ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9746925

FEDERATED LEARNING CHALLENGES AND OPPORTUNITIES: AN OUTLOOK

Jie Ding, Eric Tramel, Anit Kumar Sahu, Shuang Wu, Salman Avestimehr, Tao Zhang

Alexa AI, Amazon

ABSTRACT

Federated learning (FL) has been developed as a promising framework to leverage the resources of edge devices, enhance customers' privacy, comply with regulations, and reduce development costs. Although many methods and applications have been developed for FL, several critical challenges for practical FL systems remain unaddressed. This paper provides an outlook on FL development as part of the ICASSP 2022 special session entitled "Frontiers of Federated Learning: Applications, Challenges, and Opportunities." The outlook is categorized into five emerging directions of FL, namely algorithm foundation, personalization, hardware and security constraints, lifelong learning, and nonstandard data. Our unique perspectives are backed by practical observations from large-scale federated systems for edge devices.

Index Terms— Distributed learning, nonstandard data.

1. INTRODUCTION

Federated learning [1, 2] is a popular distributed learning framework developed for edge devices. It allows the private data to stay locally while leveraging large-scale computation from edge devices. Its main idea is to learn a joint model by alternating the following in each so-called federated, or communication, round: 1) a server pushes a model to clients, who will then perform multiple local updates, and 2) the server aggregates models from a subset of clients. The design of practical FL systems is highly nontrivial since FL often involves millions of devices, unknown heterogeneity from various cohorts, limited on-device capacity, evolving data distributions, and partially labeled data. Inspired by our practical observations, we will list some critical challenges in the following five sections (demonstrated in Figure 1).

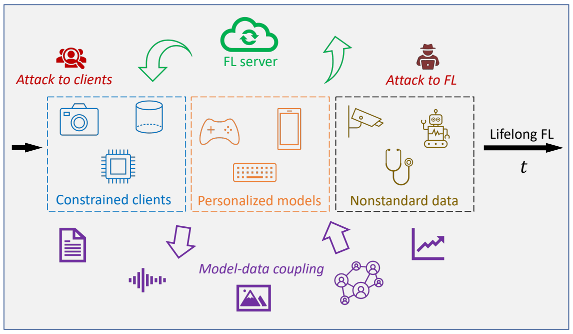

Fig. 1: Illustration of an FL system in our outlook.

2. ALGORITHMIC FUNDAMENTALS OF FL

FL goals. There are two general goals of supervised learning, namely prediction, which aims to learn a model with a small out-sample prediction risk, and detection, which will maximize the statistical power under a fixed false-detection rate. A principled FL design needs to consider unique perspectives that depend on the particular goal. We take the binary classification task as an illustrating example. Similar observations apply to more complicated tasks. Let the training data (Y_i, X_i), $i = 1, \dots, n$, and test data (Y, X) be IID random variables with values in $\mathbb{R}^d \times \{1, -1\}$.

1) *Prediction*. For the prediction purpose, one often looks for a classifier C such that the risk $\mathbb{P}(C(X) \neq Y)$ is small. It is known that the optimal classifier is $C_* : x \mapsto \text{sign}(f_*(x)) \in \{1, -1\}$ and $f_*(x) = \mathbb{E}(Y = 1 \mid x) - 1/2$. To train a classifier $\hat{C}_n$, a general approach is to learn $\hat{f}_n : \mathbb{R}^d \to \mathbb{R}$ and then let $\hat{C}_n(x) \triangleq \text{sign}(\hat{f}_n(x))$. Specifically, we operate the estimation from an (often) infinite-dimensional function space with only finite sample through sieve estimators, to trade-off between approximation errors and estimation variance, e.g., the empirical risk minimization

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(Y_i, f(X_i)) + \lambda_n \Omega(f), \qquad (1)$$

where $\mathcal{F}$ is a function class, Ω is a regularization functional, and L is a loss function. The fundamental problem is to choose a proper modeling vehicle to attain a rate-optimal $\hat{f}_n$ regarding the prediction risk. In the centralized setting, asymptotically optimal rules for rather general models have been studied (see [3, 4] and the references therein). In FL settings, however, there may not exist a way to equivalently operate (1), not to mention optimality, for general $\mathcal{F}$, L, Ω, and λ_n (except for finite-dimensional $\mathcal{F}$ and fixed-shape parameters). Thus, it is important to develop a deeper understanding of whether (1) admits an FL-based solution.

