Communication-efficient Federated Learning and Permissioned Blockchain for Digital Twin Edge Networks

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

Abstract—Emerging technologies such as Mobile Edge Computing (MEC) and next generation communications are crucial for enabling rapid development and deployment of Internet of Things (IoT). With the increasing scale of IoT networks, how to optimize the network and allocate the limited resources to provide high-quality services remains as a major concern. Existing work in this direction mainly relies on models that are of less practical value for resource limited IoT networks, and can hardly simulate the dynamic systems in real-time. In this paper, we integrate digital twins with edge networks and propose the Digital Twin Edge Networks (DITEN) to fill the gap between physical edge networks and digital systems. Then, we propose a blockchain empowered federated learning scheme to strengthen communication security and data privacy protection in DITEN. Furthermore, to improve efficiency of the integrated scheme, we propose an asynchronous aggregation scheme and use digital twin empowered reinforcement learning to schedule relaying users and allocate bandwidth resources. Theoretical analysis and numerical results confirm that the proposed scheme can considerably enhance both communication efficiency and data security for IoT applications.

Index Terms—Communication efficiency, Blockchain, federated learning, Digital twin, Edge networks

I. INTRODUCTION

The rapid development of Internet of Things (IoT) applications leads to the generation of a large amount of data at the edge networks. Analyzing and mining data from the edge network is an effective way to improve the quality of applications and the user experience of services, such as autonomous driving, virtual reality, and smart home. Meanwhile, in the 5th generation (5G) or 6th generation (6G) mobile networks, delay-sensitive applications raise new requirements for ultra-reliable and low-latency communications (URLLC) to transmit and analyze IoT data through network-wise sampling [1], cloud computing or edge computing. However, the stochasticity in wireless networks due to dynamic traffic load and uncertain channel conditions make it challenging to process such data in real-time.

The Digital Twin technology offers great potential to bridge the gap between the rate of data generation from IoT sensors and the required level of rapid and real-time data analysis. A digital twin is a digital representation of a physical item or assembly using integrated simulations and service data [2]. By continuously collecting on-line data and information from physical devices, digital twin maps Cyber Physical Systems (CPS) [3] to living digital models, that can help improve data analytics performance correspondingly also result in informed and precise decision making. In view of the above benefits, we propose integrating the digital twin model to edge networks to mitigate issues such as high transmission latency and low connection reliability caused by stochastic IoT networks.

Traditional cloud-based computing architecture can be used to build digital twin models, by collecting data and executing machine learning algorithms at the centralized server. For example, in IoT networks, the server can collect the running states and environment information of IoT devices, and develop their behavior model for different states of the environment to build their digital twins. However, the centralized computing scheme incurs much communication load consequently also leading to data security issues. To construct a digital twin network at the edge, the key challenge is to find a computing scheme that: a) can execute machine learning algorithms at the edge; b) can be applied to the distributed end IoT devices; c) can address the issues related to data security and user privacy protection.

In this regard, we propose integrating blockchain and federated learning in edge networks. Blockchain has attracted wide attention from academia and industry. By maintaining a tamper-proof, immutable distributed ledger, blockchain can establish trust among distributed users and enhance data security in a network. Many studies have exploited blockchain for resource management in Mobile Edge Computing (MEC) networks. For instance, in [4], the authors presented an edge intelligence and blockchain empowered Industrial IoT (IoT) framework, that achieves flexible and secure edge resource scheduling using cross-domain sharing and a credit-differentiated transaction approval mechanism. The integration of Artificial Intelligence (AI) and Blockchain has been studied.
for secure and intelligent resource allocation in 5G beyond networks. The authors in [5] exploited Deep Reinforcement Learning (DRL) for optimal and secure content caching policy, and proposed a Proof-of-Utility based blockchain verification scheme for vehicular edge computing. Although the integration of blockchain and AI has achieved enhanced performance, how to perform distributed learning with respect to data privacy, remains an open problem.

Federated learning [6] is an emerging paradigm that has received considerable attention recently. In traditional machine learning algorithms, user data is collected and processed by the centralized server, which results in a large amount of communication load and high possibility of data leakage. Federated learning mitigates the privacy concerns by allowing users to train machine learning models locally and requires the users to only send the model parameters to the server. It provides a privacy-preserving collaborative machine learning framework for distributed users. Considering such characteristics, federated learning has been applied as a key enabling component for IoT applications. The integration of federated learning and blockchain can be seen in some recent works such as [7]. In [7], the authors proposed a combined scheme for data sharing in IoT networks that leverages blockchain to store and verify the parameters of federated learning and accelerates the blockchain consensus process based on the federated learning results. Thus, the integration of blockchain and federated learning can enable a secure and reliable edge computing framework among untrusted distributed users.

However, due to limited computing and communication resources in IoT systems, integrating blockchain with federated learning for IoT applications faces new issues. While the computing performance can be improved through technologies such as MEC and computation offloading, the communication efficiency of the integrated scheme becomes a key challenge in IoT networks. With the increasing number of IoT devices and the limitation of wireless bandwidth, communication efficiency becomes one of the bottlenecks for performing large-scale federated learning in IoT scenarios.

In this paper, we first introduce the concept of digital twin edge networks (DITEN). By using DITEN, the communication and computation efficiency can be significantly improved to support the URLLC applications. Then, we propose a blockchain empowered federated learning scheme for DITEN, and formulate the DITEN model optimization problem. Finally, to improve the communication efficiency of the proposed scheme, we formulate the data relaying optimization problem and use Deep Neural Networks (DNN) as the strategy scheduler. The main contributions of this paper are summarized as follows.

- We introduce digital twin edge networks (DITEN), which contains the running states and behavior models of IoT devices, and interacts with IoT devices.
- We propose a blockchain empowered federated learning scheme for DITEN to enhance the learning security and data privacy of users. We adopt asynchronous model aggregation for the scheme which contributes in reducing delay due to straggler users.
- To improve the communication efficiency of our proposed blockchain empowered federated learning scheme further, we develop a reinforcement learning based algorithm for optimal user scheduling and bandwidth allocation.

