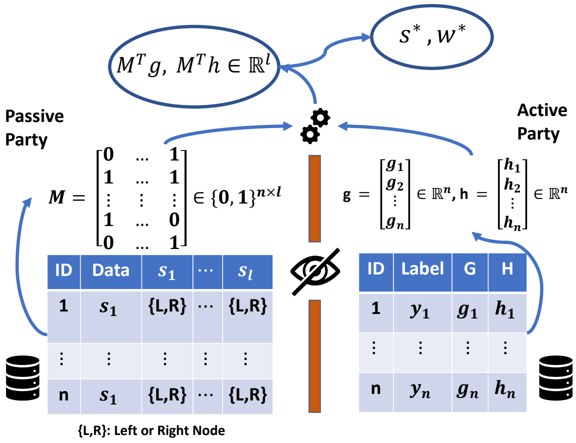
$M^T g,\ M^T h \in \mathbb{R}^l$
s^*, w^*
Passive Party
Active Party
$M = \begin{bmatrix} 0 & \dots & 1 \\ 1 & \dots & 1 \\ \vdots & \vdots & \vdots \\ 1 & \dots & 0 \\ 0 & \dots & 1 \end{bmatrix} \in \{0,1\}^{n \times l}$
$g = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix} \in \mathbb{R}^n$, $h = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix} \in \mathbb{R}^n$
ID | Data | s_1 | ⋯ | s_l
1 | s_1 | {L,R} | ⋯ | {L,R}
n | s_1 | {L,R} | ⋯ | {L,R}
ID | Label | G | H
1 | y_1 | g_1 | h_1
n | y_n | g_n | h_n
{L,R}: Left or Right Node

Figure 1: From Federated Learning XGBoost to SMM

After obtaining the result of the matrix multiplication, each candidate s_i is then evaluated by comparing the splitting score $\mathcal{L}^{s_i}_{split}$ from (5). The best candidate s^* has the highest splitting score.

$$s^* = \arg\max_{s_i \in \mathcal{S}} \frac{1}{2}\left[\frac{(G^L_{s_i})^2}{(H^L_{s_i}) + \lambda} + \frac{(G^R_{s_i})^2}{(H^R_{s_i}) + \lambda} - \frac{G^2}{H + \lambda}\right] - \gamma \tag{7}$$

If a splitting operator constructs the last layer of the tree, i.e., it constructs the tree leaves, the optimal weight for each leaf is computed as shown in [Chen and Guestrin, 2016] as

$$w^*_L = -\frac{G^L_{s^*}}{H^L_{s^*} + \lambda}, \quad w^*_R = -\frac{G^R_{s^*}}{H^R_{s^*} + \lambda} \tag{8}$$

Remark 2. *The previous studies use the addition operation to compute the aggregated gradients and hessians between multiple parties. We instead introduce the splitting matrix to mathematically formulate the private data and the functionality of the PP.*

3.2 Problem formulation - The relation between FedXGBoost and Secure-Matrix Multiplication

From the concept of the splitting matrix, we reformulate the challenges of FedXGBoost into an SMC problem that can be solved by SMM protocols. Particularly, according to (6), the AP has $g,\ h$, while the PPs own M and they desire to compute $M^T g,\ M^T h$ securely. Then the results are applied to (6) and (7) to find the best splitting candidate and the optimal leaf weights. Figure 1 illustrates the concept. The next subsections use the introduced notations of $M \in \{0,1\}^{n \times l},\ g,\ h \in \mathbb{R}^n$ to design two secure protocols, which are FedXGBoost-SMM and FedXGBoost-LDP, and analyze the information leakage of these two protocols.

3.3 FedXGBoost-SMM

FedXGBoost-SMM is motivated by the SMM protocols that allow the participants to determine the optimal splitting candidate securely. Applying the Protocol 1, D^A represents the gradient and the hessian $g,\ h$ vectors of the AP or combined as a matrix $[g\ h] \in \mathbb{R}^{n \times 2}$, while D^B is the splitting matrix of the PP.