2) *Detection*. The other goal is to develop a powerful test of $H_0 : y = -1$ against $H_1 : y = 1$ given x. This detection problem is critical for many applications, e.g., keyword detection, anomaly detection, and drug discovery. Given a fixed false-detection rate $\mathbb{P}(f(x) > 0 \mid y = -1)$, we want to minimize the false-rejection rate $\mathbb{P}(f(x) < 0 \mid y = 1)$. The optimal decision rule is then given by the Neyman-Pearson Lemma and represented as a ROC curve. Though both goals require estimating a function f, they are fundamentally differ-

ent. An optimal decision rule for the prediction goal is often not suitable for detection. An example concerns a highly imbalanced dataset, say $\mathbb{P}(Y = 1) \gg \mathbb{P}(Y = -1)$. In that case, the detection rule only depends on $X \mid Y = -1$ and $X \mid Y = 1$, which is not affected by $\mathbb{P}(Y = 1)$, while the prediction rule uses $Y \mid X$, which is sensitive to the observed frequency of $Y = 1$. Consequently, a larger imbalance tends to produce a model that favors the majority class and leads to varying detection thresholds (if using the probability of $Y \mid X$) under a fixed false-detection rate. An example is the detection of occurrences of a keyword in a long data stream. Although the conflict and reconciliation of prediction- and detection-oriented modeling have been studied for centralized settings [3,5], little is known in FL settings. A particular question is whether an FL model of good prediction performance, trained from (1), will inevitably sacrifice detection power.

Limits of communication efficiency (in rounds). Empirical evidence shows that the standard FL that averages over suitably many local updates for communication efficiency can still converge to a desirable model. Meanwhile, overly aggressive local updates are likely to harm the performance due to the heterogeneity of finite data across clients. The fundamental problem is understanding lower limits of the number of communication rounds needed to converge to a model that performs similarly to centralized training. Consider two extremes of FL training. One is to let each client perform one step of local update, which corresponds to a centralized SGD training [6]. Under reasonable conditions, the training directly corresponds to the optimization of (1) but is not communication efficient. The other extreme is to let each client perform sufficiently many local steps until its convergence, and then the server aggregates only once. For example, suppose that two clients hold n_1 and n_2 different data, and they individually solve (1) to obtain $\hat{f}_{n_1}$ and $\hat{f}_{n_2}$, respectively. We say the problem (1) meets *α-divisibility* if the risk (after subtracting the Bayes optimal risk), denoted by R, satisfies $R(w_1\hat{f}_{n_1} + (1-w_1)\hat{f}_{n_2}) \leq \alpha R(\hat{f}_n)$ for all $w_1 = n_1/n$. Clearly, a smaller divisibility α implies less risk degradation in FL. For a constant α, the order of the risk rate is maintained with only *one round of communication*, representing the most communication efficiency. It is easy to show that α is asymptotically fixed for parametric $\mathcal{F}$ and some standard regularity conditions. In general, however, α may not be small, depending on many factors such as the function class, regularization, and loss types. The limits of communication rounds needed for rate-optimality in FL (if it exists) have yet to be studied.

3. PERSONALIZATION OF FL

Many FL applications aim to improve the clients of heterogeneous data. Examples are personalized voice assistants, precision medicine, cohort-specific autonomous driving. Unlike the conventional FL that aims to train one model, the goal of personalized FL is to learn a collection of client-specific models that reduce test errors beyond what is possible with a single global model.

Nature of personalization. Existing work on personalized FL often derives algorithms based on a heuristic optimization formulation, which aims to regularize the discrepancy between local parameters and global parameters. We consider the following generic formulation to operate the personalized FL from the server's perspective.

