A Literature Study on Federated Learning

by Leyu Liu
Email: l3.liu@student.vu.nl
Supervisor: Adam Belloum

September 24, 2021

Abstract

Federated Learning(FL) is a decentralized learning approach that trains machine learning models on multiple devices collaboratively under the coordination of a server[1, 2]. It has been facing many challenges that are affecting the model performance, such as heterogeneous data challenge, heavy communication cost and privacy issues. In this paper, different FL algorithms are reviewed that highlight many existing FL problems along with various optimization techniques, aiming to raise attention for further development of FL.

Index Terms: Machine Learning, Federated Learning, Non-IID Data, Data Partitioning, Optimization, Privacy Preserving

1 Introduction

As IT is developed rapidly, Machine Learning (ML) has been extensively used and shows effectiveness in various fields. However, to train a ML model with excellent performance, it is required to have a large number of training data while these datasets might contain sensitive information that are not supposed to be revealed to the public (e.g. medical data). With increasing concern of data security, Federated Learning (FL) has emerged recently to enable privacy-preserving machine learning. However, with the decentralized setting, the amount of data in each device might be insufficient. It is difficult to train a robust model in FL with small amount of data, compared with centralized training. Thus, The main goal of FL is to preserve data privacy while maintaining good performance [3, 2, 1, 4, 5]. The general FL framework consists of a central server and multiple clients. Each client holds a local

dataset that is not shared with other parties. The server has a global model that is designed to train on the clients' data. During the model training, the global model is distributed by the server to each client to train a local model. Then, the server aggregates all the model parameters and results from all clients and updates the global model while the decentralized training process repeats until convergence [1]. As data is not shared in the FL process, the desired model can be trained without leaking any private information. Many recent research works have adopted this FL framework and try to improve it with better performance and enhanced security.

This literature study aims to explore the potential of FL and thus it focuses on highlighting the current trends, the challenges facing FL, and the potential of the FL as an approach for privacy-preserving machine learning. Thus, we propose the following research questions:

- RQ1: what are the current research trends for Federated Learning?
- RQ2: what are the challenges faced by Federated Learning?
- RQ3: what are the possible future research directions for Federated Learning?

The methodology used in this paper includes selecting literature, defining research questions, classifying literature into different categories, presenting findings and summarizing the paper. To select literature, we used academia tools such as Scopus and Google Scholar. As shown in the Appendix, 20 papers were selected based on the large number of citations, how recent they were published and whether they were published by high-ranked journals or conferences (e.g. IEEE Conference on Computer Communications, ACM SIGSAC Conference on Computer and Communications Security, IEEE Internet of Things Journal and etc.).

As FL uses data from different user devices, it is highly likely that the local data in each client is not independent and identically distributed (non-IID) data. Because of the difficulty to train non-IID data, the non-IID data challenge becomes an important factor to study. Another interesting perspective for FL research is data partitioning. Because of different data characteristics, data samples can can be partitioned differently in clients. Based on the overlapping data samples (common users or common features) among clients, the FL training process can be designed differently to keep good performance. FL also has many other existing problems such as heavy communication cost, slow convergence and low model accuracy. It has been a popular trend for researchers to investigate different kinds of federated optimization techniques and improve the FL process. Besides, as mentioned above, data privacy has

always been the primary concern in FL studies. Thus, we chose to review these scientific publications based on how they tackle non-IID data, how they partition data, their optimization techniques and methods used to preserve data privacy.

We structure this paper into 4 sections. After **Introduction**, some general concepts related to FL are introduced in **Overview**, including the definition of data partitioning, IID data and non-IID data, applications to machine learning and privacy risks and security mechanisms. Different scientific publications are compared and discussed in details in **Discussion**. Lastly, the summary of the paper is shown in **Conclusion** as well as the possible future research directions for FL.

