Federated Learning in the Lens of Crowdsourcing

Yongxin Tong, Yansheng Wang, Dingyuan Shi
SKLSDE Lab, BDBC and IRI, Beihang University, China
{yxtong, arthur_wang, chnsdy}@buaa.edu.cn

Abstract

The success of artificial intelligence (AI) is inseparable from large-scale and high-quality data, which is not always available. Involving human forces like crowdsourcing can help provide more training data and improve data quality for AI tasks. But with more privacy concerns and stricter laws, the data isolation problem is becoming worse, just when federated learning (FL) has emerged as a promising solution. In this article, we highlight the core issues in federated learning in the lens of crowdsourcing, including privacy and security, incentive mechanism, communication optimization and quality control. We expect to inspire the design of federated learning systems with existing crowdsourcing techniques. We also discuss emerging future challenges to implement a fully fledged federated learning platform.

1 Introduction

Artificial intelligence (AI) has come to a golden age. With the help of big data, new learning algorithms and powerful computing hardware, AI has shown huge potential in many real-life applications, such as image recognition and text processing. However, its success highly relies on large-scale and high-quality training data, which is not always available.

Involving human forces proves effective in either providing more training data or improving the data quality for AI tasks. In particular, crowdsourcing [1, 2], is one of the most practical solutions to data problems in AI. It is a computation paradigm where humans are gathered to collaboratively accomplish easy tasks. A representative example of crowdsourcing empowered AI is the famous ImageNet project [3], where most pictures are labeled by crowdsourced workers. The Amazon Mechanical Turk (AMT) is one of the most successful commercial crowdsourcing platforms, where a large number of data labeling tasks with monetary rewards are provided by AI practitioners for freelancers.

The lack of large-scale training data is becoming more severe in recent years. In many industries, data are often isolated by different companies or organizations. Because of commercial competition and administrative issues, they would not like to share their data. They have to train models separately with their own data but the performance is often unsatisfactory due to the lack of data. Meanwhile, with people's increasing awareness on data security and individual privacy, data privacy in AI is becoming increasingly important. Many countries are enacting strict laws to protect the data privacy of their citizens. For example, EU's General Data Protection Regulation (GDPR) which was enforced on May 25, 2018, has stipulated that any use of personal data in a company must be authorized by the data owners. Therefore, privacy issues exacerbate the data isolation problem.

Copyright 0000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

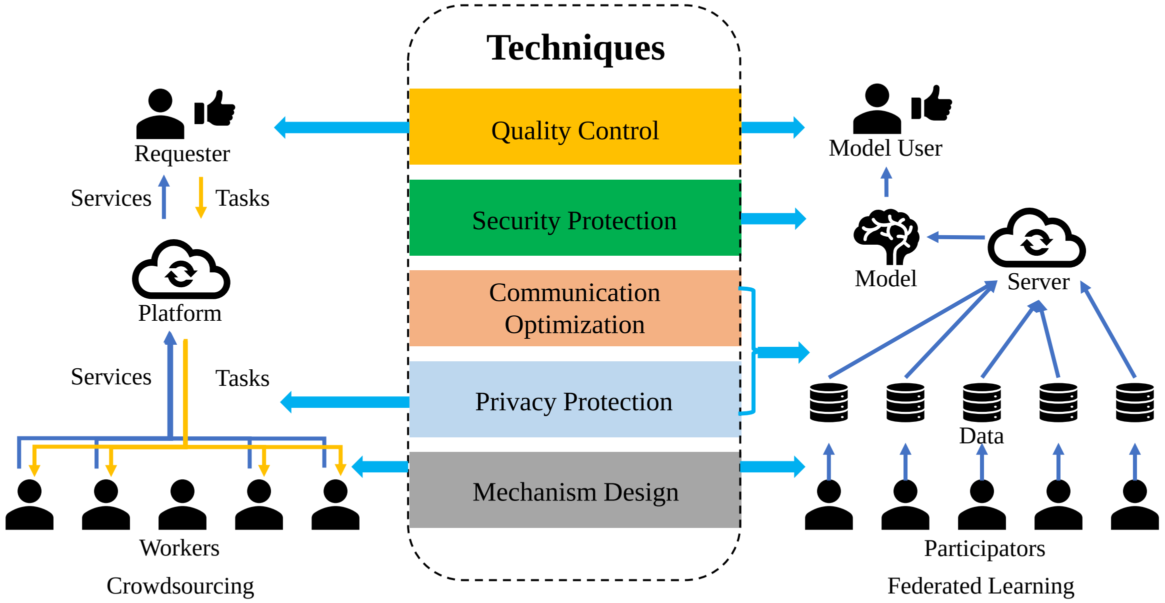

Figure 1: Comparison between crowdsourcing and federated learning.

