Federated Learning Application on Telecommunication-Joint Healthcare Recommendation

Yong Song
Telecom Artificial Intelligence Lab
AsiaInfo Technologies
Beijing, China
songyong@asiainfo.com

Yuchen Xie
Telecom Artificial Intelligence Lab
AsiaInfo Technologies
Beijing, China
xieyc@asiainfo.com

Hongwei Zhang
Information Technology Division
Novogene
Beijing, China
zhanghongwei@novogene.com

Yuxin Liang
Telecom Artificial Intelligence Lab
AsiaInfo Technologies
Beijing, China
liangyx@asiainfo.com

Xiaozhou Ye
Telecom Artificial Intelligence Lab
AsiaInfo Technologies
Beijing, China
yexz@asiainfo.com

Aidong Yang
Telecom Artificial Intelligence Lab
AsiaInfo Technologies
Beijing, China
yangad@asiainfo.com

Ye Ouyang
Telecom Artificial Intelligence Lab
AsiaInfo Technologies
Beijing, China
ye.ouyang@asiainfo.com

2021 IEEE 21st International Conference on Communication Technology (ICCT) | 978-1-6654-3206-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCT52962.2021.9657870

***Abstract*—Federated Learning (FL) is proposed to overcome data silos in model training of Machine Learning (ML) through joint modeling with privacy-preserving. Mobile Network Operators (MNOs) and Healthcare Providers (HPs) share common users with different features. Data silo exists due to the inaccessibility from MNOs to HPs for privacy-preserving. Therefore, the data of MNOs and HPs can be utilized together in FL settings to empower each other's service. In this paper, we established a Federated Gradient Boosting Decision Tree (FGBDT) system based on SecureBoost, a lossless federated ensemble model found on GBDT, to improve user classification for HPs in Healthcare Recommendation with MNOs' data. Our experiment shows that FGBDT has a 9.71% of precision increase and a 4% of F1 score improvement, and a 10.45% of cumulative precision improvement in a real operational practice than GBDT.**

***Index Terms*—Federated Learning, Gradient Boosting Decision Tree, User Classification, Healthcare Recommendation**

I. Introduction

Federated Learning (FL) was first proposed by Google, 2016. It aims to build a joint Machine Learning (ML) model based on the data located at multiple sites to solve data silos. Instead of transferring data from sites to sites, FL transfer ML model parameters in a secure way [1] [2]. Therefore, data owned by different organizations with common set can be utilized together to improve ML modeling through FL on account of its privacy-preserving, security, and regulatory compliance.

FL has plenty of applications in many fields such as sales, finance, healthcare, education, urban computing, edge computing, and block chain. FL was first practiced by Google in Gboard application on Android smartphones in 2016 [3]. Niknam et al. provided an overview of how federated learning addresses key challenges and improve performance of 5G mobile networks [4]. In addition, FL was also leveraged in Finance, Education, Edge Computing and IoT, Blockchain, Urban Computing and Smart City [5], etc.

Healthcare recommendation decreases the pressure on medical system through improving service efficiency of healthcare, especially during the pandemic. Since patients can receive healthcare service through mobile applications instead of going to hospital in person after obtaining an accepted recommendation. In addition, user classification is an inevitable step prior to recommendation, which is described as a typical classification problem with two classes of whether a user will accept a specific recommendation. Precise user classification in healthcare recommendation helps HPs to improve the recommendation success rate and highly increase user engagement. Therefore, user classification plays a significant role in healthcare recommendation. However, HPs possess limited authorized user data, and it is increasingly hard to improve classification accuracy through modeling only without any data improvement. In the meantime, MNO service is involved everywhere in our lives with a lot of data generated in different fields. Moreover, MNOs have most common user data

978-1-6654-3206-1/21/$31.00 ©2021 IEEE

with abundant features such as demographics, behavior, and geographic information of users. They are a good complement of HP data which can be applied to FL modeling.

In order to introduce Telecommunication data to user classification problem in healthcare recommendation, we developed Telecommunication-Joint FL platform based on Federated AI Technology Enabler (FATE) on account of its security guaranteed by being without data transfer or leakage [6]. Then, we proposed a Federated Gradient Boosting Decision Tree (FGDBT) model on Telecommunication-Joint FL platform to improve user classification. In addition, we deployed our model on an MNO and an HP and applied to online operation.

