This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3010798, IEEE Transactions on Industrial Informatics

Communication-Efficient Federated Learning for Digital Twin Edge Networks in Industrial IoT

Yunlong Lu, *Student Member, IEEE*, Xiaohong Huang, *Member, IEEE*, Ke Zhang, Sabita Maharjan, *Senior Member, IEEE*, and Yan Zhang, *Fellow, IEEE*

***Abstract*—The rapid development of Artificial Intelligence (AI) and 5G paradigm, opens up new possibilities for emerging applications in Industrial Internet of Things (IIoT). However, the large amount of data, the limited resources of IoT devices, and the increasing concerns of data privacy, are major obstacles to improve the quality of services in IIoT. In this paper, we propose the Digital Twin Edge Networks (DITEN) by incorporating digital twin into edge networks to fill the gap between physical systems and digital spaces. We further leverage the federated learning to construct digital twin models of IoT devices based on their running data. Moreover, to mitigate the communication overhead, we propose an asynchronous model update scheme and formulate the federated learning scheme as an optimization problem. We further decompose the problem and solve the subproblems based on the Deep Neural Network (DNN) model. Numerical results show that our proposed federated learning scheme for DITEN improves the communication efficiency and reduces the transmission energy cost.**

***Index Terms*—Communication efficiency, energy cost, federated learning, digital twin, Industrial IoT**

I. Introduction

Emerging technologies such as 5G and edge computing pave the way for the rapid development of Industrial Internet of Things (IIoT) [1]. Devices in IIoT, which are equipped with smart chips and wireless sensors, generate huge volumes of running data. Due to the limited resources, one major issue is how to process and mine these data to improve the quality of services in IIoT.

Recent years have seen the great success of Artificial Intelligence AI, which can be used in IIoT for data analysis and mining. Conventional cloud-based architectures [2] transmit the user data to a cloud server, and executing AI algorithms on the centralized server. However, due to the large amount of transmitted data and long distance between end users and cloud servers, the cloud-based solutions can hardly satisfy the delay requirement of various applications. Mobile Edge Computing (MEC) [3] is proposed to mitigate the communication delay and support latency-critical applications. For example, the computation offloading problem is studied in [4] and [5] to enable the computation-intensive applications. In [6], a deep Q-learning approach was proposed for task offloading in MEC. The radio resource allocation and computation offloading were jointly considered in [7] for the IoT fog computing system. The energy consumption and task processing delay were modeled into a constrained optimization problem by authors in [8] for task offloading in MEC. With the assistant of MEC, new technologies such as blockchain [9] and Deep Reinforcement Learning (DRL) [10] have been widely adopted for optimized resource allocation in IIoT [11]–[14]. For example, in [11], the authors presented a DRL and blockchain empowered resource scheduling framework for IIoT. In [13], the authors formulated the multi-tenant cross-slice radio resource orchestration problem as a multi-agent Markov decision process and leveraged DRL to learn the optimal policies. Besides IoT scenarios, the DRL model has also been widely studied for MEC in smart city [14] and cognitive vehicular communications [15].

However, in Industry 4.0, real-time interactions are required to fill the gap between physical and virtual domains. The stochastic communication latency and the continuous-growing running data in the IIoT network make it hard for MEC servers to perform online optimization by collecting and analyzing the running data such as the Channel State Information (CSI) from IoT devices. Therefore, the paradigm of digital twin is proposed [16] to connect the physical machines with cyber systems for better optimization of the manufacturing processes [17]. As the mapping between physical entities and virtual digital systems, digital twin has been listed by Gartner as one of the most promising technologies in the next decade. Despite the importance of digital twin, there remains a paucity of evidence on the modeling and application of digital twins in wireless networks [18]. The massive data to be synchronized and the limited computing and communication resources hinder the modeling of digital twins in IIoT networks. Moreover, the rising concerns of data privacy and security, raise new challenges for the construction of digital twins.

We leverage federated learning to alleviate the above issues in digital twin modeling. Federated learning [19] enables distributed machine learning over edge devices without collecting their raw data for training. The client users train models locally and only transmit their parameters to the server. With respect to data privacy, there has been growing interest in

This work was partially supported by Joint Funds of National Natural Science Foundation of China and Xinjiang under Project U1603261, and in part by the National Natural Science Foundation of China under Grant 61941102.

Y. Lu, X. Huang are with the Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing, China, (e-mail: yunlong.lu@ieee.org; huangxh@bupt.edu.cn;).

K. Zhang is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, (email:zhangke@uestc.edu.cn).

Sabita Maharjan is with Simula Metropolitan Center for Digital Engineering, 0167 Oslo, Norway, and also with the University of Oslo, 0316 Oslo, Norway (e-mail: sabita@simula.no).

Yan Zhang is with the University of Oslo, 0316 Oslo, Norway, and also with the Simula Metropolitan Center for Digital Engineering, 0167 Oslo, Norway (e-mail: yanzhang@ieee.org).

applying federated learning in wireless networks recently [20]. For example, the authors in [21] exploited federated learning to achieve proactive content caching in MEC scenarios. In [22], the authors proposed to integrate federated learning with blockchain for secure data sharing in industrial IoT, which provides a secure and efficient data sharing scheme for distributed IoT devices. In [23], the authors applied federated learning to wireless networks and provided the optimal solution for balancing the energy consumption and learning time cost. Some works have also considered addressing the security of federated learning, such as gradient leakage [24] and user data privacy [25]. To enhance security and privacy, the authors in [26] proposed to employ Bayesian differential privacy to provide sharper privacy loss bounds in federated learning.

However, a critical challenge posed to federated learning in IIoT is the communication overhead. With the development of AI techniques, machine learning models such as Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) are becoming more and more complex, which leads to growing size of model parameters [27]. Moreover, the training in federated learning is an iterative process, which requires frequent parameter exchange between clients and the server. Therefore, to apply federated learning in resource-constrained IIoT scenarios, reducing the communication overhead is a fundamental problem that requires in-depth investigation. Some works have explored to mitigate resource consumptions in federated learning by reducing the parameter transmission load between the server and clients. In [28], the authors proposed a control algorithm to determine the best trade-off between local update and global parameter aggregation under a given resource budget. In [29], the authors addressed the issue of inefficient training caused by client with limited resources and selected clients based on their resource conditions. However, directly skipping the global aggregations may degrade the quality and convergence of the global models. In the learning process, the allocation of computation and communication resources, the model update scheme of users, require to be jointly considered to improve the communication efficiency of federated learning.