$$\min_{f_1,\ldots,f_M \in \mathcal{F}} \sum_{m=1}^{M} G_{m,n_m}(f_m) + \lambda_n \Omega_n(f_1, \ldots, f_M), \qquad (2)$$

where G_{m,n_m} denotes client m's local loss, $\mathcal{F}$ is the common function class, and Ω_n is a regularization term. Without Ω_n, it is equivalent to optimizing M personal objectives separately. The above formulation raises two concerns. First, it is unclear how to choose Ω. Even if one heuristically specifies a regularization, (2) does not tell what an upper-bound baseline is for a client's local performance by participating in FL. There are two direct lower-bound baselines of personalized FL, namely, each client performs local training without FL, and all clients participate in conventional FL training. Understanding the gap between lower and upper limits is practically essential and theoretically intriguing. The second concern is whether the current heuristic approaches can indeed leverage the server-side global model as an intermediary to exchange helpful information across clients. Designing a personalized FL requires calibrating the tradeoff between a client's local training and inheritance from the aggregated model. Sieve estimators beyond the empirical risk minimization (2) have yet to be developed. It is also interesting to establish connections between personalized FL with other frameworks such as assisted learning [7], meta learning [8], multi-task learning [9], and knowledge distillation [10].

Connections of personalization, fairness, and robustness. Suppose that there are different cohorts of users to which personalized FL is applied. It is foreseeable that the model trained for one cohort may not be systematically biased due to the influence of other cohorts. Thus, better personalization can be associated with improved fairness (in a proper sense). Meanwhile, the global model trained from personalized FL is expected to be more robust to the heterogeneity of particular clients, which leads to enhanced robustness. Thus, fundamental connections between personalization, fairness, and robustness in the FL setting deserve further research.

Is a personalized model sufficiently good? There are two kinds of quality assessment of a model in general. One is based on a utility (say accuracy), and the other is based on the systematic discrepancy with the underlying data generating process. For example, if Y is almost independent of X, the optimal predictive model has accuracy near one-half. Thus, a model with unsatisfactory accuracy does not necessarily mean we could further improve it. Instead, it depends on the nature of the task. The assessment of systematic detects of a model is often studied in the context of goodness-of-fit tests [11] and is an underexplored problem in the personalized FL setting. With more studies along this direction, we can understand whether an undesirable prediction is due to the incompleteness of FL designs or fundamental limitations of the data and task.

4. CONSTRAINED FL: MEMORY, COMPUTATION, SECURITY, AND PRIVACY

The predictive performance of a general FL system is determined by the approximation error (bias), estimation error (variance), and optimization error (gap with the intended minimum), which are further subject to the specified model, the nature of $Y \mid X$, the optimization solver, and the way data are distributed. While a deeper understanding of their tradeoffs in various settings is yet to be studied, there are additional practical constraints that add to the complication of FL.

Memory and computation constraints. A practical FL system for edge devices has to accommodate on-device hardware capacity. We discuss two tradeoff factors. First, under memory constraints, each on-device model needs to be small in size. This poses a methodological challenge for the server to leverage a large function class for large data (in hindsight). Promising directions include the use of knowledge transfer [12] and knowledge distillation [10], heterogeneous on-device models with neural parameter sharing [13], or neural architecture search [14]. Second, under computation constraints, devices may perform only a limited number of gradient updates [15]. This leads to the challenge of prolonged FL training time, which requires developing an efficient optimization solver on the client-side and an effective aggregation rule on the server-side. Also, lower-energy devices are infrequently activated, leading to a low client-sampling rate and potentially significant fluctuations in the aggregated model.

Security and privacy constraints. Since FL does not require raw data transmission, such as customers' images and audios, it is naturally suitable for improving data privacy. However, some research has shown that clients' identifiable information can be extracted from gradients [16]. It has motivated privacy-preserving techniques such as differentially private SGD. Furthermore, a device that stores data locally becomes a high-value security target for hackers. To alleviate such a risk, one may only keep small data on a device in a period, causing additional challenges in FL training (in Section 5). A related but distinct privacy concern is to deploy an already-developed but proprietary model API to the open world. In that case, we need a deeper understanding of tradeoffs between utility and model privacy [17]. Also, during the FL training, there has been a surge of recent algorithms to provide secure aggregation [18–20], where the general idea is to only learn aggregated parameters over cohorts of devices instead of single devices for the server. From the adversarial learning perspective, since an FL system is built on numerous participants, it can be vulnerable in situations outside the intended design, e.g., when some participants maneuver the training rule. It is critical for the FL designer to understand tradeoffs between prediction and resilience against various attacks and quantify an FL system's vulnerability in practice.

