Low-Latency Federated Learning and Blockchain for Edge Association in Digital Twin Empowered 6G Networks

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

Abstract—**Emerging technologies, such as digital twins and 6th generation (6G) mobile networks, have accelerated the realization of edge intelligence in industrial Internet of Things (IIoT). The integration of digital twin and 6G bridges the physical system with digital space and enables robust instant wireless connectivity. With increasing concerns on data privacy, federated learning has been regarded as a promising solution for deploying distributed data processing and learning in wireless networks. However, unreliable communication channels, limited resources, and lack of trust among users hinder the effective application of federated learning in IIoT. In this article, we introduce the digital twin wireless networks (DTWN) by incorporating digital twins into wireless networks, to migrate real-time data processing and computation to the edge plane. Then, we propose a blockchain empowered federated learning framework running in the DTWN for collaborative computing, which improves the reliability and security of the system and enhances data privacy. Moreover, to balance the learning accuracy and time cost of the proposed scheme, we formulate an optimization problem for edge association by jointly considering digital twin association, training data batch size, and bandwidth allocation. We exploit multiagent reinforcement learning to find an optimal solution to the problem. Numerical results on real-world dataset show that the proposed scheme yields improved efficiency and reduced cost compared to benchmark learning methods.**

Index Terms—**Blockchain, communication efficiency, digital twin, federated learning, wireless networks.**

Yunlong Lu and Xiaohong Huang are with the Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yunlong.lu@ieee.org; huangxh@bupt.edu.cn).

Ke Zhang is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Beijing 610051, China (e-mail: zhangke@uestc.edu.cn).

Sabita Maharjan and Yan Zhang are with the Department of Informatics, University of Oslo, 0316 Oslo, Norway, and also with the Simula Metropolitan Center for Digital Engineering, 0167 Oslo, Norway (e-mail: sabita@ifi.uio.no; yanzhang@ieee.org).

I. INTRODUCTION

RECENT advancements in digital twins and 6th generation (6G) mobile networks have paved the way for industrial Internet of Things (IIoT) [1]. A series of edge intelligence use cases, such as intelligent transportation [2], [3] and smart city [4], are envisioned to provide high-quality services to users in the IIoT network. Data processing based on artificial intelligence (AI) algorithms lays the foundation for enabling the provision of intelligent services in the IIoT framework. Despite such enabling technologies and recent developments, the gap between the outcome of data analysis and the true reflection of the status of their physical systems remains a major hurdle in designing robust control algorithms for physical systems [5]. In this regard, the digital twin paradigm [6] emerges as one of the most promising technologies in 6G networks that can enable the near-instant wireless connectivity and extreme-reliable wireless communication[7]. Empowered by digital twin, the 6G networks can bridge the physical systems with the digital world in real time for realizing robust edge intelligence in IIoT.

Digital twins represent real objects or subjects with their data, functions, and communication capabilities in the digital space [8]. To reduce latency and enhance reliability for edge computing applications, we incorporate digital twins into wireless networks and propose the digital twin wireless networks (DTWN). By mapping IoT devices to digital twins in edge servers, DTWNs improve the efficiency of AI algorithms and mitigate the impact of unreliable communication between users and edge servers. Moreover, DTWNs interact with the physical IIoT systems in real time through data collection and analytics to keep the digital plane synchronized with the physical system. Thus, the running state analysis and optimization of user devices can be conducted directly in the DTWN, which reduces the system cost. We have noted few works that discuss integrating digital twins with physical systems to improve their performance for different applications. For instance, Schluse *et al.* [8] proposed the experimentable digital twins that can be used to realize complex control algorithms or mental models. In particular, the authors constructed a general digital twin for complex equipment and proposed a new method for health management using digital twins [9]. In this article, we propose to integrate digital twins into edge networks to mitigate the resource gap between end users and base stations (BSs) for efficient edge computing.

Conventional centralized AI algorithms require data to be transmitted to the server, which incurs high data leakage risks. To enhance data security and protect user privacy, federated learning [10], [11] has emerged as a new paradigm for distributed machine learning. Different from conventional AI algorithms, federated learning based models require transmitting only the parameters from locally trained models to the server while the original data is stored with the user to alleviate security and privacy risks [12]. To this end, the distributed framework and the enhanced data privacy assurance offered by federated learning have attracted considerable attention and have been well studied for typical MEC applications [13], such as content caching [14] and vehicular data sharing recently [15]. Nonetheless, the need of conducting the iterative training process on distributed entities each with possibly limited computing resources can adversely affect the efficiency of federated learning in edge networks. To address the issue, Wang *et al.* [16] proposed a control algorithm to determine the tradeoff between local update and global aggregation under a given resource budget. Tran *et al.* [17] formulated resource consumption and learning performance into an optimization problem and provided the approximate optimal solution.

Moreover, the lack of mutual trust among users is another key factor that can hinder them from participating in the learning process to realize edge intelligence. To deal with this issue, blockchain has been widely explored to build a secure collaborative learning mechanism among untrusted users. For example, Alcaraz *et al.* [18] proposed a three layer-based architecture to manage reliable and secure connections among entities, processes, and critical resources for the smart grid. Meanwhile, the efficiency and intelligence of blockchain needs to be further improved. Several studies have explored the synergy between AI algorithms and blockchain to enhance the data security and system reliability aspects in edge intelligence enabled networks. Zhang *et al.* [19] presented an edge intelligence and blockchain empowered framework to enhance the security of edge service management in an IoT scenario. Similarly, Dai *et al.* [20] integrated blockchain with deep reinforcement learning (DRL) for content caching in vehicular networks. As conventional AI algorithms can suffer much data transmission load while also leading to higher data leakage risk, we propose to integrate blockchain with federated learning to achieve secure and efficient edge intelligence. In our proposed framework, the federated learning parameters are recorded in the blockchain instead of the parameter server, thus considerably enhancing the security of parameters and improving the reliability of models from users.