Federated learning (FL) [4, 5, 6] has emerged as a promising solution to the data isolation problem in AI. First proposed by Google, FL aims to collaboratively build machine learning models with data from massive mobile devices without violating data privacy. As with crowdsourcing, FL also organizes humans, their devices and data, to collaborate on specific tasks. Therefore, FL can be considered as a special form of crowdsourcing. In this paper, we propose to understand federated learning from the lens of crowdsourcing. The characteristics of FL in the core issues of generic crowdsourcing and the unique issues of FL are summarized as below.

- **Privacy and security.** The most important issue in FL is privacy protection, whereas privacy is of less concerns in general crowdsourcing. Secure machine learning algorithms often play a central role in FL.

- **Incentive mechanism.** Both FL and general crowdsourcing need to motivate participants. However, there are two differences. The incentive mechanism in many crowdsourcing applications is based on the Business-to-customer (B2C) mode while in FL it can also be the Business-to-business (B2B) mode, as the participators in FL can be different companies. The incentive procedure in crowdsourcing is often a single round, while in FL it takes multiple rounds following the training steps in machine learning, which makes the design of incentive mechanisms more difficult.

- **Communication optimization.** In crowdsourcing the communication overhead is usually not a problem as each participant only has to submit small data (like the label of a picture) for a single round. However, in FL it can be a core issue because the training process often involves frequent communication of high dimensional gradients.

- **Quality control.** Optimizing the collaboration results is crucial both in crowdsourcing and FL. The difference is that in crowdsourcing the task is simple and the focus is to improve the accuracy of integrated results with a constrained budget. The task in FL is more complicated, and the focus lies in how to deal

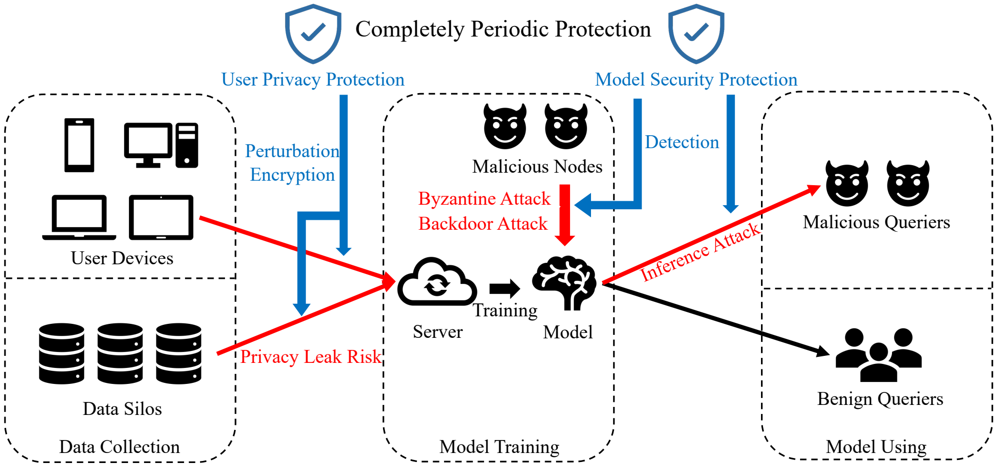

Figure 2: Completely periodic protection in federated learning.

with the heterogeneity of different parties such as non independently and identically distributed data and imbalanced computation resources.

- **Task assignment.** Task assignment is a core component in a crowdsourcing platform [7, 8], which needs to manage massive tasks and workers and to effectively allocate the resources. The assignment results can decide the practical performance of the platform. In FL, task assignment may not be an issue, as there is only one learning task in most cases.

This paper discusses the above core issues in federated learning from the perspective of crowdsourcing. The aim is to inspire the design of federated learning systems with existing techniques in crowdsourcing platforms. We also pinpoint future challenges to implement a fully fledged federated learning platform following the principles of crowdsourcing.

2 Privacy and Security

In general crowdsourcing, workers and users only provide some necessary information such as worker skills or positions which only leaks little privacy. Existing privacy protection techniques like anonymization and encryption are sufficient to protect such information. However, in federated learning, the protection object becomes massive user data which is more sensitive and easier to leak privacy when external knowledge is used. Furthermore, compared with crowdsourcing, federated learning makes it harder to judge the benignity of user uploads. This is because machine learning models are black boxes and it is non-trivial to explain the contribution of user uploads. Accordingly, it is possible for malicious users to upload information and thwart model training.

A federated learning system is expected to offer periodic protection on both user data privacy and model security, as shown in Figure 2. Specifically, a safe-to-use federated learning framework should *(i)* collect and use user data privately and *(ii)* ensure that the model converges without poisoning and will not be stolen.