5. LIFELONG FL

Conventional FL systems are designed to learn only a prespecified task using data collected from a fixed channel. Thus, their generalization capability is quite limited for most real-world learning situations where the underlying data and tasks vary over time. For example, the on-device data distribution evolves due to user behavioral changes, or the label sets are updated due to new product releases. A general goal of lifelong FL is to develop systems that can *learn continuously during execution*, *improve performance stably over time*, and *adapt existing models to dynamic environment* without forgetting previous learning. We highlight two critical challenges.

Online updates with single-pass data. During the continuous execution of an FL system, a client often receives new data online instead of holding a static local dataset. Correspondingly, the local training in each round involves both existing and newly observed data, causing extra complications to the convergence of an FL model. Moreover, due to memory or security constraints, a client may only store a small window of online data, so the local training cannot replay historical data. In other words, the training has to be made from a single pass of data, bringing a major challenge for reliable online incremental updates of an FL model.

Coupling of model training and data generation. In many applications, new local data are collected through a client-side model (e.g., according to the soft-max values), which tend to be selectively biased towards the most recent inference model. Take the keyword detection problem as an example. Suppose that at time t, the FL-updated on-device model is $X \mapsto f_t(X)$. It will then be used to identify new data (denoted by $D_{t+1}(f_t)$) to feed into the FL training at time $t+1$. If another FL system were used, say f_t were replace with $f_t^{'}$, the newly fed data would become different, namely $D_{t+1}(f_t^{'})$. A generic formulation is

$$D_t \text{ and momentum } \xRightarrow[\text{Update}]{\text{FL}} f_t, \text{ Inference of } f_t \quad \xRightarrow[\text{Collect}]{\text{Data}} D_{t+1},$$

where Y may denote a ground-truth label or a machine-generated label (Section 6). Consequently, the training data cannot be treated as IID, and the test data based on predictive validation [3] tends to favor the most recently trained model. These cause practical hazards to evaluate (and thus improve) FL performance, at least in the following two aspects.

1) *Data distribution shift*. The data fed into the FL system at different rounds are Markovian, so any performance evaluation of FL based on historical data may not be reliable.

2) *Model procedure selection for large-scale dynamical systems*. Model procedure selection in the broad sense refers to the selection of learning models, tuning parameters, and evaluation tools as a part of a modeling procedure [3]. It is the key to lifelong FL, as we need to develop a sequence of models over time instead of a single model. Due to *model-data coupling*, standard predictive validation or A/B testing may not apply. We need new methods for efficient deployment and evaluation of multiple FL procedures simultaneously.

Rethinking of Batch Normalization (BN) layers to mitigate catastrophic forgetting. Catastrophic forgetting means that adaptions to new data or tasks tend to degrade the test performance on the original data or task domain. This is particularly a concern for lifelong FL because the clients often have heterogeneous data, and their model aggregation can easily introduce large noises and lose useful information. A potential mitigation of the curse of catastrophic forgetting is to rethink the role of BN layers [21] in FL to overcoming data heterogeneity across clients. Recall that during the training phase, each BN layer standardizes a data batch x_b by $\tilde{x}_b = (\sigma^2 + \epsilon)^{-1/2}(x_b - \mu) \cdot \gamma + \beta$, where γ, β are part of the model parameters and ϵ is a small constant. Here, μ, σ^2 are running statistics being tracked with momentum during the training, as a default BN option in popular frameworks that is often taken for granted. However, it is unclear how to specify μ, σ^2 during training and prediction in the FL setting. It was recently shown that an adaptation of BN named Static Batch Normalization (sBN) [13] can significantly accelerate the convergence and improve the performance of FedAvg [6] compared with BN and other forms of normalization. Similar observations were also made in [22]. In FL training with sBN, the affine parameters γ and β are aggregated as usual, but μ and σ^2 are only calculated from local data batch, namely the above x_b. After training, the server will query local clients and calculate the global statistics of μ and σ^2 for prediction. An intuition is that re-calibrating the BN statistics (as in sBN) can help data of heterogeneous nature leverage an already-trained feature extractor consisting of BN layers.

6. DATA INCOMPLETENESS, POLARITY, AND COMPLEX DEPENDENCY

Conventional FL for supervised learning often assumes that labeled data in the standard form of $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ are readily available to train. Here, $\mathcal{X}$ and $\mathcal{Y}$ denote the data space and label space (e.g., $\mathbb{R}^d$ and $\{1, -1\}$ for the example in Section 2), respectively. However, in many real-world applications, we often see data in nonstandard forms due to resource constraints, collection restrictions, or domain-specific formats. We summarize three critical but underexplored challenges regarding partially labeled X, partially observed $\mathcal{Y}$, and irregular X, respectively.

