Toward Cooperative Federated Learning over Heterogeneous Edge/Fog Networks

Su Wang, Seyyedali Hosseinalipour, Vaneet Aggarwal, Christopher G. Brinton, David J. Love, Weifeng Su, and Mung Chiang

The authors propose a set of core methodologies that form the foundation of device-to-device and device-to-server cooperation and present preliminary experiments that demonstrate their benefits.

Abstract

Federated learning (FL) has been promoted as a popular technique for training machine learning (ML) models over edge/fog networks. Traditional implementations of FL have largely neglected the potential for inter-network cooperation, treating edge/fog devices and other infrastructure participating in ML as separate processing elements. Consequently, FL has been vulnerable to several dimensions of network heterogeneity, such as varying computation capabilities, communication resources, data qualities, and privacy demands. We advocate for cooperative federated learning (CFL), a cooperative edge/fog ML paradigm built on device-to-device (D2D) and device-to-server (D2S) interactions. Through D2D and D2S cooperation, CFL counteracts network heterogeneity in edge/fog networks through enabling a model/data/resource pooling mechanism, which will yield substantial improvements in ML model training quality and network resource consumption. We propose a set of core methodologies that form the foundation of D2D and D2S cooperation and present preliminary experiments that demonstrate their benefits. We also discuss new FL functionalities enabled by this cooperative framework such as the integration of unlabeled data and heterogeneous device privacy into ML model training. Finally, we describe some open research directions at the intersection of cooperative edge/fog and FL.

Introduction

Recently, much attention has been given to the implementation of data analytics and machine learning (ML) techniques at the network edge to handle the complexity of emerging Internet of Things (IoT) services, ranging from user-oriented (e.g., object recognition) to network-oriented (e.g., signal classification) applications [1]. IoT devices are now capable of gathering data from various sources, connecting to the internet, and performing computation tasks. Collectively, they form edge/fog networks capable of producing machine intelligence insights. Traditionally, data insights were produced via centralized computing, where network devices send all of their local measurements to a single central server for ML training. Such methods led to system-wide latency and resource inefficiencies as a result of data transmissions from edge devices to the server, for centralized ML tasks specifically. These limitations have led to the emergence of distributed ML techniques and in particular federated learning (FL).

Standard FL shifts the processing portion of ML training from the server to the edge/fog devices [2]. As shown in the upper left corner of Fig. 1, it involves a "star" server-to-device communication topology, inside of a three-part cyclical process:

- Edge devices independently and locally train an ML model.
- ML models are sent to the central server for global aggregation.
- The server synchronizes devices' ML models into an aggregated ML model called the global model.

While standard FL features global aggregations, this is the only form of cooperation among network elements. Inter-device and inter-network communications — key features of IoT networks — can also facilitate cooperation and are missed opportunities in FL. For example, direct device-to-device (D2D) communication links that are otherwise underutilized could be employed for faster and communication-efficient ML model training [3]. Traditional FL therefore does not exploit the full potential of cooperation in large-scale edge/fog networks.

We propose cooperative federated learning (CFL), a cooperative edge/fog ML paradigm that jointly orchestrates device, server, and network infrastructure resources to enhance FL while considering its core trade-offs, as shown in Fig. 1. CFL extends the notion of cooperation to address the key missed opportunities in standard FL, which are summarized below:

- Edge devices with powerful local processors or small local datasets idly wait for network stragglers to finish training [4].
- Powerful network infrastructure elements such as edge/fog servers are underutilized in FL [5].
- Edge devices without direct connectivity to the central server are neglected during ML model training and synchronization processes [2].
- IoT devices, which may have diverse privacy requirements [6], are all discouraged from sharing data/models over the network.

These shortcomings are the result of prohibiting powerful devices, idle edge servers, and network

Digital Object Identifier: 10.1109/MCOM.005.2200925

Su Wang, Vaneet Aggarwal, Christopher G. Brinton, David J. Love, and Mung Chiangare with Purdue University, USA; Seyyedali Hosseinalipour and Weifeng Su are with the University at Buffalo — SUNY, USA.