In this paper, we first propose the architecture of Digital Twin Edge Networks (DITEN), by integrating digital twin with edge computing to establish an efficient mapping between IoT devices and cyber systems. Then, we adopt federated learning to build the models in DITEN. We further present the federated learning system model and formulate the problem of reducing communication costs to an optimization problem. The problem is then decomposed into two subproblems and solved by a Deep Neural Network (DNN) model. The main contributions of this paper are summarized as follows.

- We propose the architecture of DITEN, which integrates digital twin with edge computing to make efficient and appropriate optimization of the IIoT networks.
- We leverage the federated learning scheme to construct the DITEN models, which can reduce the data transmission overhead and protect data privacy. Furthermore, to improve the communication efficiency, we propose an asynchronous model update scheme, and formulate the problem of reducing communication cost to an optimization problem.
- We solve the communication cost optimization problem by decomposing it into two subproblems and determine the optimal strategies for allocating the communication resources based on a deep neural network model.

II. System Model

In this section, we introduce the model of our digital twin edge network in IIoT. The communication model and computation model are also presented to formulate the federated learning problem.

A. Digital Twin Edge Network Model

Fig. 1 shows the architecture of our DITEN. There are mainly three layers in our framework: user layer, edge layer, and digital twin layer. The user layer consists of client devices in IIoT such as smart machines, vehicles, and IoT devices. The client users are denoted by $\mathcal{U} = \{u_1, u_2, ..., u_N\}$. Each client device generates and holds the local dataset denoted by $\mathcal{D}_i$, with size D_i. The edge layer is composed of Base Stations (BSs) that are equipped with MEC servers. The BSs, which are indexed by $\mathcal{B} = \{B_1, B_2, ..., B_M\}$, connect with user devices under their coverage via wireless communications. The digital twins are constructed in the BSs. For a client user u_i, its digital twin DT_i is modeled in its nearby BS based on the local data of u_i. As shown in Eq. (1), $DT_i(t)$ consists of its model $\mathcal{M}_i$, historical data $\mathcal{D}_i$, running state s_i and interaction state data Δs_i from u_i.

$$DT_i(t) = \Gamma(\mathcal{M}_i, \mathcal{D}_i, s_i, \Delta s_i, t). \tag{1}$$

The DT_i collects the running data from the user device and constructs a model $\mathcal{M}_i$ of u_i. Moreover, DT_i continuously interact with u_i to keep their consistency. The digital twins may also interact with each other, which forms the digital twin edge network. The virtual network can reflect the real network states in the user layer.

Our proposed DITEN constructs a mapping scheme that connects the physical devices in the user layer with the digital systems in the edge layer. Based on the DITEN, further network optimization and resource allocation strategies can be explored in DITEN and be implemented to the real networks.

We leverage federated learning to model digital twins in our DITEN. The goal of federated learning is to establish a machine learning model $\mathcal{M}$ for DITEN, which can give reaction to the states based on the rules and states of real devices. Denote the loss function as $f(w)$, which quantifies the difference between estimated and true values for instances of running data $\mathcal{D}_i$. The loss function $F_i(w)$ of user u_i on dataset $\mathcal{D}_i$ is defined as:

$$F_i(w) = \frac{1}{D_i} \sum_{x_j, y_j \in \mathcal{D}_i} f(w, x_j, y_j), \tag{2}$$

where x_j, y_i is the samples of training data. The aggregated loss function is

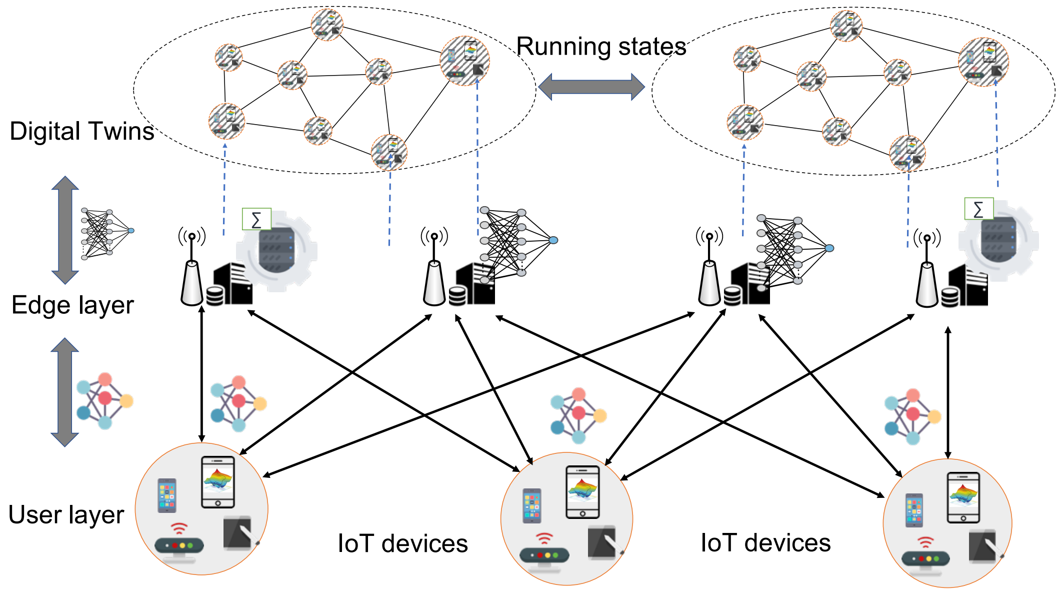


Fig. 1: Digital Twin Edge Networks for IIoT

$$F_g(w) = \frac{1}{D_g}\sum_{i=1}^{N} F_i(w), \tag{3}$$

where $D_g = \sum_{i=1}^{N} \mathcal{D}_i$ is the total size of data from participating users. The goal of federated learning is to minimize the global loss:

$$\min_{w \in \mathbb{R}_d} F_g(w) \tag{4}$$

In practical scenarios, due to limited computation and communication resources and storage capabilities, we can select the data for some specific applications that are most useful to build digital twins. For example, in the traffic prediction scenario, the digital twin models can be constructed only from the traffic data.