Contributions:

- Developing an Telecommunication-Joint FL platform with MNO data introduced to improve healthcare recommendation.
- Improving 9.71% of precision and 4% of F1 score for user classification in healthcare recommendation through proposed FGBDT model on FL platform experimentally.
- Implementing FGBDT model on both an MNO and an HP to increase user classification of the HP in a specific recommendation with an accumulative 10.45% of precision improvement in 28 days.

II. Related works

A. Federated Learning

Federated Learning (FL) employed decentralized datasets and distributed training. There are three different types of FL, including Horizontal FL (HFL), Vertical FL (VFL), and Federated Transfer Learning (FTL). On one hand, HFL is a joint-entity modeling with more common features and less common entities. For example, MNOs in different regions have almost same business but different users. Hence, they are a good fit for HFL. On the other hand, VFL is a joint-feature modeling with more common entities and less common features. For instance, hospitals and MNOs in a same location are a good fit for VFL because they provide same group of people different service. In addition, Federated Transfer Learning (FTL) is used for very limited data overlap in both entities and features.

There are three popular algorithms of VFL including Federated GBDT, and Federated Logistic Regression. Logistic Regression is a powerful ML algorithm to classify data and have been proved to achieved in a federated setting through only additively homomorphic encryption with accepted information loss in training [7] [8]. However, SecureBoost, as a Federated FBDT algorithm, provides the same level of accuracy as non-federated approach through constructing boosting trees across multiple parties with privacy-preserving [9]. According to the lossless of SecureBoost and non-linear features in the datasets of MNO and HP, SecureBoost is the best option for FL in user classification of Healthcare Recommendation.

B. Frameworks

PySyft was the first reliable and general Deep Learning Frame with privacy-preserving introduced by FL, Differential Privacy, and Encryption [10]. Google proposed an open-source FL frame named TensorFlow Federated with plenty of algorithms but only available for HFL. It was utilized a lot in Mobile Devices. Rosetta was proposed as a FL frame based on TensorFlow to provide fast AI solution with privacy-preserving to users without knowledge of cryptography and hardware security. Baidu developed PaddleFL based on PaddlePaddle with many FL strategy applied to Computer Vision, Nature Language Processing, and Recommendation without support to FTL. ByteDance developed Fedlearner with Tensorflow to support different applications in E-commerce, Finance, and Education [11]. USC proposed FedML in 2020 with distributed training, mobile training, and local simulation [12]. Meanwhile, there are other close source frames with limited algorithms like Morse, Avatar, JUGO, and PrivPy [13]. Although they are safer than open source frames, a slow product iteration exists.

Therefore, an open source frame with the most algorithms stands out. WeBank proposed industrial frame FATE based on Secure Multi-party Computation and Homomorphic Encryption [6]. It provides tools like federated modeling visualization, process schedule, and life cycle management and supports HFL, VFL, and FTL. FATE replaces traditional data encryption of model encryption integration to improve security without any data transfer. It is famous for its capability of computation frame distribution, information interaction audit, and interface scalability. Therefore, FATE is the most popular FL frame with community support as a prior choice of FL platform establishment.

III. System Design

A. Platform Design

An FATE-based FL platform is developed with three parts including Guest, Host, and Arbiter shown in Fig. 1. Guest is described as a model builder in joint modeling with training, visible training evaluation, deploy-able inference service, and inference. Host is described as a data provider with training assigned, deploy-able inference service, and inference. In this paper, HPs participate in FL as Guests and MNOs participates in FL as Hosts. Arbiter is described as a collaborator with encrypted parameter calculation. Data of all parties participated in FL modeling is stored in core domain to avoid exposure to public network instead of demilitarized zone.

B. Model Design

a) Entity Alignment: The first step in FGBDT is to find a common set of data samples which is identified by unique IDs through utilizing a privacy-preserving protocol with RSA and Hash for inter-database intersections [14]. According to the RSA and Hash encryption, it significantly cut the cost of computation and communication through creating public and private keys of RSA by the party with much more IDs for two-side alignment. In FGDBT, Host creates RSA keys as a data provider with massive data and Guest creates random number for blind RSA encryption. The alignment is achieved by following steps:

