New Approaches to Federated XGBoost Learning for Privacy-Preserving Data Analysis

Fuki Yamamoto[1], Lihua Wang[2], and Seiichi Ozawa[1,3](✉)

[1] Graduate School of Engineering, Kobe University, Kobe, Japan
tamafuki929@gmail.com, ozawasei@kobe-u.ac.jp
[2] National Institute of Information and Communications Technology, Tokyo, Japan
lh-wang@nict.go.jp
[3] Center for Mathematical and Data Sciences, Kobe University, Kobe, Japan

Abstract. In this paper, we propose a new privacy-preserving machine learning algorithm called *Federated-Learning XGBoost* (FL-XGBoost), in which a federated learning scheme is introduced into XGBoost, a state-of-the-art gradient boosting decision tree model. The proposed FL-XGBoost can train a sensitive task to be solved among different entities without revealing their own data. The proposed FL-XGBoost can achieve significant reduction in the number of communications between entities by exchanging decision tree models. In our experiments, we carry out the performance comparison between FL-XGBoost and a different federated learning approach to XGBoost called FATE. The experimental results show that the proposed method can achieve high prediction accuracy with less communication even if the number of entities is increase.

Keywords: Machine learning · Federated learning · Privacy preserving · Big data analysis · Ensemble tree classifier

1 Introduction

When we analyze big data owned by multiple entities, conventional data mining technology can effectively work on conditioned that all the entities cooperatively share their own data each other. In reality, however, this condition does not always hold. On the other hand, there exist many social problems that should be cooperatively solved by sharing sensitive data among multiple entities for such as crime deterrence, medical care, and health care for elderlies. Obviously, solving such sensitive tasks could provide a big impact to our society. On the contrary, we should keep in mind that it could expose us to great danger, causing serious incidents of personal data leak. Recently, to alleviate the current difficulty in big data analysis, privacy-preserving data mining (PPDM) has attracted considerable attention.

There have been developed several PPDM approaches such as homomorphic encryption [5] and differential privacy [2]. The former allows us to conduct specific calculations (e.g., addition and multiplication) over encrypted data. The latter provides us a mechanism of adding noise to ensure a certain level of privacy from a statistical point of view. On the other hand, however, the above PPDM approaches may sacrifice accuracy in prediction or impose us some restriction in data usage. Different from the conventional PPDM approaches where data are collected and processed at a place, federated learning gives a new possibility to learn cooperatively among different entities without revealing their own data [11]; that is, each entity conduct the learning of data locally and provide only their model updates to a center server. Then, the center server distributes the whole update information to each entity. If data include confidential information, a rigorous process is generally required to follow some regulations, resulting in reducing usability and convenience. However, if only model update information is shared in learning, it would facilitate to carry out the analysis of sensitive data.

In many practical machine learning methods, Gradient Boosting Decision Tree (GBDT) [3] is applicable to many situations. Chen et al. proposed XGBoost [1] as a more scalable and accurate GBDT method, which improved the computation speed by using the sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning. Unlike other black box approaches such as deep learning models, the prediction by XGBoost can give a certain level of explainability, because it gives importance score for feature conditions in decision trees. Yang et al. [10] introduces federated learning into XGBoost where gradient information on each tree nodes should be frequently communicated between a center server and entities of data owners. FATE (Federated AI Technology Enabler) [9], an open-source federated learning framework developed by WeBank, incorporated secret computation into the Yang et al.'s method. Although FATE achieves both secure and accurate computations, it generally requires frequent communications among entities to build each of decision trees. When the depth of a decision tree is d, it is estimated that about $2^d - 1$ times communications are required to carry out model update. Zhao et al. [12] introduced federated learning and differential privacy into GBDT by communicating models. Although this method can train a model with only tree-by-tree communication, the noise mechanism to ensure differential privacy often deteriorates the prediction accuracy of a model.

To solve the above-mentioned problems, we propose a new privacy-preserving machine learning algorithm called *Federated-Learning XGBoost* (FL-XGBoost). The contribution of this paper lies in the development of a practical federated learning scheme that reduces communications among entities without scarifying prediction accuracy. In the proposed FL-XGBoost, only one-time communication is necessary for keep high prediction accuracy in the model update.

This paper is organized as follow. Section 2 provides the preliminaries for the proposed federated learning approach. Then, we present the proposed FL-XGBoost in Sects. 3, In Sect. 4, after explaining experimental setups, we show the performance comparison between FATE and the proposed FL-XGBoost for some benchmark data sets. We also give a security analysis in Sect. 5, and Sect. 6 gives our conclusions and future work.