B. Communication Model for DITEN

We consider the finite state model to capture the states of wireless channels in our DITEN. The channel state is obtained based on the received signal-to-noise ratio (SNR). The possible values of SNR are partitioned into K intervals as $\mathcal{H} = \{H_1, H_2, ..., H_K\}$. The channel state at time period t is denoted as $\zeta(t) = H_k$, which can be considered to be constant during period t, and varies in different periods. The transition probability from channel state $\zeta(t) = H_{k1}$ to channel state $\zeta(t) = H_{k2}$ is derived according to Markov transition probabilities.

We consider the multicast Orthogonal Frequency Division Multiple access (OFDMA) protocol between end users and BSs in our DITEN. Assume that there are C_0 sub-channels over the whole bandwidth W. The number of sub-channels allocated to user u_i is c_{u_i}, and allocated to BS B_j is c_{B_j}. The achievable data rate of user u_i at BS B_n is

$$r_{u_i,B_j} = \frac{c_{u_i}}{C_0} W log(1 + \zeta_{u_i,B_j}). \tag{5}$$

The wireless spectrum allocated to all users and BSs should not exceed the total bandwidth, which follows the constraint:

$$\sum_{i=1}^{N} c_{u_i} + \sum_{j=1}^{M} c_{B_j} \leq C_0, \tag{6}$$

where $u_i \in \mathcal{U}$ and $B_j \in \mathcal{B}$.

C. Federated Learning Empowered Computation Model for DITEN

All the participating users share the same machine learning model $w(t-1)$ obtained from the BS at the beginning of iteration t. Each u_i then trains the model $w(t)$ based on its local data $\mathcal{D}_i$, denoted as:

$$w_i(t) = w(t-1) - \eta \nabla F_i(w(t-1)) \tag{7}$$

The trained model parameters w_i, and the running state s_i, are transmitted to the nearby BS by user u_i. The BS, which also act as the aggregator, collects the parameters from participating users and aggregates them into a global model, as:

$$w(t) = \frac{1}{D_g}\sum_{i=1}^{N} D_i w_i(t), \tag{8}$$

where $\eta > 0$ is the learning step. The process is repeated until the global model parameters $w(t)$ achieve the minimum global loss in Eq. (3).

For user u_i, we denote the number of CPU cycles needed to execute one unit of data by ξ_i, which is decided by the CPU chips of u_i. The CPU-cycle frequency of user u_i is denoted as f_{u_i}. Thus, the CPU energy consumption of user u_i in one iteration is

$$E_{u_i}^{cmp} = \alpha \xi_i D_i f_{u_i}^2, \tag{9}$$

where α is the energy consumption coefficient of user i's device. The computation time of user u_i is

$$T_{u_i}^{cmp} = \frac{\xi_i D_i}{f_{u_i}}, \tag{10}$$

In the transmission phase of user u_i, the transmission time is

$$T_{u_i}^{com} = \frac{|w_i(t)|}{r_i}, \tag{11}$$

where $|w_i|$ is the size of transmitted model parameters of u_i. The energy consumption of u_i's transmission is

$$E_{u_i}^{com} = \frac{\beta P_i |w_i|}{r_i}, \tag{12}$$

where β is u_i's energy consumption coefficient for transmission.

The users apply the standard gradient descent method to train their local models with the learning rate η as in Eq. (7). We consider that the gradient $\nabla F(w_g)$ is uniformly Lipschitz continuous for a positive constant L, that is

$$||\nabla f(w_{t+1}) - \nabla f(w_t)|| \leq L||w_{t+1} - w_t||. \tag{13}$$

We also consider that $F(w)$ is strongly convex with parameter μ and is twice-continuously differentiable. According to [30], we can obtain

$$F(w_{t+1}) \leq F(w_t) - \frac{1}{2L}||\nabla F(w_t)||^2 \tag{14}$$

Since $F(w)$ is strongly convex , for any w, we can also obtain

$$F(w_*) \geq F(w) - \frac{1}{2\mu}||\nabla F(w)||^2 \tag{15}$$

In the case $w = w_t$,

$$||\nabla F(w)||^2 \geq 2\mu[F(w_t) - F(w_*)]. \tag{16}$$

By subtracting $F(w_*)$ from both sides of Eq.(14) and (16), we obtain the upper bound of $\mathbb{E}[F(w(t+1)) - F(w^*)]$

$$\mathbb{E}[F(w(t+1)) - F(w^*)] \leq (1 - \frac{\mu}{L})\mathbb{E}[F(w(t)) - F(w^*)]. \tag{17}$$

Thus, the federated learning algorithm converges to the optimal global model in the training process.

Since the BSs in our system have sufficient power resources, the energy consumption of BSs has little effect on our system. Thus we only consider the energy consumption of end users in our scheme. In addition, since the downlink bandwidth is much larger than the uplink, we also do not consider the downlink time.

The proposed federated learning scheme can also be applied to multi-BS scenario, where the MBS (or the BS) collects all parameters from various BSs and aggregates them to a global model. In the multi-BS federated learning scenario, the communication between BSs and the MBS can also be modeled by the communication model in our scheme. The achievable data rate of BS B_{j_1} to BS B_{j_2} is

$$r_{B_{j_1},B_{j_2}} = \frac{c_{B_{j_1}}}{C_0} W log(1 + \zeta_{B_{j_1},B_{j_2}}). \tag{18}$$

The increased communication cost is

$$\sum_{m=1}^{M} \frac{|w_m|}{r_{B_m,B_s}} \tag{19}$$

where M is the number of BSs, r_{B_m,B_s} is the achievable data rate between BS B_m and the MBS B_s.

III. Communication-Efficient Federated Learning for DITEN

To reduce the communication overhead in our DITEN, we propose a communication-efficient federated learning scheme in this section.

The processes of conventional federated learning are shown in Fig. 2. For each participating user, there are two processes: local training and parameter transmission. Due to the heterogeneous computation and communication resources, the execution time varies in different users. The aggregator, which is a BS with the MEC server, waits for all the users to finish their execution, including the slowest one. The dynamic execution time of different users straggles the whole synchronous scheme.