0163-6804/23/$25.00 © 2023 IEEE

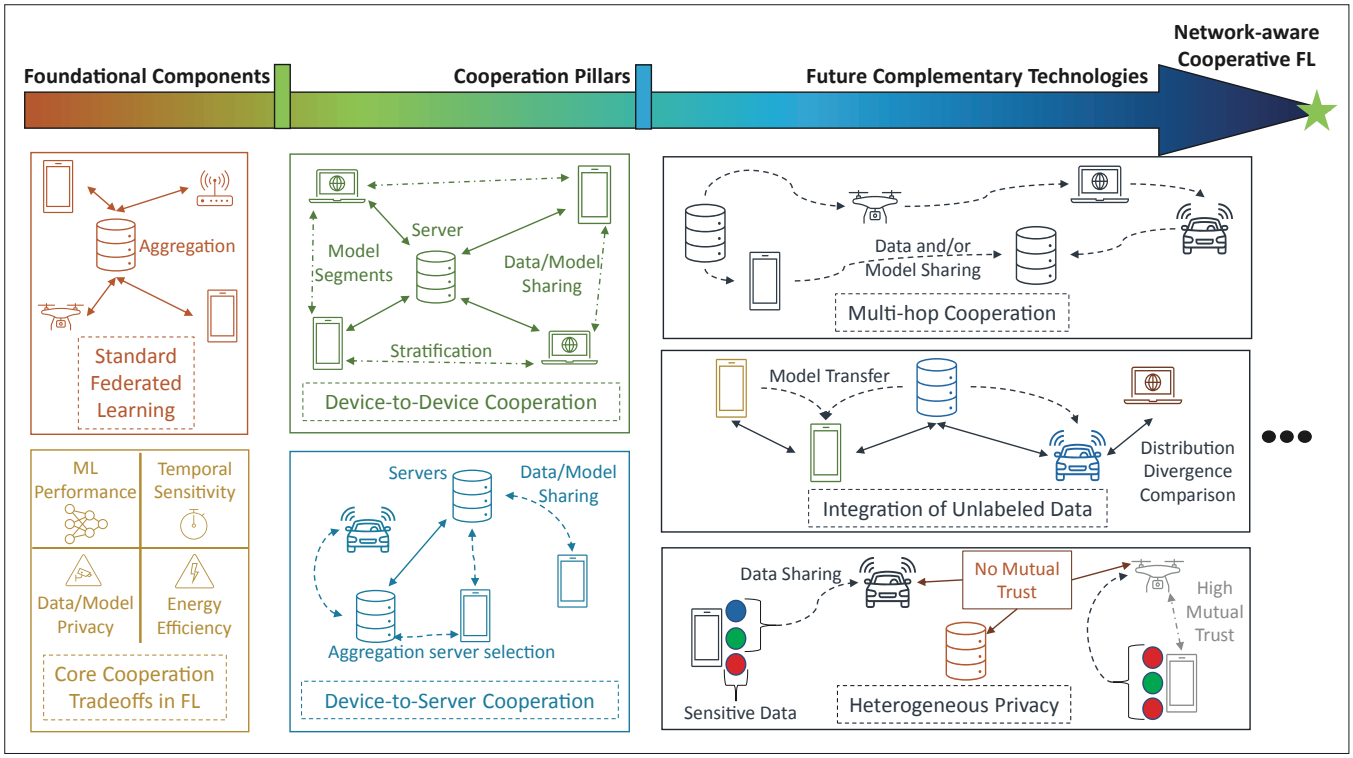

FIGURE 1. The progression from standard FL toward network-aware cooperative FL. Starting from standard FL and its associated trade-offs, we develop the pillars of cooperative FL frameworks, namely device-to-device and device-to-server cooperation. We envision that future work toward integrating collaboration into FL involves ideas such as multi-hop cooperation, integration of unlabeled data, and heterogeneous privacy.

infrastructure from helping computationally weak and/or overburdened devices in FL. Simultaneously, D2D and device-to-server (D2S) cooperation over such edge/fog networks has been shown to be feasible and beneficial to learning processes [7]. Well-designed cooperation mechanisms can thus unlock the full potential of edge/fog networks for FL, leading to:

- Improved ML performance
- Energy efficiency
- Temporal efficiency (e.g., faster ML training)
- Diverse data/model privacy.

Cooperative Federated Learning

We propose cooperative federated learning (CFL), a novel paradigm that expands the dimensions of cooperation in FL beyond global aggregations. CFL develops inter-element cooperation mechanisms including selective data sharing, computation resource sharing, ML model sharing, and data distribution comparisons. FL driven by such multi-faceted cooperation better exploits the availability of links among network devices, servers, and infrastructure in contemporary edge/fog systems. Through such links, CFL unlocks the potential of cooperative edge/fog networks for ML.

Whereas the state-of-the-art in FL [2] has mostly considered data offloading as a mechanism for improving local statistical properties in FL, our vision for CFL involves a broader look at cooperation in FL, including D2D- and D2S-driven data, model, and resource cooperation. These proposed cooperation technologies aim to improve the balance among the trade-offs in FL shown in Fig. 1. For example, intelligent cooperation can lead to better ML model performance with less energy usage and system delay.

Specifically, CFL leverages D2D links for data and model sharing, which we term D2D cooperation, and leverages D2S links to incorporate edge servers, routers, and other network infrastructure into the FL ecosystem through data processing and ML model transmission tasks. We consider data transfers in CFL noting that while some applications of FL (e.g., healthcare analytics) discourage data sharing, other applications have milder data privacy restrictions, especially when the data is generated with ML as the primary purpose (e.g., FL for self-driving vehicles with sensor measurements).