2.1 Data Privacy Protection

In federated learning, for each participator, the server and other participators cannot be easily trusted. Direct uploading of raw data can lead to privacy leaks. Uploading model parameters instead of raw data seems safer, which is also allowed in the most recognized algorithm FedAvg [5]. Some recent findings, however, show that only with model parameters, a malicious attacker can still deploy inference attack to judge membership or even reconstruct the raw data [9]. Therefore, we need to design privacy protection protocols for federated learning, which mainly include two techniques: perturbation and encryption.

Perturbation. Perturbation techniques require participators to inject noise to their raw data or intermediate results, making others hard to infer what they have uploaded. To quantitatively analyze the degree of noise injection, a widely accepted measure is differential privacy (DP), proposed by Dwork [10]. Its idea is to quantify the degree of indistinguishability by probability distributions. The DP measurement is firstly brought up and applied in database. In the field of federated learning, however, a big challenge derives from the long iterative process and from massively distributed data. For each participator and in each iteration, DP should always be satisfied. Ensuring such strict privacy protection requires strong noise to be injected, which severely deteriorates data accuracy. To prevent privacy cost from boosting wildly with iteration rounds, moments accountant [11] has been proposed. It can make the privacy budget increase sub-linearly (square root) to iteration rounds. As for large node numbers, shuffle model [12] techniques effectively cut down privacy budgets by ensuring anonymity. Besides supervised learning, DP has also been applied to unsupervised algorithms like topic modeling [13, 14].

Encryption. Perturbation technique can be considered as a balance between privacy levels and data accuracy. Nevertheless, perfect secrecy cannot be achieved as exposure of little private information always exists. Encryption techniques, on the other hand, aim to directly circumvent such exposure, *i.e.*, to calculate via ciphertexts. Several existing encryption techniques can be applied to federated learning, such as secure multiparty computation (SMC), homomorphic encryption (HE) and garbled circuits (GC). Bonawitz et al use pseudo random vectors as a mask to cover the raw updates and those random vectors can neutralize each other after aggregation [15]. Using HE to protect user privacy demands a protocol via which the server can aggregate the ciphertexts from each device and finally decipher the result [16]. Besides software level solutions, hardware solutions are also worth considering such as the trusted execution environment (TEE).

2.2 Model Security Protection

Model security refers to that the model converges to a global optimum and the results can be used safely. Model security can be violated by multiple types of attacks.

Byzantine Attack. Byzantine attack is a classical attack in distributed networks. Attackers aim to disturb the model training process and make the model unable to converge. In FL, the malicious nodes (Byzantine nodes) will upload random vectors to mislead the aggregated gradient descent direction and thus obstruct the model convergence. The core idea to detect Byzantine attack is to evaluate and to spot the outliers among user uploads. Intuitively, examining the angle between different vectors' directions may be a solution [17]. Another intuitive way is to use the distance to median values as an outlier judgement and its effectiveness is also verified in [18]. By maintaining a non-Byzantine node set while training, the later solution reaches lower time complexity.

Backdoor Attack. By deploying Byzantine attack detection techniques, we can ensure the convergence of model training. However, some higher-level attackers (nodes) can cheat the Byzantine detector by uploading plausible updates and force the model to converge to a point where some subtasks or intentionally designed misclassification are achieved. For example, the attacker may hope to make a spelling prompt model always provide some specific words (A restaurant owner hopes her restaurant name be prompted after a user types "My favorite restaurant is..."). To realize this, unlike Byzantine attackers who simply upload random vectors, they will use data poisoning or model poisoning techniques. The purpose of data poisoning is to train models with

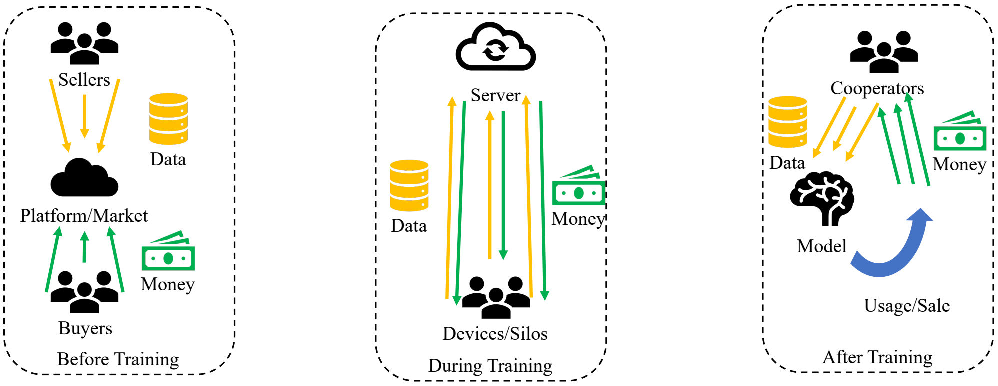

Figure 3: Mechanism design over all periods.

intentionally mislabeled or polluted data while model poisoning refers to that malicious nodes train and upload local models with training goals different from the global model. To tackle data or model poisoning, we can evaluate the approximated loss upper bound and remove outliers before local model aggregation [19].

