# Wireless Communications for Collaborative Federated Learning

Mingzhe Chen, H. Vincent Poor, Walid Saad, and Shuguang Cui

***Abstract*—To facilitate the deployment of machine learning in resource and privacy-constrained systems such as the Internet of Things (IoT), federated learning (FL) has been proposed as a means for enabling edge devices to train a shared learning model while preserving privacy. However, Google's seminal FL algorithm requires all devices to be directly connected with a central controller, which significantly limits its applications. In contrast, this paper introduces a novel FL framework, called *collaborative FL* (CFL), which enables edge devices to implement FL with less reliance on a central controller. The fundamentals of this framework are developed and a number of communication techniques are proposed so as to improve the CFL performance. Besides, an overview of centralized learning, Google's FL, and CFL is presented. For each type of learning, the basic architecture as well as its advantages, drawbacks, and operation conditions are introduced. Then, four CFL performance metrics are presented and a suite of communication techniques ranging from network formation, device scheduling, mobility management, to coding are introduced to optimize the performance of CFL. For each technique, future research opportunities are discussed. In a nutshell, this article will showcase how CFL can be effectively implemented at the edge of large-scale wireless systems.**

## I. Introduction

Machine learning (ML) finds a wide range of applications in wireless networks ranging from data analytics to network monitoring and optimization [1]. However, centralized ML requires edge devices to transmit their data to a central controller for learning. In practical deployments of ML in wireless systems, such as the Internet of Things (IoT), due to privacy issues and stringent resource (e.g., bandwidth and power) constraints, edge IoT devices may not be able or willing to share their collected data with other devices or a central controller. To enable edge devices in a wireless network in training a shared ML model without data exchanges, federated learning (FL) was proposed by Google [2].

FL is a distributed ML scheme that allows IoT devices to collaboratively perform on-device training of a shared ML task while only exchanging model parameters with a central controller. Keeping the data at IoT devices not only preserves privacy but may also reduce network congestion. Due to the unique features of FL, a number of existing works (e.g., see [3]–[6]), studied its use for wireless network optimization.

In practice, to implement FL over IoT networks, edge devices must repeatedly transmit their trained ML models to a central controller via wireless links. Due to limited wireless resources in an IoT, only a subset of devices can use FL. Meanwhile, ML models that are transmitted from IoT devices to a central controller (e.g., a base station) are subject to errors and delays caused by the wireless channel, which affects the learning performance. Therefore, it is necessary to consider the optimization of wireless networks to improve FL performance, as pointed out in [7]–[9]. This emerging "*communications for FL*" research area is the key focus of this work.

Recently, several surveys and tutorials related to FL over wireless networks appeared in [4]–[6] and [10]. First, the works in [4]–[6] investigated the use of FL for communications, rather than the impact of wireless networking on FL. Moreover, all prior works in [4]–[6] and [10] focused on the original FL from Google in [2] (called *original FL* hereinafter), which requires all IoT devices to transmit their ML models to a central controller. Hence, these existing surveys did not consider the implementation of FL with less or even no reliance on the central controller. Furthermore, they did not analyze how to use wireless techniques to optimize FL performance.

The main contribution of this article is a novel FL framework, dubbed *collaborative FL*, that combines collaborative learning [11] with FL to enable edge devices to perform FL without a central controller. We first provide a detailed overview of centralized learning (CL), original FL (OFL), and collaborative FL (CFL), and summarize their advantages, drawbacks, and operation conditions in Section II. Then, in Section III, we introduce four important performance metrics to quantify the CFL performance over IoT systems. Further, in Section IV, we introduce several important communication techniques to optimize the CFL performance metrics. For each communication technique, we first introduce the motivation for optimizing CFL performance and then present future research opportunities. Conclusions are drawn in Section V.

## II. Preliminaries and Overview

*1) Centralized Learning:* As shown in Fig. 1, CL needs only one ML model located a central controller. All devices must be connected and send their data to the BS for training this ML model. Then, the BS will transmit the trained ML model to all devices. Hence, CL only requires the base station (BS) to communicate with all devices once so as to collect all devices' datasets.