In our proposed framework, DTWNs mitigate the unreliability of learning parameter transmission caused by long communication distance and limited communication resource between end users and edge servers (BSs). However, some new challenges also arise for the maintenance of digital twins. First is the associations of digital twins to various servers. It consumes considerable amount of resources to maintain digital twins by synchronizing real-time data and building corresponding models. Due to the dynamic computing and communication resources available with edge servers, the association of digital twins to corresponding servers is a crucial factor in determining the performance of DTWNs, which is a largely unexplored dimension in the context of edge networks. Moreover, due to the multiple communication rounds in federated learning, the limited bandwidth available with BSs requires to be optimally allocated to improve the communication efficiency of associated edge servers. In this article, we first design the new DTWN model by incorporating digital twins into wireless networks to mitigate the unreliable and long distance communication between users and BSs, where the user data are synchronized to BSs to construct corresponding digital twins. Based on the DTWN, we propose a permissioned blockchain empowered federated learning framework for realizing robust edge intelligence. Data analysis can be executed on their digital twins in the BS plane directly instead of in the resource-constrained end users. The security and privacy of digital twin data are also enhanced by. To improve the efficiency of the proposed architecture, we formulate an optimization problem for edge association and bandwidth allocation, and design a multiagent reinforcement learning based algorithm to find the optimal solution to it. The main contributions of this article are summarized as follows.

1) We design a new DTWN model, which uses digital twins to mitigate the unreliable and long distance communication between end users and edge servers.
2) We define the edge association problem for the DTWN, and propose a permissioned blockchain empowered federated learning framework for edge computing in the DTWN.
3) To improve the efficiency of the proposed scheme, we formulate an optimization problem by jointly considering edge association and communication resource allocation to minimize the time cost, and develop a multiagent reinforcement learning based algorithm to find the optimal solution to the problem.

The rest of this article is organized as follows. In Section II, the blockchain and federated learning model for the DTWN is provided. In Section III, we formulate a new edge association problem in the DTWN to minimize the total system latency. In Section V, a multiagent reinforcement learning based algorithm is developed to find the optimal solution to the problem. Numerical results are presented in Section V. Finally, Section VI concludes this article.

II. Blockchain and Federated Learning Model for Digital Twin Networks

In this section, we first present our DTWN model. Then, we describe the edge association problem in a DTWN in Section II-A, which is followed by the federated learning model in Section II-B. Then, we introduce our blockchain model in Section II-C and the wireless communication model in Section II-D.

We consider a blockchain and federated learning empowered digital twin network model as depicted in Fig. 1. The system consists of N end users, such as IoT devices and mobile devices, M BSs, and a macro-BS (MBS). The BSs and the MBS are equipped with MEC servers. The end devices generate running data and synchronize their data with corresponding digital twins
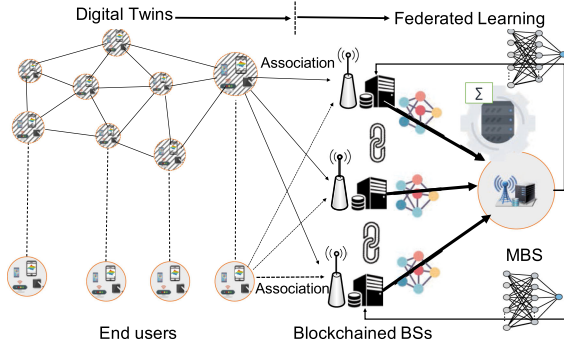
Fig. 1.   Proposed DTWNs.

that run on the BSs. We use $\mathcal{D}_i = \{(x_{i1}, y_{i1}), ..., (x_{iD_i}, y_{iD_i})\}$ to denote the data of end user i, where D_i is the data size, x_i is the data collected by end users, and y_i is the label of x_i. The digital twin of end user i in the BSs are denoted as DT_i, which is composed of behavior model $\mathcal{M}_i$, static running data $\mathcal{D}_i$, and real-time dynamic state s_t, i.e., $DT_i = (\mathcal{M}_i, \mathcal{D}_i, s_t)$. $\mathcal{D}_i$ and s_t are the essential data required to run digital twin applications. Instead of synchronizing all raw data to digital twins, which incurs great communication load and data leakage risk, we use federated learning to learn model $\mathcal{M}$ from user data. In various application scenarios, the end users may communicate with other end users to exchange running information and share data, through, e.g., device to device communications. Thus, the digital twins also form a network based on the connections of end users. Based on the constructed DTWN, we can obtain the running states of physical devices and make further decisions to optimize and drive the running of devices by analyzing digital twins directly.

In our proposed digital twin network model, we use federated learning to execute the training and learning process collaboratively for edge intelligence. Moreover, since the end users lack mutual trust and the digital twins consist of private data, we use permissioned blockchain to enhance the system security and data privacy. The permissioned blockchain records the data from digital twins and manages the participating users through permission control. The blockchain is maintained by the BSs, which are also the clients of the federated learning model. The MBS runs as the server for the federated learning model. In each iteration of federated learning, the MBS distributes the machine learning model parameters to BSs for training. The BSs train the model based on data from digital twins and return the model parameters to the MBS.

A. Edge Association in a DTWN

The end devices or users are mapped to the digital twins in BSs in a DTWN. The maintenance of digital twins consumes a large amount of computing and communication resources for synchronizing real-time data and building corresponding models. However, the computation and communication resources in wireless networks are very limited, which should be optimally used to improve the resource utility. Thus, to associate various IoT devices with different BSs according to their computation

capabilities and states of the communication channel is a key problem in the DTWN. As depicted in Fig. 1, the digital twins of IoT devices are constructed and maintained by their associated BSs. The training data and the computation tasks for training are distributed to various BSs based on the association between digital twins and the BSs. We provide the formal definition of the edge association in Definition 1.

Definition 1 (Edge Association): Consider a DTWN with N IoT users and M BSs. For any user u_i, $i \in \mathcal{N}$, the goal of edge association is to choose the target BS $j \in \mathcal{M}$ to construct the digital twin DT_i of user i. The association $\langle DT_i, BS_j \rangle$ is denoted as $\Phi(i, j)$. If DT_i is associated to BS j, then $\Phi(i, j) = D_i$, where D_i is the size of data used to construct DT_i; otherwise, $\Phi(i, j) = 0$.