Algorithm 1 FedXGBoost-SMM

Input:

- Active party (**AP**) has $g,\ h \in \mathbb{R}^n$
- Total p passive parties (**PPs**), k^{th} PP has splitting matrix $M^k \in \{0,1\}^{n \times l}$

Output of protocols: The optimal splitting operation
Procedures:

```
AP : Find the set orthonormal null-space vectors of $[g\ h]^T$: $\mathcal{U} \leftarrow \{u_i \in \mathbb{R}^n | [g\ h]^T(u_i) = 0\},\ |\mathcal{U}| = r$
AP : Transmit $\mathcal{U}$ to all PPs
PP$_k$: $W^k \in \mathbb{R}^{n \times l} \xleftarrow{Algo.2}$ Secure-Response($M^k,\ \mathcal{U}$)
PP$_k$ : Transmit $W^k$ to AP
AP : $G \leftarrow \sum_{i=1}^n g_i,\ H \leftarrow \sum_{i=1}^n h_i$
/* AP finds optimal score from all PP */
$(L^k)^* \leftarrow -\infty,\ (s^k)^* \leftarrow 0$
for k = 1 to p do
  for i = 1 to l do
    $\{G^L_{s_i}, H^L_{s_i}\} \leftarrow \{(W^k)^T g\}_i, \{(W^k)^T h\}_i$
    $G^R_{s_i} \leftarrow G - G^L_{s_i},\ H^L_{s_i} \leftarrow H - H^L_{s_i}$
    $L \leftarrow \frac{1}{2}\left[\frac{(G^L_{s_i})^2}{H^L_{s_i}+\lambda} + \frac{(G^R_{s_i})^2}{H^R_{s_i}+\lambda} - \frac{G^2}{H+\lambda}\right]$
    if $(L^k)^* < L$ then
      $(L^k)^* \leftarrow L,\ (s^k)^* \leftarrow i$
    end if
  end for
end for
AP: $k^* = \text{argmax}_{k \in \{1 \cdots p\}} L^k,\ (s^k)^*$
```

Return: PP$_{k^*}$ and its optimal splitting operation

3.3.1 Protocol description

Algorithm 1 describes the procedure of FedXGBoost-SMM. The AP first determines the set of users being analyzed. Then it announces this set to all PPs and transmits the generated orthonormal null-space vectors of $[g\ h]^T \in \mathbb{R}^{n \times 2}$. Each PP analyzes the feature of the announced user set and constructs its private splitting matrix. Then it applies Algorithm 2 to generate a secure response $W \in \mathbb{R}^{n \times l}$. The AP requests W from all PPs to compute the aggregated gradients and hessians of each splitting candidate, which are the elements of $W^T g,\ W^T h \in \mathbb{R}^l$. Afterwards, it computes the splitting score and finds the optimal score between all PPs. The PP with optimal score is requested to reveal the corresponding splitting operation. The AP then constructs new nodes and repeats the process with the new set of users.

3.3.2 Analysis of privacy-preserving

The study of SecureBoost provides a thorough analysis of potential privacy leakage for the AP. In our study, we focus on the private splitting matrix of the PPs. Despite these matrices do not contain the real feature data, they reveal the user's distribution. Such information allows the curious party to infer the range of feature values. The potential privacy leakage of FedXGBoost-SMM are caused by 1) The revealed null-space vectors set $\mathcal{U}$ of AP; 2) The response W of PP; 3) AP knows the aggregated gradients and hessians of all splitting candidates;

```
Algorithm 2 Secure-Response
Input:
  • Private data $X \in \mathbb{R}^{n \times p}$
  • Received orthonormal $\mathcal{U} = \{u_i \in \mathbb{R}^n, i \in \{1, \cdots, r\}\}$
Output: $W \in \mathbb{R}^{l \times n}$
Procedures:
  1: Select random $r'$ vectors $u_i \in \mathcal{U}$, with $r' \leq r$
  2: $Z \leftarrow [u_1 \cdots u_{r'}] \in \mathbb{R}^{n \times r'}$
  3: $W \leftarrow (I^{n \times n} - ZZ^T)X \in \mathbb{R}^{n \times p}$
Return: $W$
```