Combined, D2D and D2S cooperation form the two pillars of CFL, which together enable many complementary technologies, we only examine a few of which for brevity. As depicted in Fig. 1, we examine multi-hop cooperation due to its key influence on improved resource efficiency (i.e., energy efficiency and temporal sensitivity), integration of unlabeled data as it enhances ML performance for devices, and devices heterogeneous privacy demands because it focuses on the data/model privacy aspects of CFL.

Overarching Technologies for CFL

In the following, we explain how D2D and D2S cooperation exploit the network characteristics inherent in edge/fog networks, and fulfill the missed opportunities of FL.

Device-to-Device (D2D) Cooperation: In edge/fog networks, devices are heterogeneous statistically (i.e., different dataset characteristics) and structurally (i.e., varying computation/communication capabilities). In standard FL, such heterogeneity leads to isolated resource-abundant and resource-scarce devices, some of which may introduce straggler effects and delay ML model

D2D and D2S cooperation can better represent and facilitate the contribution of devices with partially labeled or unlabeled datasets in FL.

aggregations. In the worst-case, resource-scarce devices may be unable to complete ML model training iterations, possibly due to insufficient battery, or result in the existence of unused local data, as model training in straggler devices use smaller batches of data. To cope with device heterogeneity, we exploit D2D cooperation as discussed below.

Cooperation as Resource Pooling: Cooperative edge/fog networks can reallocate the intensity of local ML training by leveraging D2D links for data transfers from resource-scarce to resource-abundant devices. Through this process, the impact of stragglers (i.e., resource-scarce devices) on ML model training is then reduced. Simultaneously, D2D cooperation shifts the burden of ML model training to devices with energy-efficient processors, and thus can lead to network energy savings.

D2D Driven Model Offloading: Similar to data and subsequent ML training offloading, D2D cooperation can mitigate some of the model aggregation overhead. Rather than only transmitting local ML models to the aggregation server, devices can transfer different parts/chunks of their model to their neighboring devices. In this way, those devices that are far away from the server or have limited communication resources (e.g., limited bandwidth) can communicate with their neighboring devices. Devices that receive models from their neighbors then combine received models with their local one and then transmit these partially combined ML parameters to the server. Thus, D2D model offloading can reduce the number of devices engaging in resource intensive uplink transmissions.

Device-to-Server (D2S) Cooperation: Current FL research presumes conducting ML model training solely on the devices and neglects the underutilized network infrastructure elements such as edge servers. While edge servers may not gather data themselves, they can add value to the FL process owing to their powerful local processors and dedicated communications equipment which can be exploited through network collaboration. We will refer to all cooperation, aside from global model aggregations, between devices and edge servers as D2S cooperation. Two potential use cases of edge servers are provided below.

Computational Resources: D2S cooperation allows resource-scarce devices to transfer their local data to a physically stable and computationally powerful edge server. These edge servers then function similarly as a resource-abundant device, enabling the network to process more training data and lessening straggler effects.

Flexible Data Caches: Using the data they receive from nearby devices, edge servers can act as local data caches, which can carry globally representative distributions of high quality data, to mitigate the impact of non-i.i.d. data across the network. Additionally, this functionality enables better tracking of the distribution shifts in the data via comparing old data with newly arriving data at the edge servers, which enables more informative decisions on ML model training.

Enhancing the Core Properties of CFL

While many applications/extensions are possible from the foundation of D2D/D2S cooperation, we focus on a subset of techniques that can enhance the core trade-offs in CFL. We present high-level explanations of:

- Multi-hop cooperation due to its benefits for resource efficiency (energy efficiency and time sensitivity)
- Integration of unlabeled data as it extends FL to benefit a wider range of devices (e.g., devices with unlabeled data)
- Heterogeneous privacy for its enhancements to data/model privacy.

Multi-Hop D2D and D2S Cooperation: Multi-hop D2D and D2S cooperation refers to extending the above concepts developed for single-hop D2D and D2S cooperation to multiple, sequential links in between devices. This envisioned technology can greatly improve the resource efficiency (i.e., less energy consumption and/or faster model training) of FL. In particular, multi-hop cooperation enables greater connectivity/reach from resource-scarce to resource-abundant edge devices or servers.

Integration of Unlabeled Datasets: D2D and D2S cooperation can better represent and facilitate the contribution of devices with partially labeled or unlabeled datasets in FL. Unlabeled data refers to data samples that have not been tagged with a ground-truth, for example, images taken by cameras mounted on smart cars without pre-assigned or pre-identified types of objects within the image. In standard FL, only devices with labeled data are engaged in ML model training. Consequently, edge devices with unlabeled datasets are unlikely to have their data properly represented at the global ML model and so are likely to suffer from poor ML performance. Roughly speaking, D2D and D2S cooperation can enable approximations of the local data distribution of each device at its neighboring devices/network elements, even if the device has fully or mostly unlabeled data. Subsequently, the type of distributed learning method being applied can be tuned to improve ML performance across the network.

