Federated Learning for Edge Networks: Resource Optimization and Incentive Mechanism

Latif U. Khan, Shashi Raj Pandey, Nguyen H. Tran, Walid Saad, Zhu Han, Minh N. H. Nguyen, and Choong Seon Hong

The authors present the primary design aspects for enabling federated learning at the network edge. They model the incentive-based interaction between a global server and participating devices for federated learning via a Stackelberg game to motivate the participation of the devices in the federated learning process.

Abstract

Recent years have witnessed a rapid proliferation of smart Internet of Things (IoT) devices. IoT devices with intelligence require the use of effective machine learning paradigms. Federated learning can be a promising solution for enabling IoT-based smart applications. In this article, we present the primary design aspects for enabling federated learning at the network edge. We model the incentive-based interaction between a global server and participating devices for federated learning via a Stackelberg game to motivate the participation of the devices in the federated learning process. We present several open research challenges with their possible solutions. Finally, we provide an outlook on future research.

Introduction

Emerging Internet of Things (IoT) applications such as augmented reality, autonomous driving, surveillance, and industry 4.0 generate significant amounts of data. The effective deployment of such applications is thus reliant on the use of advanced machine learning techniques so as to properly exploit the generated data. However, traditional machine learning schemes use centralized training data at a data center, which requires data transfer from a massive number of distributed IoT devices to a third-party location, which raises serious privacy concerns and can be inefficient in its use of communication resources. To overcome these privacy and communication concerns, it is important to introduce distributed, edge-deployed learning algorithms such as federated learning (FL). FL allows privacy preservation by enabling distributed training without raw data transfer [1].

An overview of how FL can enable IoT-based applications is presented in Fig. 1. To benefit from FL at the network edge, several challenges must be addressed that include resource management and incentive mechanism design to motivate the participation of users in the learning of a global FL model. Learning in IoT has been studied in [2–6]. References [2, 3] rely on centralized learning solutions that have limited scalability and privacy preservation. In [4], the authors presented the challenges of FL along with its existing solutions and applications in mobile edge network optimization. In [5], the authors proposed an FL framework to provide efficient resource management at the network edge. However, [5, 6] do not discuss the important challenges pertaining to incentive design and network optimization under edge-based FL. In contrast, the overarching goal of this article is to comprehensively review a resource optimization and incentive mechanism for FL. In contrast to [4], which focuses only on high-level challenges, we present a new perspective related to the development of incentive-based FL over edge networks using game theory. We also identify new challenges and open problems, different from [4]. Our key contributions include:

- We present the key design aspects for implementing FL in edge networks.
- We present a Stackelberg-game-based approach to develop an FL incentive mechanism. In this game, FL users can strategically set the number of local iterations to maximize their utility. Meanwhile, the base station (BS), acting as leader, uses the best response strategies of the users to maximize the FL performance. The BS's utility is modeled as a function of key performance metrics such as the number of global iterations and global accuracy level in the FL setting.
- Finally, we present some key open research challenges along with guidelines pertaining to FL in edge networks.

Federated Learning at the Edge: Key Design Aspects

Resource Optimization

Optimization of communication and computation resources is necessary to enable the main phases of FL local computation, communication, and global computation. When optimizing FL computational and communication resources, the original problem whose goal is to minimize the FL cost function can have a dual formulation without constraints. Moreover, if the original problem is convex, the dual problem has the same solution. Thus, the dual problem can be decoupled for obtaining a distributed solution in FL. Computation resources can be either those of a local device or of an edge server, whereas communica-

Latif U. Khan, Shashi Raj Pandey, Minh N. H. Nguyen, and Choong Seon Hong (corresponding author) are with Kyung Hee University; Nguyen H. Tran is with the University of Sydney; Walid Saad is with Virginia Tech, and also with Kyung Hee University; Zhu Han is with the University of Houston, and also with Kyung Hee University. Latif U. Khan and Shashi Raj Pandey contributed equally to the article.