Table I summarizes the advantages, disadvantages, and operation conditions of CL. The key advantage of CL is that it enables the BS or cloud to directly find a global ML model that minimizes the learning loss function value. Since the entire training process is completed by the BS, the ML training will not be affected by wireless network performance. However, imperfect transmissions may introduce errors to

M. Chen is with the Chinese University of Hong Kong, Shenzhen and Princeton University.

H. V. Poor is with Princeton University.

W. Saad is with Virginia Tech.

S. Cui is with the Shenzhen Research Institute of Big Data and the Chinese University of Hong Kong, Shenzhen.

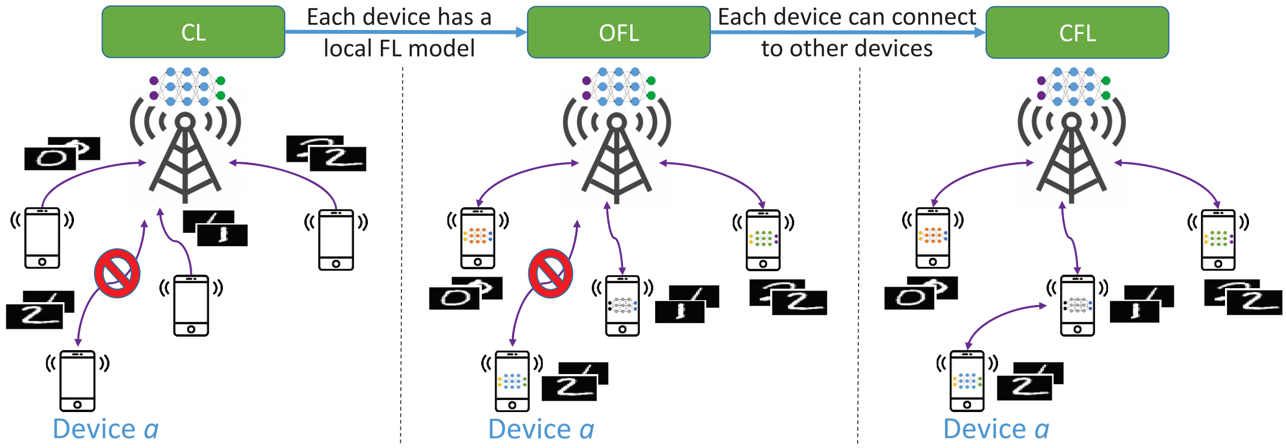


Fig. 1. Architectures of CL, OFL, and CFL. Here, to implement CFL, the BS is not necessary.

TABLE I
SUMMARY OF THE ADVANTAGES, DRAWBACKS, AND OPERATION CONDITIONS OF ML OVER WIRELESS NETWORKS.

| | Advantages | Drawbacks | Operation Conditions |
|---|---|---|---|
| **CL** | • Ability to find a globally optimal ML model.<br>• Ample computational resources and energy available for ML training.<br>• Imperfect wireless transmission has a minor impact on ML model training.<br>• Better performance for ML models with non-convex functions compared to FL. | • Private data must be shared with a centralized controller such as a BS or cloud.<br>• Significant overhead for data collection.<br>• Difficult to implement for resource and energy-limited edge devices such as IoT devices.<br>• Large delays due to long-range transmission to a remote cloud or BS. | • Each device must be willing to share its private data.<br>• All devices can transmit data to the BS. |
| **OFL** | • Privacy-preserving framework[1].<br>• Devices can learn a common ML task in a distributed manner.<br>• Ability to train ML models at device level. | • Imperfect wireless transmission affects the ML model training process.<br>• Number of users (and their data) that can perform FL is limited.<br>• All devices must have a direct and reliable wireless connection to the BS. | • All devices must be able to transmit FL model parameters to a controller or aggregator (e.g., a BS).<br>• All devices must be able to receive the FL model parameters from the BS.<br>• Devices can locally train ML models (at the edge). |
| **CFL** | • Privacy-preserving framework.<br>• Ability to include more training data samples for training compared to OFL.<br>• Amenability for implementation in large-scale systems (e.g., IoT) because CFL can accommodate more devices in the FL process compared to OFL. | • Imperfect wireless transmission affects the ML model training process.<br>• Lower convergence speed compared to OFL.<br>• The ML model of each device at convergence may be different since each device is connected to a subset of devices. | • A reliable communication link (direct or multiple hop) can be formed between any two devices that engage in CFL.<br>• Each device can locally train its ML model and aggregate the local FL models received from its associated devices. |

[1]CFL and OFL are not strictly privacy preserving since they require each device to share its trained FL model with other devices or the central controller and this model is related to the original data. However, they do guard privacy better than alternatives (such as CL) that require devices to exchange raw data. Therefore, we use the term "privacy-preserving" to capture this fact.

the data used for training. Moreover, CL requires devices to transmit their private data with the BS. Also, significant overhead and resources are needed at the network and device levels to execute CL.