2 Preliminaries

2.1 XGBoost

XGBoost [1] is a novel GBDT method, and it is faster and more accurate than traditional methods. Since GBDT is composed of multiple decision trees, the update is performed by determining the split points and leaf weights. For XGBoost, it performs update using the gradient information of loss function, instead of the feature values of data. The following Eq. (1) shows the calculation of the gradient information, denoted by g_i, h_i.

$$g_i = \partial_{\hat{y}_i^{(k-1)}} l\left(y_i, \hat{y}_i^{(k-1)}\right), h_i = \partial^2_{\hat{y}_i^{(k-1)}} l\left(y_i, \hat{y}_i^{(k-1)}\right). \tag{1}$$

The loss function $l\left(y_i, \hat{y}_i^{(k-1)}\right)$ calculates the error between the current prediction and the true value. In the conventional GBDT methods, the split points are determined by using the gradient information as impurity and the leaf weights are determined to minimize errors. Differently, in XGBoost, the split points are determined to minimize the cost function $\mathcal{L}^{(k)}\left(f_k\right)$ as in Eq. (2), and the leaf weights $\hat{\omega}_j$ are analytically determined by quadratic approximation as shown in Eq. (3).

$$\mathcal{L}^{(k)}\left(f_k\right) = \sum_{i=1}^{n}\left[l\left(y_i, \hat{y}_i^{(k-1)}\right) + g_i f_k\left(x_i\right) + \frac{1}{2}h_i f_k^2\left(x_i\right)\right] + \Omega\left(f_k\right). \tag{2}$$

$$\hat{\omega}_j = -\frac{\Sigma_{i\in I_j} g_i}{\Sigma_{i\in I_j} h_i + \lambda}. \tag{3}$$

Furthermore, the data set in each node is divided into left and right nodes, and the gradient information of the nodes are summed up to calculate G_L, H_L and G_R, H_R, respectively. As shown in Eq. (4), these values are used to calculate *score*, and the split point is where maximizes this *score*.

$$score = \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda}. \tag{4}$$

2.2 Related Works

FATE [9] uses a scheme consisting of multiple data owners and a central server, and aggregates the gradients of loss functions at the central server to realize federated learning of GBDT model. This approach requires the feature values to be discretized in advance with a common criterion among the data owners. The discrete feature values are discretized candidate split points. Therefore, as the width of the discretization is reduced, the number of candidate split point increases and the loss is reduced, but the security is also reduced, correspondingly. Each data owner calculates the sum of the gradients of all the data divided at each candidate split point and sends it to the central server. The central server sums these results up and determines the split point. To prevent the central server from obtaining information from the data owners, the process to aggregate the gradient information is under encrypted form. What the algorithm depends on is not the amount of data held by each data owner, but the total amount of data from all data owners. However, this scheme requires the same times of communications as XGBoost's nodes to train the model.

Zhao et al. [12] achieved federated learning of the GBDT model in a scheme consisting of multiple data owners only, by learning a common model in turn. Specifically, each data owner updates the model with its own data set and sends the updated model to the next data owner. Since only its own data and the updated model are required for model training, the only information required for communication is the model information. However, there is still a problem that the model prediction accuracy is sacrificed due to the noise added to satisfy the differential privacy requirement.

3 Overview of FL-XGBoost

In this section, the details of the proposed FL-XGBoost are described. The proposed scheme consists of multiple data owners $U = \{u_1, u_2, u_3, ..., u_D\}$ and the central server S. Throughout this paper, we use the symbols defined in Table 1.

Table 1. The notations used in this paper.

Notation	Description
S	Central server
D	Number of data owners
U	Data owners set
u_{tr}	Data owner to train the model
T_i	The model updated for i times
$iter$	Target number of model updates
$\text{Enc}_{pk}(\cdot)$	Encryption with a public key pk
$\text{Dec}_{sk}(\cdot)$	Decryption with a secret key sk

Zhao et al. [12] realized federated learning and differential privacy by communicating models among the data owners. This method can train the model with less communication than existing federated GBDT schemes. However, in this study, the model is assumed to be shared only among data owners. Therefore, the focus should be on the performance of the model rather than differential

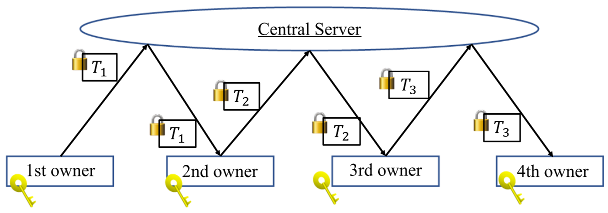

Fig. 1. The outline drawing of FL-XGBoost. T_i denotes the model that has been updated for i times.

privacy guarantees. Based on that, we propose a practical federated learning scheme for XGBoost that significantly reduces the number of communications with minimal loss. To accomplish this, we implemented a method to guarantee security without impacting the performance of the model and an algorithm to select the best data owner for the next training.