2 Overview

2.1 Federated Learning Algorithms and Data Partitioning

Based on how data is distributed in each client, FL can be classified into three categories, namely *horizontal* federated learning, *vertical* federated learning and federated transfer learning [4, 5, 2]. In horizontal federated learning, each client holds a local dataset with different users but same features, while in vertical federated learning, each dataset can have many common users but different features of the same users. Federated transfer learning is a hybrid approach of horizontal federated learning and vertical federated learning that typically deals with insufficient or unlabelled data where there is not much overlap in users and features among different datasets [2, 5].

2.2 Applications of Federated Learning to Machine Learning Models

FL is widely applied to machine learning problems. Table 1 lists the applications of FL to various ML models to solve various problems. A ML model is trained in the FL process for every client with the objective of minimizing the averaging model training loss [5, 4]. Most of the published research works use Neural Network(NN) models, such as convolutional neural network (CNN) and long-short-term memory (LSTM) to demonstrate the effectiveness of their FL algorithms with high learning accuracy for various ML problems. The most widely used ML use case is image classification. Some other ML use cases such as sentiment analysis and synthetic dataset generation, next

word prediction, speech recognition, document classification, financial problem classification and credit score prediction are also used to illustrate the applicability of FL under different ML scenarios. Compared with NN models, linear models, including support vector machines (SVM) and logistic regression, have less model complexity and easier to implement, though the performance may not always be the best [5]. Thus, linear models are usually used with certain optimizations in FL systems. Lastly, tree models, such as the tree boosting algorithm, are also efficient machine learning models that can be combined with FL framework.

Ref	Paper	ML model	ML Use case
[6]	McMahan, et al.	CNN & LSTM	Image classification & Next word prediction
[7]	Chen et al.	CNN & LSTM	Handwritten digit image recognition & Human activity recognition
[8]	Kang et al.	CNN	Image classification
[9]	Yang et al.	CNN	Image steganalysis
[10]	Gao et al.	CNN	Electroencephalography (EEG) classification
[11]	Mills et al.	CNN	Image classification
[12]	Xu et al.	CNN	Image classification
[13]	Li et al.	LSTM & multinomial logistic regression	Image classification & Sentiment analysis & Synthetic Dataset generation
[14]	Dimitriadis et al.	LSTM	Speech recognition
[15]	Yu et al.	SVM	Cardiac image classification
[16]	Yang et al.	Parallel Distributed Logistic Regression	Image classification & Document classification
[17]	Cheng et al.	Gradient Tree Boosting algorithm	Financial problem classification & Credit score prediction

Table 1: Various ML models used in FL algorithms

2.3 Federated Learning and Non-IID Data Challenges

According to Ting et al. [4], IID data refers to independent and identically distributed data. In a FL setting, the non-IID data is often more challenging to work with as training data. Non-IID means these datasets from different clients are independently generated and often heterogeneous with different distributions. With the statistical heterogeneity, Ting et al. pointed out that the non-IID data in the FL training process could cause problems in terms of the FL performance. Non-IID can also affect the predicted accuracy which could be largely dependent on the divergence of non-IID data. As consequence researchers in this field proposed many state-of-art approaches to deal with problems caused by non-IID data, such as reducing the communication overhead and speeding up the model convergence [6, 11, 7, 18] and the details are discussed further in Chapter 3.

2.4 Privacy and Security

The main purpose of FL is to protect the sensitive data in each client from being exposed. However, there exit many adversary attacks targeting machine learning models, the learning process or input data, leading to the leakage of private information [5]. To enhance security and preserve user privacy, many proposed FL systems adopted different privacy mechanisms. For instance, differential privacy is a widely used security mechanism where random noise is added to data or model (e.g. Gao et al. [10]). As for cryptographic methods, the input and output messages are usually encrypted during communication (e.g. Bonawitz et al. [19]). While the privacy mechanisms secures the data privacy, they might have side effects that limits the performance of FL systems. Hence, recent novel FL systems have often seek a trade-off between security and performance [5].The details of each security mechanism and how it is applied in different FL algorithms are shown in Chapter 3.

3 Discussion

3.1 Non-IID Data Challenges

With data heterogeneity, FL model training can be problematic. In this section, several FL algorithms that dealt with some performance issues with non-IID data are briefly introduced. Table 3 includes the optimization techniques for non-IID data. The corresponding FL algorithms are introduced in the following, while the optimization details are described in later sections.