Data incompleteness. In many FL applications, clients cannot access all the ground-truth labels. For instance, a medical center wishes to use FL to significantly improve the diagnosis quality without transmitting sensitive data from clinics distributed in rural areas. In contrast, clinics do not have advanced medical resources to label all the data. In general, suppose that client m holds n_m^{la} labeled data $(X_m^{\text{la}}, Y_m^{\text{la}})$ and n_m^{un} unlabeled data X_m^{un}, $m = 1, \ldots, M$. Admittedly, one could use the labeled data only for FL training. However, when n_m^{un} is larger in orders of magnitudes than n_m^{la}, using the unlabeled data wisely may significantly boost the FL performance. This is a barely studied but worthwhile direction of FL. In the centralized setting, several recent works on semi-supervised learning have already shown the great potential of leveraging unlabeled data. Among the partially-labeled FL settings, a practical scenario of particular interest concerns the "fully-unlabeled clients," where $n_m^{\text{la}} = 0$ for all m, but the server may have some labeled data. Back to the example, a medical center could have a few labeled data due to its rich resources, while distributed clinics have only unlabeled data. For this challenge, recent work on semi-supervised FL [23] has shown promising baseline results that with properly designed consistency regularization techniques, one could achieve performance close to the centralized and fully-labeled training, but with unlabeled clients. Its extensions to address the challenges in Sections 3-5 are worth investigating. Also, since the existing work mostly focused on image data, there is much to be explored in audio and text domains.

Data polarity. We say a set of data (X_i, Y_i), $i = 1, \ldots, n$, exhibits *polarity* if $Y_i \in \mathcal{Y}^{\text{obs}}$ for some $\mathcal{Y}^{\text{obs}} \not\subseteq \mathcal{Y}$ and all i. For example, a machine based on natural language only stores data that contains a detected keyword for subsequent recognition, so only positive data are collected but negative ones are dropped. In such cases, the collected (X, Y) does not represent the whole data distribution, potentially causing severe biases in prediction. This challenge is further complicated by issues mentioned in Section 5, which limits the use of historical data (even if they are balanced).

Complex data dependency. While it is convenient to study IID $X \in \mathbb{R}^d$, in many FL applications, data are collected from time series with inevitable dependency. For example, suppose that $\{Z_t\}_{t=1,2,\ldots}$ is a data sequence, and client m observes $\{Z_{m,t}\}_{t=1,2,\ldots}$, where $Z_t \in \mathbb{R}^k$ and $Z_{m,t}$ is a sub-vector of Z_t. For the task of one-step ahead prediction of Z_t, client m's labeled data are $(X_{m,t}, Y_{m,t})$, where $X_{m,t} = [Z_{m,t-1}, \ldots, Z_{m,t-\ell}]$ and $Y_{m,t} = X_{m,t}$ for a properly chosen lag order ℓ [24]. Though one could still apply (1), it may not be justified since $X_{m,t}, t = 1, 2, \ldots$ are non-IID or even non-stationary, and the label $Y_{m,t}$ is only a sub-vector of the intended Y_t. This is further complicated if one considers a spatio-temporal structure where Z_t represent a time-varying graph, e.g., transportation data, weather data, and internet traffic data observed from different locations.

7. CONCLUDING REMARKS

Federated learning is an exciting and growing research area. Significant progress has been made in this area over the past couple of years. However, as we discussed throughout the paper, many challenging and interesting research problems are still left to be studied. In particular, from algorithmic aspects, the issues of communication rounds, personalization, lack of labels, robustness, and continuation in federated learning are not explored much. Furthermore, from system design perspectives, the challenges of resource constraints at the edge nodes, security, and privacy are widely open areas. In parallel to algorithmic and system-design challenges, FL research is also advancing rapidly in application domains beyond computer vision [2] and natural language processing [25]. Examples are graph-structured data [26] and time-series data that arise in drug discovery, social network, recommendation system, and advertisement domains.