3.1 Secure Model Updates Among Multiple Data Owners

The scheme of Zhao et al. [12] consists of multiple data owners only and updates the model in a predetermined order. Therefore, it is possible to identify the data owner that updated the model. Furthermore, each data owner can easily obtain statistical information of other data owners, since each decision tree is learned by a single data owner and such trees are generally considered as statistical information.

In order to solve the above problems, we propose a scheme in which a central server S is introduced. The outline of FL-XGBoost is shown in Fig. 1. In FL-XGBoost, $u \in U$ learns T_i from its own data set, similar to the method of Zhao et al.

The difference is that the communication is done via S to prevent identifying data owner u_{tr} who updated the model. Since this u_{tr} is selected from U by S according to the algorithm described below, the other $u \in U$ cannot identify u_{tr}. The model is learned by repeatedly communicating model information between S and each u_{tr} selected by S to update the model. Here, all data owners share the encryption keys in advance and send the encrypted information to S so that S cannot obtain the information. In addition, communication is performed using a secure communication channel, and the communication content is not intercepted.

3.2 FL-XGBoost with Random and Uniform Data Owner Selection

Random selection is the simplest way for S to select u_{tr} from U. However, such selection of u_{tr} may contain a bias, impacting the performance of the trained model. Therefore, u_{tr} needs to be selected uniformly from U. Different from that, Zhao et al.'s method selects u_{tr} in a predetermined order, and each data

Algorithm 1: Learning algorithm of FL-XGBoost-R

```
1.1:  for $i \leftarrow 1$ to $iter$ do
1.2:      if $i$ is a multiple of $D$ then
1.3:          $S$ creates $U_{order}$.
1.4:      end
1.5:      $S$ selects $u_{tr}$ according to $U_{order}$.
1.6:      $S$ sends $\mathrm{Enc}_{pk}(T_{i-1})$ to $u_{tr}$.
1.7:      $u_{tr}$ decrypts $\mathrm{Enc}_{pk}(T_{i-1})$ to obtain $T_{i-1}$.
1.8:      $u_{tr}$ updates $T_{i-1}$ with its own data to obtain $T_i$.
1.9:      $u_{tr}$ encrypts $T_i$ and obtains $\mathrm{Enc}_{pk}(T_i)$.
1.10:     $u_{tr}$ sends $\mathrm{Enc}_{pk}(T_i)$ to $S$.
1.11: end
```

owner is selected for the same number of times. This is risky because it is easy to obtain a set of trees trained by the same data owner.

Based on the above, we propose FL-XGBoost-R, which uses an algorithm to randomly and uniformly select the data owners. Algorithm 1 shows details. First, S creates U_{order}, where U is randomly sorted. Then, select u_{tr} according to U_{order} until all data owners are covered. Such cycle is repeated.

3.3 FL-XGBoost with Data Owner Selection Based on Prediction Confidence

Even if S selects u_{tr} from U uniformly, the model's learning may not proceed equally for all $u \in U$. To obtain an unbiased trained model for the U, we propose FL-XGBoost-G, which selects u_{tr} using $|g|$, the absolute value of g.

The gradient of the loss function is denoted by g, and $|g|$ represents the magnitude of the error between the predicted and true values. Hence, the sum of $|g|$ of the data held by $u \in U$ represents the confidence of the prediction of that data set.

Data sets with low confidence is considered to be untrained, so $u \in U$ with a small $\sum |g|$ should be selected preferentially. However, for an imbalanced dataset where the number of data per class of labels varies greatly, the data with majority class will dominate $\sum |g|$. Also, the comparison is difficult when the amount of data owned by each $u \in U$ is different, since $\sum |g|$ depends on the amount of data owned. Considering the above problems, we use G_{ave}, the mean value of $|g|$ per class of labels, to be the selection index, as shown in the following equation (5).

$$G_{ave} = \frac{G_{pos}}{pos} + \frac{G_{neg}}{neg}. \tag{5}$$

Here, pos, neg denote the amount of positive and negative data, respectively. G_{pos} and G_{neg} denote $\sum |g|$ for positive and $\sum |g|$ for negative data, respectively.