The rest of this paper is organized as follows. Related work is discussed in Section II. The system model of the digital twin edge networks is introduced in Section III. The communication-efficient federated learning framework for DITEN is presented in Section IV. In Section V, our proposed lightweight blockchain and federated learning scheme for DITEN is described. The proposed method is evaluated in Section VI and the paper is concluded in Section VII.

II. RELATED WORK

Our proposed scheme is a framework based on several emerging and enabling technologies and algorithms such as the integration of blockchain and federated learning into the digital twin for edge networks (DITEN). We therefore divide literature review into three parts: (a) blockchain and AI for resource management in edge networks; (b) resource optimization with federated learning; (c) digital twin.

A. Blockchain and Artificial Intelligence for Resource Management in Edge networks

Recent years have witnessed the great success of blockchain technology. Researchers have applied blockchain in edge networks mainly to enhance data security. For instance, in [8], the authors exploited consortium blockchain and smart contract for data sharing in vehicular edge computing networks, and improved the quality of shared data by deploying a reputation mechanism. In [9], the authors proposed a joint transaction relaying and block verification optimization approach by using contract theory to accelerate the transaction verification process in congested areas.

Artificial intelligence has shown its promising potential for resource optimization in edge networks [10]. In [11], the authors adopted a deep Q-learning approach to design optimal transmission scheduling strategies for IoT systems, by selecting the optimal communication modes and spectrum resources based on their characteristics in different network states. In [12], the authors developed a distributed power allocation scheme by using model-free Deep Reinforcement Learning (DRL) to maximize a weighted utility function.

The integration of blockchain and AI has led to considerable developments in achieving secure and efficient optimization of resources in wireless networks by utilizing the merits from both technologies. In [13], the authors proposed a secure and intelligent resource sharing scheme by combining AI with blockchain, and further developed a content caching scheme by using DRL. The authors in [14] proposed a privacy-preserving Support Vector Machine (SVM) training scheme - secureSVM, and leveraged blockchain technology to build a secure and reliable data sharing platform among multiple data providers in IoT networks. However, conventional AI algorithms rely much on the centralized mechanism, which consumes a large amount of communication resources and thus are vulnerable to security threats. Moreover, due to
limited resources available at the distributed devices in edge networks, conventional centralized AI algorithms can lead to poor performance for low-latency applications, as a result of the delay associated with the straggler devices. Distributed AI algorithms have shown great potential to be applied in edge networks to address the above issues. Federated learning is an example that achieves great success in distributed edge networks, which protects data privacy and reduces the data transmission load in learning process.

B. Resource Optimization with Federated Learning

Federated learning is an emerging paradigm that has attracted wide attentions. It provides a new distributed machine learning framework, which can mitigate the risks of data leakage and thus enhance data privacy [15]. In [16], the authors proposed a proactive content caching scheme based on federated learning, that does not require collecting historical data from users. In [17], the authors introduced the concept of federated learning to solve an optimization problem involving heterogeneous power constraints and local data size of user equipment.

The increasing interest of the wireless networks research community in federated learning has resulted in a good volume of work [18] in this direction to improve the performance of federated learning through resource optimization. In [19], the authors designed a reputation-based worker selection scheme and leveraged blockchain to manage the reputation of workers securely in federated learning process. The authors in [20] proposed to integrate the DRL techniques and federated learning framework with mobile edge systems. The communication burden of federated learning can be alleviated by reducing the aggregation rounds and compressing the parameters to be transmitted. In [21], the authors proposed adapting FedAvg [6] by using a distributed form of Adam optimization, which reduces the number of rounds to convergence and improves the learning efficiency. The authors in [22] proposed a control algorithm for federated learning to minimize the loss function under a given resource budget by optimally adjusting the running times of local training and global aggregation. In [23], the authors employed federated learning for optimizing resource allocation in wireless networks.

The performance of the above work greatly relies on choosing the correct threshold, which is not an easy task. AI algorithms can adaptively decide the optimal strategy according to the dynamic network states. In studies such as [24], the authors utilized AI algorithms to improve the performance of federated learning. In [24], the authors proposed a blockchain empowered asynchronous federated learning architecture for data sharing in Internet of Vehicles (IoV) [25], and improved the convergence efficiency by optimally selecting the participating users based on the DRL algorithm. The integration of blockchain and federated learning provides an optimal distributed solution for edge computing and resource management in wireless networks. Due to typically limited resources in edge networks, however, the spectrum efficiency of the integrated scheme is still a major issue that demands considerable work for the idea of integrating AI and blockchain to be of practical value.

C. Digital Twin

Digital twin is an emerging paradigm that integrates the physical systems with cyber spaces. Digital twins interact with physical systems to remain synchronized with changes to the devices. The construction of digital twins is based on the analysis of a huge volume of data generated by devices, and the analysis results of digital twins can in turn improve the performance of physical devices. Digital twins have been applied in many industrials such as smart manufacturing [26],
health management [27], and system simulation [28]. For instance, in [26], the authors discussed the state-of-the-art of the key components and industry applications of digital twins. In [28], the authors introduced experimentable digital twins in different application scenarios, which provides new foundations for simulation-based systems engineering. In [29], the authors proposed a bi-level hybrid model named ManuChain, that consists of lower-level blockchain and upper-level digital twin models to deal with unbalance between holistic planning and local execution in manufacturing systems. The integration of digital twin technology into the edge networks can yield considerable improvement in both the latency performance and the computing efficiency of applications running on IoT devices and edge servers. Since the digital twins reflect the running states of the IoT devices, further computation and analysis can be executed directly on these digital twins. However, in edge networks such as IoT networks, due to the limited resources and distributed IoT devices, the application of digital twin remains unexplored.

III. SYSTEM MODEL

In this section, we consider a multi-user edge computing scenario in 5G beyond wireless networks. To enhance security and to ensure the edge computing performance, we propose a blockchain empowered federated learning architecture. Then we further formulate the edge computing problem based on our proposed architecture, that jointly takes into account both computation and communication aspects. The main notations in this paper are summarized in Table I.