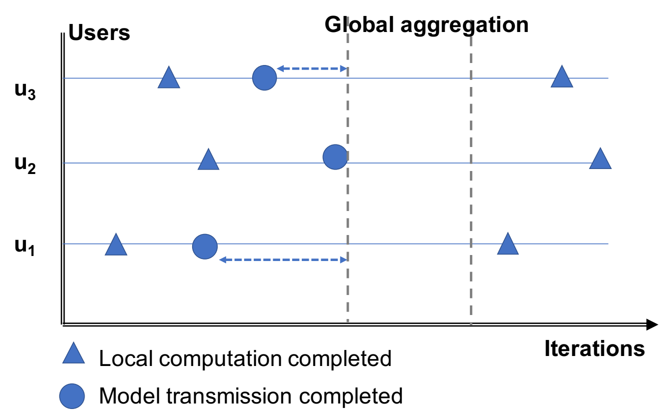


Fig. 2: Conventional federated learning processes

Towards the above issue, we improve communication efficiency and reduce overall energy cost from the following aspects:

- Decreasing the size of overall parameter data transmitted to the aggregator to improve update efficiency;
- Improving communication efficiency by optimizing the allocation of communication resources.

A. Asynchronous Weighted Model Update

We propose to use asynchronous model update scheme to reduce the size of transmitted data and improve transmission efficiency. The transmission overhead is calculated as:

$$\phi = \sum_{t=1}^{T} \sum_{i=1}^{N} |w_i(t)|, \tag{20}$$

where T is the total rounds, N is the number of total participants, $|w_i(t)|$ is the size of user i's updating model in iteration t. From Eq. (20) we can see that the communication overhead is decided by the size of model parameters, the rounds of communications, and the number of participating users. The $|w_i(t)|$ can be reduced by data compression or reconstruction [31], [32].

In our DITEN, the update includes the device state $s(t)$ and model parameters $w_i(t)$ of federated learning. The model parameters are used to build the global model, while the device running states $s(t)$ are used to synchronize the digital twins. In the gradient descent based algorithms, the model training process usually converges smoothly. For some participants, they do not need to frequently update their model parameters to the BS. Instead, they only need to update their running state to the digital twins in BSs, which can reduce the overhead of total transmission. We define the update weight $\tau_i(t)$ in iteration t as:

$$\tau_i(t) = (t - t_l) \cdot (\frac{||w_i(t) - w_g(t-1)||}{w_g(t-1)})\frac{r_i(t)}{f_{u_i}}, \tag{21}$$

where $t - t_l$ is the round gap of u_i since its last global aggregation, $||w_i(t) - w_g(t-1)||$ is the absolute model difference between u_i's local model and global model, $\frac{r_i(t)}{f_{u_i}}$ is u_i's ratio of communication capability to computation capability. The users with high $\tau_i(t)$ that exceeds a predefined value should participate in the global aggregation process in iteration t. Otherwise, the users only transmit its current state to update the digital twins, and continue to execute the local training process.

In the edge layer, the BS aggregates the local models it received according to Eq. (22)

$$w_g(t+1) = \sum_{k=1}^{K} \frac{D_k}{\sum_{k=1}^{K} D_k}(1 - e^{-\tau_i(t)}) \cdot w_k(t), \tag{22}$$

where $k \in \{1, ..., K\}$ denotes the k-th user that participants in the global aggregation.

The complete processes of our asynchronous weighted model update scheme is shown in Algorithm 1.

Algorithm 1 Asynchronous Weighted Model Update

Input: Last global update round t_l, achievable data rate $r_i(t)$, achievable CPU frequency f_{u_i}

Initialize the global model M_0
for each iteration t **do**
  for each user $u_i \in U$ **do**
    u_i trains its local model on its data D_i
    u_i calculates its update weight $\tau_i(t)$ according to Eq. (21)
    If $\tau_i(t) > \tau_0$, u_i updates its local model for global aggregation. Otherwise, u_i continues its local training
  end for
  BS B_j collects all running states and updates the digital twins
  B_j aggregates the local models it received according to Eq. (22)
  B_j broadcasts the new global model $w_g(t)$ to users
end for

B. Problem Formulation

To improve communication efficiency, we allocate the communication resources in our DITEN according to the system states such as the current computing capabilities and the channel state information of IoT devices.

We adaptively allocate our communication resources to participating users, to mitigate the imbalance of communication performance. As shown in Fig. 3, the "straggler" users such as u_2 are assigned with more communication resources, while the "fast" users such as u_1 and u_3 are assigned with less communication resources. The aggregation data rate of the

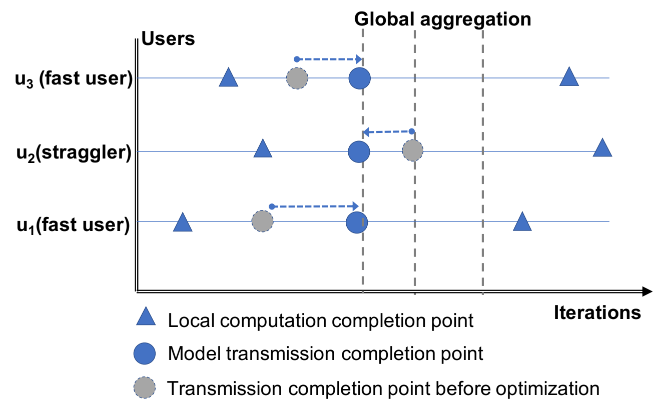


Fig. 3: Optimized federated learning processes

i-th user is

$$R_i = \sum_{k=1}^{C_0} \theta_{i,k} r_{i,k}, \tag{23}$$

where $\theta_{i,k} \in \{0, 1\}$ represents whether the n-th subcarrier is assigned to user i. $\theta_{i,k} = 1$ indicates that subcarrier n is allocated to user i. Otherwise, $\theta_{i,k} = 0$. The $\theta_{i,k}$ satisfies

$$\sum_{i \in \mathcal{N}} \theta_{i,k} \leq 1, \forall k \in C_0, \tag{24}$$

which denotes that one subcarrier can only assigned to at most one user. The transmit power of the user devices are also limited, which satisfies

$$\sum_{k \in \mathcal{C}_0} P_{i,k} \leq P_i^{max} \tag{25}$$

The expected average execution time of users in iteration t is:

$$T_{ave}(t) = \frac{1}{N} \sum_{i=1}^{N} (T_i^{cmp}(t) + T_i^{com}(t)) \tag{26}$$