McMahan et al. [6] proposed *Federated Averaging*(*FedAvg*) algorithm that showed robustness on unbalanced non-IID data. *FedAvg* uses an iterative synchronous update process for the convergence of the global model. A fixed set of clients are selected for each round to execute the global algorithm. Then, clients compute model updates and exchange intermediate results with server. Server averages the resulting model parameters and updates the global model to send back to clients to continue the iterative process. The algorithm was tested on both IID and non-ID datasets and the results proved its effectiveness on non-IID data. Besides, Mills et al. [11]improved *FedAvg* and proposed a *communication-efficient FedAvg* (*CE-FedAvg*). *CE-FedAvg* performed well in non-IID data scenarios using robust optimization and compression methods. It achieved less communication cost and less number of communication rounds to reach the same test accuracy. Similarly, Chen et al. [7] proposed another novel approach that focused on non-IID data and communication and computation optimization by using asynchronous learning and temporally weighted aggregation. Chen et al. partitioned datasets to fit the non-IID data requirements and outperformed *FedAvg* with fewer communication rounds given the same test accuracy. Karimireddy et al. [18] showed that *FedAvg* had the disadvantages of unstable and slow convergence when dealing with non-IID data. Because heterogeneous data has different distributions, each client will have have different local optimum for their own loss function, which is far away from the global optimum. Thus, it can cause the problem of client drift where the average of the local updates is the average of the local optimums, which is different from the true global optimum. To solve the problem, Karimireddy et al. proposed *SCAFFOLD*. It used correction term to make the local gradient move towards the global optimum after each client update. Thus, the algorithm had faster convergence for heterogeneous data.

Unlike the previous FL algorithms that are based on *FedAvg*, Smith et al. proposed *MOCHA* [20] which combined FL algorithm with multi-task learning (MLT) to extract relations among non-IID data. *MOCHA* takes each client as a task and the weights updating is regarded as a dual problem that is applied to the MLT framework such that every client can obtain a model targeting their own local dataset for solving the non-IID data problem. Apart from it, there have been previous research works that were targeting data heterogeneity of specific data, such as electroencephalography (EEG) data. Gao et al. [10] proposed a heterogeneous FL algorithm for EEG data classification specifically, aiming to preserve EEG data privacy and improve classification accuracy.

3.2 Data Partitioning

As described in Overview, FL can be classified based on data partitioning. In this section, different FL algorithms with different data partitioning are introduced in details, including horizontal FL algorithms, vertical FL algorithms and federated transfer learning algorithms (Table 2).

Ref	Paper	Data Partitioning	Method
[21]	Kim et al.	Horizontal FL	BlockFL
[10]	Gao et al.	Horizontal FL	hierarchical heterogeneous horizontal FL
[15]	Yu et al.	Horizontal FL	privacy-preserving SVM
[17]	Cheng et al.	Vertical FL	SecureBoost
[16]	Yang et al.	Vertical FL	Parallel distributed logistic regression for vertical FL
[22]	Feng et al.	Vertical FL	multi-Participant multi-Class vertical FL
[8]	Kang et al.	Vertical FL	Semi-supervised Vertical Federated Learning with MultiView Training
[23]	Romanini et al.	Vertical FL	PyVertical
[14]	Dimitriadis et al.	Federated Transfer Learning	Federated transfer learning with dynamic gradient aggregation
[9]	Yang et al.	Federated Transfer Learning	FedSteg