TABLE I: Summary of main notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F(w)$</td>
<td>The global function in federated learning</td>
</tr>
<tr>
<td>$w_i(t)$</td>
<td>Local model parameters learned by vehicle $i$ in slot $t$</td>
</tr>
<tr>
<td>$n_i(t)$</td>
<td>Local model learned by vehicle $i$ in slot $t$</td>
</tr>
<tr>
<td>$M(e)$</td>
<td>Global model in episode $e$</td>
</tr>
<tr>
<td>$T_{cmp}^{l}(t)$</td>
<td>The local learning time cost of user $i$ in slot $t$</td>
</tr>
<tr>
<td>$T_{cmp}^{c}(t)$</td>
<td>The communication time cost of user $i$ in slot $t$</td>
</tr>
<tr>
<td>$E_{u}^{cmp}(t)$</td>
<td>The local learning cost of user $i$ in slot $t$</td>
</tr>
<tr>
<td>$E_{cmp}(t)$</td>
<td>The communication energy cost of user $i$ in slot $t$</td>
</tr>
<tr>
<td>$f(u_i)$</td>
<td>The CPU-cycle frequency of user $u_i$</td>
</tr>
<tr>
<td>$n_i(t)$</td>
<td>The communication time allocated to user $i$ in slot $t$</td>
</tr>
<tr>
<td>$\lambda_i$</td>
<td>The transmission relaying policy from user $i$ to user $j$</td>
</tr>
<tr>
<td>$T_p$</td>
<td>The waiting time threshold of the BS in each iteration</td>
</tr>
<tr>
<td>$s(t)$</td>
<td>The system state of slot $t$</td>
</tr>
<tr>
<td>$\Theta$</td>
<td>The parameters of DNN</td>
</tr>
</tbody>
</table>

A. Digital Twin Edge Networks

We consider a DITEN framework shown in Fig. 1. The DITEN consists of two planes: user plane and edge plane. The IoT devices $U = \{u_1, u_2, \ldots, u_N\}$ run in the user plane, and connect to the edge plane through wireless communications. The computation and communication resources are limited in the user plane. In the edge plane, each Base Station (BS) $B = \{B_1, B_2, \ldots, B_M\}$ is equipped with a Mobile Edge Computing (MEC) server, which has enough computing resource. The BSs are connected to each other through wired links. All BSs collaboratively maintain a permissioned blockchain, which securely stores related data and manages the participating nodes of DITEN. The digital twins of the IoT devices are constructed in the edge plane by edge servers. The running states of IoT devices such as the available computing and communication resources, are collected and processed by edge servers to build the corresponding digital twin models. Thus, the DITEN can be denoted as

$$\Gamma(t) = (s(t), M, \Delta s),$$  

where $M$ is the behavior model of IoT devices built by analyzing their historical running data, $s(t)$ is the running states of IoT devices. Moreover, the digital twin models keep interacting with the IoT devices to update the running information $\Delta s$ of devices in real time. The changes in IoT devices lead to the corresponding changes in digital twins, and the digital twins also provide feedback to IoT devices. The original construction of digital twins is executed offline at the initialization phase of the network, while the iterative incremental update of digital twins is executed online. By using the constructed digital twins, the analysis and computation of IoT devices can be executed directly on their digital twins. The long-distance transmission of the large amount of running data is replaced by the iterative incremental update of digital twins, which can further support low latency applications.

In our proposed scheme, the computation of digital twin models consists of two phases: the local computation and the aggregation. Local computation is executed at the user side, while aggregation is executed at the edge server. Let $f(u_i)$ be the CPU-cycle frequency of user $u_i$, and $f(s_j)$ be the CPU-cycle frequency of edge server $s_j$. The number of CPU cycles for $u_i$ to execute one sample of training data is denoted by $C_i$.

At the user side, the computation time for user $u_i$ to train the local model in one iteration is given by Eq. (2):

$$T_{u_i}^{cmp} = C_i |D_i| / f(u_i).$$  

Where $D_i$ is the local dataset of user $u_i$, which is a collection of input-output data samples $\{x_i, y_i\}$. The energy consumption for one iteration can be written as

$$E_{u_i}^{cmp} = \alpha_{u_i} C_i D_i (f(u_i))^2,$$

where $\alpha_{u_i}$ is the effective switched capacitance depending on the chip architecture [30]. At the edge server $s_j$, the computation time of aggregation is:

$$T_{s_j}^{cmp} = C_0 \sum_{i=1}^{N} |w_i| / f(s_j),$$

where $|w_i|$ is the size of collected local parameters from user $i$. The energy consumption can be written as:

$$E_{s_j}^{cmp} = \alpha_{s_j} C_0 \left( \sum_{i=1}^{N} |w_i| \right) (f(s_j))^2,$$

In our system model, the computing capability $f(s_j)$ of the BS is large, while the data size $\sum_{i=1}^{N} |w_i|$ is small. In addition, the aggregation time is much less compared with the training and update process. Thus, we consider the aggregation time as a negligible value.
B. Blockchain and Federated Learning Model for DITEN

We propose a blockchain empowered federated learning architecture for DITEN as shown in Fig. 1. The proposed architecture is a hierarchical solution with multiple levels. In the user plane, a set $U$ of $N$ users consists of data providers and requesters, which can be mobile devices and IoT devices. Transmitting all running data from IoT devices to construct the digital twins in edge servers incurs a large amount of communication load and may also lead to data privacy issues for IoT users. In our proposed architecture, the learning process is executed locally to train machine learning models on the data of various users. In the edge plane, the local models are aggregated to construct the digital twin models. The BSs, equipped with computing and caching resources, maintain the permissioned blockchain and store the parameters of federated learning collected from the IoT devices. The parameters are aggregated by the BSs to update the global model. All BSs execute the consensus process of the permissioned blockchain to achieve consistency in the global model. The blockchain also stores the registration information and key characteristics of digital twins.

- **Federated Learning in User Plane:** At the user side, the whole process can be divided into two phases: local training, and model parameter transmission. In the local training phase, each user trains a local model for caching policy based on its local state (e.g., location) and its content request history. The loss function on the dataset $D_i$ of user $i$ is

$$F_i(w) = \frac{1}{|D_i|} \sum_{j \in D_i} f_j(w)$$

(6)

where $w$ is the parameter vector of the trained model and $f_j(w)$ is the loss function depending on the machine learning algorithm.