A BS can associate with multiple digital twins, whereas a digital twin can only be associated with at most one BS. That is, $\sum_{j=1}^{M} \Phi(i, j) = D_i$. We perform edge association according to the datasets D_i of IoT users, the computation capability of BSs f_j, and the transmission rate $R_{i,j}$ between u_i and BS_j denoted as

$$\Phi(i, j) = f(D_i, f_j, R_{i,j}). \qquad (1)$$

The objective of the edge association problem is to improve the utility of resources and the efficiency of running digital twins in the DTWN.

B. Federated Learning Model

The goal of federated learning is to train a global machine learning model $\mathcal{M}$ based on the data from various digital twins DT_i without transmitting the original training data. The learning process involves finding the parameter w for global model $\mathcal{M}$ that minimizes the global loss function $f(w, x, y)$ denoted as

$$\min_{w} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{D_j} \sum_{j=1}^{D_j} f(w, x_{ij}, y_{ij}). \qquad (2)$$

The client BSs train their local models based on their own data. Each client minimizes the loss function on its local data through gradient descent based methods, such as gradient descent and stochastic gradient descent, with a predefined learning rate. The clients then transmit the local gradients or the local models to the server MBS to update the global model, through algorithms, such as federated averaging [10] and federated stochastic gradient descent (SGD). We denote the aggregation of the global model as

$$\mathcal{M} = \frac{1}{N} \sum_{i=1}^{N} D_i w_i. \qquad (3)$$

In the training process, the BSs train the local models based on the digital twins DT_i that they maintained according to (2). A BS may contain several digital twins constructed from the end users under its coverage. Different from conventional federated learning, the BSs in our proposed scheme first aggregate their local models from multiple digital twins instead of sending the original local models to the MBS, which can reduce the transmission load. As BS m has K_i digital twins, the aggregation
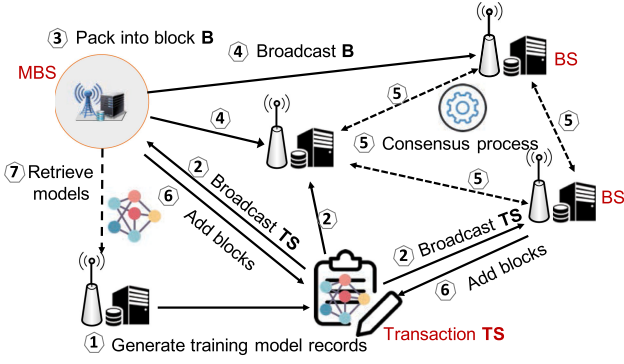
Fig. 2. Blockchain scheme for federated learning.

at BS i is

$$\mathcal{G}_m = \frac{1}{K_i} \sum_{j=1}^{K_i} D_{\mathrm{DT}_j} w_{\mathrm{DT}_j} \qquad (4)$$

where D_{DT_i} is the training data of digital twin DT_i and w_{DT_i} is the trained model of DT_i. BS m then sends $\mathcal{G}_m$ to the MBS, and the MBS updates the global model as

$$\mathcal{M} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{G}_m. \qquad (5)$$

The local models of digital twins are transmitted to the MBS through wireless links. The MBS collects the model parameters from all participating BSs and updates the global model. In this process, since all BSs transmit their parameters in the same period and the wireless bandwidth is limited, the communication efficiency is of vital importance for the learning phase to converge.

C. Blockchain Model

To enhance the security and reliability of digital twins from untrusted end users, the BSs act as the blockchain nodes and maintain the running of the permissioned blockchain. The digital twins are stored in the blockchain and their data are updated as the state of the corresponding user changes. Moreover, the local models of the BS are also stored in the blockchain, which can be verified by other BSs to ensure the quality of local models. Thus, there are three types of records, namely digital twin model records, digital twin data records, and training model records.

The overall blockchain scheme for federated learning is shown in Fig. 2. The BSs first train the local training models on their own data, and then upload the trained models to the MBS. The trained models are also recorded as blockchain transactions, and broadcasted to other BSs for verification. Other BSs collect the transactions and pack them into blocks. The consensus process is executed to verify the transactions in blocks. Our consensus process is executed based on Delegated Proof of Stakes protocol, where the stakes are the training coins. Initial training coins are allocated to BS i according to its data from

digital twins denoted as

$$S_i = \frac{\sum_{j=1}^{K_i} D_{\mathrm{DT}_j}}{\sum_{k=1}^{M} D_k} S_{\mathrm{ini}} \qquad (6)$$

where S_{ini} is an initial value and K_i is the number of digital twins associated to BS i.

The coins of each BS are then adjusted according to their performance in the training process. If the trained model of a BS passes the verification of other BSs and the MBS, coins will be awarded to the BS. Otherwise, the BS will get no pay for its training work. A number of BSs are elected as the block producers by all BSs. In the voting process, all the BSs vote for the candidate BSs by using their own training coins. The elected BSs take turns to pack the transactions in a time interval T into a block B, and broadcast block B to other producers for verification.

In our federated learning scheme, we leverage blockchain to verify the local models before embedding them into the global model. Due to high resource consuming for block verification, the interval T should be set to multiple times of the local training period, that is, the BSs execute multiple local training iterations before transmitting local models to the MBS for global aggregation.

D. Communication Model

We use the orthogonal frequency division multiple access for wireless transmission in our system. To upload trained local models, all BSs share C subchannels to transmit their parameters. The achievable uplink data rate from BS i to the MBS is

$$R_i^U = \sum_{c=1}^{C} \tau_{i,c} W^U \log_2 \left(1 + \frac{P_{i,c}^U h_{i,c}^U r_{i,m}^{-\alpha}}{\sum_{j \in \mathcal{N}'} P_{j,c}^U h_{j,c}^U r_{i,m}^{-\alpha} + N_0} \right) \quad (7)$$

where C is the total number of subchannels, $\tau_{i,c}$ is the time fraction allocated to BS i on subchannel c, and W^U is the bandwidth of each subchannel, which is a constant value. The transmission power is $P_{i,c}^U$ and the uplink channel gain on subchannel c is $h_{i,c}^U$. $r_{i,m}^{-\alpha}$ is the path loss fading of the channel between BS i and the MBS, $r_{i,m}$ is the distance between BS i and the MBS, and α is the path loss exponent. N_0 is the noise power and $\sum_{j \in \mathcal{N}'} P_{j,c}^U h_{j,c}^U r_{i,m}^{-\alpha}$ is the interference caused by other BSs using the same subchannel. In the download phase, the MBS broadcasts the global model with the rate