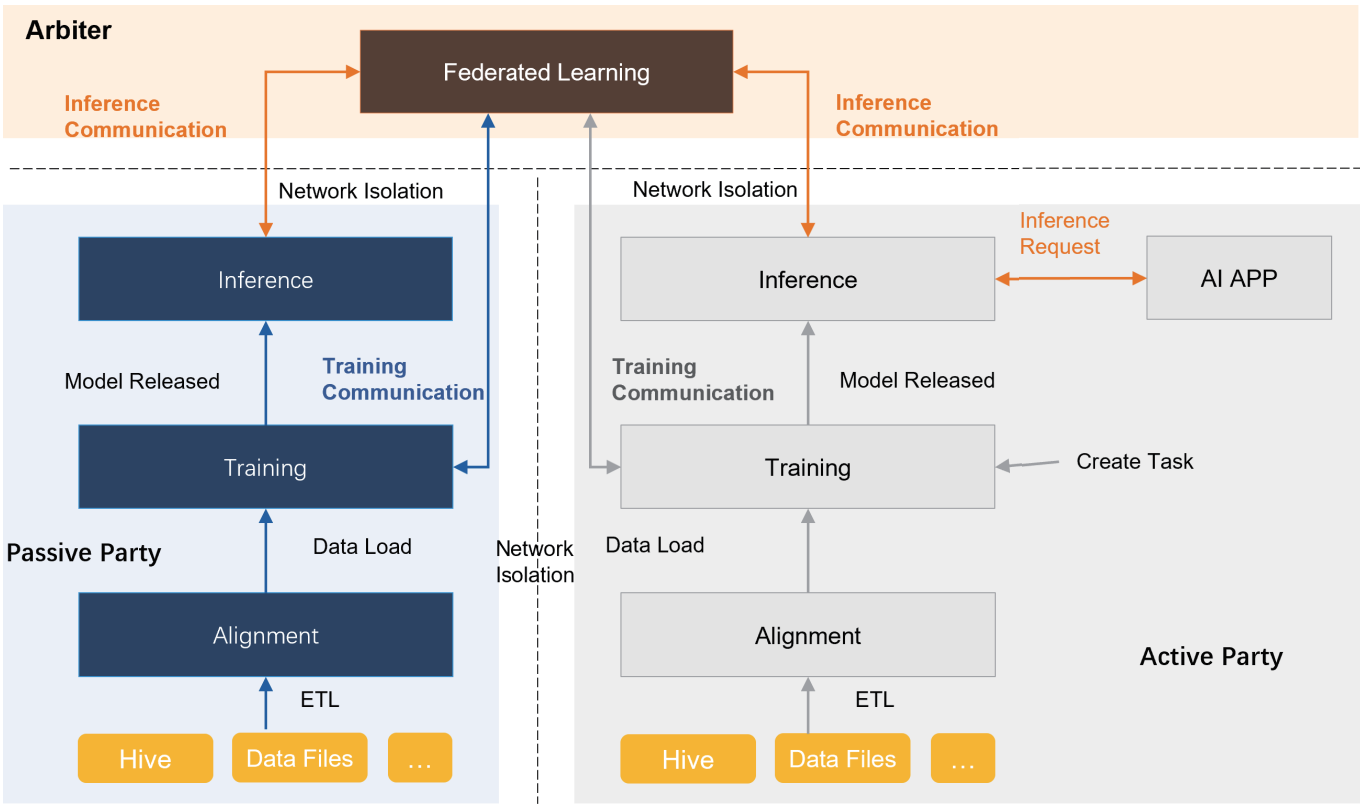

Fig. 1. Federated Learning Platform.

1) Host sends public RSA key to Guest.
2) Guest encrypts Hashed IDs with public RSA key and sends back to Host.
3) Host decodes the IDs of Guest, and encrypts Hashed IDs of Host with private RSA key to send to Guest.
4) Guest updates Hash map of IDs and calculates the encrypted intersection of IDs.
5) Guest decodes the common IDs with Hash map and sends the encrypted intersection back to Host.
6) Host decodes the intersection to obtain common IDs.

b) Modeling Training: According to SecureBoost [9], FGBDT predicts the output through using K regression/classification trees:

$$\widehat{y_l} = \sum_{k=1}^{K} \left(f_k\left(x_i\right)\right), \ \forall \ x_i \ \in \ \mathbb{R}^d, \ i = 1, 2, \ldots n. \tag{1}$$

where $\widehat{y_l}$ represents the prediction of dependent variable y, $f_k\left(x_i\right)$ describes the base joint Classification and Regression Tree (CART) models learned together by all participants. The loss function of k-th base tree to minimize is shown below.