We train the model parameters by using gradient descent algorithm, i.e.,

$$w_i(t) = w_i(t-1) - \eta \cdot \nabla F_i(w_i(t-1)),$$  (7)

where $w_i(t)$ is the learned model parameters of iteration $t$, $\eta$ is the learning rate, and $\nabla F_i(w_i(t-1))$ is the gradient of the loss function with parameters $w_i(t-1)$.

Then the model parameters are transmitted to the edge plane through wireless uplinks to the nearby BS.

- **Blockchain Empowered Global Aggregation in Edge Plane:** The BSs in the edge plane, which are nodes of the blockchain, collect all model parameters from participating users and store them as transactions. The BS aims at minimizing the global loss function through global aggregation, defined as:

$$F(w(t)) = \frac{1}{|D|} \sum_{i=1}^{N} \sum_{j=1}^{|D_i|} f_j(w_i, x_{ij}, y_{ij}).$$  (8)

In the global aggregation, the BS aggregates parameters to update the global model $w(t)$, according to Eq. (9):

$$w(t) = \frac{1}{\sum_{i=1}^{N} |D_i|} \sum_{i=1}^{N} |D_i| \cdot w_i(t).$$  (9)

The BS which gets the right to generate a block performs the computation of global aggregation. The updated global model is verified in the consensus process and is broadcasted to the participating users by the BS, to start another training iteration. The process is repeated until the loss function defined in Eq. (8) converges or the expected learning accuracy $\alpha_i (0 < \alpha < 1)$ is achieved.

The detailed time sequences of our proposed architecture are shown in Fig. 2. Each iteration can be split into two slots, local training and global aggregation, respectively. A local user $i$ first trains its local model in time $T_i^{\text{cmp}}$, and transmits the parameter vector to edge plane in time $T_i^{\text{com}}$. Due to the heterogeneity in computing and communication resources, the computing and communication time of different users may vary considerably. In some cases, the transmission channel of a user may be congested, which may hinder the convergence of the global learning process. To address this issue, we design the transmission scheduling policy to relay parameters from the users that can suffer unreliable communication channels to their nearby users that have better communication channels.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig2.png}
\caption{Time phases of the proposed architecture}
\end{figure}

C. Communication Model

The learned model parameters are transmitted to the edge plane by users after local computation. We consider using time sharing model such as Time Division Multiple Access (TDMA) for data transmission from users to the BSs. Note that the transmission model is not limited to time sharing model. Other schemes such as OFDMA can also be adopted in our scheme. The available time slots in transmission is denoted as $M$ and the time of each slot is $\tau_0$. The achievable transmission rate of user $i$ is defined as:

$$r_i(t) = B \log_2(1 + \frac{h_i(t)P_i(t)}{N_0})$$  (10)

where $B$ is the transmission bandwidth, $h_i(t)$ is the channel gain of user $i$ at time slot $t$, $N_0$ is the noise power, and $P_i(t)$ is the transmission power of user $i$. The duration time of the status message update $\Delta s(t)$ for a digital twin is denoted as:

$$T_{upd} = \frac{|\Delta s(t)|}{r_i(t)},$$  (11)

where $|\Delta s(t)|$ is the size of update state, and $r_i(t)$ is the transmission rate. Since the size of update $|\Delta s(t)|$ is a small value, the message update duration is much less compared to transmission of the model parameters. For example, the update message can be the CPU frequency of an IoT device.
or the location of a mobile device. Thus, we only consider the transmission time of model parameters in our scheme. Let us denote the size of model parameters $w_i(t)$ and $\nabla F(w_i(t))$ by $s_i(t)$, and time slots allocated to user $i$ at iteration $t$ by $m_i(t)$. The transmission time constraint of user $i$ is

$$\tau_0 m_i(t) \cdot r_i(t) \geq s_i(t)$$

(12)

where $\sum_{i=1}^{N} m_i = M$. The energy consumption $E_i^{\text{com}}$ for one iteration is

$$E_i^{\text{com}} = \tau_0 m_i(t) P_i(t) = \tau_0 m_i(t) \frac{N_0}{h_i(t)} e^{r_i(t) \lambda} - 1$$

(13)

IV. COMMUNICATION-EFFICIENT FEDERATED LEARNING FOR DITEN

A. Communication Resource Optimization

The limited computing and communication resources with the users demands methods to train the federated learning model with reduced resource consumption. The resource optimization is a combinatorial problem that introduces a tradeoff between learning accuracy and efficiency in terms of resource consumptions. The total time cost can be expressed as

$$T_{\text{total}} = \sum_{t=1}^{t_{\text{max}}} \left( \max_{u_i} T_i^{\text{com}} + \sum_{i=1}^{N} (\tau_0 m_i) \right),$$

(14)

where $t_{\text{max}}$ is the maximum iteration number of the federated learning process. The total energy consumption will be:

$$E_{\text{total}} = \sum_{t=1}^{t_{\text{max}}} \left( \sum_{i=1}^{N} (E_i^{\text{com}} + E_i^{\text{comp}}) + \sum_{j=1}^{M} E_j^{\text{com}} \right)$$

(15)

Our objective here is to learn an optimal federated model from distributed users, which also minimizes the weighted cost function. The optimization problem can be formulated as

$$\min_{f,m,\lambda} \frac{1}{|D|} \sum_{i=1}^{N} \sum_{j=1}^{|D_i|} f_j(w) + \beta(T_{\text{total}} + kE_{\text{total}})$$

(16)

s.t.

$$\lambda \in \{0, 1\},$$

(16a)

$$\sum_{i=1}^{N} m_i \leq M,$$

(16b)

$$f_i^{\text{min}} \leq f_i \leq f_i^{\text{max}}, \forall i \in \mathbb{N},$$

(16c)

$$P_i^{\text{min}} \leq P_i \leq P_i^{\text{max}}, \forall i \in \mathbb{N},$$

(16d)

$$0 \leq \beta \leq 1,$$

(16e)

where $\beta, k \in [0, 1]$ is the control factor that decides the trade-off between loss function and cost functions. The total execution time is $T_{\text{total}}$ and the total energy cost is $E_{\text{total}}$. $F(w)$ denotes the loss function of the model parameters learned by the federated learning scheme. $s$ is the delay factor, and $\lambda$ is the user scheduling policy vector that decides the relay users to transmit the nearby training parameters.