Algorithm 2: Learning algorithm of FL-XGBoost-G

```
2.1:  for $i \leftarrow 1$ to $iter$ do
2.2:      if $i \leq D$ then
2.3:          $S$ selects $u_{tr}$ with FL-XGBoost-R.
2.4:      end
2.5:      else
2.6:          $S$ selects $u \in U$ with the largest $G_{ave}$ as $u_{tr}$.
2.7:      end
2.8:      $S$ sends $\mathrm{Enc}_{pk}(T_{i-1})$ to $u_{tr}$.
2.9:      $u_{tr}$ dectypts $\mathrm{Enc}_{pk}(T_{i-1})$ and gets $T_{i-1}$.
2.10:     $u_{tr}$ updates $T_{i-1}$ with its own data to obtain $T_i$.
2.11:     $u_{tr}$ encrypts $T_i$ and obtains $\mathrm{Enc}_{pk}(T_i)$.
2.12:     $u_{tr}$ computes $G_{ave}$.
2.13:     $u_{tr}$ sends $\mathrm{Enc}_{pk}(T_i)$ and $G_{ave}$ to $S$.
2.14: end
```

Algorithm 2 shows details. FL-XGBoost-R is used for the first round. Then u_{tr} transmits $\mathrm{Enc}_{pk}(T_i)$ as well as G_{ave} to S. This implies that after the first round, S has G_{ave} of all $u \in U$. Thereafter, S selects $u \in U$ with the largest G_{ave} as u_{tr}.

4 Experiments

In this section, we describe experiments using open data sets to verify the usefulness of the proposed methods. In the scheme of the proposed methods, all elements of U own independent datasets and share the same feature space and label space. For this purpose, we divided the dataset into D subsets horizontally, and performed the experiments assuming that each subset is a dataset owned by $u \in U$. We also compared the proposed methods with the federated XGBoost implemented in FATE and the XGBoost trained on the whole dataset.

The experimental environment is UBuntu 18.04 with 64[Gb] RAM, and the programming language is Python.

4.1 Data Set

Table 2 shows information on the four binary classification datasets used in the experiment. For brevity, we named the datasets as ‘Arcene’ for the Arcene Data Set [6], ‘Biodeg’ for the QSAR biodegradation Data Set [8], ‘Credit’ for the Credit Card Fraud Detection [4], ‘German’ for the German Credit Data [7] in this paper. Basically, accuracy is the evaluation index for the models, but only for ‘Credit’, we use the F1-score to take account for the imbalance of the number of data per class.

Table 2. Information of the dataset used in the experiment.

Dataset	Number of data	Number of features
Arcene [6]	729	10000
Biodeg [8]	854	41
Credit [4]	230693	30
German [7]	810	20

4.2 Results

We conducted verification experiments with open data sets for the existing and proposed methods and the results of the experiments are shown in Table 3. Table 3a shows the results of XGBoost trained on the entire dataset, and Table 3b shows the results of FATE's federated XGBoost with the number of candidate split points set to 10000. The number of candidate split points is a hyperparameter, which controls the trade-off between security and prediction accuracy and was set as 10000 because the result was almost the same as Table 3a at 10000. Table 3c shows the results for FL-XGBoost-R and Table 3d shows the results for FL-XGBoost-G. The proposed methods were tested for various values of D because D is considered to affect the results.

First, we compared the results of the proposed methods. From Table 3c and Table 3d, FL-XGBoost-G shows better results for all datasets in case $D \leq 10$. However, in case $D = 15, 20$, FL-XGBoost-R shows better results depending on the dataset. This is because the effect of outliers on the value of G_{ave} increases as D increases. Some data sets have outliers that deviate from the distribution of other data, and generally such data are prone to large prediction errors. The number of outlier data is generally small and does not affect G_{ave} in large data sets, but in the case of small data sets, such as when D is large, the outlier affects G_{ave}. Therefore, the data owner with outliers is preferentially selected. This may deteriorate accuracy depending on the dataset.

Next, we compared the results of the existing methods with those of the proposed methods. FATE's federated XGBoost realized model training with little or no loss, given that the results in Tables 3a and 3b are almost identical. Comparing this with Tables 3d, FL-XGBoost-G achieves the same level of prediction accuracy when D is small. However, as the value of D increases, the prediction accuracy of the proposed methods deteriorates in all datasets.