Table 2: FL algorithms classified in terms of data partitioning

3.2.1 Horizontal Federated Learning Algorithms

Many FL algorithms were proposed to deal with horizontally partitioned data. As each client holds different datasets with common features but different users, the total user sample size is increased in the whole FL system compared with local training. They usually upload local gradients to server for aggregation and use the increased user sample size to improve model accuracy [2]. *BlockFL* [21] was a horizontal FL framework that used the blockchain network. Kim et al. pointed out that traditional FL system might suffer from the vulnerability of the single central server and clients

or devices with more data were not promoted to contribute in FL process. Thus, *BlockFL* was designed to solve these issues. It allows each device to compute local model updates and global model updates to prevent the system suffering from single-server malfunctioning. It also provides data rewards to encourage devices with larger size of datasets. Besides, Gao et al. proposed *Hierarchical heterogeneous horizontal FL* (*HHHFL*) that targets on the data heterogeneity of EEG data. It uses manifold projection for high dimensionality reduction and it adopts the FL process for privacy preserving. In addition, Yu et al.[15] proposed *privacy-preserving SVM* (*PP-SVM*) that was designed for horizontally partitioned data. It constructs a privacy-preserving global SVM model efficiently. The key information is secured using secure computation of the kernel matrix. The results showed that *PP-SVM* preserved data privacy through secure computation with linear communication cost.

3.2.2 Vertical Federated Learning Algorithms

In vertical FL, the vertically partitioned dataset consists of data with overlapped users. Researchers tend to make use of the overlapped user data and aggregate different features [2]. Cheng et al. [17] proposed *SecureBoost* that trained a federated gradient tree boosting model with high accuracy. *SecureBoost* follows the general FL workflow. Sensitive data is encrypted before communication. In the local data, some users are the same in different clients while there exist distinct users in each client. Thus, data alignment was performed for finding common data samples. To construct the tree boosting algorithm, the optimal tree is built by computing local optimal split and optimal leaf weight in each client. The encrypted results will be sent to server for aggregation. Experiment results proved that *SecureBoost* secured data privacy with accuracy score as good as the non-federated tree boosting algorithm.

In addition, Yang et al. [16] proposed vertical FL approach for parallel distributed logistic regression models. Unlike other vertical FL algorithms that use a coordinator as a third party for updating global model, Yang et el. came up with a new approach that removed the coordinator. It reduces the risk of information leakage and simplifies the system. Thus, every two clients could communicate directly with each other for secure model updates. In each client, there is a parameter server and several worker nodes where the communication within a client only happens between every worker node and the parameter server. Results showed that the system architecture worked efficiently with large-scale dataset with fast training process.

To work with multiple clients and multiple classes in vertical FL setting, Feng et al.[22] proposed *multi-Participant multi-Class vertical FL* (*MMVFL*)

based on the concept of multi-view learning while preserving data and label information. Each client learns local model parameters and pseudo-label matrix that represent data points till the model converges. As the clients only send pseudo-label matrix updates to server for the purpose of secure label sharing, the original data information is well preserved.

In vertical FL, most algorithms tended to work with the overlapped data samples. To deal with the problem of limited common data, Kang et al. [8] proposed a vertical FL algorithm with *multi-view learning* (*FedMVT*). It utilized the semi-supervised learning method to generate data representations and predict pseudo-labels. Given a training dataset, two NN models are used to learn feature representations of overlapped data and non-overlapped data in each client. Three softmax classifiers are trained to improve the representations and predict pseudo-labels for unlabelled data.

In addition, Romanini et al. [23] proposed *PyVertical*. It a vertical FL framework that can be combined with split NN models for privacy-preserving MNIST classification. The NN model is split into multiple segments and sent to each client to train on their local data samples. The results are concatenated in the server for the global model updates while the communication between any two parties is secured by using a cryptographic method called Private Set Intersection (PSI) for computing the intersection of data elements.

3.2.3 Federated Transfer Learning Algorithms

Federated Transfer Learning (FTL) frameworks have been widely used to deal with unlabelled data or datasets between two parties that do not have much overlap [2]. Previous researchers have proposed many novel FTL approaches with optimizations of communication or privacy preserving. Dimitriadis et al. [14] proposed a FTL framework. It processes N random sampled clients obtained from the client pool in each iteration step for model training. The intermediate results are aggregated by the server for model updates. It adopts unsupervised training method with N-best hypothesis to predict data labels. It also combines a TTS-based (text-to-speech-based) audio generation method to improve the model training for speech recognition task.