$$R_i^D = \sum_{c=1}^{C} W^D \log_2 \left(1 + \frac{P_{i,c}^D h_{i,c}^D}{\sum_{j \in \mathcal{N}''} P_{j,c}^D h_{j,c}^D r_{i,m}^{-\alpha} + N_0} \right) \quad (8)$$

where $P_{i,c}^D$ is the downlink power of BS i and $h_{i,c}^D$ is the channel gain between BS i and the MBS, and $\sum_{j \in \mathcal{N}''} P_{j,c}^D h_{j,c}^D r_{i,m}^{-\alpha}$ is the downlink inference.

III. EDGE ASSOCIATION IN THE DTWN: PROBLEM FORMULATION

In this section, we first analyze the delay performance of the proposed model, and then formulate the optimization problem

for edge association in the DTWN to minimize the total system time cost under the given learning accuracy requirement.

Now, we start to derive the formulation of the edge association problem. We consider that the gradient $\nabla f(w)$ of $f(w)$ is L-Lipschitz smooth, that is

$$||\nabla f(w_{t+1}) - \nabla f(w_t)|| \leq L||w_{t+1} - w_t|| \tag{9}$$

where L is a positive constant and $||w_{t+1} - w_t||$ is the norm of $w_{t+1} - w_t$. We consider that the loss function $f(w)$ is strongly convex, that is

$$f(w_{t+1}) \geq f(w_t) + \langle \nabla f(w_t), w_{t+1} - w_t \rangle + \frac{1}{2}||w_{t+1} - w_t||^2. \tag{10}$$

Many loss functions for federated learning can satisfy the above-mentioned assumptions, for example, the logic loss functions. If (9) and (10) are satisfied, the upper bound of global iterations can be obtained as [21]

$$\mathcal{T}(\theta_L, \theta_G) = \frac{\mathcal{O}(\log(1/\theta_L))}{1 - \theta_G} \tag{11}$$

where θ_L is the local accuracy such that $\frac{||\nabla f(w_{t+1})||}{||\nabla f(w_t)||} \leq \theta_L$, θ_G is the global accuracy, and $0 \leq \theta_L, \theta_G \leq 1$. As in [17], we consider θ_L as a fixed value, so that the upper bound $\mathcal{T}(\theta_L, \theta_G)$ can be simplified to $\mathcal{T}(\theta_G) = \frac{1}{1 - \theta_G}$. Denote the time of one local training iteration by T_{cmp}, then the computation time in one global iteration is $\log(1/\theta)T_{\text{cmp}}$, and the upper bound of total learning time is $\mathcal{T}(\theta_G)T_{\text{glob}}$.

The time cost in our proposed scheme mainly consists of the following.

1) *Local Training on Digital Twins:* The time cost for the local training of BS i is decided by the computing capability and the data size of its digital twins. The time cost is

$$T_i^{\text{cmp}} = \frac{\sum_{j=1}^{K_i} b_j D_{\text{DT}_j}}{f_i^C} f^C \tag{12}$$

where f^C is the number of CPU cycles required to train one sample of data, f_i^C is the CPU frequency of BS i, and b_j is the training batch size of digital twin DT_j.

2) *Model Aggregation on BSs:* The BSs aggregate its local models from various digital twins according to (4). The computing time for local aggregation is

$$T_i^{la} = \frac{\sum_{j=1}^{K_i} |w_j|}{f_i^C} f_b^C \tag{13}$$

where $|w_j|$ is the size of local models and f_b^C is the number of CPU cycles required to aggregate one unit of data. Since all the clients share the same global model, $|w_1| = |w_2| = \cdots = |w_j| = |w_g|$. Thus, the time cost for local aggregation is

$$T_i^{la} = \frac{K_i|w_g|}{f_i^C} f_b^C. \tag{14}$$

3) *Transmission of Model Parameters:* The local models aggregated by BS i are then broadcasted to other BSs as transactions. The time cost is related to the number of

blockchain nodes and the transmission efficiency. Since other BSs also help to transmit the transaction in the broadcast process, the time function is related to $\log_2 M$, where M is the size of the BS network. The required time cost is

$$T_i^{\text{pt}} = \xi \log_2 M \frac{K_i|w_g|}{R_i^U} \tag{15}$$

where ξ is a factor of transmission time cost that can be obtained from historical running records of the transmission process.

4) *Block Validation:* The block producer BS collects the transactions and packs them into a block. The block is then broadcasted to other producer BSs, and validated by them. Thus, the time cost is

$$T_{\text{bp}}^{\text{bv}} = \xi \log_2 M_p \frac{S_B}{R_i^D} + \max_i \frac{S_B f^v}{f_i s} \tag{16}$$

where M_p is the number of block producers, and S_B is the size of a block.

Note that in the aggregation phase, the size of model parameters $|w_g|$ is small and the computing capability f_i is high. Thus, compared to other phases, the time for aggregation is very small that can be neglected. Based on the above analysis, the time cost for one iteration is denoted as

$$T = \max_i \left\{ \frac{\sum_{j=1}^{K_i} b_j D_{\text{DT}_j}}{f_i^C} f^C \right\} + \max_i \left\{ \xi \log_2 M \frac{K_i|w_g|}{R_i^U} \right\}$$
$$+ \xi \log_2 M_p \frac{S_B}{R_i^D} + \max_i \frac{S_B f^v}{f_i s}. \tag{17}$$

