ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 8, Special Issue 1, August 2021

FEDERATED LEARNING FOR PRIVACY PRESERVATION

Cina Mathew
Sathyabama University of Science and Technology

Dr. P. Asha
Sathyabama University of Science and Technology

Abstract: One of the significant challenges of Artificial Intelligence (AI) and Machine learning models is to preserve data privacy and to ensure data security. Addressing this problem lead to the application of Federated Learning (FL) mechanism towards preserving data privacy. Smart healthcare is another domain which we expect will greatly benefit from the rising of federated learning techniques. Medical datas are very sensitive and private, yet medical data are difficult to collect and they exist in isolated medical centers and hospitals. The insufficiency of data sources and the lack of labels have led to an unsatisfactory performance of machine learning models, which becomes the bottleneck of current smart healthcare. We envisage that if all medical institutions are united and share their data to form a large medical dataset, then the performance of machine learning models trained on that large medical dataset would be significantly improved. Federated learning combining with transfer learning is the main way to achieve this vision. Transfer learning could be applied to fill the missing labels thereby expanding the scale of the available data and further improving the performance of a trained model. Therefore, federated transfer learning would play a pivotal role in the development of smart healthcare and it may be able to take human health care to a whole new level. In this paper, we present in detail understanding of Federated Machine Learning, various federated architectures along with different privacy-preserving mechanisms. Finally, we also depict how Federated Learning is an emerging area of future research that would bring a new era in AI and Machine learning.

Keywords- Federated Learning, Artificial Intelligence, Machine Learning, Privacy, Security, Distributed Learning.

I. INTRODUCTION

Due to the emergence of AI and Machine learning over the past few decades, Preserving the data privacy is of utmost importance in these days. Data leaks on publicly available data and the private data of the companies lead to alarming increase towards data privacy. Utilizing the data which is isolated as data islands by maintaining specific privacy standards is very crucial for better data security. Misusing the personal data of the user may cause overhead to the user forcing him not to enclose his personal details. Even in the companies and industries, it is essential to protect data from data leaks as it would lead to grave consequences for the company. The data leaks, in turn, would affect the financial and commercial aspects of the company on a large scale leading to huge losses.

Many machine learning and AI models need sufficient data for training and to produce high- quality models. Although the models need to use user data if they need to build good prediction models for the user, there should be a way to ensure user privacy. In edge devices where users interact with different applications like in mobile phones, there is an ample amount of private data related to the user which is being exposed every day. To solve the problem of using data to train models, ensuring data privacy, we have a new approach known as Federated Learning (FL) [1]. Federated Learning is a collaborative Machine learning technique where the machine learning models are trained on edge devices instead of a central server to ensure data privacy. The data is not exchanged between the devices. However, only the model updates (gradient updates) are sent to the server to build a global model using the aggregated gradients from all the computing edge devices. Thus, the server has no information about the raw data that the edge devices have been trained on, maximizing the data privacy of the users. Federated Learning has been evolving over the past few years due to the increasing demand for data privacy and security.

The rest of the survey work is organized as follows: Section 2 details various related works in the area of Federated Learning. Section 3 details about Federated Learning, its working principle, training process, categorization of Federated Learning architectures along with various implementation frameworks, and Section 4

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 8, Special Issue 1, August 2021

elaborates more about the privacy- preserving mechanisms in FL. Section 7 concludes the survey work and suggests a few possible directions for the future area of research.

II. RELATED LITERATURE

Because data is dispersed across numerous devices, machine learning and AI models need to be able to access it reliably in order to construct efficient models. Federated Learning is a recent trend for training machine learning models in a decentralised manner without access to raw data other than the updated gradients from client models. The focus of Federated Learning is on the edge and mobile computing [2, 3] devices and then extended its application to large scale production systems.

For AI models to be helpful, data scattered everywhere as data islands must be combined on a broad scale. Integrating data from these islands is difficult because it incurs a cost. Federated Learning has shown to be a lifesaver when it comes to lowering the cost of data integration by executing AI models on data stored on edge computing devices.