Digital Object Identifier:
10.1109/MCOM.001.1900649

0163-6804/20/$25.00 © 2020 IEEE

tion resources are mainly radio resources of the access network. In the local computation phase, every selected device iteratively performs a local model update using its dataset. The allocation of local device computational resources strongly depends on the device energy consumption, local learning time, and local learning accuracy. Further, the heterogeneity of the local dataset sizes significantly affects the allocation of local computational resources. Device energy consumption and local learning time are strongly dependent on CPU capability. Increasing the device CPU frequency can increase the energy consumption and decrease the learning time. Similarly, the local computing latency increases for a fixed frequency with an increase in local learning accuracy. Evidently, there is a need to study the trade-off between computation energy consumption, computational latency, learning time, and learning accuracy. Moreover, the access network and core network resources must be allocated optimally during the communication phase.

Learning Algorithm Design

FL uses local and global computation resources along with communication resources. Several machine learning techniques, such as long short-term memory, convolutional neural network, and Naive Bayes schemes can be used at each local device. To enable FL, numerous optimization schemes, such as federated averaging (FedAvg) and FedProx can be used to train non-convex FL models [7]. FedProx is a modified version of FedAvg that captures both statistical and system heterogeneity among end devices. FedAvg runs stochastic gradient descent (SGD) on a set of devices to yield local model weights. Subsequently, an averaging of the local weights is performed at the edge computing server located at the BS. FedProx has similar steps as FedAvg, but the difference lies in local device minimizing of objective function that considers the objective function of FedAvg with an additional proximal term. By doing so, FedProx limits the impact of non-independent and identically distributed (non-i.i.d.) device data on the global learning model. FedAvg does not guarantee theoretical convergence, while FedProx shows theoretical convergence.

In FedAvg and FedProx, all devices are weighted equally in global FL model computation without considering fairness, despite the differences in the device capabilities (e.g., hardware). To capture such fairness among devices, a so-called fairness-enabled FedAvg algorithm was proposed [8]. Fairness-enabled FedAvg assigns higher weights to devices with poor performance by modifying the objective function of the typical FedAvg algorithm. To introduce potential fairness and reduce training accuracy variance, local devices having a high empirical loss (local loss function) are emphasized by assigning higher relative weight in the fairness-enabled FedAvg. Meanwhile, in [9], an adaptive control scheme was proposed to adapt the global FL aggregation frequency. This adaptive control scheme offers a desirable trade-off between global model aggregation and local model update to minimize the loss function with resource budget constraint. All of the above-discussed methods are used for a single task global FL model. In real-world IoT systems, it is also of interest to use multi-task FL for handling multiple tasks, whose data is distributed among multiple edge nodes. A federated multi-task learning scheme was proposed in [10] by modifying the so-called communication-efficient distributed dual coordinate ascent (CoCoA) framework. To enable a wide variety of machine learning models, CoCoA supports objectives for linear regularized loss minimization [11]. In CoCoA, partial results from local computation are effectively combined using optimization problems primal-dual structure. In each round, CoCoA enables the use of any arbitrary optimization algorithm on a local dataset to solve a local learning problem by using distributed optimization for coping with system-level and statistical heterogeneity.