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

where $\Omega\left(f_k\right) = \gamma T + \frac{1}{2}\lambda \left\|w\right\|^2$ describes complexity of model structure as regulation, $g_i = \partial_{\widehat{y_i}^{(k-1)}} l(y_i, \ \widehat{y_i}^{(k-1)})$, and $h_i = \partial^2_{\widehat{y_i}^{(k-1)}} l(y_i, \ \widehat{y_i}^{(k-1)})$. It is generated by a Taylor approximation to keep quadratic term like the typical case of square loss. After removing the constants in loss function, it is simplified as follow:

$$\begin{aligned} \mathcal{L}^{(k)} &\cong \sum_{i=1}^{n} \left[g_i f_k\left(x_i\right) + \frac{1}{2} h_i {f_k}^2\left(x_i\right) \right] + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\ &\quad + \gamma T \\ &= \sum_{i=1}^{n} \left[g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2 \right] + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\ &\quad + \gamma T \\ &= \sum_{j=1}^{T} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] \\ &\quad + \gamma T \end{aligned} \tag{3}$$

where i describes the size of sample, j describes the number of leaf nodes. In addition, $f_k\left(x_i\right) = w_{q(x_i)}$ which describes the leaf weight of the tree. We can calculate the w_j^* to optimize the $\mathcal{L}^{(k)}$ when $\partial_{w_j} \mathcal{L}^{(k)} = 0$ shown below:

$$w_j^* = -\frac{G_j}{H_j + \lambda} \tag{4}$$

where $G_j = \sum_{i \in I_j} g_i$ and $H_j = \sum_{i \in I_j} h_i$. We can obtain an updated object function with w_j^* below:

$$\mathcal{L}^{(k)} = -\frac{1}{2}\sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T \tag{5}$$

It starts from the root node and add a split for each leaf node till reaching the maximum depth to construct a base CART. Therefore, it maximizes a specific equation to determine the best split through

$$\mathcal{L}_{sp} = \frac{\left[\sum_{i \in I_L} g_i\right]^2}{2\sum_{i \in I_L} h_i + \lambda} + \frac{\left[\sum_{i \in I_R} g_i\right]^2}{2\sum_{i \in I_R} h_i + \lambda} - \frac{\left[\sum_{i \in I} g_i\right]^2}{2\sum_{i \in I} h_i + \lambda} - \gamma, \tag{6}$$

where I_L and I_R are the instance spaces of left and right split nodes.

Both split object function and loss function only involves g_j and h_j. Hence, Paillier [15] encryption is introduced to achieve privacy-preserving. In Paillier cryptosystem, a message m is encrypted as $\langle m \rangle = g^m r^n \ mod \ n$ for random $r \in \{0, 1, \ldots, n-1\}$. It is easy to prove that

$$\begin{aligned} \langle m_1 \rangle \cdot \langle m_2 \rangle &= \left(g^{m_1} r_1^c\right)\left(g^{m_2} r_2^c\right) \ mod \ n \\ &= g^{m_1 + m_2} \left(r_1 r_2\right)^c \ mod \ n \\ &= \langle m_1 + m_2 \rangle . \end{aligned} \tag{7}$$

Therefore, we have $\langle h_L \rangle = \prod_{i \in I_L} \langle h_i \rangle$ and $\langle g_L \rangle = \prod_{i \in I_L} \langle g_i \rangle$. The best split can be found in 2 steps.

- Each Host party calculates all possible $\langle h_L \rangle$ and $\langle g_L \rangle$ locally and sent back to Guest party.
- Guest party deciphers all $\langle h_L \rangle$ and $\langle g_L \rangle$ to compute global optimal split.

The training process can be concluded as follow.

1) Guest first calculates g_i and h_i, $i \in \{1, \ldots, N\}$ encrypted with Paillier homomorphic scheme.
2) Guest generates encrypted buckets and sends them to Host.
3) Host aggregates the encrypted gradients based on buckets with features mapping and then sends back to Guest.
4) Guest decrypts the aggregated result to determine global optimal split and returns corresponding parameters k_{opt} and v_{opt} to Host.
5) Host determines the attributes' value based on k_{opt} and v_{opt} and returns corresponding records to Guest.
6) Repeat steps 2-4 iteratively until reaching the maximum score.

c) Model Inference: According to the training process mentioned above, a party-specific lookup table with [feature, threshold] is generated to support inference. The inference can be achieved as follow steps.