Yang et al. [9] proposed *FedSteg* for federated transfer learning. It improves the performance of steganographic image detection with CNN-based models. The proposed model follows the general FL process: a cloud model is distributed to each client for local training and a server aggregates the results and parameters for updating the global cloud model. All the communication is protected by encryption methods. By adopting transfer learning, each client is able to train a personalized model based on their own data through

the integration of the global cloud model and the previous local model. The results showed *FedSteg* outperformed other non-federated steganalysis methods with data privacy secured effectively.

3.3 Optimization

With limited communication bandwidth between client and server, large communication cost becomes the major problem in federated optimization [6]. Besides, the accuracy of the FL algorithm and the convergence rate have always been important factors to consider when developing FL algorithms. Previous researchers have proposed various optimization techniques (Table 3) to improve FL learning process such as speed up convergence, improve model accuracy and reduce communication overhead, which is described in details as follows.

To optimize the FL process, *FedAvg* [6] increased computation in each client before communicating with server for averaging. The results showed the most effective approach to have additional computation was to increase the number of epochs for local stochastic gradient descent (SGD) updates. The optimization successfully reduces the number of communication rounds while keeping good test accuracy. However, it results in heavy communication overhead in each round and the model has slow convergence rate due to data heterogeneity. Hence, Mills et al. [11] proposed *CE-FedAvg* that improved the *FedAvg* by using distributed Adam optimization [24] and state-of-art compression methods. In *CE-FedAvg*, Adam optimization is used instead of a normal stochastic gradient descent (SGD) for faster convergence. The model uploaded from client to server is compressed first which reduces the total number of uploaded data significantly. Hence, both the communication rounds and the communication cost in each round are reduced. According to Chen et al. [7], NN shallow layers have a much smaller number of important parameters for learning performance compared with deep layers and should have more regular updates. Therefore, instead of updating the entire model synchronously (e.g. *FedAvg*), Chen et al. proposed asynchronous model updates that allowed parameters in NN shallow layers to be updated more frequently. Such optimization leads to less data exchange in each communication round. *SCAFFOLD* [18] is another optimization algorithm based on *FedAvg*. In addition to the original *FedAvg* steps, two state variables are maintained for the client ("client control variate") and server ("server control variate") to compute gradients and approximate the ideal updates by overcoming gradient dissimilarity. Thus, it provides fast and strong convergence.

Dimitriadis et al. [14] proposed a hierarchical optimization method in their FTL framework that significantly improved the convergence speed and

Ref	Paper	Algorithm	Optimization Technique	Non-IID data
[6]	Mcmahan et al.	FedAvg	reduce communication rounds with additional computation	✓
[11]	Mills et al.	CE-FedAvg	distributed Adam optimization and state-of-art compression for less communication and faster convergence(based on FedAvg)	✓
[7]	Chen et al.	enhanced FL technique	temporarily weight aggregation for better convergence and asynchronous model updates for less communication (based on FedAvg)	✓
[18]	Karimireddy et al.	SCAFFOLD	use control variables to overcome gradient dissimilarity for better convergence (based on FedAvg)	✓
[14]	Dimitriadis et al.	federated transfer learning framework	hierarchical optimization method with optimizers on both client and server side for faster convergence and less communication	
[13]	Li et al.	FedSaE	self-adaptively predict affordable workload and select clients for fast convergence and better accuracy	
[20]	Smith et al.	MOCHA	efficient weight updating for FL multi-task learning to reduce communication cost and speed up the learning process	✓
[10]	Gao et al.	HHHFL	target EEG data and improve model accuracy	✓

Table 3: FL Optimization Techniques

lowered the communication overhead. To be more specific, it uses a client-side optimizer and server-side optimizer. It also aggregates the estimated

local gradients to boost the training process.

FedSaE [13] is a self-adaptive framework. It can predict affordable workload for each client and select clients adaptively for high-quality model training. It deals with the problem where different client has different system configurations(system heterogeneity). Though calculating workload and selecting clients can cause slight overhead, the results showed that the overall speed of convergence was significantly increased as well as the model accuracy.