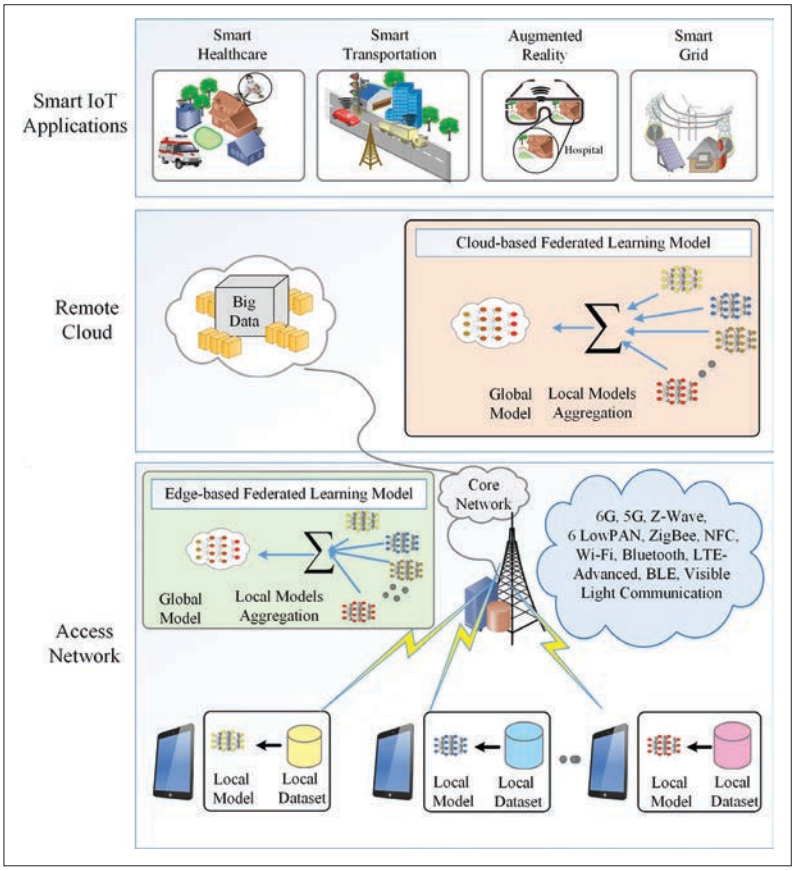

Figure 1. An overview of FL in enabling IoT-based smart applications.

Hardware-Software Co-Design for Federated Learning

For a fixed hardware design, one can find optimal software design by searching for different architectures. However, this approach poses limitations on the design because neural network design is strongly dependent on the used dataset. Therefore, there is a need to jointly consider both hardware design space and neural architecture search space for a more flexible design of the end device for FL [12]. One promising approach for efficient design of end devices involved in FL is hardware-software co-design. Several approaches such as high-level synthesis, co-verification-based embedded systems, and virtual prototyping can be used for hardware-software co-design of IoT devices. A design based on virtual prototyping uses computer-aided engineering, computer-automated design, and computer-aided design for

The design of mechanisms that incentivize users to participate in FL is a key challenge. Incentives are possible in different forms, such as user-defined utility and money-based rewards. Several frameworks such as game theory and auction theory can be used in the design of FL incentives.