*2) Original Federated Learning:* To maintain privacy, Google's OFL framework allows each edge device to cooperatively train a shared ML model without data transmission. In OFL, both devices and the BS own an ML model with the same architecture, as in Fig. 1. OFL is trained by an iterative learning process. First, all devices use their local data to train their local ML models and transmit their trained models to the BS. Then, the BS aggregates the received ML models, generated a new aggregate ML model, and transmits it back to all devices. Hereinafter, the ML model trained by an edge device is called *local FL model* while the ML model generated by the BS is called *global FL model*. At convergence, the global FL model will be equal to all local models, which means that devices find a shared FL model and the locoal FL model at convergence can be used to analyze all devices' datasets.

The advantages, disadvantages, and operation conditions of OFL are summarized in Table I. The key advantage of OFL is that it preserves data privacy and can be implemented over devices with less overhead than CL. However, OFL still requires all devices to transmit their local FL model parameters to a BS. Hence, imperfect and dynamic wireless transmission will significantly impact the OFL convergence time and performance.

*3) Collaborative Federated Learning:* OFL requires all devices to send their local models to a BS, however, in an IoT, devices may not be able to be connected to the BS due to energy limitations or to a potentially high transmission delay. To overcome this challenge, we propose the concept of CFL using which devices can engage in FL without being connected to a BS or a cloud.

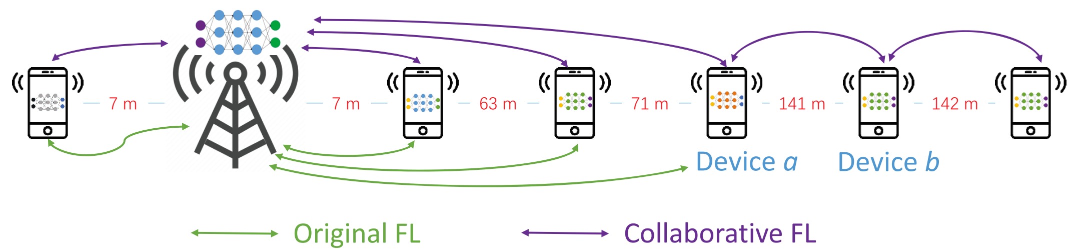


Fig. 2. Simulation system for the implementation of CFL and OFL. In this figure, a red digit is the distance between two adjacent devices.

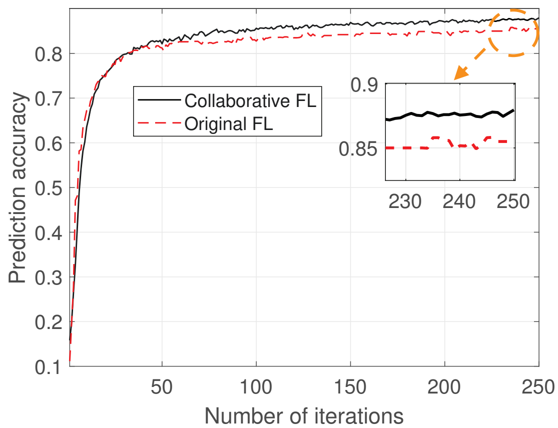


Fig. 3. Simulation results to show the performance of CFL and OFL.

In CFL, some devices are directly connected to the BS while the others are associated with a certain number of neighboring devices. For example, as shown in Fig. 1, for OFL, device $b$ cannot be directly connected to the BS and perform FL due to a potentially high transmission delay. However, in CFL, device $b$ can be connected to device $a$ for performing FL. Since each device in CFL can use broadcast techniques to transmit its local FL model parameters to multiple other devices, CFL does not occupy more resources compared to OFL. CFL is also trained iteratively. First, each device transmits its trained local FL model to its connected devices or the BS. Then, the BS generates the global FL model and transmits it to the associated devices. Finally, each device updates its local FL model based on the local FL models received from other devices or the BS. In OFL, each device trains its local FL model using gradient descent (GD) methods while the BS aggregates the local FL models. However, in CFL, each device must both aggregate the local FL models received from other devices and train its local FL model.

To show the difference between CFL and OFL, we implemented a preliminary simulation for a network having one BS and six devices, as shown in Fig. 2. The local FL model of each device consists of a shallow feedforward neural network with 50 neurons. The MNIST dataset [12] is used for training the local models at each device and each device has 500 data samples. We assume that device $a$ is the farthest device that could be connected directly to the BS for FL training.

Fig. 3 shows how the identification accuracy changes over time. Fig. 3 demonstrates that CFL outperforms OFL. This is because, for OFL, only four devices can participate in FL and the other two devices have a delay larger than 0.23 s. Since CFL allows devices to be connected to other devices and the transmission delay between any two neighboring devices is smaller than 0.23 s, six devices can participate. In fact, CFL can also reduce the energy consumption for device $b$ since it only needs to transmit its ML model parameters to device $a$ instead of the BS. The energy consumed by device $a$ in CFL is similar to that of device $a$ in OFL. This is because, although in CFL device $a$ needs to exchange its local FL model with device $b$ and the BS, it can broadcast its FL model. Meanwhile, it is known that the energy consumed by any wireless device for receiving data is much smaller than the energy needed for data transmission. Therefore, we can reasonably assume that the energy consumed by device $a$ to receive the FL model of device $b$ is negligible.