HHHFL [10] maps the heterogeneous EEG data to one common embedding space(manifold projection). It trains a privacy-preserving classifier with NN. A combined loss function is used that is composed of both domain loss and classification loss. Model accuracy is significantly improved compared with the non-FL EEG classification algorithm. Apart from it, *MOCHA* [20] optimizes the FL process by considering data heterogeneity and system heterogeneity. The algorithm gives each node flexibility of defining their own parameters to perform efficient weight updating. It increased the speed of the FL multi-task learning greatly.

3.4 Privacy and Security

Data privacy protection has been a primary focus in all FL framework. In addition to improving the model performance, researchers often take a great effort exploring different privacy-preserving approaches to prevent data leakage. In this section, various security mechanisms are discussed (Table 4) as well as how they deal with different privacy issues under different circumstances.

Ref	Paper	Security Mechanism/ Advanced Attack
[19]	Bonawitz et al.	cryptographic methods (use authentication, public-private key pair and encrypted messages)
[9]	Yang et al.	secure multi-party computation and differential privacy
[25]	Lu et al.	differential privacy(use noise and laplace mechanism)
[26]	Wang et al.	GAN-based user-level attack
[12]	Xu et al.	use a double-masking protocol with public-private key pair, encryption and verification of result correctness

Table 4: Privacy-Preserving Mechanisms used in FL algorithm

Bonawitz et al. [19] proposed a FL protocol with cryptographic methods. Its security mechanism includes secret sharing, key agreement, authenticated encryption, pseudorandom generator (PRG), signature scheme and public key infrastructure. The secret information is split into n segments before sharing. Besides, a public-private key pair is generated to protect against active adversaries. In each communication round, the messages are encrypted and sent to the other party with a private key for authentication. Besides, PRG can produce indistinguishable random seed. The signature scheme can verify the validity of a public key with a message and a signature, while the public key infrastructure is designed to register clients for identification.

Besides, in *FedSteg* [9], the shared parameters among different parties are encrypted. Several encryption approaches are exploited, such as secure multi-party computation and additively homomorphic encryption, to ensure privacy-preserving steganalysis. Results showed that using differential privacy provided similar protection to additively homomorphic encryption method, proving the extensibility of different security mechanisms applied to FedSteg.

Lu et al. [25] proposed their FL algorithm using differential privacy(DP) mechanism to ensure data privacy. During model training, noise is added to local data for the local model to be trained on each client. Laplace mechanism is used along with a sensitivity "s" on local data model to be broadcasted to other clients. The whole DP training process repeats until model converged or timeout.

Privacy attacks are well-developed using generative adversarial networks (GAN) to reconstruct data representatives among all clients or even targeting a specific client. To illustrate the risk of privacy leakage at user level, Wang et al. [26] proposed a client-specific multi-task GAN model. It can identify a specific client and perform strong attacks to recover client data precisely. The attacks include a passive attack that analyzes updates from clients and an active attack that sends an isolated model to the victim client for recovering data samples. Results proved that such user-level attack was more effective than the regular GAN-based attacks and inversion attacks. Hence, it raises attention on the improvement of privacy-preserving mechanisms in future FL development.

Additionally, when FL is combined with deep neural networks, privacy leakage could happen during model training and it might be difficult to verify the correctness of the aggregated results, thus, Xu et al. [12] proposed VerifyNet to solve these security problems with a double-masking protocol. Except the server and clients, the proposed system includes a trusted authority that assigns the public-private key pair to each client. The communication between server and clients is encrypted and the server will need to generate

a proof for each aggregated result for the verification of correctness. Results showed the security of FL was enhanced, though it caused large communication cost due to the verification and masking process.

4 Conclusions

FL enables prediction models to be collaboratively learned on distributed devices. In this paper, we include some important factors of FL including data partitioning, data heterogeneity, privacy preserving, optimization and applications. There are yet some other issues that are not covered by the paper are still worth studying, such as the application of FL in the industry and business related issues. With the summary and discussion of various state-of-art FL algorithms, we hope to raise attention and motivate further research in FL. To answer RQ1 for the current trends of FL, recent FL algorithms address the issues of model learning, privacy concerns and data challenges. Many FL systems focus on developing efficient and practical FL with consideration of real-world datasets. Besides, federated optimization techniques are widely applied to improve the model accuracy, reduce overhead and enhance security.