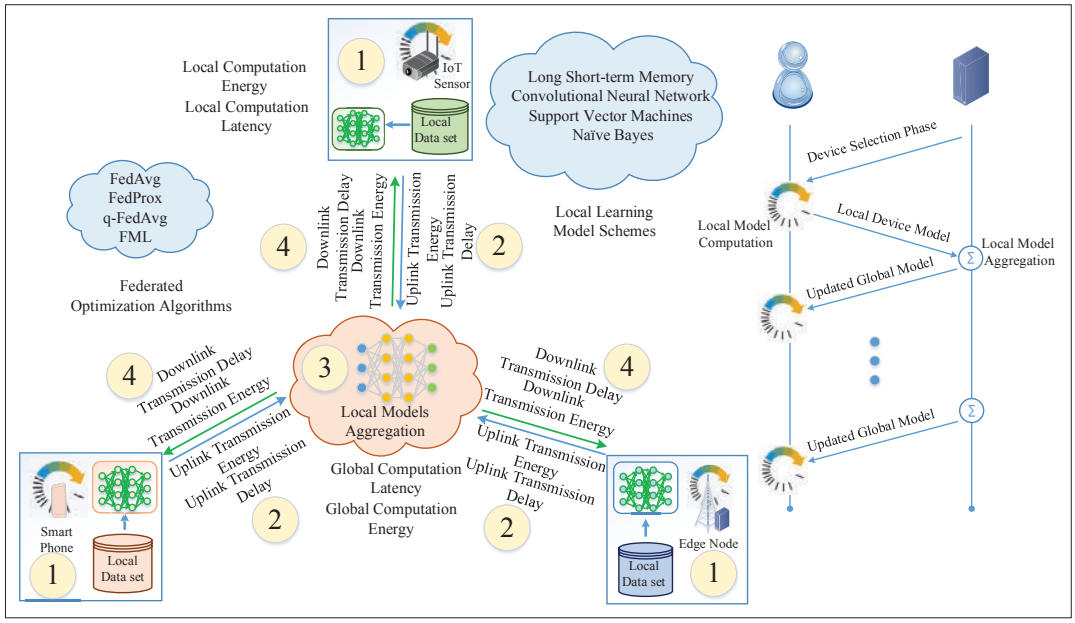

Figure 2. FL sequence diagram.

the validation of a design before prototype implementation, whereas a high-level synthesis offers an automated design process by creating digital hardware based on the algorithmic description for the desired behavior. The prominent challenges of high-level synthesis-based design are wired signal and multiplexer delays. Moreover, co-verification-based embedded systems enable concurrent testing and debugging of both software and hardware design; however, such designs require successful interactions between hardware and software teams.

Incentive Mechanism Design

The design of mechanisms that incentivize users to participate in FL is a key challenge. Incentives are possible in different forms, such as user-defined utility and money-based rewards. Several frameworks such as game theory and auction theory can be used in the design of FL incentives [13, 14]. One can design an incentive mechanism using game theory while considering both communication and computation costs. The communication cost can be defined as the total number of rounds used for the interactions between the edge server and end devices, whereas the computational cost can be the number of local iterations required to compute the local learning model [4]. For synchronous aggregation, given a fixed number of global FL rounds between end devices and edge server, the convergence rate of the global FL model has a proportional relationship with the number of local iterations. An increase in the number of local iterations minimizes the local learning model error, and thus, few global FL rounds are required to reach a certain global FL model accuracy. Therefore, for a fixed global FL model accuracy, an increase in computational cost reduces communication cost and vice versa. For instance, consider an incentive mechanism game whose players are the edge server and edge users. The edge server announces a reward as an incentive to the participating users while maximizing its benefits in terms of improving global FL model accuracy. Meanwhile, the edge users maximize their individual utilities to improve their benefit. One example of a user utility could be the improvement of local learning model accuracy within the allowed communication time during FL training. An improvement in the local learning model accuracy of the end user increases its incentive from the edge server and vice versa. This process of incentive-based sharing of model parameters continues until convergence to some global model accuracy level.

Incentive-Based Federated Learning over Edge Networks

System Model

Consider a multi-user system comprising a BS and a set of user devices with non-i.i.d. and heterogeneous data sizes. Enabling FL over such edge networks involves the use of the computational resources at both device and cloud levels, as well as network communication resources. In a typical FL environment, participating user equipment (UE) must iterate over their local (possibly non-i.i.d.) data to train a global model. However, UEs are generally reluctant to participate in FL due to limited computing and communication resources. Thus, enabling FL requires some careful design considerations:

- First, to motivate UEs for participation, it is necessary to model the economic interaction between the BS and the UEs. Within each global iteration, the BS can offer a reward rate (e.g., iterations) to the UEs for selecting the optimal local iteration strategy (i.e., CPU-frequency cycle) that can minimize the overall energy consumption of FL, with a minimal learning time.
- The set of resource-constrained UEs involved in FL has numerous heterogeneous parameters: computational capacity, training data size, and channel conditions. This heterogeneity significantly affects the local learning

model computation time for a certain fixed local model accuracy level. For a synchronous FL setting, the local learning model accuracy will be different for different UEs due to both data and system heterogeneity. Therefore, it is necessary to tackle the challenge of heterogeneous local learning model accuracy for the UEs in synchronous FL.