Table I summarizes the advantages, disadvantages, and operation conditions of CFL. The key advantage of CFL is that it enables the devices to perform the FL without transmitting local FL models to the BS, as shown in Fig. 4. We can now remark:

- Choosing between CL or FL depends on: a) willingness of data sharing, b) ML model data size, and c) size of the collected data of each device. For example, when devices agree to share the data and the size of the collected data is smaller than the ML model data size, CL is recommended.
- Choosing between OFL or CFL depends on: a) whether the BS performs FL and b) the connection and transmission delay between devices and the BS. For example, if all wireless devices need to implement FL without the BS, then CFL is more suitable.
- OFL can be considered as a special case of CFL. In a network, if each device is connected to all other devices, CFL is equivalent OFL.

## III. Performance of CFL over Wireless Networks

*1) Loss Function Value:* The loss function value is used to evaluate the CFL performance. The goal of CFL training is to find an ML model that minimizes the loss function which depends on the local FL models of all participating devices. When those models are transmitted over wireless links, they experience transmission errors and delays, which can negatively impact the loss function during training. Due to limited energy and computing resources, only a subset of devices can be engaged in CFL, which decreases the total number of training data samples used for training the local FL models and increases the loss function value. Table II summarizes the wireless factors that affect the FL loss function along with suggested solutions.

*2) Convergence Time:* For CFL, the convergence time has three components: a) FL model parameter transmission delay,

TABLE II
SUMMARY OF THE WIRELESS FACTORS THAT AFFECT THE PERFORMANCE METRICS AND SUGGESTED SOLUTIONS.

| | **Wireless Factors** | **Effects on FL** | **Suggested Solutions** |
|---|---|---|---|
| **Loss function value** | • Limited bandwidth and computational resources. | • Number of devices that can perform FL at each iteration is limited. | • Probabilistic user scheduling.<br>• Over the air techniques allowing devices to aggregate local FL models over wireless transmission [10].<br>• Optimized network formation. |
| | • Limited transmit power. | • Errors in local FL models. | • Channel coding and decoding.<br>• Intelligent retransmission. |
| **Convergence time** | • Limited bandwidth, energy, and transmit power. | • Use of more time for local FL model parameter transmission. | • Coding and decoding of FL model.<br>• FL model parameter prediction.<br>• Over the air techniques.<br>• Optimized network formation. |
| | • Limited computational resources. | • Number of local FL model updates at each CFL iteration is limited. | • Use of more global FL model updates.<br>• Partial local FL model training. |
| **Energy consumption** | • Limited wireless resources, e.g., bandwidth.<br>• Wireless channel conditions. | • Use of more energy for local FL model transmission. | • Channel coding.<br>• Optimized network formation.<br>• Use of more local FL model updates. |
| **Reliability** | • Limited transmit power. | • Errors in local FL models. | • Channel coding.<br>• Improved device connection policy.<br>• Use of more local FL model updates.<br>• Optimized network formation. |

b) time needed by each device to train its local FL model, and c) number of iterations for FL convergence (i.e., the number of global FL model updates). The model transmission delay depends on the FL model data size and the data rate of the wireless link. The time used to train each device's local FL model depends on the FL model data size, the computational resources of each device, and the number of iterations (called number of local FL model updates hereinafter) that each device uses to train its local FL model (using GD) at each FL iteration. Note that as the number of local FL model updates increases, the number of global FL model updates decreases. The number of global FL model updates also depends on the limited spectrum resources that restrict the number of devices that engage in FL. Table II summarizes the wireless factors that affect the convergence time and the suggested solutions.

*3) Energy Consumption:* The energy consumption needed for training a CFL algorithm has four components: a) local FL model transmission, b) local FL model update, c) global FL model transmission, and d) global FL model aggregation. Each device will spend energy for local FL model transmission and update while the BS needs to spend energy for global FL model transmission and aggregation. A tradeoff exists between the energy consumption of the local FL model update and the transmission energy. The energy consumption of CFL depends on the FL model data size, the distance between the BS and the devices, the convergence time requirement, and the target loss function value. Table II summarizes the wireless factors that affect the energy consumption along with suggested solutions.