Due of the privacy-preserving characteristic of Federated Learning, much of the academic work is currently focused in FL. For vertically partitioned data, Clifton and Vaidya provided safe k-means [4], secure association mining algorithms [5], and a naive Bayes classifier [6]. The authors of [7] developed a privacy-preserving linear regression methodology based on homomorphic encryption. For vertically partitioned data, the authors of [8,9] developed a linear regression technique. The linear regression problem was handled immediately using FL. The authors of [10] used Stochastic Gradient Descent (SGD) to solve the problem and also proposed privacy-preserving methods for neural networks and logistic regression. The authors of [11] presented a new algorithm for association rules on the horizonally partitioned data. For horizontally partitioned data [12] and vertically partitioned data [13], secure Support Vector Machines (SVM) methods have been implemented. [14] developed a number of secure multi-party linear regression and classification techniques. The authors of [15] suggested multi-party gradient descent methods that are both efficient and secure. Secure Multiparty Computation (SMC) was used in all of these studies [16]. The authors of [17] suggested a homomorphic encryption-based secure logistic regression methodology. Shokri and Shmatikov [18] proposed that neural networks be trained on horizontally partitioned data with parameter updates exchanged. The authors of [19] used homomorphic encryption to improve the overall security of the system while maintaining gradient privacy.

III. FEDERATED MACHINE LEARNING

Federated Learning is a decentralised machine learning environment in which all participating clients train a shared global model without disclosing data to a central server. The server receives only the model updates from each participating device. To generate an efficient global model, the model updates are pooled using the Federated Averaging technique [3]. As a result, it is collaborative machine learning, in which all clients contribute model updates in order to reach a common learning goal. FL enables smarter models, lower latency, and reduced battery usage while maintaining privacy.

a) DEFINITIONS

To understand the term Federated Learning, it is essential to know the terms distributed learning [20,21], centralized and decentralized Federated Learning [3].

- ✓ **Distributed Machine Learning:** We train a model on a huge dataset via distributed machine learning. Clients are computational nodes in a single cluster or datacenter in this case. Any part of the dataset is accessible to all clients. In a datacenter, the data is dispersed among several computing nodes. The goal of distributed learning is to parallelize computer capacity by distributing data or models.
- ✓ **Centralized Federated Learning:** It has a central server that orchestrates the entire training process and coordinates all of the learning nodes. The central server is in charge of initial node selection before training begins, as well as the aggregation of model updates received. The server may become a bottleneck here as all the selected nodes have to send updates to a single entity.
- ✓ **Decentralized Federated Learning:** The computing nodes in this Federated Learning scenario can work together to compute the global model. This configuration prevents single-point failures by exchanging model updates only across networked nodes without the orchestration of the central server.

b) FEDERATED LEARNING LIFE CYCLE

The life cycle of a Federated Learning (FL) model consists of six stages, as shown in Figure 1.

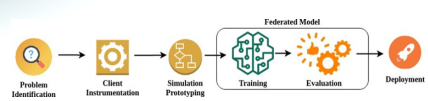

Figure 1:Life cycle of Federated Learning

Initially, in the FL process, a model engineer develops a model for a particular application. In natural language processing, a domain expert may develop a prediction model for next word prediction to use in a virtual keyboard application. Figure 1 depicts the primary components and actors involved in the FL process. A typical workflow of the FL model can be realized, as shown in Figure 1. The life cycle of a Federated Learning (FL) model consists of six stages, as shown in Figure.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 8, Special Issue 1, August 2021

- Problem identification: Here, the model engineer finds a problem that needs to be solved with Federated Learning at this point.
- Client instrumentation: The essential training data is stored locally by the clients. Additional data or metadata may be required in a few circumstances, such as the labels for a supervised learning task.
- Simulation prototyping: The model engineer prototypes the model architectures and then uses a proxy dataset to test the learning hyperparameters in a Federated Learning simulation.
- Federated model training: Typically, all federated training processes are started to train multiple model versions. For additional training, we might employ other optimization hyperparameters.
- Federated model evaluation: The models are then assessed, and the top candidates are chosen when the Federated Learning tasks have been suitably trained. Various metrics computed on standard datasets in the datacenter are used in the analysis. The models are pushed to held-out clients after federated evaluation is completed on local client data.
- Deployment: Following the selection of a good model, it proceeds through the usual model launch procedure, which includes live A/B testing, manual quality assurance, and a staged deployment. The model's launch process is determined by the application owner, who is unaffected by how the model is trained.