1. *PP knows the null-space vectors of AP*
   Let the set of r orthonormal null-space vectors construct a matrix $U \in \mathbb{R}^{n \times r}$, with $rank(U) = r,\ r \leq n-2$, reconstructing $[g\ h]$ is equivalent to the following problem

   Problem 1. *Find $x \in \mathbb{R}^{n \times 2}$ with a given $U \in \mathbb{R}^{n \times r}$, $rank(U) = r$ that satisfies*
   $$U^T x = 0^r \in \mathbb{R}^r \tag{9}$$

   There exist infinite solutions for (9) due to the rank deficiency of the linear equation system. The span of x can be inferred if $r \approx n-2$. However, it is sufficient to choose $1 \ll r \ll n-2$

2. *AP knows the response W from PP*
   As described in Algorithm 1, PP randomly selects r' vectors in $\mathcal{U}$ to construct Z, with $r' < r$. Then it computes
   $$W = (I^{n \times n} - ZZ^T)M \in \mathbb{R}^{n \times l} \tag{10}$$

   Property 1. *[Karr* et al.*, 2007] The constructed Z from the received orthonormal null-space vectors satisfies $rank(I^{n \times n} - ZZ^T) < n$ so it is not invertible.*
   In the original protocol, [Karr *et al.*, 2007] stated that the information of $rank(W)$ contributes to the privacy leakage. For this reason, the randomness introduced in Algorithm 2 conceals the information of ZZ^T.

3. *AP knows the aggregated gradients and hessians of all splitting candidates*
   Consider the AP is curious about M. The effort to reconstruct possible splitting candidates is equivalent to the following integer programming problem

   Problem 2. *Find $\{x_1, \cdots, x_n\} \in \{0, 1\}$ such that for a given $A = (a_1 \cdots a_n)^T \in \mathbb{R}^n$ and $b \in \mathbb{R}$ it satisfies*
   $$\sum_{i=1}^{n} x_i a_i - b = 0$$

   From our understanding, Problem 2 belongs to the set of NP-complete problems. This guarantees the privacy-preserving under the assumption of bounded computational capability. *This challenge also occurs in methods applying homomorphic encryption techniques, yet was not mentioned in the previous literature.* In SecureBoost, AP encrypts the gradients and hessians as $\langle g \rangle$ [1], $\langle h \rangle$ before transmitting these to PP. *Each PP aggregates the encrypted gradients and hessians and sends back to the AP, i.e., PP computes the multiplication of the splitting matrix and the encrypted vectors to obtain the encrypted $\langle M^T g \rangle$ and $\langle M^T h \rangle$, respectively.*

   Observation 1. *If there exists an efficient algorithm that solves Problem 2 in polynomial time, the splitting matrix can be reconstructed from the known aggregated gradients $\langle M^T g \rangle$ and hessians $\langle M^T h \rangle$.*

To this end, we conclude that FedXGBoost-SMM achieves equivalent privacy-preserving as SecureBoost that applies homomorphic encryption techniques.

3.3.3 Enhanced FedXGBoost-SMM

This subsection proposes a more sophisticated protocol to handle the Observation 1. We swap the role of participants, i.e, the PP with its splitting matrix will generate the set of null-space vectors. The reconstruction of the splitting matrix from the generated null-space vectors is equivalent to Problem 2. To handle this, we add calibrated random $\{0, 1\}$ columns before constructing the null-space, i.e.,
$$M^* = [M\ M'] \in \{0, 1\}^{n \times (l + l_1)},$$
where $M' \in \{0, 1\}^{n \times l_1}$ is properly generated. Assume that the curious party has unbounded computational capability, if there exists an efficient algorithm as declared in the Observation 1, the private splitting label can not be inferred with high confidence. Figure 2 illustrates the concept.