- One approach for handling the communication-computation trade-off in FL is via an appropriate client selection strategy. Selecting the IoT devices with sufficient computing power and training data jointly improves FL model accuracy and training costs [4]. In our previous work [15], we jointly optimized the computing time and energy consumption of FL over wireless networks. The problem studied in [15] captures two trade-offs: UE energy consumption and FL time via variations in device CPU cycles per second, and computational and communication latencies for FL accuracy. However, here, we use a Stackelberg-game-based incentive mechanism to select a set of IoT devices willing to join the model training process. Then the selected set will collaboratively train a global model while minimizing the overall training costs (i.e., computation and communication cost).

Stackelberg Game Solution

The BS employs an incentive mechanism for motivating the set of UEs to participate in global FL model training. However, heterogeneous UEs have different computational and communication costs for training, and thus, they expect different rewards. Moreover, the BS seeks to minimize the learning time while maximizing the accuracy level of the learning model. This complex interaction between the BS and the UEs can be cast as a Stackelberg game with one leader (BS) and multiple followers (UEs). For the offered reward, the BS maximizes its utility modeled as a function of key FL performance metrics such as the number of communication rounds needed to reach a desirable global FL model accuracy level. Correspondingly, the UEs will respond to the BS-offered reward and choose their local iteration strategy (i.e., select a CPU-frequency cycle for local computation) to maximize their own benefits [14]. Evaluating the responses from the UEs, the BS will adjust its reward rate, and the process repeats until a desired accuracy level is obtained. To this end, the BS must design an incentive mechanism to influence available UEs for training the global model. In this framework, the sequence of interactions between the BS and the UEs to reach a Stackelberg equilibrium is as follows:

- Initially, each UE submits its best response (i.e., optimal CPU-frequency) to the BS for the offered reward rate to maximize its local utility. Specifically, each UE considers the viability of the offered reward rate for their incurred computational and communication costs in FL.
- Next, the BS evaluates these responses, updates the global model, and broadcasts its offered reward rate to the UEs to maximize its own utility function. The utility of the BS is modeled as a strictly concave function of key FL performance metrics such as the number of global iterations required to reach global accuracy for a given local relative accuracy.
- Given the optimal offered reward, the UEs will correspondingly tune their strategy and update response that solves their individual utility maximization problem. This iterative process continues in each round of interaction between the BS and UEs.

In summary, we follow the best response algorithm to achieve the Stackelberg equilibrium. For this, with the first-order condition, we first find a unique Nash equilibrium at the lower-level problem (among UEs), and then use a backward induction method to solve the upper-level BS problem.

Performance Evaluation

We now evaluate the performance of our incentive-based FL model by examining the contributions of each FL-participating UE. We investigate the impact of communication channel conditions and local computational characteristics on the accuracy of the global FL model. We evaluate the impact of the offered reward in terms of communication cost vs. local relative accuracy to characterize the system performance in FL. For FL, we adopt a classification task using multinomial logistic regression and distribute the MNIST dataset among participating UEs [1]. For federated optimization, we use the modified CoCoA framework [10]. The distributed federated optimization scheme of [10] allows us to tackle both system-level and statistical heterogeneity efficiently. We consider five participating UEs having different channel conditions and an equal local data size. At each UE, we define the mean square error of the learning problem (i.e., the local relative accuracy metric). For the UEs utility, we choose a concave function of the local relative accuracy and the BS-offered reward.

In Fig. 3a, the impact of the offered reward rate on the relative accuracy for five UEs is shown. The accuracy improves when the relative accuracy value (x-axis) is smaller. Intuitively, an increase in the offered reward rate will motivate UEs to iterate more within one global iteration, resulting in better accuracy. The heterogeneous UE responses is the result of individual computational limitations, local data size, and communication channel conditions. The impact of the communication channel conditions on local relative accuracy for a randomly chosen UE with defined computational characteristics and local data size is illustrated in Fig. 3b. For clarity, we use a normalized communication time to quantify the adversity of channel conditions. Here, a unit value for the normalized communication time signifies poor channel conditions. As the communication time increases, the UEs perform more local iterations to avoid expensive communication costs. Figure 3c presents the relationship between the offered reward rate and the local relative accuracy at the UEs. The offered reward rate reveals the optimal response of the UEs that maximizes their own utilities for given channel conditions. Here, we have consistency in the normalized BS utility function for various response behaviors of the UEs to the offered reward rate. Thus, it is crucial to have an appropriate incentive design to align the responses of the participating UEs for improving the FL performance.