This is because the impact of outliers on G_{ave} increases as D increases. Some data sets have outliers that deviate from the overall data distribution, and generally such outliers may lead to large prediction errors. The number of outlier data is generally small and does not affect G_{ave} in large data sets, but when D is large and the entire data set is divided into subsets containing fewer data, the outlier in a subset may have more impact on G_{ave}. Therefore, the data owner with outliers is preferentially selected. This may deteriorate accuracy depending on the dataset.

This is because the proposed methods uses a single owner's data to update the model, and therefore the amount of data available for updating is reduced.

Table 3. Experiment results of prediction accuracy for the existing and proposed methods using open data sets. (a) and (b) show the result of the existing methods, and (c) and (d) show the result of the proposed methods.

(a) The results of XGBoost trained on the entire dataset

Arcene	0.850 ± 0.00
Biodeg	0.826 ± 0.01
Credit	0.850 ± 0.00
German	0.790 ± 0.01

(b) The result of the federated XGBoost implemented in FATE

Arcene	0.850 ± 0.00
Biodeg	0.826 ± 0.01
Credit	0.850 ± 0.00
German	0.792 ± 0.02

(c) The results with FL-XGBoost-R

D	3	5	10	15	20
Arcene	0.850 ± 0.00	0.690 ± 0.00	0.740 ± 0.02	0.700 ± 0.00	0.650 ± 0.00
Biodeg	0.831 ± 0.01	0.828 ± 0.02	0.826 ± 0.02	0.842 ± 0.01	0.840 ± 0.01
Credit	0.841 ± 0.00	0.841 ± 0.00	0.838 ± 0.00	0.838 ± 0.00	0.828 ± 0.01
German	0.772 ± 0.02	0.750 ± 0.02	0.750 ± 0.01	0.724 ± 0.02	0.716 ± 0.02

(d) The results with FL-XGBoost-G

D	3	5	10	15	20
Arcene	0.850 ± 0.00	0.700 ± 0.00	0.740 ± 0.02	0.750 ± 0.00	0.650 ± 0.00
Biodeg	0.848 ± 0.01	0.830 ± 0.00	0.826 ± 0.01	0.828 ± 0.00	0.818 ± 0.01
Credit	0.861 ± 0.01	0.841 ± 0.00	0.843 ± 0.01	0.840 ± 0.01	0.812 ± 0.01
German	0.774 ± 0.02	0.764 ± 0.04	0.759 ± 0.02	0.731 ± 0.01	0.720 ± 0.01

In contrast, FATE's federated XGBoost algorithm is unaffected by the amount of data owned by each data owners as long as the total amount is the same. Improving the accuracy with large D is an issue for future works.

5 Security Analysis

In this section, with the precondition that a secure communication channel is used so that the transmitted contents are not intercepted, we provide security analysis of the proposed FL-XGBoost with the following assumptions of attackers.

- Server S being honest-but-curious
- Data owners U being honest-but-curious
- *Passive Attack* being the only attack method

Here, *Passive Attack* means that there is no collusion among the participants of the scheme, and each attacker attacks based on the information obtained through correctly following the scheme. In addition, the purpose of each attacker is as follows.

1. S aims to obtain:
   - model information of T_i,
   - personal labels and feature values, and/or
   - number of data per class of label of $u \in U$.
2. U aim to obtain:
   - u_{tr} for each update, except where the attacker data owners themselves are u_{tr},

- number of data per class of label of $u \in U$ other than the attacker data owners themselves, and/or
- labels and feature values of individuals belonging to $u \in U$ other than the attacker data owners themselves.

Here, the trained models are intended to be shared only with $u \in U$, so they do not want this information to be known by S. Also, each tree in the model is considered to be statistical information of $u \in U$, so it is necessary to prevent the identifying the data owner who trained each tree.

In addition, the information received by each attacker is assumed to be as follows.

1. Information received by S:
   - u_{tr} for each update,
   - $\mathrm{Enc}_{pk}(T_i)(i = \{0, 1, 2, ...\})$,
   - G_{ave} (only in FL-XGBoost-G).

   So we need only consider security of FL-XGBoost-G.
2. Information received by U:
   - T_i $(i = \{0, 1, 2, ...\})$.