Problem (16) is a non-convex problem for which it is difficult to find a closed-form solution. The minimization of $\frac{1}{|D|} \sum_{i=1}^{N} F_i(w)$ is addressed by gradient descent algorithm, and the accuracy is determined by the exact learning algorithm executed on training data and the quality of local data. Thus, we simplify problem (16) to

$$\min_{f,m,\lambda} \left( T_{\text{total}} + kE_{\text{total}} \right)$$

(17)

It can be observed that solution $f$ is decided by the end users, while the user scheduling strategies $\lambda$ is decided by the BSs. Compared with end users, the BSs have sufficient power supply and we donot consider the energy consumptions of BSs. Thus, the objective of the end users is

$$\min_{f,p} (T_{\text{comp}} + kE_{\text{total}})$$

s.t.

$$f_i^{\text{min}} \leq f_i \leq f_i^{\text{max}}, \forall i \in \mathbb{N},$$

(18a)

$$P_i^{\text{min}} \leq P_i \leq P_i^{\text{max}}, \forall i \in \mathbb{N},$$

(18b)

The objective for communication optimization at the BS is

$$\min_{\lambda,m} T_{\text{total}}$$

s.t.

$$\lambda \in \{0, 1\},$$

(19a)

$$\sum_{i=1}^{N} m_i \leq M,$$

(19b)

$$E_{\text{total}} \leq E_{\text{Th}},$$

(19c)

where $E_{\text{Th}}$ is the expected energy consumption threshold of the total end user devices in the DITEN. Problem (18) is a CPU-cycle control problem which can be solved by algorithms in [17]. We then first utilize an asynchronous aggregation scheme to improve the efficiency of communication process, and use DRL to find the optimal relay user scheduling strategies to approximate the optimal solution to Problem (19).

B. Asynchronous Model Update and Aggregation

In conventional synchronous schemes, all users send their model parameters to a centralized server, and wait for the aggregated global model from the server. However, due to the varying communication conditions, two issues need to be addressed in our proposed scheme. First, users have to spend much time waiting for the slow users and the global aggregation in blockchain. Second, the concurrent transmission of parameters exacerbates the strain on channel resources and may cause communication congestion.

We propose an asynchronous scheme to transmit the model parameters and retrieve the global model, for reducing waiting time and mitigating communication load. Each of the $N$ users has a separate iteration index $c$ that records the progress of its training process. The edge servers also maintain a global iteration index which records the aggregation progress in the blockchain. At the beginning, $t = 0$, the servers (BSs) broadcast an initial model $w(0)$ to all participating users. In iteration $t$, the users compute new $w_i(t)$ in the next iterations by calculating the gradient-descent of the local loss function. The new $w_i(t)$ is transmitted to the edge server. The new $w_i(t)$ needs to be approved by verifiers to achieve consistency among the blockchain nodes. The process is time-consuming and the $w_i(t)$ cannot be obtained by other participating users.
immediately. We introduce a man-made delay to mitigate the problem. All transmitted models are stamped with their iteration index as timestamps. The participating users retrieve the blockchain to get a new global model for training once they send their local models. Instead of returning the global model in iteration $t$, the edge server returns the former approved global model in $t-s$ iteration, where $s$ is a delay factor. Since the blockchain records all model updates in each iteration, a user uploading its local model with index $t$ can obtain all models with index $t_0 \leq t-s-1$, where $s$ is an adaptive delay step that is designed according to the aggregation and verification progress of the blockchain. If $s$ is too large, much delay is added to the update process, which may hinder the convergence of the system. If $s$ is too small, the edge server cannot return the global model in time. Thus $s$ should be updated timely based on the newest progress of blockchain.

Each BS collects the local models from users within its coverage. Then it aggregates these local models and sends the results to other participating BSs. In conventional synchronous scheme, the BS spends much time waiting for all the models of users. In our asynchronous scheme, the BS starts aggregation after a time period $T_p$ to facilitate rapid convergence of the process. The detailed procedure of our proposed asynchronous gradient descent scheme is provided in Algorithm 1.

### Algorithm 1 Asynchronous Model Update and Aggregation

**Input:** delay step $s$, time interval $T_p$

1. Initialize the global model $M_0$
2. for each iteration $t$ do
3. Select a leader $r_0$ from delegates
4. for each user $u_i \in U$ do
5. $u_i$ trains its local model $m_{u_i}$ on its data $d_i$
6. $u_i$ sends $m_{u_i}$ to its nearby BS
7. end for
8. for each BS $s_j$ do
9. $s_j$ waits for local models for period $T_p$
10. Once receiving a model, the BS sends $M_{t-s}$ to the provider for training
11. $s_j$ aggregates the local models $m_{u_i}$ and send the results to other BSs
12. end for
13. The BSs, which also act as the blockchain nodes, run the consensus process to obtain a synchronous global model $M_t$
14. Approve $M_t$ and add it to blockchain
15. end for

The participating users can also obtain new global models from other nearby users, which can considerably reduce their communication load with BSs. Optimizing scheduling strategies of data transmission can further improve communication efficiency. Thus, the global federated learning model in iteration $t$ is given by:

$$w_g(t, T_p, \theta) = \frac{1}{\sum_{i=1}^{N} D_i} \sum_{i=1}^{N} D_i |\theta_i| \cdot w_i(t). \quad (20)$$

where $\theta_i = 1$ if $\Delta t_i \leq T_p$ and $w_i(t)$ is approved in the blockchain, $\theta_i = 0$ otherwise.

### C. DNN based User and Resource Scheduling for Parameter Transmission

Due to the heterogeneous communication resources of end devices, users with poor communication conditions may slow down the whole transmission process. To improve the communication efficiency in the transmission phase, we relay the transmission tasks of the “straggler” to the users with higher communication capability. Thus, the IoT nodes can be divided into active nodes and idle nodes according to their transmission states.