Model Stealing. Equipped with Byzantine and backdoor attack detectors, a federated learning system is able to safely train a model. However, we also need to provide protection for model use. Model stealing happens when the model is confidential and can only be used via APIs (*i.e.*, Machine-Learning-as-a-Service, MLaaS). In that case, attackers may use the APIs to infer the model structure and parameters, and the confidential information of the model is exposed. In federated learning, this may happen when one of the participators wants to forcibly occupy all the outcomes. To prevent this, we can deploy a detector to examine the frequency of API queries and judge whether the query sequence is benign [20].

3 Incentive Mechanism

Both general crowdsourcing and FL involve multiple human participators. Thus suitable incentive mechanisms are necessary to attract people to actively contribute to the tasks. In traditional crowdsourcing, some tasks may be less attractive to workers due to distance, difficulty or other reasons. Therefore, the platform needs to motivate the workers with additional rewards [21]. For FL, incentive mechanism design is more difficult. This is because the black-box nature of many machine learning models makes it tricky to evaluate each participator's contribution. Furthermore, the number of participators in FL, especially in cross-device settings, can be much larger than in traditional crowdsourcing.

The mechanisms for FL should be incentive-compatible and fairness-aware. They can be accomplished before, during and after training process, as shown in Figure 3.

Mechanisms before Training. Designing mechanisms to incite participators before training means to establish suitable rules for data trading. In the past decade several data markets and data sharing platforms such as

Dawex[1], Xignite[2] and WorldQuant[3] have been developed. They all hope to build a data trading platform where companies and users can buy, sell and exchange data with satisfactory prices to all parties. However, this goal is hard to reach. As Fernandez et al point out, different combinations of data may produce different levels of values[22], so instead of setting static price for data, people seek to find dynamic evaluation methods.

Mechanisms during Training. The goal of designing mechanisms during training is to incite the participators to use their best data for training. To achieve the goal, participators who contribute the most deserve the highest reward. Many researchers model the training process as a Stackelberg game[23]. There are two stages in this setting. In the first stage, the server receives updates from each user and distributes rewards base on their contributions to the model. In the second stage, users change their update strategies based on the rewards they receive. Then the cycle repeats until all the server and users converge to equilibrium. Chen et al study mechanism design of federated learning from a game theoretical and optimization perspective. Recent years blockchain emerges as a novel technique and raises wide-spread research enthusiasm. It is also an alternative technique to design incentive mechanisms. One way to combine blockchain with mechanism design is to build a reputation record by the blockchain[24]. Owing to the immutability and consensus of blockchain, users' reputation would be hard to recover once it gets damaged and thus the participators will behave honestly to maintain their reputations.

Mechanisms after Training. In cross-silo federated learning settings the companies cooperate with each other to train a global model and the rewards (profits) mainly come from the model use. Distributing the profits fairly requires designing mechanisms that can evaluate each entity's contribution to the final model. Shapley value is a classical concept in game theory to evaluate contributions and can also be applied in profit sharing in federated learning. The drawback is that the calculation of Shapley values is time-consuming. As a result, Song et al propose a novel accleration technique to make it practical[25].

4 Communication Optimization

Crowdsourcing platforms usually do not care the communication overhead. Tasks like labeling the images require the workers to upload very few data for a single round. The workers can submit their results separately and occasionally. Hence there is not much pressure on the server's communication bandwidth. However, the communication cost becomes a primary bottleneck for FL. Model training algorithms like the stochastic gradient descent (SGD) take a large number of rounds to converge. Besides, unlike the powerful servers in distributed learning, the nodes in FL are massively distributed mobile devices with limited communication bandwidth and active time, which makes the problem even more challenging.

Many methods have been proposed to improve the communication efficiency of FL. Two basic ideas are to either reduce the number of interactions between parties and the server or to compress the data in transmission. These two ideas can also be combined. Some representative techniques are explained as below.

4.1 Interactive Number Reduction

There are three different ways to reduce the number of interactions during the federated learning process: client sampling, local updating, and decentralized training.