We define the time variance to quantify the execution imbalance between participants, as:

$$T_{var}(t) = \frac{\sum_{i=1}^{N} |T_i(t) - T_{ave}(t)|}{N} \tag{27}$$

Base on the above models, we now formulate an optimization problem to minimize the communication cost in federated learning process. We optimize the resource allocation for participating users in federated learning. The minimization problem is:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3010798, IEEE Transactions on Industrial Informatics

$$\min_{\boldsymbol{f},\boldsymbol{\lambda},\boldsymbol{\theta}} \frac{1}{\sum_{i=1}^{N}\lambda_i}\sum_{i=1}^{N}\lambda_i(T_i(\theta,t)-T_{ave}(\boldsymbol{\lambda},\boldsymbol{\theta},t))^2 \quad (28)$$

$$\text{s.t.} \quad \lambda_i, \theta_i^k \in \{0,1\}, \forall i \in \mathbb{N}, \quad (28a)$$

$$\sum_{i\in\mathbb{N},k\in\mathbb{C}_0} \theta_{i,k} \le C_0, \quad (28b)$$

$$T_{ave}(\boldsymbol{\lambda},\boldsymbol{\theta},t) \le T_{thd}, \forall i \in \mathbb{N}, \quad (28c)$$

$$f_i^{min} \le f_i \le f_i^{max}, \forall i \in \mathbb{N}, \quad (28d)$$

$$P_i^{min} \le P_i \le P_i^{max}, \forall i \in \mathbb{N}, \quad (28e)$$

$$E_i(r_i, P_i) \le E_{thd}, \quad (28f)$$

where $\lambda_i(t)$ is the index that denotes whether user i participates in the t-th global aggregation of federated learning. If $\tau_i(t) \ge \tau_0$, $\lambda_i(t) = 1$. Otherwise, $\lambda_i(t) = 0$. T_{thd} is the execution time threshold. The execution time of u_i is given by:

$$T_i(\theta, P_i, t) = \frac{\xi_i D_i}{f_{u_i}} + \frac{|w_i(t)|}{\sum_{k=1}^{C_0}\theta_{i,k} r_{i,k}} \quad (29)$$

The average execution time of participating users in iteration t is:

$$T_{ave}(\boldsymbol{\lambda},\boldsymbol{\theta},\boldsymbol{P},t) = \frac{1}{N}\sum_{i=1}^{N}\lambda_i \cdot (\frac{\xi_i D_i}{f_{u_i}} + \frac{|w_i(t)|}{\sum_{k=1}^{C_0}\theta_{i,k} r_{i,k}}) \quad (30)$$

Constraint (28a) denotes that the allocated bandwidth should not exceed the total bandwidth. (28d) and (28e) are the maximum CPU frequency and transmission power of users. Constraint (28f) is the energy consumption limit of users, where E_{thd} is decided by the power supply of user devices.

IV. Optimization of Communication Resources for Federated Learning

Since it is hard to obtain the T_{ave} in advance, we leverage digital twins to estimate the expected execution time of T_{ave}. At the beginning of each iteration t, the states $s(t)$ of each IoT devices are mapped to the digital twins. We calculate the expected execution time of each user device according to Eq. (30) based on the communication model and computation model in Section II.

In problem (28), the optimization variables can be divided into two phases: f_i in the computation phase and $\theta_{i,k}$, $\lambda(\tau_i)$ in the communication phase. In the computation phase, the variable f_i of user devices determines the energy cost and computation time. Thus, the objective of this phase is to jointly minimize the energy cost and computation time:

$$\min_{\boldsymbol{f_i}} \sum_{i=1}^{N} E_n^{cmp}(f_i) + \gamma T^{cmp} \quad (31)$$

$$\text{s.t.} \quad \frac{\xi_i D_i}{f_i} \le T_{cmp}, \forall i \in \mathbb{N}, \quad (32)$$

$$f_i^{min} \le f_i \le f_i^{max}, \forall i \in \mathbb{N} \quad (33)$$

The balance requirement between the energy consumption and the time cost is determined by the factor γ. Problem (31) is a CPU-cycle control problem that can be solved by categorizing users into three groups according to their $T_{u_i}^{cmp}$. The optimal solution is:

$$f(x) = \begin{cases} f_n^{max} & \forall i \in \mathcal{N}_1, \\ f_n^{min} & \forall i \in \mathcal{N}_2, \\ \frac{\xi_i D_i}{T_{cmp}^*} & \forall i \in \mathcal{N}_3, \end{cases} \quad (34)$$

$$T_{cmp}^* = max\{T_{\mathcal{N}_1}, T_{\mathcal{N}_2}, T_{\mathcal{N}_3}\}, \quad (35)$$

where $\mathcal{N}_1$ is the "bottleneck" user group that always run its maximum frequency, $\mathcal{N}_2$ is the "strong" group which run minimum frequency, $\mathcal{N}_3$ is the user group having the optimal frequency. $\mathcal{N}_1$, $\mathcal{N}_2$, $\mathcal{N}_3$ are produced based on their computation time $T_{u_i}^{cmp}$ and thresholds $T_{\mathcal{N}_1}$,$T_{\mathcal{N}_2}$,$T_{\mathcal{N}_3}$ [33].

In the communication phase, the computation time of each participant is fixed, denoted as $T^{cmp}(t) = \{T_1^{cmp}, ..., T_N^{cmp}\}, \forall i \in \mathcal{N}$. The optimization problem is:

$$\min_{\boldsymbol{\lambda},\boldsymbol{\theta}} \sum_{i=1}^{N}\lambda_i(T_i^{com}(\theta, P_i, t) + T_i^{cmp} - T_{ave})^2 \quad (36)$$

$$\text{s.t.} \quad \lambda_i, \theta_i^k \in \{0,1\}, \forall i \in \mathbb{N}, \quad (36a)$$

$$T_{ave} \le T_{thd}, \quad (36b)$$

$$\sum_{i\in\mathbb{N},k\in\mathbb{C}_0} \theta_{i,k} \le C_0, \quad (36c)$$

$$P_i^{min} \le P_i \le P_i^{max}, \forall i \in \mathbb{N}, \quad (36d)$$

$$E_i(r_i, P_i) \le E_{thd}, \quad (36e)$$

Conventional optimization methods require the optimization problem to satisfy strict assumptions and constraints. The performance of conventional optimization methods greatly relies on choosing the correct threshold, which is a challenging task for dynamic scenarios. To improve the system flexibility and robustness, we propose the digital twin empowered deep neural network to find the optimal solution. The trained DNN model can be used for online optimization, and can deal with dynamic system states. The states $s(t)$ are the input of DNN Θ, which is defined as:

$$\boldsymbol{s}(t) = \{\boldsymbol{T_i^{cmp}}, \boldsymbol{\theta_{t-1}}, \boldsymbol{\zeta}, \boldsymbol{\lambda}\}. \quad (37)$$

The output vector $\boldsymbol{\lambda}$ is calculated according to Eq. (38)

$$\boldsymbol{\theta}_t = \sum_{l=1}^{L} f(\boldsymbol{W}^{[l]}\boldsymbol{X}^{[l]} + \boldsymbol{b}^{[l]}). \quad (38)$$

We propose a heuristic algorithm to solve problem (36). The scheme consists of three phases: initialization, exploration, and training.