5.1 Security Against the Central Server

The G_{ave} and model information transmitted in FL-XGBoost is calculated from the gradient information of the loss function. Therefore, we first consider the information leaked from the gradient information. In the case of binary classification, when the prediction is y_{pred} and the true label is y, the gradient information g is given by following Eq. (6).

$$g = y_{pred} - y. \tag{6}$$

y_{pred} represents probability and $0 \leq y_{pred} \leq 1$. It can be inferred that the sign of g represents y, since $g \geq 0$ if $y = 0$ and $g < 0$ if $y = 1$.

No useful information from what S receives can be obtained from the encrypted information $\mathrm{Enc}_{pk}(T_i)$ without a key. Also, since G_{ave} in equation (5) is the sum of the mean values of all classes of label of $|g|$, it is not possible to obtain the sign of g and the number of data used in the calculation from its value. Hence, S cannot get the desired information.

5.2 Security Against the Data Owners

First, we consider whether it is possible to identify u_{tr} who updated the model. Since u_{tr} is selected by S, no one other than u_{tr} and S are informed of the selected $u \in U$. However, in the case of $D = 2$, u_{tr} can be identified by excluding its own learned trees. Apart from that, in the case of $D > 2$, $u \in U$ cannot identify u_{tr} because of its indistinguishability. From the above, it is not possible to obtain information on specific data owners from the trained models. Hence, the attacker cannot even get the number of data per class of labels that $u \in U$ owns.

Next, we examine the risk of personal information leakage from the model information. Specifically, we divide the model information into thresholds and leaf weights, and consider the information that can be leaked from each.

The thresholds are determined from the feature values, and thus the values correspond to the feature values of any individual. Therefore, if a feature value is unique, we can identify whose data were used to train the model.

Since λ and h in Eq. (3) which shows the leaf weights are larger than 0, the sign of the leaf weights is determined by $\sum g$ of the data in the leaf data set. Therefore, the sign of individual g cannot be identified from $\sum g$, as long as the number of data in the leaf data set is sufficiently large. However, in an extreme case where there is only one data in the leaf data set, the sign of the leaf output can indicate the label of that individual. As a result, if such individuals can be identified, the attacker can obtain the desired information.

6 Conclusion

In this study, we propose a practical federated learning scheme for XGBoost that significantly reduces the number of communication cycles compared to existing federate approaches. The proposed method achieves the same level of prediction accuracy as existing methods when each data owner has a sufficient amount of data. However, as the amount of data owned by individual data owners decreases, the performance of the models deteriorates. It is also found that in extreme cases, there is a risk of leakage of sensitive personal information from thresholds and leaf weights. This needs to be resolved by imposing some constraints on the training of the model.

Our future work is to refine the scheme to be more practical by solving the above-mentioned problems and by implementing a method of sharing encryption keys and a secure way of utilizing G_{ave}.

Acknowledgement. We would like to thank Associate Professor Toshiaki Omori and the members of the National Institute of Information and Communications Technology (NICT) for their helpful advice and support in writing this paper. This research has been accomplished through the project "Social Implementation of Privacy-Preserving Data Analytics" (JPMJCR19F6) in the JST CREST research area "Development and Integration of Artificial Intelligence Technologies for Innovation Acceleration".

References

1. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
2. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14

3. Friedman, J.: Greedy function approximation: a gradient boosting machine. Ann. Stat. pp. 1189–1232 (2001)
4. Kaggle: Credit Card Fraud Detection. https://www.kaggle.com/mlg-ulb/creditcardfraud. Accessed 14 Sep 2020
5. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
6. UCI: Arcene Data Set. https://archive.ics.uci.edu/ml/datasets/Arcene. Accessed 14 Sep 2020
7. UCI: German Credit Data. https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data. Accessed 14 Sep 2020
8. UCI: QSAR biodegradation Data Set (UCI). https://archive.ics.uci.edu/ml/datasets/QSAR+biodegradation. Accessed 14 Sep 2020
9. Webank: FATE (Federated AI Technology Enabler). https://fate.readthedocs.io/en/latest/index.html. Accessed 14 Sep 2020
10. Yang, M., Song, L., Xu, J., Li, C., Tan, G.: The tradeoff between privacy and accuracy in anomaly detection using federated XGBoost. arXiv preprint arXiv:1907.07157 (2019)
11. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. In: ACM Transactions on Intelligent Systems and Technology (TIST), New York, NY, USA, pp. 1–19. ACM (2019)
12. Zhao, L., et al.: Inprivate digging: enabling tree-based distributed data mining with differential privacy. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 2087–2095. IEEE (2018)