Heterogeneous Privacy: Similar to statistical (i.e., data-level) and structural (i.e., computation/communication resources) differences, edge/fog devices also exhibit heterogeneity with respect to their privacy needs. Heterogeneous privacy needs will motivate selective D2D and D2S cooperation. For example, D2D cooperation can involve sharing sensitive data only among mutually trusted devices (e.g., edge devices belonging to the same user or family), while among untrusted neighbors, this sharing can be limited to sharing insensitive data or even prohibited completely. This is one of the future complementary technologies depicted in Fig. 1. With such methods in place, D2D/D2S cooperative technologies can improve the resource efficiency and ML performance while meeting data/model privacy requirements of edge/fog network elements.

Toward Network-Aware CFL

As depicted in Fig. 1, our vision for CFL relies on cooperative edge/fog networks to enhance standard FL on:

- ML performance — the effectiveness of the ML model trained by the network
- Energy efficiency — the network-wide accumulated energy expenditure on data pro-

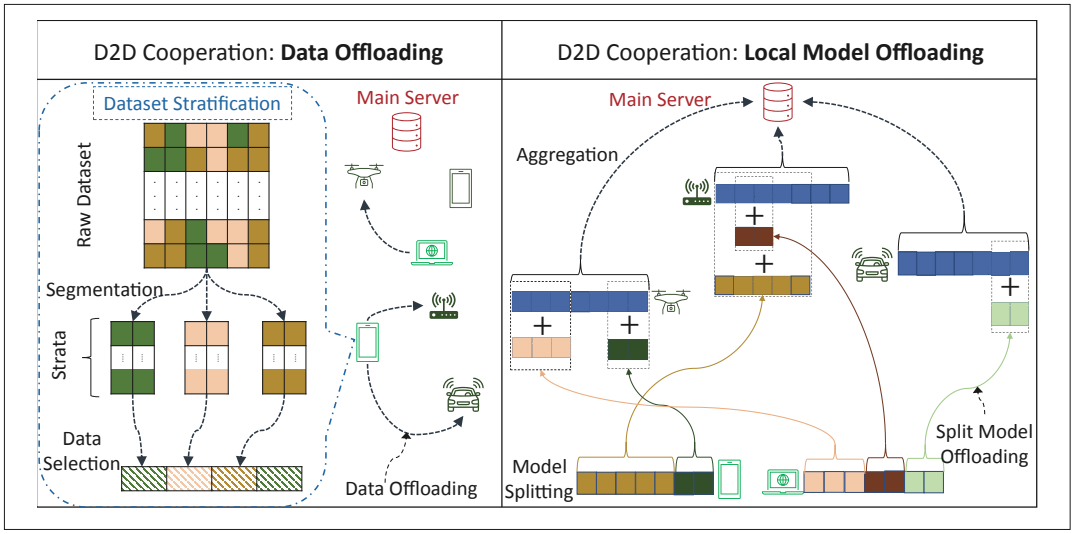

FIGURE 2. Data and local model offloading form the basis of effective D2D cooperation. We envision smart data offloading by edge devices through dataset stratification. Additionally, we propose local model offloading, where devices offload segments of their local ML models to other devices.

cessing, and data/model communication
- Temporal efficiency — the total time including idle time consumed by the FL process
- Data and model privacy — the heterogeneous privacy needs in large-scale edge/fog networks.

Methodologies that develop network-aware CFL must carefully balance their contributions to these four coupled elements, which contain design trade-offs.

The first layer of network-aware CFL consists of the pillars of D2D and D2S cooperation and their core mechanisms. D2D and D2S cooperation can leverage data and model transfers to cope with device heterogeneity.For example, data offloading through D2D links can yield energy savings, and ML model routing through D2S links can mitigate straggler effects.

With our established frameworks of D2D and D2S cooperation, we can then develop complementary technologies to enhance CFL along the four design considerations of ML performance, energy efficiency, temporal sensitivity, and data/model privacy. In Fig. 1, we depict three sample complementary techniques, with each technique primarily improving one aspect of CFL. For instance, multi-hop collaborations such as those seen in industrial IoT [8] enhance resource (energy and delay) efficiency, integration of unlabeled data such as those in autonomous driving [9] can improve ML performance, and heterogeneous privacy as seen in social trust [6] offers an alternative approach to data/model privacy. We next develop D2D and D2S cooperation as the two pillars of network-aware CFL, and, as future work, explain how complementary technologies can enhance them.

Network-Aware D2D Cooperation

The first step toward network-aware CFL is to develop and maximize the benefits arising from D2D cooperation.Well-designed D2D cooperation can enable efficient orchestration of limited network resources, leading to improvements in the trade-offs of FL from Fig. 1. To this end, we first introduce a set of core, overarching technologies to enable effective D2D cooperation, and thereafter propose future work on complementary technologies to further enhance D2D cooperation.