IV. FEDERATED LEARNING TRAINING PROCESS

Federated Learning (FL) training process consists of five steps and a central server orchestrates the training process in FL setting, by iterative execution of the steps.

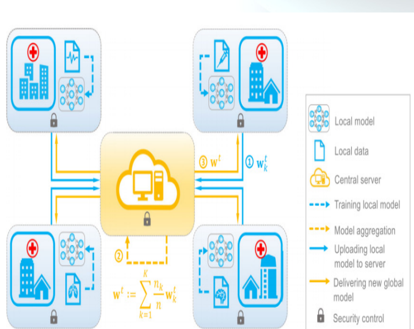

Figure 2:Federated learning

1. Train global model in the server.
2. Deploy global model to edge devices.
3. Optimize model from each edge device.
4. Upload locally trained model update.
5. Average the update values and apply the average to the global model.
6. Repeat step 2 to step 5.

a. FEDERATED LEARNING CATEGORIZATION

Horizontal Federated Learning (HFL): Because the data is spread across different devices, the data utilised to train the Federated Learning (FL) is non-identical. We divide Federated Learning (FL) into three categories based on how data is dispersed among numerous participating devices in the process: Horizontal FL, Vertical FL, and Federated Transfer Learning.

In FL, the central authority is responsible for executing the final global update of the model based on model updates from clients. We'll go into FL categorization based on data splitting in depth. Horizontal Federated Learning (HFL), also known as sample-based Federated Learning (FL), is used in the scenarios in which datasets share the same feature space but different space in samples, as shown in Figure 3.

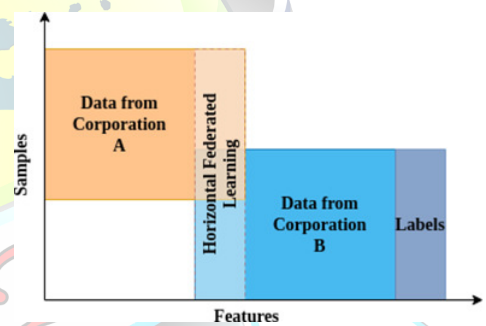

Figure 3 Horizontal Federated Learning (HFL) data partitioning.

Figure 3 depicts an example architecture for a horizontal Federated Learning (FL) system. Using a parameter or cloud server, k participants with the same data structure collaborate to develop a machine-learning model. It is assumed that no information is leaked to the server from any of the participants [25]. Sample architecture for a horizontal Federated Learning (FL) system is shown in Figure 4. The training process of the HFL system usually contains the following four steps.

- **Step 1:** Initially, all participants compute training gradients locally, and then use differential privacy [26], encryption [25], or secret sharing [24] approaches to conceal selected gradients. These masked results are then forwarded to the server.
- **Step 2:** The server then performs secure aggregation without learning any information about any participating client.
- **Step 3:** The server sends the aggregated results to all the participants.
- **Step 4:** The decrypted gradients are used to update each participant's model.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 8, Special Issue 1, August 2021

All of the steps are repeated until the loss function.

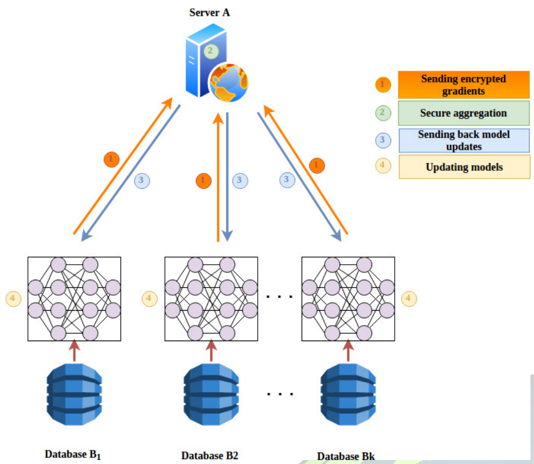

Figure 4 Horizontal Federated Learning (HFL) architecture.