8. REFERENCES

[1] Reza Shokri and Vitaly Shmatikov, "Privacy-preserving deep learning," in *Proc. CCS*. ACM, 2015, pp. 1310–1321.

[2] Jakub Konevcny, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[3] Jie Ding, Vahid Tarokh, and Yuhong Yang, "Model selection techniques: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 6, pp. 16–34, 2018.

[4] Jie Ding, Enmao Diao, Jiawei Zhou, and Vahid Tarokh, "On statistical efficiency in learning," *IEEE Trans. Inf. Theory*, vol. 67, no. 4, pp. 2488–2506, 2020.

[5] Jie Ding, Jiawei Zhou, and Vahid Tarokh, "Asymptotically optimal prediction for time-varying data generating processes," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 3034–3067, 2019.

[6] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*. PMLR, 2017, pp. 1273–1282.

[7] Xun Xian, Xinran Wang, Jie Ding, and Reza Ghanadan, "Assisted learning: a framework for multi-organization learning," *NeurIPS 2020*, 2020.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.

[9] Theodoros Evgeniou and Massimiliano Pontil, "Regularized multi–task learning," in *Proc. KDD*, 2004, pp. 109–117.

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[11] Jiawei Zhang, Jie Ding, and Yuhong Yang, "Is a classification procedure good enough? a goodness-of-fit assessment tool for classification learning," *J. Am. Stat. Assoc.*, 2021.

[12] Chaoyang He, Murali Annavaram, and Salman Avestimehr, "Group knowledge transfer: Federated learning of large cnns at the edge," 2020.

[13] Enmao Diao, Jie Ding, and Vahid Tarokh, "HeteroFL: Computation and communication efficient federated learning for heterogeneous clients," *arxiv preprint arxiv:2010.01264*, 2020.

[14] Chaoyang He, Murali Annavaram, and Salman Avestimehr, "Fednas: Federated deep learning via neural architecture search," *arXiv e-prints*, pp. arXiv–2004, 2020.

[15] Jianyu Wang et al, "A field guide to federated optimization," *CoRR*, vol. abs/2107.06917, 2021.

[16] Tribhuvanesh Orekondy, Seong Joon Oh, Yang Zhang, Bernt Schiele, and Mario Fritz, "Gradient-leaks: Understanding and controlling deanonymization in federated learning," *arXiv preprint arXiv:1805.05838*, 2018.

[17] Xinran Wang, Yu Xiang, Jun Gao, and Jie Ding, "Information laundering for model privacy," *Proc. ICLR*, 2021.

[18] Jinhyun So, Başak Güler, and A Salman Avestimehr, "Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 479–489, 2021.

[19] Jinhyun So, Ramy E. Ali, Basak Guler, Jiantao Jiao, and Salman Avestimehr, "Securing secure aggregation: Mitigating multi-round privacy leakage in federated learning," *CoRR*, vol. abs/2106.03328, 2021.

[20] Constance Beguier and Eric W Tramel, "Safer: Sparse secure aggregation for federated learning," *arXiv preprint arXiv:2007.14861*, 2020.

[21] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*. PMLR, 2015, pp. 448–456.

[22] Mathieu Andreux, Jean Ogier du Terrail, Constance Beguier, and Eric W Tramel, "Siloed federated learning for multi-centric histopathology datasets," in *Proc. DART*, pp. 129–139. Springer, 2020.

[23] Enmao Diao, Jie Ding, and Vahid Tarokh, "SemiFL: Communication efficient semi-supervised federated learning with unlabeled clients," *arXiv preprint arXiv:2106.01432*, 2021.

[24] Qiuyi Han, Jie Ding, Edoardo M Airoldi, and Vahid Tarokh, "SLANTS: Sequential adaptive nonlinear modeling of time series," *IEEE Trans. Signal Process.*, vol. 65, no. 19, pp. 4994–5005.

[25] Bill Yuchen Lin, Chaoyang He, Zihang Zeng, Hulin Wang, Yufen Huang, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr, "Fednlp: A research platform for federated learning in natural language processing," *CoRR*, vol. abs/2104.08815, 2021.

[26] Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Yu Rong, Peilin Zhao, Junzhou Huang, Murali Annavaram, and Salman Avestimehr, "Fedgraphnn: A federated learning system and benchmark for graph neural networks," *CoRR*, vol. abs/2104.07145, 2021.