Core Techniques for Effective D2D Cooperation

Effective D2D cooperation improves network resource efficiency and ML model training through innovations in data and model offloading, which we propose in Fig. 2. At the data level, we propose dataset stratification, a method which clusters local datasets for higher quality data offloading. At the ML model level, we propose local model offloading, a technique that involves partial local ML model offloading to streamline efficient ML model aggregations.

Dataset Stratification: After offloading data, the sending device (sender) continues local ML model training on a smaller local dataset, as keeping a copy of the transferred data leads to bias at ML model aggregations from counting the same data multiple times. On the other end, the receiving device (receiver) may receive data that is unrepresentative of the data gathered at the sender. As a result, random/naive data offloading may hinder rather than help the ML model training, motivating dataset stratification.

As depicted on the left subplot of Fig. 2, dataset stratification clusters datasets into strata (i.e., categories) based on task-dependent criteria. Using the example of clothing recognition (i.e., the Fashion-MNIST dataset [10]) for data collected by smartphones' cameras, each stratum may contain data belonging to a unique type of clothing (e.g., T-shirts or coats). Then, through sequential selection of the most representative data samples (i.e., those that are closest to the strata average) from the most populous strata, devices can offload datapoints which well-capture the distribution of their local dataset. In doing so, dataset stratification keeps the distribution of the dataset at senders relatively intact, and enables receivers to receive a representative sample of data from senders.

As an example, consider a smart car communicating with a drone. From its operation, the car has images of mostly stop signs and traffic lights, which the drone may not. Dataset stratification enables the car to transmit a small set of representative stop sign

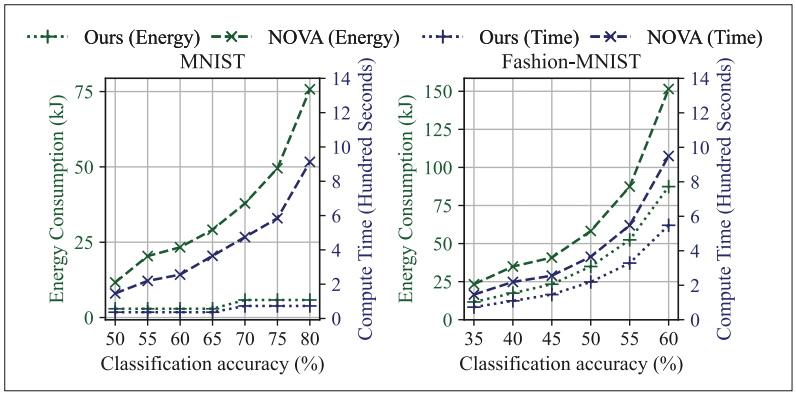

FIGURE 3. Our D2D cooperation-driven method incurs both less compute time and energy consumption relative to the state-of-the-art NOVA methodology on two commonly used machine learning datasets (MNIST for numbers and Fashion-MNIST for clothes).

and traffic light pictures, without significantly distorting its local dataset distribution, to the drone.

Local Model Offloading: We propose segmenting and sequentially transmitting the devices' local ML models at global ML model aggregations.Each segment of an ML model is a subset of model parameters. For example, in the case of neural networks, each segment may contain the parameters associated with a layer of the neural network. Local model offloading, depicted in Fig. 2, enables devices with limited access to the server (e.g., due to unsatisfactory channel conditions) to offload different segments of their local ML models to intermediary devices with a better accessibility to the server. These intermediary devices will combine their local ML models with their received partial ML models, leading to a set of partially aggregated parameters, which are then sent to the central server. In this way, the central server is able to perform the global aggregation using all ML model parameters while saving communication resources.

We have taken initial steps toward formalizing this D2D cooperation methodology in our prior work [10], including optimization formulation, theoretical results, and convergence proofs. To evaluate its potential benefits, we compare it against the algorithm Nova [11] on two common datasets used to evaluate FL methods: MNIST (numbers) and Fashion-MNIST (clothes). Simulation results are shown in Fig. 3, where both methods train a two layer convolutional neural network (CNN) across a network of 10 devices. A detailed description of the computational infrastructure, wireless channel models, and models of energy consumption (including energy from both the communication and computation processes) can be found in [10].We compare against Nova [11], a recent and well-established method for aggregations involving heterogeneous training epochs across edge/fog devices. Our proposed CFL technology is seen to yield consistent energy savings, and faster ML training times.