*4) Reliability:* For CFL, we define reliability as the probability that a CFL algorithm achieves a target loss function value. At each iteration, erroneous local FL models caused by imperfect wireless transmission must be discarded by the devices. Hence, the number of local FL models used to generate the global FL model will decrease thus increasing the CFL convergence time and the loss function value. Hence, a CFL algorithm may not be able to achieve a target FL loss function value due to imperfect wireless transmissions. Thus, the reliability of CFL depends on the wireless channel conditions. As the transmit power of each device increases, the number of erroneous local FL models decreases thus increasing CFL reliability. Table II summarizes the wireless factors that affect the reliability and the suggested solutions.

## IV. COMMUNICATION TECHNIQUES FOR COLLABORATIVE FEDERATED LEARNING

We now overview key techniques for improving the performance of CFL over wireless networks.

### *A. Network Formation*

The first fundamental step towards deploying CFL is to analyze the process of network formation using which devices could be connected to one another to engage in a CFL task. In CFL, devices can form different network topologies. For example, IoT devices can form a grid topology for CFL, as shown in Fig. 4. Naturally, the training complexity and the FL convergence time directly depend on the formed topology. Hence, for any given scenario, one must investigate the optimal CFL network topology using the metrics of Section III.

Fig. 4 shows the upper bound of the number of iterations for CFL convergence when assuming that the upper bound is derived while assuming that each device updates its local FL model using the Lazy Metropolis method and the GD method [13]. Fig. 4 shows that, when the number of links of each device increases, the number of iterations decreases because having more links increases the frequency of local FL model sharing.

CFL yields interesting network formation research questions:

- **CFL network formation fundamentals**: The optimal CFL network topology depends on the CFL performance

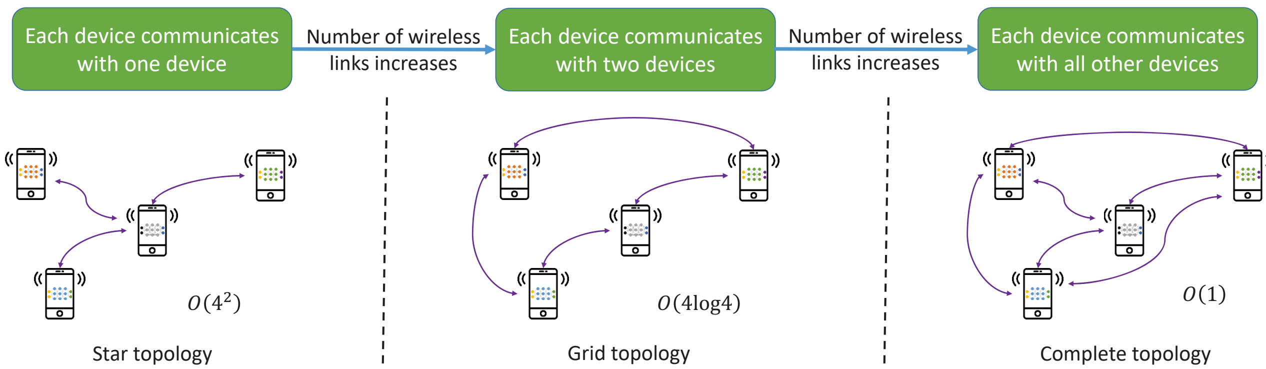


Fig. 4. Number of iterations needed to converge for a CFL algorithm with different topologies [13].

metrics being optimized. Therefore, a fundamental CFL question is that of network formation: How can the devices interact to form an optimal network topology that maximizes the various CFL performance metrics and tradeoffs? To find the optimal topology, the first step is to define a proper utility function that jointly considers multiple, dependent CFL performance metrics and network topology. Given the defined utility function, one must develop network formation algorithms to optimize the utility function. Both centralzied and distributed solutions can be developed. Centralized solutions such as searching based algorithms may be able to find the globally optimal network topology. However, the implementation of centralized solutions requires the collection of devices' information such as locations, which is impractical in a large-scale IoT. For distributed solutions, one can adopt a game-theoretic approach [14]. In a network formation game, each device is treated as an agent whose goal is to form a graph with neighboring devices to optimize the CFL performance metrics. The CFL performance (e.g., utility) depends on the entire graph and decisions of all agents, which makes the use of mean field game theory suitable. When dealing with large-scale networks, one can rely on the tools of mean-field game theory that allow capturing the presence of an infinite number of agents in a game.