Vertical Federated Learning (VFL):

When two datasets share the same sample ID space but differ in feature space, vertical federated learning (VFL) or feature-based federated learning (FL) is used. Vertically Federated Learning (VFL) combines these different features and computes gradients and training loss while maintaining privacy. Finally, it cooperatively constructs a model using data from both sides. After the learning phase, each party only has the model parameters that correspond to its features. Finally, when it comes to inference, the two sides must work together to obtain results.

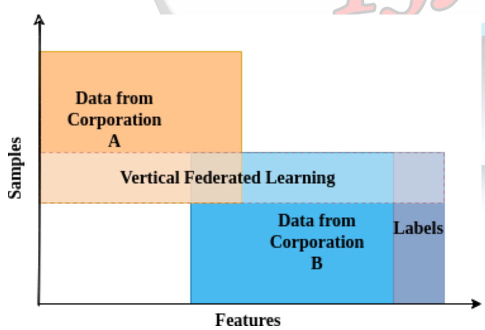

Figure 5. Vertical Federated Learning (VFL) data partitioning.

The training process of VFL can be divided into the following four steps, as shown in Figure 6.

- **Step 1:** Initially, Collaborator C creates encryption pairs and sends a public key to A and B.
- **Step 2:** Both A and B encrypt and exchange the intermediate results for gradient and loss calculations.
- **Step 3:** Companies A and B compute the encrypted gradients and add a mask, respectively. Company B also computes an encrypted loss. Both A and B send encrypted values to C.
- **Step 4:** C decrypts and send the decrypted gradients and loss back to A and B. Then A and B unmask the gradients and update the model parameters accordingly.

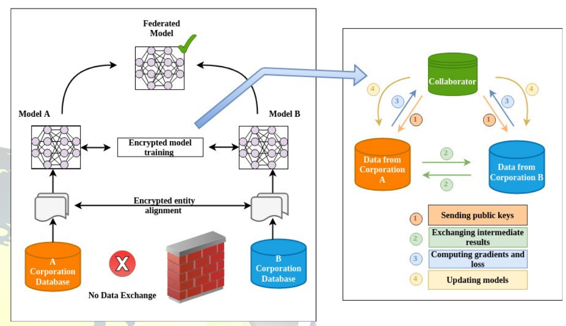

Figure 6. Vertical Federated Learning (VFL) architecture.

Federated Transfer Learning (FTL): In instances when two datasets differ in both sample and feature space, federated transfer learning is utilised. FTL is an important addition to existing Federated Learning (FL) systems since it addresses issues that aren't covered by existing FL algorithms. Learning a common representation between the features of parties A and B is known as transfer learning. It reduces the number of mistakes made while predicting labels for the target domain. Both parties must still compute the prediction results at inference time. Thus, in a federated context, transfer-learning approaches can be used to offer solutions for the full sample and feature space. In general, a common representation is learned between the two feature spaces using limited common sample sets, and then applied to only one-side feature samples to obtain predictions.

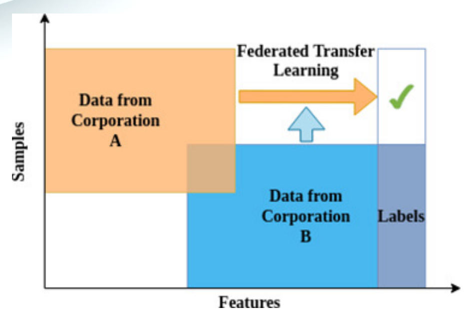

Figure 7. Federated Transfer Learning (FTL) data partitioning

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 8, Special Issue 1, August 2021

V. FEDERATED LEARNING IMPLEMENTATION FRAMEWORKS

Due to the variety of edge computing devices, federated learning is challenging to create and deploy in the real world. Different programming languages, frameworks, and hardware configurations may be used on these devices. There are numerous federated frameworks for simulating FL algorithms. TensorFlow Federated [27], PySyft [28], Federated AI Technology Enabler [29], PaddleFL [30], Leaf [31], and Clara Training Framework [64] are a few of the available tools and frameworks. TensorFlow Federated (TFF) created by Google is an extendable, powerful framework for executing Federated Learning (FL) research by simulating Federated Learning (FL) computations on realistic proxy datasets.