In a typical FL environment, participating user equipment (UE) must iterate over their local (possibly non-i.i.d.) data to train a global model. However, UEs are generally reluctant to participate in FL due to limited computing and communication resources. Thus, enabling FL requires some careful design considerations.

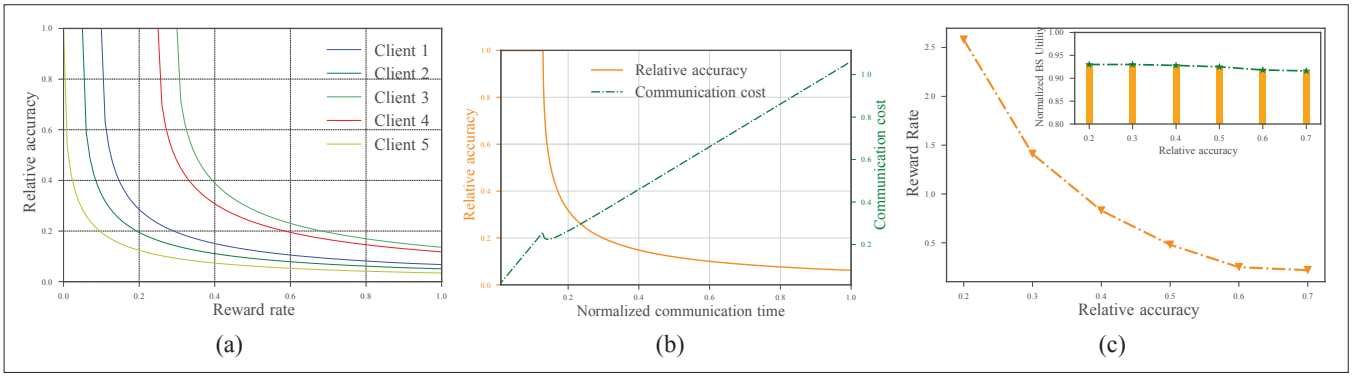

Figure 3. Impact of: a) offered reward rate on client's (UEs) iteration strategy for corresponding relative local accuracy; b) communication time with relative accuracy; c) offered reward rate, normalized BS utility vs. local relative accuracy.

Open Research Challenges

Resource Optimization for Blockchain-Based Federated Learning

An attacker might attack the centralized FL server in order to alter global model parameters. In addition, a malicious user might alter FL parameters during communication. To cope with such security and robustness issues, blockchain-based FL (BFL) can be used. BFL does not require central coordination in the learning of the global model, which yields enhanced robust operation. In BFL, all users send their local model parameters to their associated miners, which are responsible for sharing local model updates through a distributed ledger. Finally, local model updates of all the devices involved in learning are sent back by miners to their associated devices for the local models aggregation. Although BFL provides benefits of security and robustness, it faces a significant challenge of computational and communication resource optimization to reach a consensus among all miners. Static miners can be implemented at the BS, whereas wireless mobile miners can be implemented using drones. However, drone-based mobile miners pose more serious resource allocation challenges than static miners at the BS.

Context-Aware Federated Learning

How does one enable more specialized FL according to users' contextual information? Context awareness is the ability of a device/system to sense, understand, and adapt to its surrounding environment. To enable intelligent context-aware applications, FL is a viable solution. For instance, consider keyboard search suggestion in smartphones in which the use of FL is a promising solution. In this type of design, we must consider context awareness for enhanced performance. A unique globally shared FL model must be used separately for regions with different languages to enable more effective operation. Therefore, the location of the global model must be considered near that region (i.e., micro data center) rather than a central cloud.