- **Network formation with asynchronous training**: Under asynchronous FL training, IoT devices will update and transmit their local FL models at different time slots. Due to limited computing and wireless resources, each device may not want to update its local FL model until it receives all local FL models of its associated devices. Using asynchronous training can increase local FL model update frequency and the data rate of each device which reduces the convergence time. In asynchronous training, the number of devices that need to transmit the local FL models is time-varying. Hence, the network topology must be adapted to the changes in the number of devices that must transmit local FL models. Here, one must determine the frequency with which the network topology must be updated according to the number of participating devices. Each network topology update will change the wireless resource allocation and device association schemes so as to improve CFL performance metrics such as convergence time. However, network topology updates will also introduce communication overhead such as network state information sharing.
- **Network formation with partial network information**: In actual IoT, each device may not completely know the network architecture, device locations, and network composition. Due to this limited information, the number of devices that each device can be connected to is limited and hence devices may not be able to form a network topology that satisfies the CFL operation conditions (see Table I). Therefore, there is a need to investigate a globally optimal network formation for IoT devices with partial information. Since most existing complexity results on network formation (e.g., see [13]) assume that each device has complete information, they cannot be used for devices with partial network information. Meanwhile, due to partial network information, devices may form several unconnected small device groups. Hence, a multi-layer network formation must be designed. For example, in the first layer, devices exchange their local FL model parameters in their own groups while the local FL model parameters are exchanged over multiple groups in the second layer. The designed scheme must balance communication overhead and training complexity across layers.

### *B. Device Scheduling*

Due to energy constraints and wireless resource limitations, the number of devices that can engage in CFL is limited. Hence, an IoT device may update its local FL model using the local models of a subset of devices thus decreasing the CFL convergence time. Therefore, we must find an optimal device scheduling policy that can determine which iterations and the frequency that each device performs CFL thus optimizing the CFL performance metrics.

We envision several research problems related to device scheduling:

- **Data importance-aware device scheduling**: In CFL, the contribution of each device's dataset on the update of a local FL model can be seen as the data importance of that device's dataset. The data importance of each device depends on the number of training data samples and the data distribution. For instance, if a device has a large number of training data samples, its local model will be allocated a large weight during local model update. Since only a subset of devices can perform FL at each iteration, it is necessary to design data importance-aware device scheduling policies for faster convergence. In particular, one must first build a data importance model that jointly considers the number of training data samples, data distribution, and data uniqueness. In CFL, devices cannot share data and, hence, each device may not be able to directly know the data importance of other devices. Therefore, there is a need to find a method to learn the data importance of other devices from their transmitted local FL model parameters. One must determine the frequency of local FL model update for devices with different data importance. Note that increasing the update frequency of the devices with high data importance can improve convergence speed but it also increases the loss function value.

- **Device scheduling for multiple FL tasks**: In an IoT, a device may perform multiple FL algorithms simultaneously. Therefore, we must design a device scheduling policy that enables devices to efficiently train multiple FL models and transmit them to other devices simultaneously. Since each task has its specific convergence time requirement and target loss function value, the developed device scheduling policy must determine which FL model must be trained first and which FL model must be transmitted first so as to satisfy the requirements of each FL task. Since the convergence time of each FL task is different, the designed scheduling policy must be adapted to changes in the number of incomplete FL tasks.

### C. Coding

During the CFL training process, source coding, channel coding, and gradient coding can be used to improve the FL performance. Source coding can compress the high-dimensional FL model parameters so that they can be represented by a small number of bits hence reducing the FL parameter transmission delay [15]. Channel coding can protect the transmitted FL model parameters against noise and interference thus reducing packet errors and improving CFL reliability. Gradient coding is used to encode the GD parameters of machine learning algorithms to improve the ML performance.

Obviously, source, channel, and gradient coding can significantly improve CFL performance. However, a number of research questions still exists:

- **Heterogeneous source coding design**: In an IoT, the wireless link characteristics of each device are different (e.g., different rates). To efficiently use wireless resources for FL model transmission, each device may encode its local FL model using different number of bits or coding techniques. This type of coding schemes is called heterogeneous source coding. For example, some devices can use 15 bits to represent their local FL models while another can use 7 bits. Heterogeneous source coding can significantly reduce the coding energy consumption and decrease the loss function value. However, in CFL, a device must transmit its local FL model to multiple devices. Therefore, one must determine the number of local FL models that each device must encode and the number of bits used for encoding the corresponding local FL models.
- **Gradient coding for avoiding stragglers**: Due to limited wireless resources, an IoT system can have devices with extremely high transmission delay or computational delay. Such devices (called stragglers) may not be able to complete the local FL model transmission within the time duration required by the system. If a network has a large number of stragglers, the number of devices that can perform CFL will significantly decrease. Therefore, there is a need to design gradient coding schemes for addressing the problem of stragglers. However, traditional gradient coding methods require devices to share their dataset with other devices so as to remove stragglers and hence, they cannot be used for CFL since CFL does not allow devices to share their data. Hence, we need new gradient coding schemes without data sharing.