The Federated Core (FC) API is for expressing novel algorithms, while the Federated Learning (FL) API is for federated models that have been implemented. PySyft is a new open-source library for Federated Learning (FL) and privacy protection. It was created by the OpenMined community, and it integrates these several technologies for creating safe and private machine learning models.

We can start building privacy-preserving applications right away using these popular deep learning frameworks without needing to learn a new Deep Learning framework. As a result, Federated Learning (FL) and other privacy-preserving methods could be simply implemented in any application domain.

As a result, several frameworks are created to mimic FL in a server context. They do not, however, allow for large-scale client experimentation in a distributed mobile scenario. Another framework Leaf includes a set of open-source federated datasets, an evaluation framework, and a set of reference implementations using for practical federated environments.

VI. PRIVACY MECHANISMS IN FEDERATED LEARNING

Privacy is one of the crucial properties of Federated Learning (FL). Therefore, it requires analysis and security models to provide privacy guarantees. In this section, we briefly review various privacy techniques for Federated Learning (FL).

Secure Multiparty Computation (SMC): Multiple parties are involved in SMC security models, which give security evidence in a well-defined simulation framework to ensure that each party only knows its input and output. The parties have no information about the other parties in this situation. Zero-knowledge is very desirable, but it usually necessitates extremely sophisticated computation methods that are difficult to execute efficiently. . In some unusual circumstances, partial information disclosure may be permissible if security guarantees are provided. As a result, a security model using SMC can be built with fewer security criteria in exchange for efficiency.

Differential Privacy: Differential Privacy is introducing noise to the data or employing generalisation methods to obscure certain important attributes until a third party is unable to differentiate the individual, rendering the data unrecoverable and therefore protecting user privacy. The DP method is lossy as machine learning models are built after noise is injected, which can reduce much performance in prediction accuracy.

- **Local Differential Privacy:** Differential privacy can be achieved without requiring trust in a centralised server by having each client apply a differentially private transformation to their data before sharing it with the server.
- **Distributed Differential Privacy:** Here, the clients first compute and encode a minimal, focused report, and then send the encoded reports to a secure computation function, whose output is available to the central server. The output already satisfies differential privacy requirements by the time the central server can inspect it. The encoding is done to help maintain privacy on the clients. This privacy-preserving technique can be implemented via secure aggregations and secure shuffling.
- **Hybrid Differential Privacy:** This combines multiple trust models by partitioning users by their trust model preferences. There are two options before the advent of HDP like most-trusting and the least trusting model.

Homomorphic Encryption (HE): Through an exchange of parameters within the encryption method, homomorphic encryption is used to preserve user data privacy. The data and the model are not sent, and they cannot be guessed by the other party's data, unlike differential privacy protection. Certain mathematical operations can be done directly on ciphertexts without any prior decryption using homomorphic encryption (HE) techniques. Homomorphic encryption is a useful technique for Multiparty Computation (MPC) since it allows a participant to compute functions on values while keeping the data concealed.

Secure Aggregations: Secure aggregation is server-side functionality for a large number of clients. It allows each client to submit a tensor value, with the server learning just the aggregate function of the clients' values, which is usually the sum. Only an unordered collection of messages from all clients is learned by the server. Aside from the metadata in the message itself, the server is unable to link any communication to its sender Secure aggregation is server-side functionality for a large number of clients. It allows each client to submit a tensor value, with the server learning just the aggregate function of the clients' values, which is usually the sum. Only an unordered collection of messages from all clients is learned by the server. Aside from the metadata in the message itself, the server is unable to link any communication to its sender.