1) Guest refers to owner of root node with related feature-threshold tuple.
2) The retrieved party compare the value of corresponding attribute with the threshold from lookup table to decide which child node to retrieve.
3) Guest returns party-id and record-id of the retrieved node.
4) Repeat steps 2-3 until a leaf node s reached.

IV. Experiment Analysis

We collaborate with an MNO as the Host Party and a HP as a Guest Party in a same region. The MNO provides Telecom data and HP prepares user data generated by a Healthcare Mobile Application.

A. Datasets

a) Guest Party: Healthcare Mobile Application has three different types of data which includes demographics, patient behavior related, and hospital/healthcare provider related data. The whole dataset includes approximate 690,000 records with about 130 features on site including almost 80 healthcare-visit-related data and 50 appointment-related data. The demographic data includes gender, age, registration time, registration type, etc. The healthcare-visit features include medical visit times per week, visit days, time of visit, hospital visited, doctor visited, the most frequent visited departments, etc. Appointment-related data includes times of appointment, days of appointment, time, etc.

b) Host Party: he MNO own demographic, online behavior, Healthcare-related tags, Application usage, and online search data. It includes 670,000,000 user records with 60 features including 20 demographics, 25 Healthcare tags, 5 of Telecommunication behavior, location trace, and online behavior. The demographic data includes user id, income level, age, gender, premium level, etc. The Telecommunication behavior data includes device brand, device model, etc. The online behavior data includes usage frequency for specific Applications in a recent month, total quantity usage in a recent month, Guest days of usage, searched quantity, days, and times of healthcare-related key words, etc. The Healthcare tags include healthcare-related phone call records including days and numbers, related text messages, etc.

B. Experiment Setup

The hardware and software environment is provided by both the MNO and HP. It is shown in Table I.

a) Alignment: After secure alignment, we have around 440,000 shared data with about 0.5% positive data. To reduce the impact of imbalance, we random sample training set with a ratio of 1 positive data to 10 negative data as an under-sampling strategy. In other words, the dataset for our experiment owns 5.5% of total shared data with 90% of training set and 10% of test set.

TABLE I
EXPERIMENTAL ENVIRONMENT

Participant Party	Software and Hardware		
	CPU	*RAM(G)*	*Software*
MNO	64	512	Linux CentOS 7.2, Python 3.6, Tomcat 8.5, JDK 1.8
HP	64	512	Linux CentOS 7.2, Python 3.6, Tomcat 8.5, JDK 1.8

[a]Hardware and Software Setup.

TABLE II
BATCH INFERENCE TIME CONSUMPTION

Task	10	100	1000	10000	100000
Time (s)	0.1022	0.2833	0.3315	0.7516	7.6643

[c]Time Consumption of Different-Scale Inference Tasks

b) Features: After feature engineering, there are 59 features selected in the MNO and 88 features chosen by the HP.

c) Parameters: Both GBDT on only Guest site and FGBDT share the same parameters including the number of decision trees of 100, 10-fold cross validation, and threshold of 0.5 to distinguish positive and negative data.

C. Metrics

To evaluate our model for detecting 0.5% of positive data in an imbalanced dataset, we focus on precision on account of effect and F1 score because of the model performance.

D. Experiment Results

The average time of alignment is 308 minutes. In model training, it takes 281 minutes in average. The average time delay of inference API is about 0.0548 seconds. The average time spend for batch inference is describe in Table II. Our results are shown in Table III. The GBDT on Guest site has 78.71% of precision and 80% of F1 score. On other hand, FGBDT owns 87.88% of precision and 84% of F1 score. It shows the improvement of 9.71% precision and 4% F1 score of FGBDT comparing to GBDT without federated setting.

E. Operational Results

We deployed our FGBDT model trained by history data of both the MNO and HP and ran for 28 days to test it. Meanwhile, we also deployed a GBDT model on HP's site with the same data and ran for 28 days for comparison. To evaluate the model effect in operational perspective, we recorded daily prediction of both FGBDT and GBDT model and compared to the real click of the specific recommendation content generated by users. We select daily precision as comparison metric shown in Fig. 2, because it can describe the operational model effect directly. The average daily precision of FGBDT is 2.05% and of on HP's site is 1.67%, which is shown in Table IV. It improves 0.38% of average precision in the amount of 440,000 data samples.