In the 6G network, the growth of the user scale, the ultralow latency requirement of communication, and the dynamic network status make it an important issue to reduce the time cost of model training in various applications. Since accuracy and latency are two main metrics to evaluate the decision-making abilities of digital twins in our proposed scheme, we consider the edge association problem to find the tradeoff between learning accuracy and time cost of the learning process. Due to the dynamic computing and communication capabilities of various BSs, the edge association of digital twins, that is, how to allocate the digital twins of different end users to various BSs for training, is a key issue to be solved for minimizing the total time cost. Moreover, increasing the training batch size b_n of each digital twin DT_n can improve the learning accuracy. However, it will also increase the learning time cost to execute more computation. In addition, how to allocate the bandwidth resources to improve the communication efficiency should also be considered. In our edge association problem, we should carefully design these policies to minimize the total time cost of the proposed scheme. Thus, we formulate the optimization problem as minimizing the time cost of federated learning under a given learning accuracy. To solve the problem, the association of digital twins, the batch size of their training data, and the bandwidth allocation should be jointly considered according to the computing capability f_i^C and the channel state $h_{i,c}$. The optimization problem can be
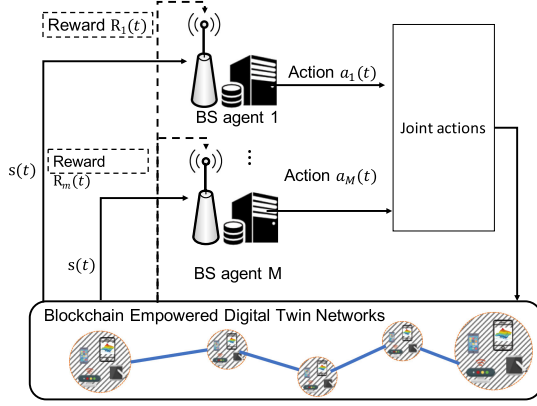
Fig. 3. Multiagent DRL for edge association

formulated as

$$\min_{\boldsymbol{K}_i, \boldsymbol{b}_n, \boldsymbol{\tau}_{i,c}} \frac{1}{1 - \theta_G} T \tag{18}$$

$$\text{s.t.} \quad \theta_G \geq \theta_{\text{th}}, \theta_G, \theta_{\text{th}} \in (0, 1) \tag{18a}$$

$$\sum_{i=1}^{M} K_i = D, K_i \in \mathcal{N} \tag{18b}$$

$$\sum_{i=1}^{M} \tau_{i,c} \leq 1, c \in \mathcal{C} \tag{18c}$$

$$b^{\min} \leq b_n \leq b_n^{\max} \quad \forall n \in \mathcal{N}. \tag{18d}$$

Constraint (18b) ensures that the sum of the number of associated digital twins does not exceed the size of total dataset. Constraint (18c) guarantees that each subchannel can only be allocated to at most one BS. Constraint (18d) ensures the range of training batch size for each digital twin. Problem (18) is a combinational problem. Since there are several products of variables in the objective function, and the time cost of each BS is also affected by the resource states of other BSs, it is challenging to solve problem (18).

IV. MULTIAGENT DRL FOR EDGE ASSOCIATION

Since the system states are only determined by network states in the current iteration and the allocation policies in the last iteration, we regard the problem as a Markov decision process (MDP) and use multiagent DRL [22] based algorithm to solve it.

A. Multiagent DRL Framework

The proposed multiagent reinforcement learning framework is shown in Fig. 3. In our proposed system, each BS is regarded as a DRL agent. The environment consists of BSs and digital twins of end users. Our multiagent DRL framework consists of multiple agents, a common environment, system state $\mathcal{S}$, action $\mathcal{A}$, and the reward function $\mathcal{R}$, which are described in the following.

1) *State Space:* The state of the environment is composed of the computing capabilities f^C of BSs, the number of digital twins K_i on each BS i, the training data size of each digital twin D_n, and the channel state $h_{i,c}$. The states of multiple agents are denoted as $s(t) = (\boldsymbol{f}^C, \boldsymbol{K}, \boldsymbol{D}, \boldsymbol{h})$, where each dimension is a state vector that contains the states for all agents.

2) *Action Space:* The actions of BS i in our system consist of the digital twin allocation K_i, the training data batch sizes for its digital twins $\boldsymbol{b}_i$, and the bandwidth allocation $\boldsymbol{\tau}_i$. Thus, the actions are denoted as $\boldsymbol{a}_i(t) = (K_i, \boldsymbol{b}_i, \boldsymbol{\tau}_i)$. BS agent i makes new action decisions $\boldsymbol{a}_i(t)$ at the beginning of iteration t based on system state $s(t)$. The system action is $\boldsymbol{a}(t) = (\boldsymbol{a}_1, ..., \boldsymbol{a}_i, ..., \boldsymbol{a}_m)$.

3) *Reward:* We define the reward function of BS i according to its time cost T_i based on (17) as

$$\mathcal{R}_i(s(t), \boldsymbol{a}_i(t)) = -T_i(t). \tag{19}$$

The reward vector of all agents is $\boldsymbol{R} = (\mathcal{R}_1, ..., \mathcal{R}_m)$. According to (18), the total time cost T is decided by the maximum time cost of agents $\max\{T_1, T_2, ..., T_m\}$. Thus, each DRL agent shares the same reward function in our scheme. In the training process, the BS agents adjust their actions to maximize the reward function, that is, to minimize the system time cost in each iteration.

The learning process of BS i is to find the best policy that maps its states to its actions, denoted as $\boldsymbol{a}_i = \pi_i(s)$, where $\boldsymbol{a}_i$ is the action to be taken by BS i for the whole system state s. The objective is to maximize the expected reward, that is

$$\mathcal{R}_t = \sum_i \gamma \mathcal{R}_i(s(t), \boldsymbol{a}_i(t)) \tag{20}$$

where γ is the discount rate, $0 \leq \gamma \leq 1$. In the conventional DRL framework, it is hard for an agent to obtain the states of others. In our DTWN, the states of digital twins and BSs are recorded in the blockchain. A BS can retrieve records from blockchain to obtain the system states and actions of other agents in the training process.

B. Multiagent DRL Algorithm for Edge Association

Since the network state variables, such as $\boldsymbol{h}$ and $\boldsymbol{f}^C$, are continuous values, the state and action spaces are large. We adopt a deep deterministic policy gradient (DDPG) algorithm to address the formulated MDP problem. The DDPG algorithm uses policy-based actor-critic to select and evaluate the actions, and uses deep Q networks (DQNs) to approximate value functions in actor and critic networks. The multiple agents share the same reward function and train their policy networks cooperatively to minimize the time cost.