## V. Conclusion

This article has proposed a novel wireless CFL framework and introduced the challenges and opportunities of using wireless communication techniques for optimizing CFL performance. The introduced techniques provide guidance for reliably deploying CFL across IoT devices. The discussed research opportunities identify important open problems that must be considered when designing and deploying CFL for IoT systems. We expect that the proposed CFL framework will fundamentally change the original FL architecture allowing it to be deployed for several future applications such as mobile keyboard prediction, device identification and monitoring, and extreme event detection for autonomous vehicles.

## VI. Acknowledgments

The work was supported in part by the Key Area R&D Program of Guangdong Province with grant No. 2018B030338001, by Natural Science Foundation of China with grant NSFC-61629101, and by Guangdong Research Project No. 2017ZT07X152 and in part by the U.S. National Science Foundation under Grant CCF-1908308 and CNS-1814477.

## References

[1] M. Chen et al., "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, Fourthquarter 2019.

[2] K. Bonawitz et al., "Towards federated learning at scale: System design," in *Proc. Systems and Machine Learning Conference*, Stanford, CA, USA, 2019.

[3] S. Savazzi et al., "Federated learning with cooperating devices: A consensus approach for massive IoT networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4641–4654, May 2020.

[4] J. Kang et al., "Reliable federated learning for mobile networks," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 72–80, April 2020.

[5] J. Park et al., "Wireless network intelligence at the edge," *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019.

[6] S. Niknam et al., "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, June 2020.

[7] K. Yang et al., “Federated learning via over-the-air computation,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.
[8] M. Chen et al., “A joint learning and communications framework for federated learning over wireless networks,” *IEEE Trans. Wireless Commun.*, to appear, 2020.
[9] M. M. Amiri and D. Gunduz, “Federated learning over wireless fading channels,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, May 2020.
[10] G. Zhu et al., “Toward an intelligent edge: Wireless communication meets machine learning,” *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 19–25, Jan. 2020.
[11] A. et al. Elgabli, “GADMM: Fast and communication efficient framework for distributed machine learning,” *J. Mach. Learn. Res.*, vol. 21, no. 76, pp. 1–39, Mar. 2020.
[12] Y. LeCun, “The MNIST database of handwritten digits,” http://yann.lecun.com/exdb/mnist/.
[13] A. Nedic et al., “Network topology and communication-computation tradeoffs in decentralised optimization,” *Proc. IEEE*, vol. 106, no. 5, pp. 953–976, May 2018.
[14] Z. Han et al., *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*, Cambridge Univ. Press, Oct. 2012.
[15] N. Shlezinger et al., “Federated learning with quantization constraints,” in *Proc. IEEE ICASSP*, Barcelona, Spain, May 2020.

**Mingzhe Chen** is currently a Post-Doctoral Researcher at the Electrical Engineering Department, Princeton University and at the Chinese University of Hong Kong, Shenzhen, China. His research interests include machine learning, virtual reality, unmanned aerial vehicles, and wireless networks.

**H. Vincent Poor** is the Michael Henry Strater University Professor of Electrical Engineering. His research interests are in the areas of information theory and signal processing, and their applications in wireless networks and related fields such as energy systems.

**Walid Saad** is a Professor at the Department of Electrical and Computer Engineering at Virginia Tech. His research interests include wireless networks, machine learning, game theory, unmanned aerial vehicles, and cyber-physical systems.

**Shuguang Cui** is the Chair Professor of the Chinese University of Hong Kong at Shenzhen and the Vice Director at Shenzhen Research Institute of Big Data. His current research interests focus on data driven large-scale system control and resource management, large dataset analysis, IoT system design, energy-harvesting-based communication system design, and cognitive network optimization.