We propose a user association scheme for relaying the parameter data to improve communication efficiency. Denote a model $w_i$ trained by $u_i$ with poor communication capability, and a potential user $u_j$ that is capable of relaying the transmission tasks of $u_i$. The scheduling policy is to decide whether $u_j$ should be relayed from $u_i$ to $u_j$, that is, to obtain the value of $\lambda_{i,j} \in \{0, 1\}$. We consider associating $u_i$ to $u_j$ as a pair $\langle u_i, u_j \rangle$. As depicted in Fig. 3, if $u_i$ is relayed from $u_i$ to $u_j$, then $\lambda_{i,j} = 1$. The bandwidth resources are then allocated to relay users denoted as $m_i$ according to their states.

![Fig. 3: User scheduling for parameter data transmission](image-url)

We use the DNN to generate the user association policy. The DNN is denoted by $\Theta = \{W[l], b[l]\}$, where $l \in \{1, ..., L\}$ denotes the $l$-th layer of the DNN, $W[l]$ is the weight vector and $b[l]$ is the bias vector. The DNN is established according to the following equation:

$$Y[l] = f(W[l] X[l] + b[l]), \quad (21)$$

where $X[l]$ is the input, $Y[l]$ is the output, and $f(x)$ is the activation function. In the learning process, the states of the system such as the achievable transmission rate and the computing capability of each IoT device, are the input in the DNN. The optimal relay policy $\lambda$ is the output of the DNN. In each epoch, the training samples $(r, f, \lambda, m, T)$ are used to train the DNN models.

In our proposed DITEN system, two main issues need to be addressed. First, how to reduce the scale of training data. The DNN requires a large amount of data to train the parameters. However, in IoT networks, obtaining sufficient training data, as well as processing a large amount of data, both are challenging. In our proposed DITEN, the storage and processing of massive data get much difficult because the
maintaining of digital twins also consumes a certain amount of resources. Second, how to get the training data. Since the network states vary with time, it is not trivial to have access to data representing network state. We proposed to use DRL to train the policy DNN, where the DRL agents are the BSs and learn the optimal transmission strategies under various states. The DRL agents obtain system states by interacting with digital twin models.

D. The DRL Agent for Transmission Optimization

The DRL agents are the BSs in our DITEN. Each DRL agent allocates its bandwidth resources to users under its coverage. At the beginning of each iteration, the resource allocation actions \((\lambda_{i}, m_{i})\) are decided based on the current system states obtained from digital twins. The performance of the action is quantified by the reward function at the end of the iteration.

- **State:** The system state at iteration \(t\) is defined as
  \[ s_{t} = (F(t), f(t), r(t), w(t)) \]  
  where \(F(t)\) is the loss value vector of each user, \(f(t)\) is the CPU capability vector that can be used to estimate the local computation time. \(r(t)\) is the achievable data rate vector that is determined by the bandwidth allocation policy. \(w(t)\) is the learned parameters of users. These state metrics reflect the running states of user devices, which provides the basis for user scheduling and bandwidth resource allocation decisions.

- **Action:** The actions in our DITEN consist of two aspects: user scheduling and bandwidth resource allocation. The user scheduling choose the active users for parameter transmission, which are also the relay nodes of inactive users. We use vector \(\lambda\) to denote the user scheduling actions. The bandwidth allocation action vector is denoted by \(m\), where \(m_{i}\) is the transmission slots allocated to user \(i\) in a time-sharing protocol. The assigned \(m_{i}\) should not exceed the total bandwidth slots \(M\).

- **Reward:** The constraints such as \(f_{i}^{\text{min}} \leq f_{i} \leq f_{i}^{\text{max}}\), \(P_{i}^{\text{min}} \leq P_{i} \leq P_{i}^{\text{max}}\) in the optimization problem are also checked at the end of an iteration. Moreover, the total energy cost \(E_{\text{total}}\) is also checked as a budget that should not exceed an expected value. We define the reward as
  \[ R(t) = \begin{cases} -\eta T_{\text{total}} + C & \text{if check } f_{i}, P_{i}, \text{ and } E \leq E_{\text{total}}, \\ -\eta T_{\text{total}} - C & \text{otherwise}. \end{cases} \]  
  The reward \(R\) is negatively correlated with the completion time of the iteration. If the metrics after taking actions all pass the constraint check, then a positive reward will be added. Otherwise a negative penalty will be added.

In our DITEN, each DRL agent learn to achieve the maximum cumulative reward
\[ E[\sum_{t=0}^{T-1} \gamma R(s_{t}, \lambda_{t}, m_{t})], \]  
where \(\gamma\) is the discounting factor for future reward. The training process leads the DRL agent to find the optimal solution to Problem (19).

E. Digital Twin Empowered Policy DNN training in Reinforcement Learning

The reinforcement learning is adopted to train the policy DNN for user scheduling and bandwidth allocation. We use the digital twin empowered exploration to obtain the experience training data of DNN, as shown in Fig. 4. The digital twins are composed of the models learned from the IoT devices and their corresponding running states. The digital twin empowered DNN consists of two phases: training and decision making. In the training phase, we collect the system running states from digital twins, including the achievable data rates \(r(t)\), and the available computing capability \(f(t)\), as
\[ s(t) = (F(t), r(t), f(t), w(t)). \]  

The relaying policies towards network states are explored and generated based on our system model. The states \(s(t)\) are the input to DNN \(\Theta\). The output vector \(\lambda\) and \(m\) are calculated according to
\[ \lambda = \sum_{i=1}^{L} f(W^{[l]}X^{[l]} + b^{[l]}). \]  

The ReLU function, \(f_{\delta}(x) = \max(0, x)\), is used as the activation function in our proposed DNN. Thus the variables in \(\lambda\) are continuous values. We convert the continuous values into relaying policies by letting each transmitter \(i\) choose the highest \(\lambda \in \sum_{j} \lambda_{ij}\) as the receiver. We denote the original output policy as \(\lambda(0)\). We then compute the cost of the transmitting policy according to the system model as in Eq. (16), denoted as \(Q(s(t), \lambda(0))\). Note that \(\lambda(0)\) is usually not the best policy.