In the training phase, each BS agent generates the training sample $\langle s, \boldsymbol{a}_i, R_i, s' \rangle$, where s' is the state of the DTWN in the next time slot. We leverage DQN with the tuples randomly sampled from replay memory to train the actor network $\pi(s_t|\theta\pi)$ and critic network $Q(s_t, \boldsymbol{a}_i|\theta_Q)$, with parameters θ_π and θ_Q. There are two sets of actor and critic networks—the primary networks and the target networks. The target networks have the

same structure as the primary networks and are used to generate target values for training primary networks.

Each agent i has an actor DNN $\pi(s_t|\theta_\pi)$ to determine its action $\boldsymbol{a}_i$ based on current state s_t as

$$a_i(t) = \pi_i(s_t|\theta_{\pi_i}) + \mathfrak{N} \tag{21}$$

where $\mathfrak{N}$ is the random noise generated by the Ornstein–Uhlenbeck process [23], and θ_{π_i} is the explored edge association policy. The parameters of the primary actor DNN are updated as

$$\theta_\pi = \theta_\pi + \alpha_\pi$$
$$\cdot \mathbb{E}[\nabla_{\boldsymbol{a}_i} Q(\boldsymbol{s_t}, \boldsymbol{a_1}, ..., \boldsymbol{a_i}|\theta_Q)|_{\boldsymbol{a_i}=\pi(\boldsymbol{s_t}|\theta_\pi)} \cdot \nabla_{\theta_\pi} \pi(\boldsymbol{s_t})] \tag{22}$$

where α_π is the learning rate of the actor DNN. We train the primary critic network of agent i by taking steps in the opposite gradient direction of the loss function as

$$\theta_{Q_i} = \theta_{Q_i}$$
$$+ \alpha_{Q_i} \cdot \mathbb{E}[2(y_t - Q(\boldsymbol{s_t}, \boldsymbol{a_i}|\theta_{Q_i})) \cdot \nabla Q(\boldsymbol{s_t}, \boldsymbol{a_1}, ..., \boldsymbol{a_i})] \tag{23}$$

where α_{Q_i} is the learning rate of the primary critic DNN, y_t is the target value generated by target networks, and $(\boldsymbol{a_1}, ..., \boldsymbol{a_i})$ are the actions of all DRL agents.

The target networks represent the old version of the primary networks. The parameters of the target action DNN θ_π^T and the target critic DNN θ_Q^T are trained in a different way. Agent i updates its target action DNN $\theta_{\pi_i}^T$ and critic DNN $\theta_{Q_i}^T$ as

$$\theta_{\pi_i}^T = \beta\theta_{\pi_i} + (1 - \beta)\theta_{\pi_i} \tag{24}$$
$$\theta_{Q_i}^T = \beta\theta_{Q_i} + (1 - \beta)\theta_{Q_i} \tag{25}$$

where β is the update rate.

The proposed multiagent reinforcement learning algorithm for an edge association problem is summarized as Algorithm 1. Each agent first initializes its primary and target critic-actor DNNs, and initializes the edge association policy. Then, the primary actor DNN of agent i generates action $\boldsymbol{a}_i$ based on current state and policy according to (21). The observed reward $R_i(e)$ and next state s_{t+1} are calculated, and the tuple $(s_t, a, R_i(e), s_{t+1})$ is stored into the replay memory as training samples. The primary actor DNN and critic DNN are then updated based on (22) and (23). The target networks are updated according to (24).

The computation cost training process of the DRL can be performed offline for large number of episodes based on the dynamic system states. The trained DRL models are deployed online to optimally allocate the resources to minimize the time cost of federated learning under given learning accuracy. The actions in our multiagent DRL for edge association are composed of digital twin association policies $\boldsymbol{K}$, batch size of training data $\boldsymbol{b}$, and the bandwidth allocation policies $\boldsymbol{\tau}$. Due to the large action and state space, we adopt the actor DNN to generate actions. The complexity of our proposed learning algorithm mainly lies in the training process of actor DNN θ_π and critic DNN θ_Q. Consider that the DNN in our algorithm has L hidden
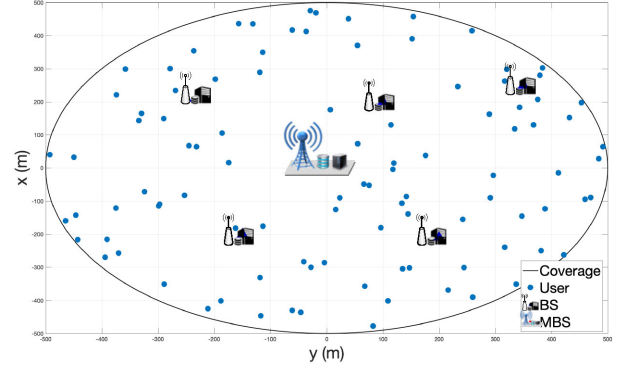

Fig. 4. Illustration of the wireless network.

Algorithm 1: The Multiagent DRL Based Algorithm for Edge Association.

1: **for** each BS $i \leq M$ **do**
2:      Initialize primary and target actor-critic networks; Initialize replay memory;
3: **end for**
4: **for** each BS $i \leq M$ **do**
5:      **for** episode $e \leq E$ **do**
6:          Initialize DTWN environment setup;
7:          **for** each time slot t **do**
8:              Observe current state s and execute action a_i according to (21);
9:              Compute reward $R_i(t)$, and transit to the next state s_{t+1} based on actions $(\boldsymbol{a_1}, \boldsymbol{a_2}, ..., \boldsymbol{a_n})$
10:             Store $(s_t, \boldsymbol{a_i}(t), R_i(t), s_{t+1})$ into replay memory
11:             Update primary actor and critic networks according to (22) and (23)
12:             Update target networks according to (24)
13:         **end for**
14:     **end for**
15: **end for**

---

layers, and the average number of neurons in each layer is L_a. Thus, the time complexity for training a DNN is $O(L_a^2)$. The total complexity for training the multiagent DRL model with M agents is $O((M \cdot L_a^2) \cdot E)$. Since the training at each agent is executed in parallel, the time complexity is $O(L_a^2 \cdot E)$, which is related to the DNN size and the number of episodes.