Future Development of Complementary Technologies

Further extension of D2D cooperation can further enhance the trade-offs in FL and subsequently CFL depicted in Fig. 1. A few open research directions are summarized below:

D2D Cooperation in Non-Stationary Networks: In non-stationary networks, devices will enter and exit the network, leading to varying network size, changing compute resource availability, and time-varying D2D links. Here effective cooperation should consider the physical stability of edge/fog devices to determine effective time-varying anchor devices.These anchors will receive nearby ML models from devices as they leave the network and transmit the latest global ML model to new devices as they join the network. In doing so, anchor devices improve ML training by enabling more devices to contribute to the training process in-between global model aggregations.

n-Hop Cooperation: n-hop cooperation aims to broaden the scope of D2D cooperation, providing resource-scarce devices with greater access to resource-abundant ones through intermediary devices. Consequently, this technology can further enhance the resource and time savings introduced by single-hop D2D cooperation. This calls for novel optimization methodologies to characterize the benefits and trade-offs of n-hop cooperation.

Heterogeneous Privacy Needs in D2D Cooperation: Edge/fog devices may have heterogeneous privacy needs. For example, D2D connections may be allowed based on trust or familiarity. In such cases, devices often band together into cliques, which are private groups with a certain level of mutual trust.Data transferring can be restricted to links between mutually trusted devices (e.g., smart devices such as a smartphone, laptop, and tablet of the same owner). In intra-clique cooperation, then, devices can share data without any restrictions, while, in inter-clique cooperation, devices may only be willing to share model parameters or insensitive data. Furthermore, dataset stratification can be designed to separate data based on sensitivity, with restricted offloading of sensitive strata (e.g., personal health data or biometrics) among intra-clique devices and unrestricted offloading of insensitive strata (e.g., weather information).

Inclusion of Devices With Unlabeled Data: In practical edge/fog networks, some devices may have mostly or fully unlabeled datasets. Standard FL neglects all these devices and obtains a global ML model by only engaging the devices with labeled datasets.Through D2D cooperation, edge devices can share small quantities of data, labeled or unlabeled, to develop estimates of data distributions at devices with unlabeled datasets. This technique, termed unlabeled distribution estimation, will then involve determining unique combinations of ML models trained by devices with labeled datasets for use at devices with unlabeled datasets.

Network-Aware D2S Cooperation

Edge servers, especially in large-scale edge/fog networks, offer an untapped resource in standard FL.D2S cooperation aims to facilitate efficient utilization of these resources, for example, by enabling devices with limited computation capabilities to transfer local training data to edge servers. In doing so, edge servers can leverage their powerful and efficient processors to improve energy efficiency and training delay of ML model training, thus enhancing the trade-off between energy consumption and ML performance. In the following, we introduce a set of core technologies that enable effective D2S cooperation, and thereafter explain complementary future technologies to further enhance it.

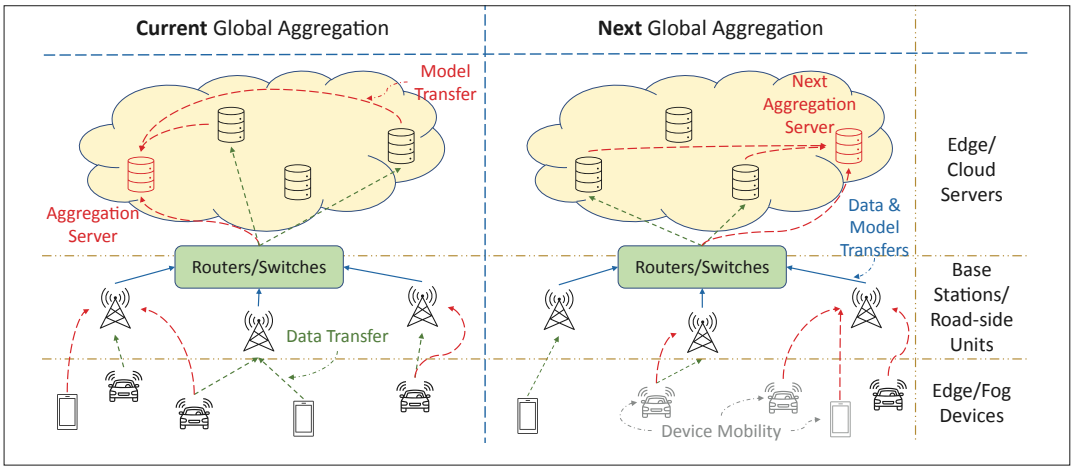

FIGURE 4. Device-to-Server (D2S) cooperation via data/model transfers can improve the resource efficiency and performance of FL. Additionally, intelligent selection of the aggregation server can further reduce aggregation delay.

Core Techniques for Effective D2S Cooperation

Edge servers can have a diverse set of functionalities when assisting ML. They can act as computational resources to train ML models, as communication gateways to reroute ML models during global aggregations, and as the model aggregation points. Starting at the computational level, we propose a novel technology called load balancing of data processing tasks, that relies on data offloading from edge devices to edge servers. Then, we propose efficient ML model parameter and data routing from edge devices to edge servers using base stations and network routers. Finally, we develop a concept called floating aggregation point, a method to optimize the selection of the aggregation server to save communication resources and minimize communication delay. We present a visual summary of these new technologies in Fig. 4, and explain them in detail below.