TABLE III
EXPERIMENTAL COMPARISON

Model Metric	Performance (%)		
	GBDT	*FGBDT*	*Improvement*
Precision	78.17%	87.88%	9.71%
F1 Score	80.00%	84.00%	4.00%

[c]Experimental Model Performance of GBDT and FGBDT

TABLE IV
OPERATIONAL COMPARISON

Model Metric	Performance (%)		
	GBDT	*FGBDT*	*Improvement*
Average Daily Precision	1.67%	2.05%	0.38%
Cumulative Precision	46.84%	57.29%	10.45%

[d]Operational Model Performance of GBDT and FGBDT

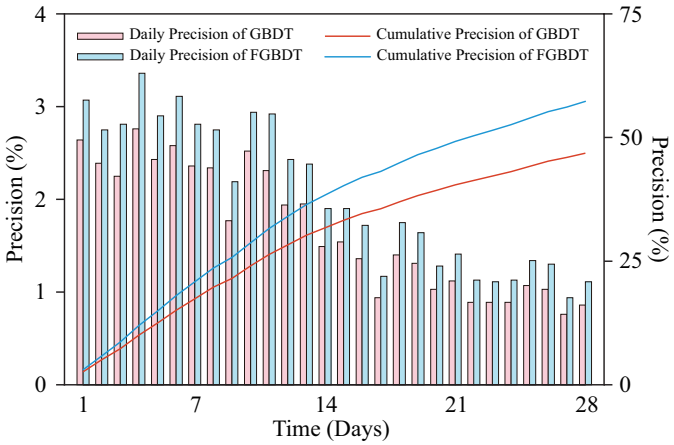

Fig. 2. Components of Federated Learning Participant.

V. CONCLUSION

In this paper, we developed an FL platform based on FATE with a host of MNO to help potential guests improve their modeling and utilize FGBDT to improve user classification accuracy of HPs in Recommendation. In an experimental perspective, FGBDT provides 9.71% of precision improvement and 4% of F1 score increase based on an imbalanced dataset. In an operational practice, FGBDT provides 0.38% average precision improvement of approximate 440,000 data in 28 days. It proves that an FL with MNO introduced can effectively help HPs to improve their modeling in Healthcare Recommendation. In addition, there are also some work we need to do in the future including comparing with more algorithms, especially the existing schemes, to determine the best one in healthcare recommendation, and introducing new parties to benefit them with telecommunication-joint FL.

ACKNOWLEDGMENT

This work was supported by AsiaInfo Technologies Limited and applied to its product AISWare AI^2.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[2] H. B. McMahan, E. Moore, D. Ramage, and B. A. Y. Arcas, "Federated learning of deep networks using model averaging," *ArXiv*, vol. abs/1602.05629, 2016.

[3] B. McMahan and D. Ramage, "Federated learning: Collaborative machine learning without centralized training data," *Google Research Blog*, vol. 3, 2017.

[4] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[5] T. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE translation journal on magnetics in Japan*, vol. 2, no. 8, pp. 740–741, 1987.

[6] WeBank AI Department, "Federated AI Technology Enabler (FATE)," https://github.com/FederatedAI/FATE, 2019.

[7] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," *arXiv preprint arXiv:1711.10677*, 2017.

[8] M. Kim, Y. Song, S. Wang, Y. Xia, X. Jiang *et al.*, "Secure logistic regression based on homomorphic encryption: Design and evaluation," *JMIR medical informatics*, vol. 6, no. 2, p. e8805, 2018.

[9] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos, and Q. Yang, "Secureboost: A lossless federated learning framework," *IEEE Intelligent Systems*, 2021.

[10] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, D. Rueckert, and J. Passerat-Palmbach, "A generic framework for privacy preserving deep learning," *arXiv preprint arXiv:1811.04017*, 2018.

[11] O. Li, J. Sun, X. Yang, W. Gao, H. Zhang, J. Xie, V. Smith, and C. Wang, "Label leakage and protection in two-party split learning," *arXiv preprint arXiv:2102.08504*, 2021.

[12] C. He, S. Li, J. So, X. Zeng, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu *et al.*, "Fedml: A research library and benchmark for federated machine learning," *arXiv preprint arXiv:2007.13518*, 2020.

[13] Y. Li and W. Xu, "Privpy: General and scalable privacy-preserving data mining," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1299–1307.

[14] G. Liang and S. S. Chawathe, "Privacy-preserving inter-database operations," in *International Conference on Intelligence and Security Informatics*. Springer, 2004, pp. 66–82.

[15] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1999, pp. 223–238.