To answer RQ2, this paper discusses many challenges encountered by researchers. The first challenge is data privacy leakage risks. The server and clients need to exchange results in the middle of FL process. It is important but challenging to protect the raw data without being revealed in the communication. Another challenge is the heavy communication cost. As the number of devices used in FL is large, communication can be a bottle neck for FL performance [2]. It is necessary to reduce communication overhead and speed up FL. Besides, local data in each client can have different characteristics and distributions, which can affect the model performance largely. Thus, effective FL optimization techniques should be developed to deal with these data challenges. Novel FL systems have addressed these challenges and made great efforts improving the general FL workflow to make it more efficient and secure.

RQ3 refers to the possible future research directions of FL. Though the performance of FL has been improved significantly in recent years compared with the original approach, there are still many remaining problems in FL that need to be solved in the future. Data security has always been a concern in FL. Despite of all the efforts adopting privacy-preserving mechanisms, the risk of leaking sensitive information is still high, especially with the enhanced adversary attacks. Thus, it leads to a future research direction of more secured FL with methods such as reliable client selection and applying client-

specific privacy restrictions [2, 5]. Besides, with an increasing number of devices participating the FL process, the communication cost could be really high and it is important to balance the trade-off between communication cost and computation cost [2]. Moreover, though FL is popular, its application is still limited. It could be also an interesting direction to extend its application (e.g. in unsupervised setting) and modify system architecture [5]. Lastly, clients might have different configurations in their systems. As FL is getting more popular and more devices are getting involved, system heterogeneity should be further studied in future federated optimization [2, 13].

References

[1] P. K. et al., "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021. [Online]. Available: http://dx.doi.org/10.1561/2200000083

[2] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021. [Online]. Available: https://doi.org/10.1016/j.knosys.2021.106775

[3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, p. 1–1, 2021. [Online]. Available: http://dx.doi.org/10.1109/COMST.2021.3075439

[4] D. Ting, H. Hamdan, K. A. Kasmiran, and R. Yaakob, "Federated learning optimization techniques for non-iid data: A review," *International Journal of Advanced Research in Engineering and Technology (IJARET)*, vol. 11, pp. 1315–1329, 2020. [Online]. Available: http://dx.doi.org/10.34218/IJARET.11.12.2020.125

[5] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," 2019, arXiv:1907.09693.

[6] H. 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 (AISTATS)*, 2017, arXiv:1602.05629.

[7] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally

weighted aggregation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, p. 4229–4238, Oct 2020. [Online]. Available: http://dx.doi.org/10.1109/TNNLS.2019.2953131

[8] Y. Kang, Y. Liu, and T. Chen, "FedMVT: Semi-supervised vertical federated learning with multiview training," in *International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with IJCAI 2020 (FL-IJCAI'20)*, 2020, arXiv:2008.10838.

[9] H. Yang, H. He, W. Zhang, and X. Cao, "FedSteg: A federated transfer learning framework for secure image steganalysis," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1084–1094, 2021. [Online]. Available: https://doi.org/10.1109/TNSE.2020.2996612

[10] D. Gao, C. Ju, X. Wei, Y. Liu, T. Chen, and Q. Yang, "HHHFL: Hierarchical heterogeneous horizontal federated learning for electroencephalography," 2020, arXiv:1909.05784.

[11] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5986–5994, 2020. [Online]. Available: http://dx.doi.org/10.1109/JIOT.2019.2956615

[12] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: Secure and verifiable federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2020. [Online]. Available: https://doi.org/10.1109/TIFS.2019.2929409

[13] L. Li, M. Duan, D. Liu, Y. Zhang, A. Ren, X. Chen, Y. Tan, and C. Wang, "FedSAE: A novel self-adaptive federated learning framework in heterogeneous systems," *International Joint Conference on Neural Network (IJCNN)*, 2021, arXiv:2104.07515.