Mobility-Aware Federated Learning

How does one enable seamless communication of smart mobile devices with an edge server during the learning phase of a global FL model? Seamless connectivity of the devices with a centralized server during the training phase must be maintained. Mobility of devices must be considered during the device selection phase of FL protocol. Deep-learning-based mobility prediction schemes can be used to ensure the connectivity of devices during FL training.

Conclusions and Future Recommendations

In this article, we have presented the key design aspects, incentive mechanism, and open research challenges for enabling FL in edge networks. We have identified four key design aspects: resource optimization, incentive mechanism, learning algorithm design, and hardware-software co-design-based end devices for FL at the network edge. We have shown that game-theoretic incentive mechanisms can be used to effectively model interaction between devices and edge server for FL. This work can potentially make FL amenable for implementation in diverse 5G-enabled smart IoT applications such as intelligent transportation systems, Industry 4.0, and digital health care. Finally, we present several recommendations for future research:

- Generally, FL involves training of a global FL model via an exchange of learning model updates between a centralized server and geographically distributed devices. However, wireless devices will have heterogeneous energy and processing power (CPU cycles per second) capabilities. Some of the devices might have noisy local datasets. Therefore, there is a need for novel FL protocols that will provide criteria for the selection of a set of local devices having sufficient resources. The selection criteria of the devices must include long-lasting backup power, sufficient memory, accurate data, and higher processing power.
- A set of densely populated devices involved in FL might not be able to have real-time access to the edge server located at the BS due to a lack of communication resources. To cope with this challenge, one can develop new FL protocols based on socially aware device-to-device (D2D) communication. Socially aware D2D communication has an advantage of reusing the occupied bandwidth by other users while protecting them by keeping the interference level below the

maximum allowed limit. Initially, multiple clusters based on social relationships and the distance between devices should be created. Then a cluster head is selected for every cluster based on its highest social relationship with other devices. Within every cluster, a sub-global FL model is trained iteratively by exchanging the model parameters between the cluster head and its associated devices. Then the sub-global FL model parameters from all cluster heads are sent to the BS for global model aggregation. Finally, the global FL parameters are sent back to cluster heads, which in turn disseminate them to their associated cluster devices.

- Exchange of learning model updates via blockchain offers enhanced security. However, reaching consensus via traditional consensus algorithms among blockchain nodes can add more latency to the learning time. Therefore, it is recommended to design novel consensus algorithms with low latency.

Acknowledgments

This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2020R1A4A1018607).