Load Balancing of Data Processing: In standard FL, edge devices perform all of the computationally intensive ML model training tasks. We propose a novel technology of load balancing for data processing tasks in order to make use of the computational power at edge servers, similar to mobile edge computing frameworks in large-scale edge/fog networks [12]. As part of load balancing, resource-constrained edge devices have the option to transfer a subset or all of their local data to nearby base stations (or road-side units), which through efficient data routing (another innovation which we explain next) relay the data to one or many edge servers.

Efficient Data and Parameter Routing: To enable D2S cooperation, we propose efficient data and model parameter routing through the use of routers/switches as shown in Fig. 4. Through a combination of base stations and routers/switches, we can finely control the routing of data and ML models based on communication factors, such as channel congestion (a base station may be serving many users), and systems factors, such as computation power availability (an edge server may be running intensive data backups). This fine-grained control of data and ML model routing enables energy efficient and fast ML training.

Floating Aggregation Point: In scenarios with many edge servers, we can improve resource efficiency by dynamically selecting the aggregation server. Specifically, we propose floating aggregation point, a novel technology that adjusts the aggregation server based on changing edge/fog network properties. As shown in Fig. 4, the choice of global aggregation server changes in response to devices' positions and dataset sizes, which influence the total communication/computation resource consumption for ML model training and parameter aggregation (uplink) and broadcasting (downlink).

We have taken initial steps toward formalizing such a D2S cooperation methodology in [13], including corresponding mathematical formulations and a proof-of-concept testbed implementation. To demonstrate the potential benefits, we compare our method to FedAvg [2] and Nova [11] on two common benchmark datasets in FL literature: Fashion-MNIST (clothes) and CIFAR-10 (common objects) in Fig. 5. In this experiment, we consider a network of 20 edge devices and 10 edge servers training a two layer CNN. Other system parameters, such as the wireless channels and edge server links, can be found in [13]. This shows that CFL obtains substantial improvements over both baselines in terms of energy and time savings for the same target ML performance.

Future Development of Complementary Technologies

D2S cooperation can be further extended to enhance the existing trade-offs in FL and CFL depicted in Fig. 1.The following outlines a few open research directions.

Non-Stationary Servers: Given current trends leveraging unmanned aerial vehicles (UAVs) as mobile communication servers (e.g., at sporting events), a natural next step for D2S collaboration involves non-stationary edge/fog servers [14]. Since mobile servers' locations can be controlled, efficient server placement methods can be pursued to improve data/model routing and offloading. These methods should carefully investigate the trade-offs involved in server positioning. For example, placing a server near a dense neighborhood of devices may ease aggregation delay, but placing a server near a few resource-scarce devices may save more computation energy.

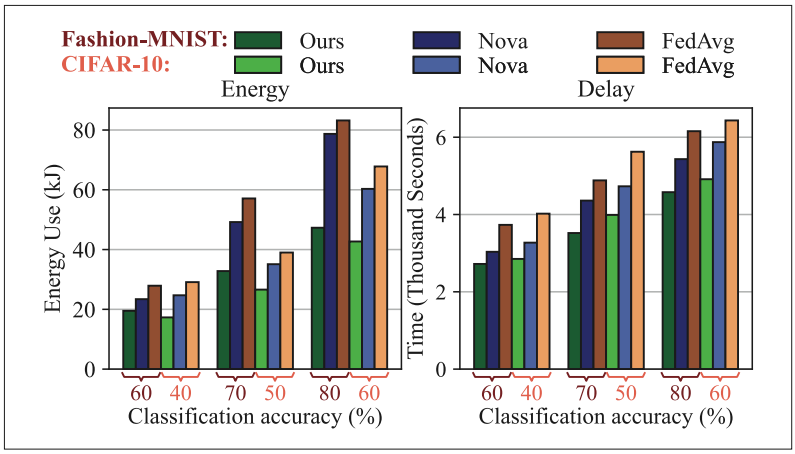

FIGURE 5. On both Fashion-MNIST and CIFAR-10, our method based on D2S collaborations enable both energy and time savings during the ML model training process.

Joint D2D and D2S Collaboration: Frameworks combining D2D and D2S can yield further benefits to ML performance, resource consumption, and time efficiency.Such frameworks can enable simultaneous D2D and D2S data/model offloading. For example, as an edge device offloads data to another device, this secondary device could simultaneously can offload data to an edge server. This combined D2D and D2S cooperation ensures that resource-abundant edge devices do not become overburdened, and resource consumption of data offloading (i.e., energy and delay) is optimized.