V. NUMERICAL RESULTS

We conduct our experiment on a real-world dataset CIFAR10[24]. The CIFAR10 dataset consists of $60\,000$ images in 10 classes, including $50\,000$ training images and $10\,000$ test images. Learning on the image dataset simulates the real edge computing scenarios such as traffic flow monitoring and image recognition of smart cameras. We consider a wireless network with 1 MBS, 5 BSs, and 100 end users. The users are randomly distributed in the coverage area of BSs, as shown in
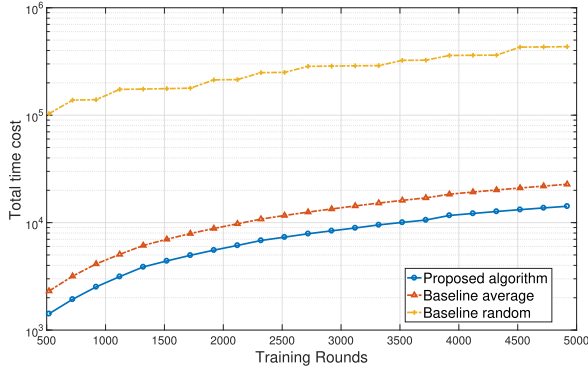
Fig. 5.   Total system time cost of different schemes in the training process.
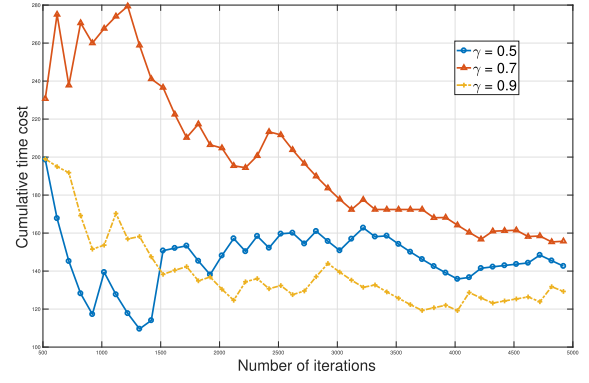

Fig. 6.   Loss performance in the learning process.

Fig. 7.   Cumulative average system cost in the training process of DRL.

Fig. 8.   Time cost of the proposed optimization algorithm.

Fig. 4. The CIFAR10 dataset is shuffled and assigned to the end users randomly. Thus, the training data of federated learning is independent and identically distributed in our experiment. The convolutional neural network (CNN) [25] is adopted as the machine learning model for federated learning. The CNN model has two 5×5 convolution layers (with 32 channels and 64 channels, respectively) and 2×2 max-pooling layers, followed by a fully connected layer with 512 units. The maximum CPU frequencies of five BSs are 2.6, 1.8, 3.6, 2.4, and 2.4GHz. The transmission powers of BSs and the MBS are 34 and 42 dB·m, respectively. The bandwidth of the subchannel is set as 30 MHz. N_0 is set to -174 dB·m.

The latency performance with respect to training rounds is shown in Fig. 5 for our proposed algorithm and the conventional learning algorithms with random edge association and with average edge association. Our proposed algorithm significantly reduces the system time cost compared with the benchmark algorithms. The multiagent reinforcement learning algorithm optimizes digital twin association, and optimally allocates communication resources, which improves the running efficiency and reduces the system latency for our scheme. Fig. 6 depicts the learning performance of the proposed algorithm, compared with the learning algorithm with full training data and the learning algorithm with random edge association. From the results, we can see that our proposed algorithm achieves a comparable convergence performance compared to fully trained federated learning, where all training data of users are utilized in each
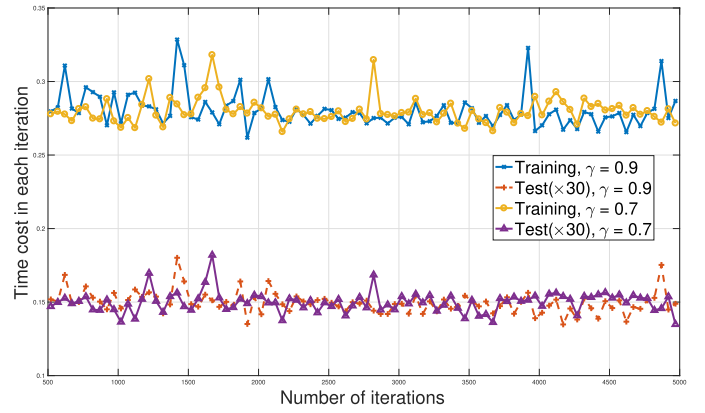
training process. The learning loss of our proposed algorithm is improved through the optimization process compared with the conventional algorithm with random edge association.

The average cumulative reward is calculated by $\mathcal{R}_n = \frac{\sum_{t=1}^{n} \sum_{i=1}^{M} R_{i,t}}{n \cdot M}$, where $R_{i,t}$ is the total reward of agent i in episode t, and M is the number of agents. The cumulative average time costs of our proposed multiagent DRL are depicted in Fig. 7 for different values of the discount factor γ. The cost of different γ converges as the number of iterations increases, which denotes that the training process of the DRL model has maximized the cumulative reward and has minimized the system time cost. The curve with $\gamma = 0.9$ achieves the best performance, which has a higher convergence rate and smaller system time cost. The time cost for running our reinforcement learning based optimization algorithm is shown in Fig. 8. We can see that the training of our proposed optimization algorithm costs much more time than the test phase. The reason is that the training of a DRL model is a process of iterative exploration. The agents take much time to interact with the environment to learn optimal policies. Although there are some fluctuations in the result curves, the overall time cost for training or testing in each iteration is comparable. The time is consumed mainly by the training process of the deep neural networks in reinforcement learning. From Fig. 8, we can also see that a larger discount factor γ slightly leads to a larger time cost in each iteration. The

reason is that a larger discount factor incurs more computation in the policy training process, thus leading to a larger time cost.