Client Sampling. In a cross-device federated learning scenario, there are usually millions of devices in participation, which makes round-robin strategies impossible to work. The server has to sample some parties in each round for faster convergence and the accuracy should be compromised. The client sampling trick usually does not work alone. In some earliest works of FL such as Federated Optimization [26] and FedAvg [5], random

[1] https://www.dawex.com

[2] https://www.xignite.com

[3] https://www.worldquant.com

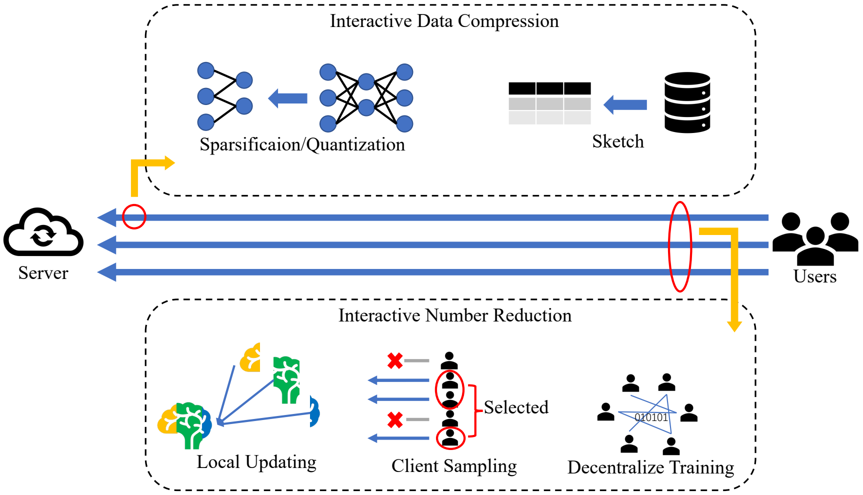
Interactive Data Compression
Sparsificaion/Quantization
Sketch
Server
Users
Interactive Number Reduction
Local Updating
Selected
Client Sampling
010101
Decentralize Training

Figure 4: Communication optimization techniques in federated learning.

sampling over multiple parties is used during the training process. Afterwards, it is applied by other works on cross-device FL by default. Most of them still use the simplest uniformly random sampling method. Some consider choosing the clients conditionally. For example, FedCS [27] actively manages the clients based on their resource conditions. Reinforcement learning techniques have also been applied to adaptively sample more suitable clients [28].

Local Updating. The local updating techniques are first proposed to improve the communication efficiency in distributed learning and they can also work well in FL. Some approaches use primal-dual methods to decompose a global objective into many subproblems that can be solved in parallel. Therefore the communication rounds can be effectively reduced. However, in FL some trivial ideas have shown good empirical performance. The most commonly used algorithm FedAvg [5] is based on local SGD and global averaging. An extension [29] uses more adaptive local updating methods like ADAGRAD, ADAM and YOGI.

Decentralized Training. Decentralized learning algorithms can effectively reduce the communication overhead on the server side by apportioning them to each client. Many existing works on decentralized learning can also be applied in FL, such as decentralized training algorithms of linear classifiers [30] and deep neural networks [31]. Some other works are based on special network structures. In [32], it considers the problem that the distributed nodes can only communicate to their neighbors on a fixed communication graph in machine learning and devises a gossip algorithm for average consensus with convergence guarantees. In [33], a dynamic peer-to-peer network environment is considered and a novel decentralized FL framework BrainTorrent is proposed.

4.2 Interactive Data Compression

Another way to reduce communication cost is to compress the data in transmission directly. Many existing FL works follow the model compression approaches in machine learning, such as sparsificaion and quantization. Some others apply data structures like sketch to re-encode the gradients.

Compression with Sparsificaion and Quantization. One of the pioneering works in FL [4] focuses on improving the communication efficiency by random mask structures as well as a combination of quantization, random rotations, and subsampling to compress the model parameters. The distributed mean estimation problem is studied in [34] and an communication-efficient compressing algorithm using constant number of bits is devised. In [35], it proposes Deep Gradient Compression (DGC) in cross-device distributed learning setting which can greatly reduce the communication bandwidth by 99%. Two different strategies named extrapolation compression and difference compression are proposed in [36], combined with a decentralized training algorithm. A special quantization-based technique for gradient compression in FL is proposed in [37], combined with a periodic averaging learning algorithm.

Compression with Sketch. A probabilistic data structure for compressing big streaming data, the sketch [38], is also used in gradient compression. To find heavy hitters (most frequent items) in the federated setting, a sketch-based algorithm with local differential privacy is proposed in [39]. To compress sparse and nonuniform gradient in distributed machine learning, MinMaxSketch is designed in [40], which uses a quantile sketch to sort gradients into buckets. Another similar approach, the Sketched-SGD, is proposed in [41]. It is demonstrated to have a 40x reduction in total communication cost with no loss in final model performance. In [42], the authors prove that Count sketch without additional noise can satisfy the notion of differential privacy (DP) under specific assumptions, which is also known as “privacy for free”.

5 Quality Control

The heterogeneity of workers in crowdsourcing results in the quality variation of aggregated results. The purpose of quality control is to quantify the heterogeneous quality of workers and tasks and effectively aggregate results to ensure high-quality task completion. Similarly, there is also the heterogeneity problem in FL participants, especially in cross-device scenarios. Thus quality control is also important for FL. Different from quality control techniques in crowdsourcing, which concentrate on evaluating the characteristics or skills of different workers to improve crowdsourced results, quality control in FL mainly deals with two unique challenges, the heterogeneity in data and the heterogeneity in resource.