Another possible problem is the sparsity of M. The constructed null-space of M is sparse so the private values of AP at some users might be revealed. If this happens, AP must refuse to response so the learning process is impaired. We handle the sparsity by adding columns with random numerical values to M before generating the null-space vectors.
$$M^* = [M\ M'\ Y] \in \{0, 1\}^{n \times (l + l_1 + l_2)}, Y \sim \mathcal{N}(\mu, \sigma) \in \mathbb{R}^{n \times l_2}$$
Each column of M^* are independent so the multiplication result of columns in M' and Y are omitted in the final evaluation. Note that M' must be properly generated to handle the Observation 2, this results in higher model complexity.

3.4 FedXGBoost-LDP

FedXGBoost-LDP is a different approach from FedXGBoost-SMM which perturbs the gradients and hessians to achieve privacy-preserving, e.g Duchi's method from [Duchi *et al.*, 2013], Piecewise or Hybrid mechanisms from [Wang *et al.*, 2019], etc. The perturbed data is then used directly for training, which reduces the training time in comparison to HE or SMM methods. Nevertheless, due to the high non-linearity of the splitting score function, the injected noise degrades the utility strongly. For this reason, we use the first-order approximation for (4) to evaluate the splitting score. Particularly, PP received the perturbed gradient and hessian values from AP, the optimal splitting score is estimated by
$$s^* = \arg\max_{s_i \in \mathcal{S}} -\frac{1}{\lambda} G_{s_i}^L G_{s_i}^R \tag{11}$$
It can be shown that the above estimator is unbiased, but its variance depends strongly on the variance of the injected

[1] $\langle . \rangle$ operator indicates encrypted data

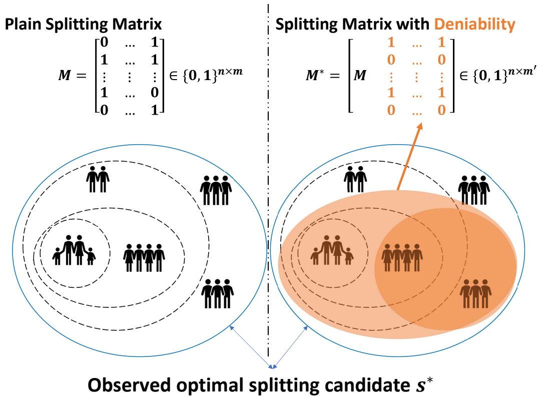
Plain Splitting Matrix
$M = \begin{bmatrix} 0 & \dots & 1 \\ 1 & \dots & 1 \\ \vdots & \vdots & \vdots \\ 1 & \dots & 0 \\ 0 & \dots & 1 \end{bmatrix} \in \{0,1\}^{n \times m}$
Splitting Matrix with Deniability
$M^* = \begin{bmatrix} M & \begin{matrix} 1 & \dots & 1 \\ 0 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 1 & \dots & 1 \\ 0 & \dots & 0 \end{matrix} \end{bmatrix} \in \{0,1\}^{n \times m'}$
Observed optimal splitting candidate s^*

Figure 2: Splitting matrix with injected random columns

Adding random $\{0, 1\}$ columns into the real splitting matrix prevents users' labels from being inferred with high confidence.

noise, which implies a compromise between privacy and the accuracy. Nevertheless, this method accelerate the training process significantly in comparison to encryption or linear algebra techniques, thus it is a heuristic approach. A detailed discussion of (11) description can be found in [Le *et al.*, 2021], which is the full version of this paper.