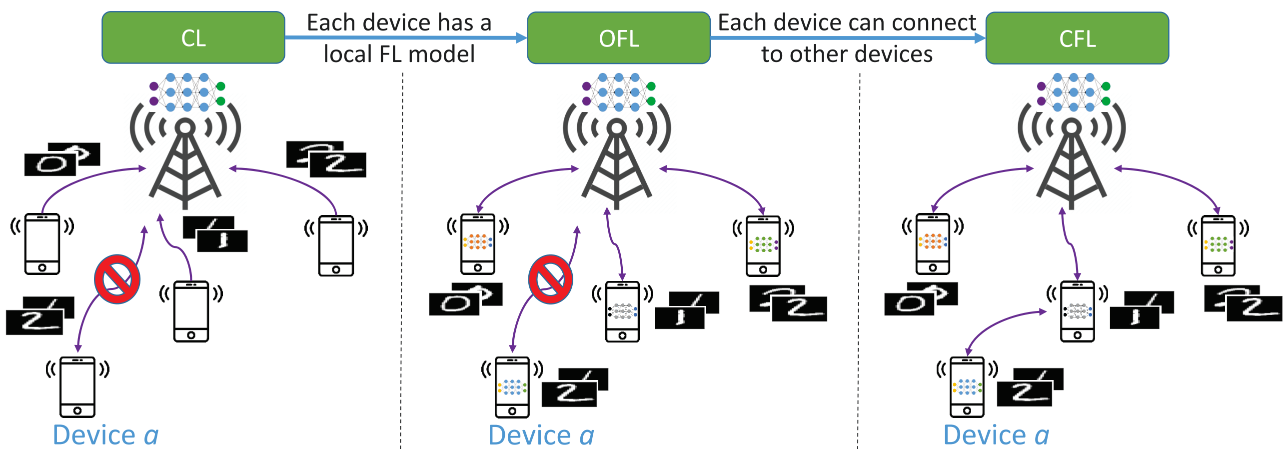


Fig. 5. Architectures of CL, OFL, and CFL. Here, to implement CFL, the BS is not necessary.

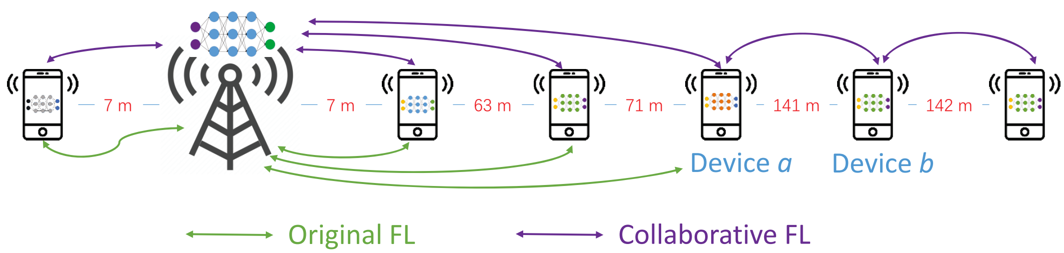


Fig. 6. Simulation system for the implementation of CFL and OFL. In this figure, a red digit is the distance between two adjacent devices.

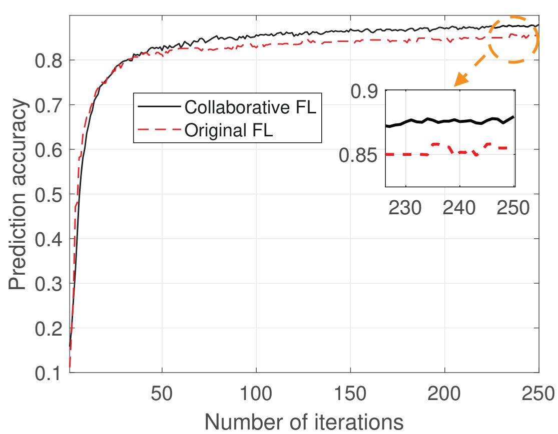


Fig. 7. Simulation results to show the performance of CFL and OFL.

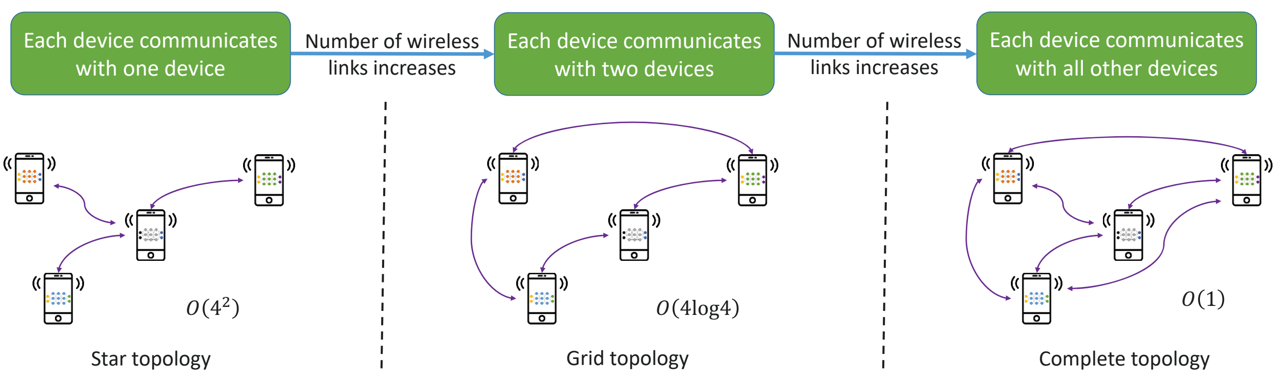


Fig. 8. Number of iterations needed to converge for a CFL algorithm with different topologies [13].