5.1 Data Heterogeneity

The data heterogeneity mainly refers to the non-IID data problem in FL. In [43], a taxonomy of non-iid data regimes is provided, including feature distribution skew, label distribution skew, etc. The most commonly adopted FedAvg [5] algorithm only considers the IID case at first and no theoretical analysis on its convergence is made. To improve the performance of FedAvg especially in non-IID settings, some approaches have been proposed [44, 45, 46]. A more rigorous theoretical proof on the convergence of FedAvg with non-IID data is provided in [47]. It establishes a convergence rate by the inverse of the number of iterations for strongly convex and smooth problems. However, in [48], it demonstrates that the accuracy of FedAVG can be largely damaged on highly skewed non-IID data where each device only has a single class of data. The solution is to create a globally shared small subset of data. But if each device shares too much data, the privacy constraints might be broken, which is contradictory to the initial purpose of FL. To avoid data sharing, in [49], it proposes federated augmentation, where each participant will collectively train a generative model. Therefore, the local data can be augmented by the generative model and gets rid of the non-IIDness. The differences between each client’s data distribution can also be taken as black boxes. A reinforcement learning-based client selection approach is proposed in [28] to deal with the non-IID data.

5.2 Resource Heterogeneity

The computation resources of different devices are commonly heterogeneous in mobile edge computing. The main objective is to optimize the model quality by resource allocation. The problems of resource heterogeneity such as uncertainty of wireless channels and devices with heterogeneous power constraints have been emphasized in [50]. It formalizes FL with heterogeneous resources as an optimization problem to capture the trade-offs between efficiency and accuracy. Then it solves the non-convex problem by decomposing it into several convex sub-problems. Control algorithms have also been proposed to realize more effective resource allocation. An adaptive control algorithm for FL with distributed SGD is proposed in [51]. It studies the convergence bound of the learning problem with the resource consumption constraints. A protocol named FedCS is proposed in [27] which performs client selection in each round according to their different resource conditions. Devices with poor computation power or low communication bandwidth will be eliminated during training. A more comprehensive survey on FL in mobile edge networks can be found in [52].

6 Future Directions

In this section, we will envision some future directions in federated learning.

Task Assignment. Until now, no work has considered the task assignment problem in FL. In most cases there is only one learning task. Meanwhile a single learning task cannot be decomposed like in crowdsourcing [53]. However, on a federated learning platform where task requesters and data providers can join and leave freely, task assignment can still be meaningful. This makes the platform similar to a data market [22]. But it still has unique challenges of FL, such as data privacy concerns, and the resource budgets of the data providers.

Acceleration of Encryption Schemes. The encryption schemes for privacy protection in FL such as homomorphic encryption and secret sharing often brings extremely large computation cost. For example, training a simple MLP on the MNIST dataset with homomorphic encryption can take 10 times slower than the original algorithm without any privacy protection [9]. The time consumption will be intolerable with larger dataset and deeper models like CNN. Therefore, how to accelerate the encryption schemes in FL has become a crucial problem.

Personalization. Most of existing works focus on training a global model rather than a personalized one for each client. With the help of multi-task learning, the participants can have personalized results by learning separate but related models [44]. However, it can only work with a small number of participants. In a large-scale cross-device scenario, novel domain adaptation techniques need to be designed to realize the full personalization.

Fairness. Some existing works already consider the fairness problem in FL [54, 55]. However, the problem is still challenging especially in scenarios with millions of mobile devices. Moreover, differential privacy is commonly used as a privacy preserving technique in FL but it may result in unfairness in model training [56]. How to compromise the strict constraints while still preserving the fairness in FL remains an open question.

General FL Platform. Like a crowdsourcing platform, a general FL platform can bring more opportunities to both data providers and task requesters. However, designing such a platform is challenging. The aforementioned key components should all be considered. The platform should support different types of learning algorithms and privacy preserving schemes in different scenarios. Also, it should provide incentive and pricing mechanisms for the participators. The communication cost and the model quality should also be optimized. Existing FL systems and benchmarks [57] only implement part of these features and cannot work as a general FL platform.

7 Conclusion

Federated Learning has gained much attention in recent years as a promising solution to the data isolation problem in artificial intelligence. Both as human-empowered AI techniques, federated learning and crowdsourcing have many similarities. In this article, we discuss four core issues in federated learning from the perspective of crowdsourcing, namely privacy and security, mechanism design, communication optimization and quality control. We find that the design of federated learning systems can be inspired by existing techniques in crowdsourcing platforms. We also envision some future directions of federated learning, which would be helpful to build a fully fledged federated learning platform.