4 The complete FedXGBoost Protocols - FedXGBoost-SMM & FedXGBoost-LDP

Firstly, the AP determines the set of users (the users in a common node) being analyzed and announces this to all PPs. Next, all parties follow either FedXGBoost-SMM or FedXGBoost-LDP to find the best splitting candidate. After determining optimal splitting score, AP requests the information from the owner of the best score. Particularly, AP requests the feature analyzed by that splitting operation and the set of users in left and right nodes. After receiving the feature information, AP constructs a look-up table to record the corresponding PP and the analyzed feature. On the other side, the chosen PP also records the chosen feature and the best splitting operation for the usage in the regression phase. The training process continues from the split users' space until the tree reaches the maximum depth. At this stage, optimal leaf weight is computed according to (8) and saved for the prediction phase. This completes the construction of one tree.

The application of the regression of the trained model is similar to the study by [Cheng *et al.*, 2019]. When the AP wants to make inference from a new instance, it uses its look-up table and cooperates with the PPs to determine in which tree leaf the instance belongs to. Next, it aggregates the optimal weight overall regression trees to obtain a final prediction.

5 Experiments and Evaluation

We provide the experiments on two dataset 1) "Give Me Some Credit" and 2) "Default of Credit Card Clients" published on Kaggle. They contain 150000 instances with 10 attributes and 30000 instances with 25 attributes respectively. The experiments are conducted by two Linux machines, with 16-Core, 64GB-memory, and network bandwidth of 25000Mb/s. We evaluate FedXGBoost-LDP by comparing the model accuracy and the time consumption with the plain XGBoost and the studied encryption techniques. FedXGBoost applies different LDP perturbation techniques, which are Laplace mechanism (LM) and Duchi's method (DM) with varying privacy budget $\epsilon = \{1, 3\}$. According to (11), we also evaluate LDP mechanisms using first-order approximation, which is expected to reduce the accuracy loss.

Dataset	#Users	XGBoost	Paillier Encryption	Fed-XGBoost LDP
1	150K	260	1560	270
2	30K	69	354	73

Table 1: Time consumption of one iteration of different approaches in seconds.

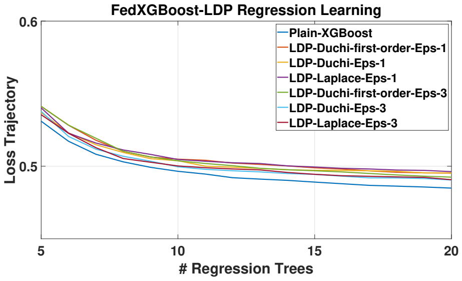

Figure 3: Learning's loss trajectory of approaches on dataset 2.

Figure 3 depicts the loss trajectory the experiment on dataset 2. Despite the injected noise causes a small performance reduction, it is almost equivalent to the plain XGBoost. Table 1 depicts that FedXGBoost-LDP have negligible overhead in comparison with plain XGBoost, both tremendously accelerate the learning process in comparison to the encryption method.

6 Conclusion

This paper studies two different protocols (FedXGBoost-SMM and FedXGBoost-LDP) that enable the state of the art tree ensemble model XGBoost to be conducted under FL settings. Different from the previous work applying homomorphic encryption, our linear algebra based protocol FedXGBoost-SMM incurs lower overhead while maintaining lossless accuracy. We also propose and empirically evaluate the accuracy of the heuristic protocol FedXGBoost-LDP, which relaxes the splitting score computation to first order approximation for computational speedup, and uses LDP noise perturbation. For future work, we will experimentally evaluate the overhead of FedXGBoost-SMM. Further study of scalable privacy-preserving XGBoost for FL is crucial for its deployment in practice.

References

[Aono *et al.*, 2016] Yoshinori Aono, Takuya Hayashi, Trieu Phong Le, and Lihua Wang. Scalable and secure logistic regression via homomorphic encryption. In *Sixth ACM Conference on Data and Application Security and Privacy (CODASPY '16). ACM, New York, NY, USA*, page 142–144, 2016.

[Chen and Guestrin, 2016] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. https://doi.org/10.1145/2939672.2939785, 2016.

[Cheng *et al.*, 2019] K. Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, and Qiang Yang. Secureboost: A lossless federated learning framework. *ArXiv*, abs/1901.08755, 2019.