We start exploring the best relaying policy towards state \(s(t)\) through the exploration process. We randomly change the 0-1 values of \(\lambda\) and allocate the corresponding transmission time \(m_{i}\) to active users. The corresponding system cost is calculated to obtain the reward. If we change \(\lambda_{ij} \in \lambda\) from 0 to 1, where \(r_{i} \leq r_{j}\) and \(f_{i} \geq f_{j}\) it means we relay \(w_{i}\) from \(u_{i}\) to \(u_{j}\), the reduced cost can be defined as
\[ \Delta r_{i}(\lambda_{ij}) = \left( \frac{C_{0}s_{0}}{f_{i}} + \frac{s_{0}}{r_{i}(t)} \right) - \left( \frac{C_{0}s_{0}}{f_{j}} + \frac{2s_{0}}{r_{j}(t)} \right) \]
Thus, the exploration is formulated as
\[
\max_{\lambda_{ij}} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \lambda_{ij} \cdot \Delta t_{ij}(\lambda_{ij})
\]
\[
\text{s.t. } \lambda_{ij} \in \{0, 1\},
\]
\[
\sum_{i=1}^{N} \sum_{j=i+1}^{N} B_{ij} \leq B_{Th},
\]
Eq. (28) is a 0-1 Knapsack problem. The approximate solution can be obtained by using a heuristic algorithm based on the greedy method. We change the relaying policy by adjusting $\lambda$ and $m$ based on Eq.(28). We then choose the $(\lambda, m)$ that minimizes the time cost in communication. The $(s(t), \lambda(t), m(t), R(t))$ is saved into replay memory as training samples. The parameters of policy DNN are then updated in the training process with the object to reduce the loss function. The training stops after the loss function satisfies an expected threshold. The trained policy DNN model is then used to generate the instant optimal relaying policies towards the real-time system states collected from digital twins.

V. LIGHT WEIGHT BLOCKCHAIN AND FEDERATED LEARNING FOR DITEN

Due to intensive resource consumption and high latency associated with traditional blockchain schemes, it is of less practical value to utilize a traditional blockchain in edge networks despite the merits that it offers. We therefore propose a light weight blockchain scheme for digital twin edge networks and improve the blockchain scheme for efficient integration with the federated learning process.

A. Distributed Consensus for Learning

In our proposed architecture, the consistency of data in permissioned blockchain is guaranteed through the form of distributed consensus. We use multiple permissioned blockchains to aggregate the local models generated by IoT devices under their coverage. We exploit blockchain for synchronizing these models and achieving consensus between different BSs. Since the global model needs to be confirmed as blockchain transactions, the running efficiency of blockchain is of vital importance to the whole learning process.

We develop a lightweight verification scheme for our permissioned blockchain based on DPoS. In DPoS, the verifiers are selected based on their stakes. In our scheme, the stakes are earned by the computing contribution to the global model. The aggregated models and corresponding local models are stored as blockchain transactions, as shown in Fig. 5. To adapt to the training process, we verify the models generated in each learning iteration by executing the transaction verification process. Different from conventional transaction verification, we not only verify the regular terms of transactions, but also verify the quality of the models, based on the historical model. Our consensus process consists of three main steps: multi-aggregation, verifier selection, model verification.

- **Multi-aggregation**: In each iteration, the participating BSs first aggregate the local models collected from IoT devices under their coverage. The aggregation process is executed in parallel, according to Eq. (9).
- **Verifier Selection**: In our scheme, all participating IoT devices act as blockchain users. They hold stakes that represent their contribution to the trained model such as the size of data and the quality of local models. The blockchain users vote for their preferred BSs as verifiers according to their computing and communication capabilities. If the number of participating BSs exceeds a threshold $k$, then the BSs with top $k$ votes are elected as verifiers. Otherwise, all participating BSs are selected as verifiers. The leader verifier is elected based on the votes and the random factor, which also has the right to generate a new block.
- **Model Verification**: In the transaction verification process, each BS sends its aggregated model to other verifiers for verification. In addition to the regular checks, the verifiers also verify the received models based on whether the model makes positive contribution to update last global model $w(t-1)$ to the new global model $w(t)$, i.e.,
\[
\frac{|w(t-1) - w_i(t)||^2}{w(t)} \leq \frac{|w(t-1)||^2}{w(t)}
\]
The leader verifier collects the verification results from all verifiers and confirms the transactions. The confirmed global model is then transmitted back to the participating users for training. The leader verifier, that is also the block manager, packs the confirmed transactions into a candidate block after a certain interval. The verifiers then verify the items of the candidate block such as its format and signature. The audited blocks are added to the blockchain and broadcasted to the BSs for storage.

B. Integrating Permissioned Blockchain into DITEN

We use digital twin at the edge plane to model the running states of IoT devices. To improve the reliability and security of digital twins, we further integrate permissioned blockchain to digital twins in our proposed scheme.

The IoT devices in our system register to the permissioned blockchain to obtain a unique identity (ID). The digital twin of each device consists of running rules and behavior models of the IoT devices, together with some running state data. We store the digital twin model in the permissioned blockchain,
and record the changing history of the models like a model chain. Storing the digital twin model instead of the original running data of IoT devices can dramatically reduce the computation and storage burden of the permissioned blockchain.

The verifiers also verify these models as transactions to achieve consistency. The time that verifier \( i \) takes to execute the consensus process will be

\[
t_i^k = \frac{|w|}{f_i} + d_i|w|N,
\]

where \( |w| \) is the size of model transactions, \( f_i \) is the CPU cycles and \( d_i \) is the average delay in transaction broadcasting and \( N \) is the number of verifiers.

The synchronization of digital twins with cyber physical devices is accomplished by using smart contracts in permissioned blockchain. When the state update data from IoT devices is received by the blockchain, the smart contract is triggered to match its corresponding digital twins and perform further analysis of the updated data. The processed data is then used to update the parameters of the digital twin to keep consistency with the physical IoT devices.

The complete process of our proposed blockchain empowered federated learning scheme for DITEN is summarized as Algorithm 2.

**Algorithm 2** The blockchain empowered federated learning for DITEN

**Input:** The registration of users as participating nodes \( U = \{u_1, u_2, ..., u_N\} \), the data of user \( i \), \( d_i \in D \).