References

[1] A. I. Chittilappilly, L. Chen, and S. Amer-Yahia, "A survey of general-purpose crowdsourcing techniques," *IEEE TKDE*, vol. 28, no. 9, pp. 2246–2266, 2016.

[2] Y. Tong, Z. Zhou, Y. Zeng *et al.*, "Spatial crowdsourcing: a survey," *VLDB*, vol. 29, no. 1, pp. 217–250, 2020.

[3] J. Deng, W. Dong, R. Socher *et al.*, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[4] J. Konecný, H. B. McMahan, F. X. Yu *et al.*, "Federated learning: Strategies for improving communication efficiency," *CoRR*, vol. abs/1610.05492, 2016.

[5] B. McMahan, E. Moore, D. Ramage *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, vol. 54, 2017, pp. 1273–1282.

[6] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM TIST*, vol. 10, no. 2, pp. 12:1–12:19, 2019.

[7] Y. Tong, J. She, B. Ding *et al.*, "Online mobile micro-task allocation in spatial crowdsourcing," in *ICDE*, 2016, pp. 49–60.

[8] Y. Tong, Y. Zeng, B. Ding, L. Wang, and L. Chen, "Two-sided online micro-task assignment in spatial crowdsourcing," *IEEE TKDE*, 2019. [Online]. Available: `doi.org/10.1109/TKDE.2019.2948863`

[9] L. T. Phong, Y. Aono, T. Hayashi *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE TIFS*, vol. 13, no. 5, pp. 1333–1345, 2018.

[10] C. Dwork, "Differential privacy," in *ICALP*, 2006, pp. 1–12.

[11] M. Abadi, A. Chu, I. J. Goodfellow *et al.*, "Deep learning with differential privacy," in *CCS*, 2016, pp. 308–318.

[12] Ú. Erlingsson, V. Feldman, I. Mironov *et al.*, "Amplification by shuffling: From local to central differential privacy via anonymity," in *SODA*, 2019, pp. 2468–2479.

[13] D. Jiang, Y. Song, Y. Tong *et al.*, "Federated topic modeling," in *CIKM*, 2019, pp. 1071–1080.

[14] Y. Wang, Y. Tong, and D. Shi, "Federated latent dirichlet allocation: A local differential privacy based framework," in *AAAI*, 2020, pp. 6283–6290.

[15] K. Bonawitz, V. Ivanov, B. Kreuter *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *CCS*, 2017, pp. 1175–1191.

[16] Z. Erkin, T. Veugen, T. Toft *et al.*, "Generating private recommendations efficiently using homomorphic encryption and data packing," *IEEE TIFS*, vol. 7, no. 3, pp. 1053–1066, 2012.

[17] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui *et al.*, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *NIPS*, 2017, pp. 119–129.

[18] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," in *NIPS*, 2018, pp. 4618–4628.

[19] J. Steinhardt, P. W. Koh, and P. Liang, "Certified defenses for data poisoning attacks," in *NIPS*, 2017, pp. 3517–3529.

[20] M. Juuti, S. Szyller, S. Marchal *et al.*, "PRADA: protecting against DNN model stealing attacks," in *IEEE EuroSP*, 2019, pp. 512–527.

[21] Y. Tong, L. Wang, Z. Zhou *et al.*, "Dynamic pricing in spatial crowdsourcing: A matching-based approach," in *SIGMOD*, 2018, pp. 773–788.

[22] R. C. Fernandez, P. Subramaniam, and M. Franklin, "Data market platforms: Trading data assets to solve data problems," *PVLDB*, vol. 13, no. 11, pp. 1933–1947, 2020.

[23] S. R. Pandey, N. H. Tran, M. Bennis *et al.*, "A crowdsourcing framework for on-device federated learning," *IEEE TWC*, vol. 19, no. 5, pp. 3241–3256, 2020.

[24] J. Kang, Z. Xiong *et al.*, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE IoTJ*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[25] T. Song, Y. Tong, and S. Wei, “Profit allocation for federated learning,” in *BigData*, 2019, pp. 2577–2586.

[26] J. Konecný, B. McMahan, and D. Ramage, “Federated optimization: Distributed optimization beyond the datacenter,” *CoRR*, vol. abs/1511.03575, 2015.

[27] T. Nishio and R. Yonetani, “Client selection for federated learning with heterogeneous resources in mobile edge,” in *IEEE ICC*, 2019, pp. 1–7.

[28] H. Wang, Z. Kaplan, D. Niu *et al.*, “Optimizing federated learning on non-iid data with reinforcement learning,” in *IEEE INFOCOM*, 2020, pp. 1698–1707.

[29] S. J. Reddi, Z. Charles, M. Zaheer *et al.*, “Adaptive federated optimization,” *CoRR*, vol. abs/2003.00295, 2020.