[Cramer *et al.*, 2015] Ronald Cramer, Ivan Damgard, and Jesper Buus Nielsen. Secure multiparty computation and secret sharing an information theoretic approach for the internet of things, 2015.

[Duchi *et al.*, 2013] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy and minimax bounds: Sharp rates for probability estimation. 2013.

[Dwork and Roth, 2014] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. DOI: 10.1561/0400000042, 2014.

[Fang *et al.*, 2020a] Wenjing Fang, Chaochao Chen, Jin Tan, Chaofan Yu, Yufei Lu, Li Wang, Lei Wang, Jun Zhou, and Alex X. A hybrid-domain framework for secure gradient tree boosting, 05 2020.

[Fang *et al.*, 2020b] Wenjing Fang, Chaochao Chen, Jin Tan, Chaofan Yu, Yufei Lu, Li Wang, Lei Wang, Jun Zhou, and Alex X. A hybrid-domain framework for secure gradient tree boosting, 2020.

[Karnin *et al.*, 2016] Zohar Karnin, Kevin Lang, and Edo Liberty. Optimal quantile approximation in streams. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 71–78, 2016.

[Karr *et al.*, 2007] Alan F. Karr, Xiaodong Lin, Jerome P. Reiter, and Ashish P. Sanil. Privacy-preserving analysis of vertically partitioned data using secure matrix products. 2007.

[Le *et al.*, 2021] Nhan Khanh Le, Yang Liu, Quang Minh Nguyen, Qingchen Liu, Fangzhou Liu, Quanwei Cai, and Sandra Hirche. Fedxgboost: Privacy-preserving xgboost for federated learning. *arXiv preprint*, 2021.

[Li *et al.*, 2008] Ping Li, Qiang Wu, and Christopher Burges. Mcrank: Learning to rank using multiple classification and gradient boosting. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2008.

[Li *et al.*, 2019] Qinbin Li, Zeyi Wen, and Bingsheng He. Practical federated gradient boosting decision trees, 2019.

[Li *et al.*, 2021] Qinbin Li, Zhaomin Wu, Zeyi Wen, and Bingsheng He. Privacy-preserving gradient boosting decision trees, 2021.

[Liang and S.Chawathe, 2004] G. Liang and S. S.Chawathe. "privacy-preserving inter-database operations". *International Conference on Intelligence and Security Informatics*, 2004.

[Liu *et al.*, 2020] Yang Liu, Zhuo Ma, Ximeng Liu, Siqi Ma, Surya Nepal, Robert. H Deng, and Kui Ren. Boosting privately: Federated extreme gradient boosting for mobile crowdsensing. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 1–11, 2020.

[Narayanan and Shmatikov, 2006] A. Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *ArXiv*, abs/cs/0610105, 2006.

[Shi *et al.*, 2021] Yuanmin Shi, Siran Yin, Ze Chen, and Leiming Yan. Xgboost algorithm under differential privacy protection. doi:10.32604/jihpp.2021.012193, 2021.

[Shingi, 2020] Geet Shingi. A federated learning based approach for loan defaults prediction. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 362–368, 2020.

[Tyree *et al.*, 2011] Stephen Tyree, Kilian Q. Weinberger, Kunal Agrawal, and Jennifer Paykin. Parallel boosted regression trees for web search ranking. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, page 387–396, New York, NY, USA, 2011. Association for Computing Machinery.

[Wang *et al.*, 2019] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. Collecting and analyzing multidimensional data with local differential privacy, 2019.

[Xu and Wang, 2019] Jie Xu and Fei Wang. Federated learning for healthcare informatics. *CoRR*, abs/1911.06270, 2019.

[Yang *et al.*, 2019a] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2), January 2019.

[Yang *et al.*, 2019b] Wensi Yang, Yuhang Zhang, Kejiang Ye, Li Li, and Cheng-Zhong Xu. *FFD: A Federated Learning Based Method for Credit Card Fraud Detection*, pages 18–32. 06 2019.

[Úlfar Erlingsson *et al.*, 2014] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response, 2014.