1. Initialize the permissioned blockchain \( B \) and global model \( M_0 \). Distribute the initial global model \( M_0 \).
2. **for** each episode \( e \) **do**
3. **for** each time slot \( t \) **do**
4. Select relay users and allocate bandwidth resources
5. **for** each relay user \( u_t \in U \) **do**
6. \( u_t \) retrieves global model \( M_{t-1} \) from permissioned blockchain \( B \)
7. \( u_t \) executes local computation on its local data \( d_t \)
8. \( u_t \) updates its local model and uploads parameters to the nearby BS \( s_j \)
9. **end for**
10. BS \( s_j \) aggregates the models and issues a transaction to the blockchain
11. \( s_j \) collects all transactions and broadcasts \( M_j(e) \) to other BSs for verification
12. **end for**
13. Delegates vote for leader verifier \( s_L \). \( s_l \) collects \( M_j(e) \) and aggregates them to a global model \( M(e) \)
14. \( s_L \) broadcasts global model to other verifiers for verification
15. The confirmed global model \( M(e) \) and local models are added to the permissioned blockchain
16. **end for**
17. **return** The parameters of the final global model \( M \).

**VI. NUMERICAL RESULTS**

We conduct a series of experiments and simulations to evaluate the performances of our proposed architecture for DITEN. The performance of federated learning is tested on the real-world MNIST dataset [31] and the Fashion-MNIST [32]. For each dataset, the training set contains 60,000 examples and the testing set contains 10,000 examples. The learning task is to recognize the images in the dataset, which is a common task in MEC applications such as traffic detection and intelligent camera. We use the neural network in TensorFlow [33] as the local training model of federated learning. We consider an edge network with 100 end users and 4 BSs. The samples in datasets are shuffled and assigned to the users randomly in our experiments.

Our proposed federated learning is also compared with the conventional federated learning, where all users participate in the aggregation synchronously, and the spectrum resource is equally allocated to all users. Compared with the benchmark algorithm, the additional complexity of the proposed algorithm is mainly caused by the DRL algorithm that is used to optimize the communication resources. The complexity of the DRL algorithm is \( O(L_n) \), where \( n \) is the number of users and \( L \) is the number of episodes.

![Fig. 6: The learning accuracy](image)

**Fig. 6:** The learning accuracy results of our proposed federated learning and the benchmark approach are shown in Fig. 8. Our proposed scheme considerably reduces the system cost compared with the benchmark approach, and the system cost increases slightly as the size of training model increases. Fig. 9 shows the communication time cost of each iteration in our proposed algorithm and in the benchmark approach. The communication time in our proposed scheme is much less than the benchmark approach. The reason is that in resource limited edge networks, all end users share the limited bandwidth resources to transmit their parameters in different time. The benchmark approach incurs...
Policy Gradient (DDPG) to train the resource scheduling and allocate more bandwidth resources to them.

much time cost to wait for the synchronous transmission. While in our proposed asynchronous scheme, we optimally relay the parameter transmission tasks to a subset of users and allocate more bandwidth resources to them.

In this paper, we focused on improving the security and efficiency of edge computing in IoT networks. We have introduced the concept of digital twin into edge networks to model the running states of IoT networks. We have proposed a blockchain empowered federated learning architecture for DITEN, which provides a secure solution for distributed edge computing with respect to data privacy. To further improve the aggregation performance of the proposed scheme, we have provided an efficient asynchronous gradient descent mechanism. In addition, we have proposed a digital twin empowered DRL approach for optimal user and resource scheduling to enhance communication efficiency of our proposed federated learning in DITEN. Extensive numerical results have shown that the integrated blockchain and federated learning scheme results in considerably higher efficiency and comparable accuracy with benchmark approach, while also enhancing the security of user data in our proposed DITEN.

Due to the huge action space, we use Deep Deterministic Policy Gradient (DDPG) to train the resource scheduling model in DITEN. Compared with conventional DRL, we improve the policy exploration module according to digital twin empowered strategies as mentioned before. The DRL agents are trained by iteratively interacting with the digital twin models. The learning results of the DRL agent are shown in Fig. 10. The agent learns to maximize the total reward by exploring different data relaying policies. The normalized cumulative rewards with different learning rates both converge to fixed values in a certain number of learning iterations. Our proposed DRL-based approach can find the optimal solutions for user scheduling and bandwidth allocation.

**VI. CONCLUSION**

In this paper, we focused on improving the security and efficiency of edge computing in IoT networks. We have introduced the concept of digital twin into edge networks to model the running states of IoT networks. We have proposed a blockchain empowered federated learning architecture for DITEN, which provides a secure solution for distributed edge computing with respect to data privacy. To further improve the aggregation performance of the proposed scheme, we have provided an efficient asynchronous gradient descent mechanism. In addition, we have proposed a digital twin empowered DRL approach for optimal user and resource scheduling to enhance communication efficiency of our proposed federated learning in DITEN. Extensive numerical results have shown that the integrated blockchain and federated learning scheme results in considerably higher efficiency and comparable accuracy with benchmark approach, while also enhancing the security of user data in our proposed DITEN.

**REFERENCES**


Sabita Maharjan (SM’19) received her Ph.D. degree in Networks and Distributed Systems from University of Oslo, and Simula Research Laboratory, Norway, in 2013. She is currently a Senior Research Scientist in Simula Metropolitan Center for Digital Engineering, Norway, and Associate Professor (adjunct position) in University of Oslo, Norway. She worked as a Research Engineer in Institute for Infocomm Research (I2R), Singapore in 2010. She was a Visiting Scholar at Zhejiang University (ZU), Hangzhou, China in 2011, and a Visiting Research Collaborator in University of Illinois at Urbana Champaign (UIUC) in 2012. She was a Postdoctoral Fellow at Simula Research Laboratory, Norway from 2014 to 2016. She publishes regularly in prestigious journals in her field such as 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. She serves/has served as the Guest Editor for journals such as IET Cyber Physical Systems and IEEE Access, and in the technical program committee of conferences including top conferences like IEEE INFOCOM and IEEE IWQoS. Her current research interests include vehicular networks and 5G, network security and resilience, smart grid communications, Internet of Things, and computational intelligence.

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