References

[1] H. B. E. Moore *et al.*, "Communication-Efficient Learning of Deep Networks From Decentralized Data," arXiv preprint arXiv:1602.05629, 2016.
[2] K. Guo *et al.*, "Artificial Intelligence-Based Semantic Internet of Things in a User-Centric Smart City," *Sensors*, vol. 18, no. 5, Apr. 2018, p. 1341.
[3] M. Chen *et al.*, "Caching in the Sky: Proactive Deployment of Cache-Enabled Unmanned Aerial Vehicles for Optimized Quality-of-Experience," *IEEE JSAC*, vol. 35, no. 5, May 2017, pp. 1046–61.
[4] W. Y. B. Lim *et al.*, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *IEEE Commun. Surveys & Tutorials*, vol. 22, no. 3, Apr. 2020, pp. 2031–63.
[5] J. Park *et al.*, "Wireless Network Intelligence at the Edge," arXiv preprint arXiv:1812.02858, 2018.
[6] X. Wang *et al.*, "In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning," *IEEE Network*, vol. 33, no. 4, July/Aug. 2019, pp. 156–65.
[7] A. Nilsson *et al.*, "A Performance Evaluation of Federated Learning Algorithms," *Proc. 2nd Wksp. Distributed Infrastructures for Deep Learning*, New York, NY, Dec. 2018, pp. 1–8.
[8] T. Li *et al.*, "Fair Resource Allocation in Federated Learning," arXiv preprint arXiv:1905.10497, 2019.
[9] S. Wang *et al.*, "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," *IEEE JSAC*, vol. 37, no. 6, June 2019, pp. 1205–21.
[10] V. Smith *et al.*, "Federated Multi-Task Learning," *Proc. Advances in Neural Info. Processing Systems 30*, Long Beach, CA, May 2017, pp. 4424–34.
[11] M. Jaggi *et al.*, "Communication-Efficient Distributed Dual Coordinate Ascent," *Advances in Neural Info. Processing Systems*, 2014, pp. 3068–76.
[12] K. Guo *et al.*, "Software-Hardware Codesign for Efficient Neural Network Acceleration," *IEEE Micro*, vol. 37, no. 2, Mar. 2017, pp. 18–25.
[13] Z. Han *et al.*, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*, Cambridge Univ. Press, 2012.
[14] S. R. Pandey *et al.*, "A Crowdsourcing Framework for On-Device Federated Learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, May 2020, pp. 3241–56.
[15] C. Dinh *et al.*, "Federated Learning Over Wireless Networks: Convergence Analysis and Resource Allocation," arXiv preprint arXiv:1910.13067, 2019.

Biographies

Latif U. Khan is pursuing his Ph.D. degree in computer engineering at Kyung Hee University (KHU), South Korea. He received his M.S. (electrical engineering) degree with distinction from the University of Engineering and Technology, Peshawar, Pakistan, in 2017. His research interests include analytical techniques of optimization and game theory to edge computing, federated learning, and end-to-end network slicing.

Shashi Raj Pandey is currently pursuing his Ph.D. degree in the Computer Engineering Department at KHU.. He received his B.E degree in electrical and electronics engineering with specialization in communication from Kathmandu University, Nepal, in 2013. His research interests include network economics, game theory, wireless communications and networking, edge computing, and machine learning.

Nguyen H. Tran [S'10, M'11, SM'18] is currently working as a senior lecturer in the School of Computer Science, University of Sydney. He received his B.S. degree from Hochiminh City University of Technology and his Ph.D. degree from KHU in electrical and computer engineering in 2005 and 2011, respectively. His research interest is in applying analytic techniques of optimization and game theory to cutting-edge applications.

Walid Saad [S'07, M'10, SM'15, F'19] received his Ph.D. degree from the University of Oslo in 2010. Currently, he is a professor in the Department of Electrical and Computer Engineering at Virginia Tech. His research interests include wireless networks, machine learning, game theory, cybersecurity, unmanned aerial vehicles, and cyber-physical systems. He is the author/co-author of eight conference best paper awards and of the 2015 IEEE ComSoc Fred W. Ellersick Prize.

Zhu Han [S'01, M'04, SM'09, F'14] received his Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park. Currently, he is a professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. He is an AAAS Fellow since 2019. He has been a 1 percent highly cited researcher since 2017 according to Web of Science.

Minh N. H. Nguyen is currently pursuing his Ph.D. degree in computer engineering at KHU. He received his B.E. degree in computer science and engineering from the Hochiminh City University of Technology in 2013. His research interests include network optimization, mobile edge computing, and the Internet of Things.

Choong Seon Hong [S'95, M'97, SM'11] is working as a professor with the Department of Computer Science and Engineering, KHU. He is currently an Associate Editor of *Future Internet*, the *International Journal of Network Management*, and the *Journal of Communications and Networks*, and an Associate Technical Editor of *IEEE Communications Magazine*. His research interests include machine learning, edge computing, future Internet, and UAV networks.

We have shown that game-theoretic incentive mechanisms can be used to effectively model interaction between devices and edge server for FL. This work can potentially make FL amenable for implementation in diverse 5G-enabled smart IoT applications such as intelligent transportation systems, Industry 4.0, and digital health care.