SecureBoost: In the context of Federated Learning, SecureBoost is a unique gradient-tree boosting technique (FL). There are two key steps to it. The data is first aligned under the privacy constraint. Second, it learns a shared gradient-tree boosting model collaboratively while securing all training data across different private parties.

SecureBoost is advantageous because it achieves the same degree of accuracy as a non-privacy-preserving technique while revealing zero information about each private data provider. The SecureBoost framework is theoretically demonstrated to be as accurate as other non-federated gradient tree-boosting techniques that bring data into one place.

VII. CONCLUSIONS AND FUTURE SCOPE

The emphasis on data privacy and security with the isolation of data has become the next challenges for AI, but Federated Learning (FL) has emerged with the solution. It could establish a united model for multiple enterprises and institutions while local data is protected so that enterprises could work together on data security. Thus, Federated Learning (FL) provides a platform to build a cross-enterprise, cross-data, and cross-domain ecosphere for AI, Machine learning and big data. This paper generally introduces the basic working of Federated Learning (FL), various architectures, privacy-preserving techniques of Federated Learning (FL), and discusses its potential in industrial applications. In the near future, Federated Learning (FL) would break the barriers between industries and establish a new community, wherein the data and knowledge could be shared. It ensures the safety and the benefits would be equally distributed based on the contribution of each participant. Finally, the essence and need of AI would be brought to every corner of our lives through Federated Learning (FL).

REFERENCES

[1] H. B. McMahan & Daniel Ramage. (2017), “Federated learning: Collaborative machine learning without centralized training data”, Google AI Blog. [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[2] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017), “Communication- efficient learning of deep networks from decentralized data”, *In Artificial Intelligence and Statistics,* ppl. 1273-1282.

[3] H. B. McMahan, Eider Moore, Daniel Ramage, Seth Hampson, & Blaise Aguera y Arcas. (2017), “Communication-efficient learning of deep networks from decentralized data”, *In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ppl 1273–1282.

[4] Jaideep Vaidya & Chris Clifton. (2003), “Privacy-preserving K-means clustering over vertically partitioned data”, *In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’03). ACM*, ppl 206–215. [Online]. Available: https://doi.org/10.1145/956750.956776

[5] Jaideep Vaidya & Chris Clifton. (2002), “Privacy preserving association rule mining in vertically partitioned data”. *In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’02). ACM*, ppl 639–644. [Online]. Available: https://doi.org/10.1145/775047.775142

[6] Jaideep Vaidya & Chris Clifton. (2004), “Privacy preserving naïve Bayes classifier for vertically partitioned data”, *In Proceedings of the 4th SIAM Conference on Data Mining*, ppl 330–334.

[7] Murat Kantarcioglu & Chris Clifton. (2004), “Privacy-preserving distributed mining of association rules on horizontally partitioned data”, *IEEE Trans. on Knowl. and Data Eng*, ppl 1026–1037. [Online]. Available: https://doi.org/10.1109/TKDE.2004.45

[8] Adrià Gascón, Phillipp Schoppmann, Borja Balle, Mariana Raykova, Jack Doerner, Samee Zahur, &David Evans. (2016), “Secure linear regression on vertically partitioned datasets”, *IACR Cryptology*,892.

[9] Irene Giacomelli, Somesh Jha, Marc Joye, C. David Page, & Kyonghwan Yoon. (2017), “Privacy- preserving ridge regression with only linearly-homomorphic encryption”, *IACR Cryptology*, 979. [Online]. Available: https://eprint.iacr.org/2017/979

[10] Payman Mohassel & Yupeng Zhang. (2017), “SecureML: A system for scalable privacy-preserving machine learning”. *IACR Cryptology*.