- *Initialization:* since the communication time of i-th user only depends on its own transmitted data and the subcarriers it occupied, we iteratively assign the subcarriers to participating users to find the best allocation policy. Note that the number of participating users can be controlled by Algorithm 1. We consider the number of subchannels is larger than the number of participating users. The initial allocation strategy is determined as follows. First, we rank the subchannels according to their current channel states. The good channels are assigned to the users with high

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3010798, IEEE Transactions on Industrial Informatics

T_i^{cmp}, and each user is assigned with one subchannel. Then, we calculate the current $T_i(\theta, P_i, t)$ of each user and assign the good channels to the users with high $T_i(\theta, P_i, t)$.

- *Exploration:* We then explore the possible allocation policies to validate their performance. The estimated average time cost T_{ave} is calculated first. For user u_i that $T_i^{cmp} + T_i^{com} > T_{ave}$, more subcarriers are allocated to u_i. Otherwise, if $T_i^{cmp} + T_i^{com} < T_{ave}$, fewer subcarriers are allocated to u_i. We iteratively change the assignment of subcarriers, while the others remain the same as in the initialized states, to obtain the new allocation policy $\boldsymbol{\theta}$. We calculate the object value $\Psi(s, \theta)$ based on the system parameters in digital twin according to Eq. (36).
- *Training:* According to the values of $\Psi(s, \theta)$, we choose the one that minimizes the objective function in Eq. (36), and save the tuple $\{s(t), \theta_t\}$ into training buffer memory. We adopt the experience replay technique to train the DNN model. The trained DNN model can generate the optimal communication resource allocation strategy towards various system states in real-time.

The complete communication resource optimization process is shown in Algorithm 2.

Algorithm 2 Communication Resource Optimization

Input: Computation time of each participant $T_i^{cmp}(t)$, size of trained models $|w_i(t)|$, channel state information $\zeta(t)$

Initialize the original allocation strategy $\boldsymbol{\theta}_0$
Calculate the current object value $\Psi(s, \theta)$ according to Eq. (36), where s is obtained from the digital twins in edge layer
for each iteration t **do**
  for each $\boldsymbol{\theta}_j$ **do**
    Run the exploration and obtain the new allocation strategy as $\boldsymbol{\theta}_j$
    Calculate the current object value $\Psi(s, \theta)$ according to Eq. (36) based on states of digital twins
    Record the states s, allocation strategy, and the $\Psi(s, \theta_j)$
  end for
  Calculate the best allocation strategy $\Psi(s(t), \theta(t))$
  Store the states and optimal allocation strategy $\Psi(s(t), \theta(t))$
end for
Train the DNN model with stored samples (s_t, θ_t)

Since the DNN model can make real-time decisions, we mainly consider the complexity in the training process. For a given system state s, we denote the size of subchannels as K and the number of participants as N. Each subchannel can be assigned to one of K users, while every user should occupy one subchannel. Thus, there are $N + (N-1) + ... + (N-K) + K * (K-N)$ types of allocation strategies. Thus, the complexity to find the optimal strategy is $\mathcal{O}(K * (K-N))$. Due to the dynamic network states, the computation time T_i^{cmp} and the update factor λ_i vary in each iteration. The dynamic state data s can be obtained by interacting with the digital twin networks in each iteration. For T iterations, the total complexity for obtaining the training samples is $\mathcal{O}(KT(K-N))$. The DNN model is trained on the data samples off-line and updated periodically with the change of network states. The trained DNN model is then used to generate real-time resource allocation policies for improving the communication efficiency of federated learning.

V. Numerical Results

We conduct evaluations of the proposed communication-efficient federated learning on the real-world dataset MNIST [34]. The MNIST training set contains 60,000 examples and the testing set contains 10,000 examples. The learning on the dataset is similar to the image recognition applications in IIoT such as autonomous driving and intelligent camera. In our federated learning, we take the CNN model [35] as the initial model to be trained. We also compare our asynchronous model update scheme with the conventional synchronous update scheme. We set the number of end users to 100 in our evaluation, while their device states are randomly derived from different Gaussian distributions. The example network scenario is shown in Fig. 4. We mainly consider the application scenario with single BS as the aggregation server. The proposed scheme can also be extended to the application scenarios with multiple BSs and one MBS as the aggregation server. Compared with single-BS scenarios, applying the proposed scheme to multi-BS scenarios will slightly increase the communication cost between BSs and the MBS to transmit model parameters. We shuffle the 60,000 training samples and distribute them to the 100 users, where the number of samples on each user follows $N(\mu = 600, \sigma = 200)$.

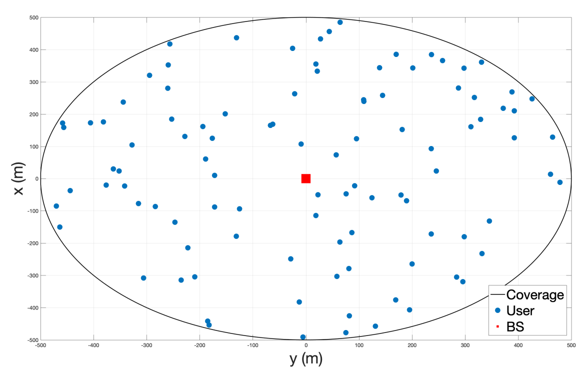


Fig. 4: The example network scenario

We evaluate the accuracy of our proposed federated learning with different numbers of participants on the MNIST dataset. We set the number of participants to 30, 50, 70, 90 by adjusting the weight threshold for participating in the update process. The accuracy quantifies the ratio of correctly predicted samples by the learned global model. From Fig. 5 we can see that the learning scheme converges to high accuracies after 20 iterations. The final accuracy results show that the accuracy increases with the number of participants. However, the performance gap between groups with different numbers of participants is small, which shows the good scalability of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3010798, IEEE Transactions on Industrial Informatics

the proposed federated learning scheme. The results of training loss in Fig. 6 further confirm the good accuracy and scalability as in Fig. 5, where the loss results in different groups with different numbers of participants converge to similar small values.