VI. Conclusion

In this article, we proposed a new DTWN model based on federated learning and blockchain, and improved the running efficiency of the proposed scheme. We first presented a DTWN model in a wireless edge network consisting of end users and BSs. We further developed a blockchain empowered federated learning scheme in DTWN for realizing edge intelligence. To improve the efficiency of the proposed scheme and better allocate the limited resources, we formulated an optimization problem for given learning accuracy. Specifically, we formulated the problem of edge association between digital twins and BSs and derived an optimal solution to the problem by exploiting the multiagent reinforcement learning. Numerical results based on the real-world dataset demonstrated that the proposed scheme effectively reduces the learning latency and achieves good learning convergence.

References

[1] A. H. Sodhro, S. Pirbhulal, and V. H. C. de Albuquerque, "Artificial intelligence-driven mechanism for edge computing-based industrial applications," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4235–4243, Jul. 2019.

[2] N. Kumar, S. S. Rahman, and N. Dhakad, "Fuzzy inference enabled deep reinforcement learning-based traffic light control for intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2020.2984033.

[3] K. Zhang, S. Leng, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1987–1997, Apr. 2019.

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

[5] T. H.-J. Uhlemann, C. Lehmann, and R. Steinhilper, "The digital twin: Realizing the cyber-physical production system for industry 4.0," *Procedia CIRP*, vol. 61, pp. 335–340, 2017.

[6] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.

[7] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6G be?" *Nature Electron.*, vol. 3, no. 1, pp. 20–29, 2020.

[8] M. Schluse, M. Priggemeyer, L. Atorf, and J. Rossmann, "Experimentable digital twins—Streamlining simulation-based systems engineering for Industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1722–1731, Apr. 2018.

[9] F. Tao, M. Zhang, Y. Liu, and A. Nee, "Digital twin driven prognostics and health management for complex equipment," *CIRP Ann.*, vol. 67, no. 1, pp. 169–172, 2018.

[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[11] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4424–4434.

[12] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IOT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4177–4186, Jun. 2020.

[13] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surv. Tut.*, vol. 22, no. 3, pp. 2031–2063, 2020.

[14] Z. Yu *et al.*, "Federated learning based proactive content caching in edge computing," in *Proc. IEEE Global Commun. Conf.*, 2018, pp. 1–6.

[15] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4298–4311, Apr. 2020.

[16] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 1205–1221, Jun. 2019.

[17] 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 *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1387–1395.

[18] C. Alcaraz, J. E. Rubio, and J. Lopez, "Blockchain-assisted access for federated smart grid domains: Coupling and features," *J. Parallel Distrib. Comput.*, vol. 144, pp. 124–135, Oct. 2020.

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

[20] 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 Trans. Vehicular Technol.*, vol. 69, no. 4, pp. 4312–4324, Apr. 2020.

[21] C. Ma *et al.*, "Distributed optimization with arbitrary local solvers," *Optim. Methods Softw.*, vol. 32, no. 4, pp. 813–848, 2017.

[22] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6379–6390.

[23] O. E. Barndorff-Nielsen and N. Shephard, "Non-Gaussian Ornstein–Uhlenbeck-based models and some of their uses in financial economics," *J Roy. Statist. Soc., Ser. B (Statist. Methodol.)*, vol. 63, no. 2, pp. 167–241, 2001.

[24] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., Univ. of Toronto, 2009.

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

Yunlong Lu (Member, IEEE) received the M.S. degree in computer science and technology from the School of Computer Science from Beijing University of Posts and Telecommunications, Beijing, China, in 2015, where he is currently working toward the Ph.D. degree in computer science and technology.

From 2018 to 2019, he was a visiting student with the University of Oslo, Oslo, Norway. His current research interests include blockchain, wireless networks and privacy-preserving machine learning.

Xiaohong Huang (Member, IEEE) received the B.E. degree in mechanical and electronic engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2000, and the Ph.D. degree in electrical and electronic engineering from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2005.

Since 2005, he has been with BUPT, where she is currently a Professor and Director of Network and Information Center in Institute of Network Technology. He has authored or coauthored more than 50 academic papers in the area of wavelength division multiplexing (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 communication and information system from the University of Electronic Science and Technology of China, Beijing, China, in 2017.

He is currently a Lecturer with 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.
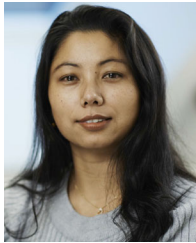
Sabita Maharjan (Senior Member, IEEE) received the Ph.D. degree in networks and distributed systems from the University of Oslo, Oslo, Norway, and Simula Research Laboratory, Oslo, Norway, in 2013.

She is currently an Associate Professor with the Department of Informatics, University of Oslo, and a Senior Research Scientist with Simula Metropolitan Center for Digital Engineering, Oslo, Norway. She authored or coauthored regularly in prestigious journals in her field such as the IEEE TRANSACTIONS ON SMART GRID, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *IEEE Communications Magazine*, *IEEE Network Magazine*, *IEEE Wireless Communications Magazine*, and IEEE INTERNET OF THINGS JOURNAL. Her current research interests include vehicular networks and 5G, network security, smart grid communications, Internet of Things, and computational intelligence.

Dr. Maharjan is currently the Vice-Chair of the IEEE Communications Society Technical Committee on Green Communications and Computing SIG on Green AI. She is an Associate Editor for the IEEE INTERNET OF THINGS JOURNAL, and has served as the Guest Editor for journals such as IEEE ACCESS, and in the technical program committee of conferences including top conferences such as IEEE INFOCOM and IEEE IWQoS.

Yan Zhang (Fellow, IEEE) received the Ph.D. degree in electrical and electronic engineering from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore, in 2005.

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).

Prof. Zhang is a Fellow of IET, an elected member of Academia Europaea, and an elected member of Norwegian Academy of Technological Sciences. He is an Editor (or Area Editor, Associate Editor) for several IEEE publications, including *IEEE Communications Magazine*, *IEEE Network Magazine*, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, 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, and 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. He is an elected member of CCF Technical Committee of Blockchain. In both 2018 and 2019, he was a recipient of the global Highly Cited Researcher Award.