[11] Murat Kantarcioglu & Chris Clifton. (2004), “Privacy-preserving distributed mining of association rules on horizontally partitioned data”, *IEEE Trans. on Knowl. and Data Eng*, ppl 1026–1037. [Online]. Available:

https://doi.org/10.1109/TKDE.2004.45

[12] Hwanjo Yu, Xiaoqian Jiang, & Jaideep Vaidya. (2006), "Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data", *In Proceedings of the 2006 ACM Symposium on Applied Computing (SAC'06)*, ppl 603–610. [Online]. Available: https://doi.org/10.1145/1141277.1141415

[13] Hwanjo Yu, Jaideep Vaidya, & Xiaoqian Jiang. (2006), "Privacy-preserving SVM classification on vertically partitioned data", *In Proceedings of the 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD'06),* ppl 647–656. [Online]. Available: https://doi.org/10.1007/11731139_74

[14] Wenliang Du, Yunghsiang Sam Han, & Shigang Chen. (2004), "Privacy-preserving multivariate statistical analysis: Linear regression and classification", *In SDM*, Vol. 4, ppl 222–233.

[15] Li Wan, Wee Keong Ng, Shuguo Han, & Vincent C. S. Lee. (2007), "Privacy-preservation for gradient descent methods", *In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07)*, ppl 775–783. [Online]. Available: https://doi.org/10.1145/1281192.1281275

[16] O. Goldreich, S. Micali, & A. Wigderson. (1987), "How to play any mental game", *In Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC'87)*, ppl 218–229. [Online]. Available: https://doi.org/10.1145/28395.28420

[17] Yoshinori Aono, Takuya Hayashi, Le Trieu Phong, & Lihua Wang. (2016), "Scalable and secure logistic regression via homomorphic encryption", *In Proceedings of the 6th ACM Conference on Data and Application Security and Privacy (CODASPY'16)*, ppl 142–144. [Online]. Available: https://doi.org/10.1145/2857705.2857731

[18] Reza Shokri & Vitaly Shmatikov. (2015). "Privacy-preserving deep learning", *In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15)* , ppl 1310– 1321. [Online]. Available: https://doi.org/10.1145/2810103.2813687

[19] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, & Shiho Moriai. (2018). Privacy- preserving deep learning via additively homomorphic encryption. *IEEE Trans. Information Forensics and Security* (2018), ppl 1333–1345.

[20] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, & Jeroen Klop. (2020), " A survey on Distributed Machine Learning", ACM Computing Surveys, Vol. 53, [Online]. Available: https://doi.org/10.1145/3377454

[21] Diego Peteiro-Barral & Bertha Guijarro-Berdiñas. (2013), "A survey of methods for distributed machine learning", *Progress in Artificial Intelligence,* Vol. 2, No. 1, ppl 1–11.

[22] Abhijit Guha Roy, S. Siddiqui, S. Pölsterl, Nassir Navab, & Christian Wachinger. (2019), "BrainTorrent: A Peer-to-Peer Environment for Decentralized Federated Learning," arXiv: 1905.06731

[23] Chenghao Hu, Jingyan Jiang, & Zhi Wang, (2019), "Decentralized Federated Learning: A Segmented Gossip Approach", *1st International Workshop on Federated Machine Learning for User Privacy and Data Confidentiality*

[24] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, & Karn Seth. (2017), "Practical secure aggregation for privacy- preserving machine learning", *In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*, ppl 1175–1191. [Online]. Available: https://doi.org/10.1145/3133956.3133982

[25] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, & Shiho Moriai. (2018), "Privacy- preserving deep learning via additively homomorphic encryption", *IEEE Trans. Information Forensics and Security,* Vol. 13, No. 5, ppl 1333–1345.

[26] Reza Shokri & Vitaly Shmatikov. (2015), "Privacy-preserving deep learning", *In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15),* ppl 1310– 1321. [Online]. Available: https://doi.org/10.1145/2810103.2813687

[27] The TFF Authors. (2019), TensorFlow Federated. [Online]. Available: https://www.tensorflow.org/federated

[28] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, & Jonathan Passerat-Palmbach. (2018), A generic framework for privacy preserving deep learning.

[29] The FATE Authors. (2019) Federated AI technology enabler,

[30] The PaddleFL Authors. (2019) PaddleFL. [Online]. Available: https://github.com/PaddlePaddle/PaddleFL

[31] The Leaf Authors. (2019), Leaf. [Online]. Available: https://leaf.cmu.edu/

[32] The Clara Training Framework Authors. (2019), NVIDIA Clara. [Online]. Available: https://developer.nvidia.com/clara.