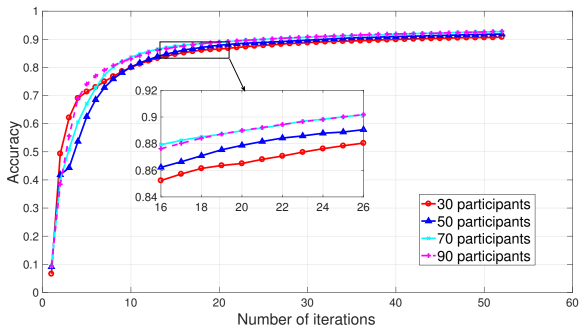


Fig. 5: The accuracy of our proposed scheme with different numbers of participants

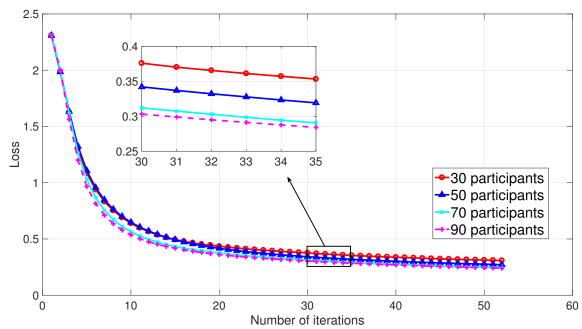


Fig. 6: The loss of proposed scheme with different numbers of participants

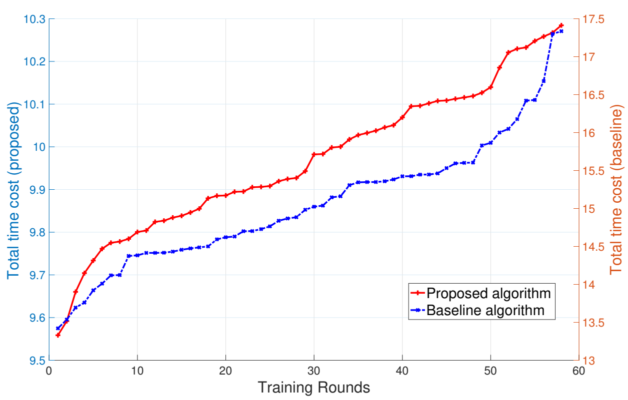


Fig. 7: The accumulative time cost in the training process

The accumulative time cost results of our proposed communication resource optimized update scheme, and the conventional synchronous update scheme, are shown in Fig. 7. Compared with the baseline algorithm, our proposed method achieves better time cost results. The accumulative time cost of our proposed method increases much slower with the training round than the baseline algorithm. The reason is that the execution time of the synchronous update scheme is determined by the slowest user with poor computing and communication capability. The results show that the proposed scheme reduces the overall communication time cost in the model update process, which can further reduce the overall energy consumption in communication.

Fig. 7 shows the comparing time cost with different numbers of participants. The time cost of our proposed method increases slightly with the number of participants, while the time cost of the baseline algorithm increases considerably with the participation of new users. Since the time cost also denotes the global consumption of computation and communication resources, the results indicate that the allocation of resources can reduce the average resource consumption. The loss values and inference accuracy of our proposed DNN model in the training process are provided in Fig. 9. From Fig. 9 we can see that the DNN model can be trained in dozens of rounds with high inference accuracy. Therefore, the trained DNN model can be used in real-time optimization of communication resources in the federated learning process to reduce overall energy consumption.

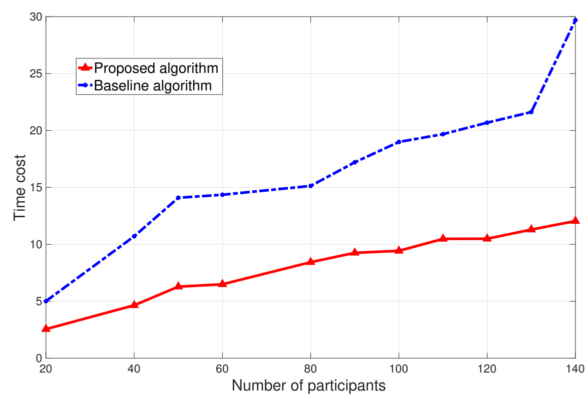


Fig. 8: The comparing cost with various numbers of participants

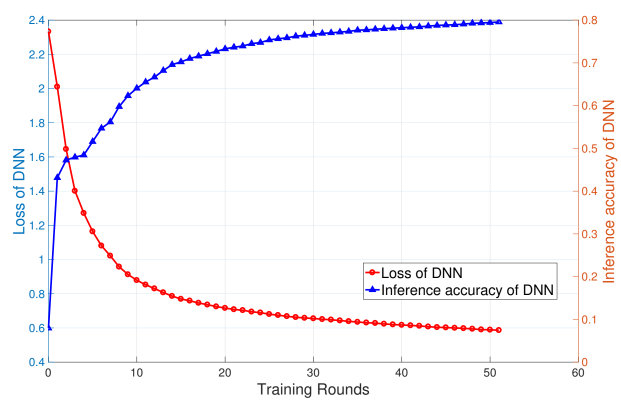


Fig. 9: The training loss and inference accuracy of our proposed DNN model

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3010798, IEEE Transactions on Industrial Informatics

VI. Conclusion

In this paper, we proposed the architecture of DITEN, which incorporates digital twins into edge networks for real-time data analysis and network resource optimization. To model digital twins, we used federated learning to build digital twins from the historical running data of devices. The raw data transmission is avoided and data privacy is enhanced in federated learning. We then formulated an optimization problem that aims at reducing the communication cost of federated learning, and provided the solution by decomposing it and using DNN for communication resource allocation. Numerical results on the benchmark real-world dataset corroborated that our proposed mechanism can improve the communication efficiency and reduce the overall energy cost.

References

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[2] S. E. Mahmoodi, R. N. Uma, and K. P. Subbalakshmi, "Optimal joint scheduling and cloud offloading for mobile applications," *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 301–313, 2019.