[14] D. Dimitriadis, K. Kumatani, R. Gmyr, Y. Gaur, and S. E. Eskimez, "Federated transfer learning with dynamic gradient aggregation," 2020, arXiv:2008.02452.

[15] H. Yu, X. Jiang, and J. Vaidya, "Privacy-preserving svm using nonlinear kernels on horizontally partitioned data," in *Proceedings of the 2006 ACM Symposium on Applied Computing*, ser. SAC '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 603–610. [Online]. Available: https://doi.org/10.1145/1141277.1141415

[16] S. Yang, B. Ren, X. Zhou, and L. Liu, "Parallel distributed logistic regression for vertical federated learning without third-party coordinator," 2019, arXiv:1911.09824.

[17] 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. [Online]. Available: http://dx.doi.org/10.1109/MIS.2021.3082561

[18] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 5132–5143, arXiv:1910.06378v4. [Online]. Available: https://proceedings.mlr.press/v119/karimireddy20a.html

[19] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1175–1191. [Online]. Available: https://doi.org/10.1145/3133956.3133982

[20] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 4427–4437, arXiv:1705.10467.

[21] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "On-device federated learning via blockchain and its latency analysis," 2018, arXiv:1808.03949.

[22] S. Feng and H. Yu, "Multi-participant multi-class vertical federated learning," 2020, arXiv:2001.11154.

[23] D. Romanini, A. J. Hall, P. Papadopoulos, T. Titcombe, A. Ismail, T. Cebere, R. Sandmann, R. Roehm, and M. A. Hoeh, "PyVertical: A vertical federated learning framework for multi-headed splitnn," in *Workshop on Distributed and Private Machine Learning (DPML) - ICLR*, 2021, arXiv:2104.00489.

[24] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2014, arXiv:1412.6980.

[25] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2020. [Online]. Available: https://doi.org/10.1109/TII.2019.2942190

[26] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019, pp. 2512–2520. [Online]. Available: https://doi.org/10.1109/INFOCOM.2019.8737416

A Appendix

Ref	Year	Method & Novelty	Paper
[6]	2017	propose FedAvg	Mcmahan et al.
[21]	2019	propose BlockFL for horizontal FL	Kim et al.
[10]	2020	propose HHHFL to deal with heterogeneous EEG data	Gao et al.
[15]	2006	combine horizontal FL with SVM	Yu et al.
[20]	2017	propose MOCHA as a federated multi-task learning approach for non-IID data	Smith et al.
[18]	2021	propose SCAFOLD for heterogeneous data and solve the "client-drift" issue	Karimireddy et al.
[11]	2020	improve FedAvg with Adam optimization and compression techniques	Mills et al.
[7]	2019	Use temporarily weighted aggregation and asynchronous model updates	Chen et al.
[17]	2021	propose SecureBoost and combine vertical FL with boosting trees	Cheng et al.
[16]	2019	combine vertical FL with parallel distributed Logistic Regression	Yang et al.
[22]	2020	propose MMVFL for multi-class and multi-participant problems	Feng et al.
[8]	2020	propose FedMVT as vertical FL in semi-supervised setting	Kang et al.
[23]	2021	propose PyVertical and combine verticla FL with split NN	Romanini et al.
[14]	2020	propose a federated transfer learning framework with hierarchical optimization and dynamic gradient aggregation	Dimitriadis et al.
[9]	2021	propose FedSteg as a FTL for image steganalysis	Yang et al.
[13]	2021	propose FedSaE to select participants self-adaptively and deal with the system heterogeneity issue	Li et al.
[19]	2017	propose a privacy-preserving protocol with cryptographic methods	Bonawitz et al.
[26]	2019	propose mGAN-AI attack that targets specific user	Wang et al.
[25]	2020	integrate FL with blockchain and apply DP	Lu et al.
[12]	2020	propose VerifyNet with double-masking protocol	Xu et al.