[30] L. He, A. Bian, and M. Jaggi, “COLA: decentralized linear learning,” in *NIPS*, 2018, pp. 4541–4551.

[31] M. Kamp, L. Adilova, J. Sicking *et al.*, “Efficient decentralized deep learning by dynamic model averaging,” in *PKDD*, vol. 11051, 2018, pp. 393–409.

[32] A. Koloskova, S. U. Stich, and M. Jaggi, “Decentralized stochastic optimization and gossip algorithms with compressed communication,” in *ICML*, vol. 97, 2019, pp. 3478–3487.

[33] A. G. Roy, S. Siddiqui, S. Pölsterl *et al.*, “Braintorrent: A peer-to-peer environment for decentralized federated learning,” *CoRR*, vol. abs/1905.06731, 2019.

[34] A. T. Suresh, F. X. Yu, S. Kumar *et al.*, “Distributed mean estimation with limited communication,” in *ICML*, vol. 70, 2017, pp. 3329–3337.

[35] Y. Lin, S. Han, H. Mao *et al.*, “Deep gradient compression: Reducing the communication bandwidth for distributed training,” in *ICLR*, 2018.

[36] H. Tang, S. Gan, C. Zhang *et al.*, “Communication compression for decentralized training,” in *NIPS*, 2018, pp. 7663–7673.

[37] A. Reisizadeh, A. Mokhtari, H. Hassani *et al.*, “Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization,” in *AISTATS*, vol. 108, 2020, pp. 2021–2031.

[38] S. Muthukrishnan, “Data streams: algorithms and applications,” in *SODA*, 2003, pp. 413–413.

[39] W. Zhu, P. Kairouz, B. McMahan *et al.*, “Federated heavy hitters discovery with differential privacy,” in *AISTAS*, S. Chiappa and R. Calandra, Eds., vol. 108, 2020, pp. 3837–3847.

[40] J. Jiang, F. Fu, T. Yang *et al.*, “Sketchml: Accelerating distributed machine learning with data sketches,” in *SIGMOD*, 2018, pp. 1269–1284.

[41] N. Ivkin, D. Rothchild, E. Ullah *et al.*, “Communication-efficient distributed SGD with sketching,” in *NIPS*, 2019, pp. 13 144–13 154.

[42] T. Li, Z. Liu, V. Sekar *et al.*, “Privacy for free: Communication-efficient learning with differential privacy using sketches,” *CoRR*, vol. abs/1911.00972, 2019.

[43] P. Kairouz, H. B. McMahan, B. Avent *et al.*, “Advances and open problems in federated learning,” *CoRR*, vol. abs/1912.04977, 2019.

[44] V. Smith, C. Chiang, M. Sanjabi *et al.*, “Federated multi-task learning,” in *NIPS*, 2017, pp. 4424–4434.

[45] M. Yurochkin, M. Agarwal, S. Ghosh *et al.*, “Bayesian nonparametric federated learning of neural networks,” in *ICML*, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97, 2019, pp. 7252–7261.

[46] H. Wang, M. Yurochkin, Y. Sun *et al.*, “Federated learning with matched averaging,” in *ICLR*, 2020.

[47] X. Li, K. Huang, W. Yang *et al.*, “On the convergence of fedavg on non-iid data,” in *ICLR*, 2020.

[48] Y. Zhao, M. Li, L. Lai *et al.*, “Federated learning with non-iid data,” *CoRR*, vol. abs/1806.00582, 2018.

[49] E. Jeong, S. Oh, H. Kim *et al.*, “Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data,” *CoRR*, vol. abs/1811.11479, 2018.

[50] N. H. Tran, W. Bao, A. Y. Zomaya *et al.*, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM*, 2019, pp. 1387–1395.

[51] S. Wang, T. Tuor, T. Salonidis *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE JSAC*, vol. 37, no. 6, pp. 1205–1221, 2019.

[52] W. Y. B. Lim, N. C. Luong, D. T. Hoang *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE ComSur*, vol. 22, no. 3, pp. 2031–2063, 2020.

[53] Y. Tong, L. Chen, Z. Zhou *et al.*, "SLADE: A smart large-scale task decomposer in crowdsourcing," *IEEE TKDE*, vol. 30, no. 8, pp. 1588–1601, 2018.

[54] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *ICML*, vol. 97, 2019, pp. 4615–4625.

[55] T. Li, M. Sanjabi, A. Beirami *et al.*, "Fair resource allocation in federated learning," in *ICLR*, 2020.

[56] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," in *NIPS*, 2019, pp. 15 453–15 462.

[57] K. Bonawitz, H. Eichner, W. Grieskamp *et al.*, "Towards federated learning at scale: System design," in *MLSys*, 2019.