[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[4] B. Yang, X. Cao, J. Bassey, X. Li, and L. Qian, "Computation offloading in multi-access edge computing: A multi-task learning approach," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[5] Z. Zhao, R. Zhao, J. Xia, X. Lei, D. Li, C. Yuen, and L. Fan, "A novel framework of three-hierarchical offloading optimization for mec in industrial iot networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5424–5434, 2020.

[6] K. Zhang, Y. Zhu, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Deep learning empowered task offloading for mobile edge computing in urban informatics," *IEEE Internet of Things Journal*, 2019.

[7] Z. Chang, L. Liu, X. Guo, and Q. Sheng, "Dynamic resource allocation and computation offloading for iot fog computing system," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.

[8] A. Bozorgchenani, F. Mashhadi, D. Tarchi, and S. Salinas Monroy, "Multi-objective computation sharing in energy and delay constrained mobile edge computing environments," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[9] Z. Chang, W. Guo, X. Guo, Z. Zhou, and T. Ristaniemi, "Incentive mechanism for edge computing-based blockchain," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.

[10] Y. Dai, D. Xu, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2020.

[11] K. Zhang, Y. Zhu, S. Maharjan, and Y. Zhang, "Edge intelligence and blockchain empowered 5g beyond for the industrial internet of things," *IEEE Network*, vol. 33, no. 5, pp. 12–19, 2019.

[12] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, 2019.

[13] X. Chen, Z. Zhao, C. Wu, M. Bennis, H. Liu, Y. Ji, and H. Zhang, "Multi-tenant cross-slice resource orchestration: A deep reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2377–2392, 2019.

[14] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J. Oh, "Semisupervised deep reinforcement learning in support of iot and smart city services," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 624–635, 2018.

[15] K. Zhang, S. Leng, X. Peng, P. Li, S. Maharjan, and Y. Zhang, "Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks," *IEEE Internet of Things*, to be published.

[16] E. Glaessgen and D. Stargel, *The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles*. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2012-1818

[17] Y. Ma, H. Zhou, H. He, G. Jiao, and S. Wei, "A digital twin-based approach for quality control and optimization of complex product assembly," in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, 2019, pp. 762–767.

[18] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.

[19] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[20] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[21] Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas, "Federated learning based proactive content caching in edge computing," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[22] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.

[23] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019, pp. 1387–1395.

[24] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 747–14 756.

[25] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2512–2520.

[26] A. Triastcyn and B. Faltings, "Federated learning with bayesian differential privacy," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 2587–2596.

[27] C. M. Gamanayake, L. A. Jayasinghe, B. Ng, and C. Yuen, "Cluster pruning: An efficient filter pruning method for edge ai vision applications," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–1, 2020.

[28] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, 2019.

[29] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.

[30] M. P. Friedlander and M. Schmidt, "Hybrid deterministic-stochastic methods for data fitting," *SIAM Journal on Scientific Computing*, vol. 34, no. 3, pp. A1380–A1405, 2012.

[31] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright, "Atomo: Communication-efficient learning via atomic sparsification," in *Advances in Neural Information Processing Systems*, 2018, pp. 9850–9861.

[32] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," *arXiv preprint arXiv:1812.07210*, 2018.

[33] C. Dinh, N. H. Tran, M. N. H. Nguyen, C. S. Hong, W. Bao, A. Y. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," 2019.

[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[35] "Convolutional neural network," https://www.tensorflow.org/tutorials/images/cnn, 2019.

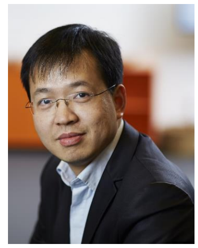
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3010798, IEEE Transactions on Industrial Informatics

Yunlong Lu received the M.S degree in School of Computer Science from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2015, where he is currently working towards the Ph.D. degree in Computer Science and Technology. His current research interests include blockchain, wireless networks, and privacy-preserving machine learning.

Xiaohong Huang received her B.E. degree from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2000 and Ph.D. degree from the school of Electrical and Electronic Engineering (EEE), Nanyang Technological University, Singapore in 2005. Since 2005, Dr. Huang has joined BUPT and now she is a professor and director of Network and Information Center in Institute of Network Technology of BUPT. Dr. Huang has published more than 50 academic papers in the area of WDM optical networks, IP networks and other related fields. Her current interests are Internet architecture, software defined networking, and network function virtualization.

Ke Zhang received the Ph.D. degree in University of Electronic Science and Technology of China, 2017. He is currently a lecturer in the School of Information and Communication Engineering, University of Electronic Science and Technology of China. His research interests include scheduling of mobile edge computing, design and optimization of next generation wireless networks, smart grid, and the Internet of Things.

Sabita Maharjan (SM'20) received the Ph.D. degree in networks and distributed systems from the Simula Research Laboratory, and University of Oslo, Norway, in 2013. She is currently a Senior Research Scientist at the Simula Metropolitan Center for Digital Engineering, Norway, and an Associate Professor at the University of Oslo. Her current research interests include wireless networks, network security and resilience, smart grid communications, Internet of Things, machine-to-machine communication, software-defined wireless networking, and the Internet of Vehicles.



Yan Zhang (F'20) received the Ph.D. degree from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. He is currently a Full Professor with the Department of Informatics, University of Oslo, Oslo, Norway. His research interests include next-generation wireless networks leading to 5G beyond/6G, green and secure cyber-physical systems (e.g., smart grid and transport). Dr. Zhang is an Editor for IEEE publications, including IEEE Communications Magazine, IEEE Network Magazine, IEEE Transactions on Vehicular Technology, IEEE Transactions on Industrial Informatics, IEEE Transactions on Green Communications and Networking, IEEE Communications Survey and Tutorials, IEEE Internet of Things Journal, IEEE Systems Journal, IEEE Vehicular Technology Magazine, and IEEE Blockchain Technical Briefs. He is a symposium/track chair in a number of conferences, including IEEE ICC 2021, IEEE Globecom 2017, IEEE PIMRC 2016, IEEE SmartGridComm 2015. He is an IEEE Vehicular Technology Society Distinguished Lecturer during 2016-2020 and he is named as CCF 2019 Distinguished Speaker. He is the Chair of IEEE Communications Society Technical Committee on Green Communications and Computing (TCGCC). He is a member of CCF Technical Committee of Blockchain. Since 2018, Prof. Zhang has been a recipient of the Highly Cited Researcher Award (Web of Science top 1