Inclusion of Unlabeled Data: D2S cooperation can also be used to extend FL to edge/fog networks with fully unlabeled data, such as autonomous driving where camera-equipped cars take images without labels. One possible approach is to extended the well-known concept of contrastive learning [15] to FL. Standard contrastive learning differentiates among datapoints in centralized settings via determining their similarities and differences. However, in federated settings, edge devices' datasets may simply be too small or lack sufficient data for standard contrastive learning to be effective. Through D2S collaboration, contrastive learning can be enabled in FL by leveraging edge servers as caches of data. As caches, edge servers can then supplement edge devices' local datasets with their cached data so that the compare and contrast steps of contrastive learning are feasible/effective in FL.

Conclusion

We proposed cooperative federated learning (CFL), a paradigm that extends the notion of cooperation in federated learning (FL) and unlocks the potential of edge/fog networks in the execution of distributed machine learning tasks. Through device-to-device (D2D) and device-to-server (D2S) cooperation, CFL counteracts the heterogeneity of edge/fog networks to improve ML model performance, energy efficiency, temporal sensitivity, and data/model privacy. We proposed novel technologies that enable efficient D2D and D2S cooperation in CFL. Finally, we illustrated how CFL can extend the frontiers of research in FL.

Acknowledgment

This work was supported in part by the National Science Foundation (NSF) under Grant CNS-2146171; in part by DARPA under Grant D22AP00168-00; and in part by the Office of NavalResearch (ONR) under Grant N00014-23-C-1016.

References

[1] B. Salau, A. Rawal, and D. B. Rawat, "Recent Advances in Artificial Intelligence for Wireless Internet of Things and Cyber-Physical Systems: A Comprehensive Survey," *IEEE Internet Things J.*, 2022.
[2] P. Kairouz *et al.*, "Advances and Open Problems in Federated Learning," *Found. Trends Machine Learn.*, vol. 14, no. 1–2, 2021, pp. 1–210.
[3] Z. Su *et al.*, "Secure and Efficient Federated Learning for Smart Grid With Edge-Cloud Collaboration," *IEEE Trans. Industrial Inform.*, vol. 18, no. 2, 2021, pp. 1333–44.
[4] A. Reisizadeh *et al.*, "Straggler-Resilient Federated Learning: Leveraging the Interplay Between Statistical Accuracy and System Heterogeneity," *IEEE J. Sel. Areas Inf. Theory*, 2022.
[5] V. S. Mai, R. J. La, and T. Zhang, "Federated Learning With Server Learning: Enhancing Performance for Non-Iid Data," arXiv:2210.02614, 2022.
[6] X. Lin *et al.*, "Friend-as-Learner: Sociallydriven Trustworthy and Efficient Wireless Federated Edge Learning," *IEEE Trans. Mobile Comput.*, 2021.
[7] B. Kar *et al.*, "Offloading Using Traditional Optimization and Machine Learning in Federated Cloud-Edge-Fog Systems: A Survey," *IEEE Commun. Surveys & Tuts.*, 2023.
[8] Z. Li *et al.*, "Data Heterogeneity-Robust Federated Learning via Group Client Selection in Industrial IoT," *IEEE Internet Things J.*, 2022.
[9] C. Luo, X. Yang, and A. Yuille, "Self-Supervised Pillar Motion Learning for Autonomous Driving," *Proc. CVPR*, 2021, pp. 3183–3192.
[10] S. Hosseinalipour *et al.*, "Parallel Successive Learning for Dynamic Distributed Model Training Over Heterogeneous Wireless Networks," arXiv:2202.02947, 2022.
[11] J. Wang *et al.*, "A Novel Framework for the Analysis and Design of Heterogeneous Federated Learning," *IEEE Trans. Signal Process.*, vol. 69, 2021, pp. 5234–49.
[12] R. Kaewpuang *et al.*, "A Framework for Cooperative Resource Management in Mobile Cloud Computing," *IEEE JSAC*, vol. 31, no. 12, 2013, pp. 2685–2700.
[13] B. Ganguly *et al.*, "Multi-Edge Server-Assisted Dynamic Federated Learning With an Optimized Floating Aggregation Point," arXiv:2203.13950, 2022.
[14] S. Savazzi *et al.*, "Opportunities of Federated Learning in Connected, Cooperative, and Automated Industrial Systems," *IEEE Commun. Mag.*, vol. 59, no. 2, 2021, pp. 16–21.
[15] T. Chen *et al.*, "A Simple Framework for Contrastive Learning of Visual Representations," *Proc. ICML*, 2020, pp. 1597–1607.

Biographies

Su Wang [S'19] is a Ph.D. candidate at Purdue University.

Seyyedali Hosseinalipour [M'20] is an Assistant Professor at University at Buffalo (SUNY).

Vaneet Aggarwal [SM'15] is a Professor at Purdue University.

Christopher G. Brinton [SM'20] is the Elmore Rising Star Assistant Professor at Purdue University.

David Love [F'15] is the Nick Trbovich Professor at Purdue University.

Weifeng Su [F'18] is a Professor at University at Buffalo (SUNY).

Mung Chiang [F'12] is the President of Purdue University.