Digital Twin-Empowered Communication Network Resource Management for Low-Carbon Smart Park

Xiaoyu Su*, Zehan Jia*, Zhenyu Zhou*, Zhong Gan†, Xiaoyan Wang‡, Shahid Mumtaz§
*State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources,
School of Electric and Electronic Engineering, North China Electric Power University, 102206, Beijing, China.
Email: suxiaoyu, zh_jia, zhenyu_zhou@ncepu.edu.cn
†State Grid Shanghai Municipal Electric Power Company, Shanghai, China.
Email: wxhua75@sina.com
‡Graduate School of Science and Engineering, Ibaraki University, Ibaraki-ken, Japan.
Email: xiaoyan.wang.shawn@vc.ibaraki.ac.jp
§Instituto de Telecomunicações, Aveiro, Portugal.
Email: smumtaz@av.it.pt

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838789

***Abstract*—The low-carbon operation of smart park requires to deploy massive internet of things (IoT) devices to provide real-time monitoring and control services. Digital twin (DT) provides accurate guidance for communication network resource management in low-carbon smart park by establishing a digital representation of physical entities. Facing the strict requirements of DT on delay and accuracy, as well as the constraints of access priority and energy consumption, we propose a federated learning-based DT framework and a Latency-awarE diGital twIn assisted resOurce maNagement algorithm (LEGION). LEGION can achieve a well tradeoff between delay and accuracy performances under the long-term constraints of access priority and energy consumption. Compared with existing algorithms, LEGION has superior performance in average iteration delay, DT loss function, energy consumption, and access priority deficit.**

***Index Terms*—Digital twin, low-carbon smart park, federated learning, resource management**

I. Introduction

In recent years, environment pollution has posed challenges to the rapid economy development. Building low-carbon smart park has been investigated recently to improve the energy utilization efficiency and reduce carbon emissions [1]. Massive internet of things (IoT) devices such as temperature and current sensors are deployed in the low-carbon smart park to provide real-time monitoring and control, which impose critical requirements on low-latency data transmission and processing [2]. To further improve resource utilization efficiency and satisfy quality of service (QoS) demands, the multi-dimensional resources in terms of communication, computing, energy, and storage should be dynamically and intelligently managed.

Digital twin (DT) can assist low-carbon smart park to realize communication network resource management from the perspective of digital world [3]. DT collects real-time states from massive devices to establish a digital representation of all the physical entities, thereby enabling accurate decision guidance for multi-dimensional resource management [4]. However, when constructing DT and utilizing it for assisting resource management, two critical challenges have to be addressed, which are summarized as follows.

Strict Requirements of DT on Delay and Accuracy: DT needs to be updated in low latency to maintain real-time consistency with physical entities. Moreover, the loss function which reflects the deviation of the DT from the actual value has to be minimized to improve the accuracy of resource management.

Access Priority and Energy Consumption Constraints: Considering the service importance and the limited battery capacity of devices, it is necessary to establish long-term access priority and energy consumption constraints to ensure the access priority of important devices and reduce the energy consumption. However, in these previous works, the long-term constraints are coupled with the short-term resource management optimization.

Federated learning (FL) addresses privacy disclosure concerns through isolating global model training from raw data uploading. Each device trains a local model and only uploads the model parameters to the server instead of the entire data set. There exist some works on employing FL for communication delay reduction and model accuracy improvement. In [5], Shi *et al.* proposed a FL-based joint device scheduling and resource allocation policy to maximize model accuracy under delay constraint. In [6], Saha *et al.* proposed a fog-enabled FL framework to reduce communication delay and energy consumption without affecting the global model's convergence rate. Some research attempts have focused on employing FL for DT construction. In [7], FL was combined with DT to mitigate the conflict between model training and privacy protection. In [8], Lu *et al.* proposed a blockchain-empowered FL architecture to achieve reliable collaborative computing in a DT-assisted wireless network. However, the coupling between resource management and long-term constraints of energy consumption and access priority are not considered.

Considering the above challenges, we propose a Latency-awarE diGital twIn assisted resOurce maNagement algorithm (LEGION) for low-carbon smart park. First, we develop a

FL-based DT construction framework, and formulate the DT-assisted resource management problem. The objective is to minimize the total iteration delay under the long-term access priority and energy consumption constraints through the joint optimization of access scheduling and computational resource allocation. Then, we decouple the short-term resource management optimization and long-term constraints based on virtual queues. Next, we combine deep Q-learning (DQN) [9] and Lagrange dual decomposition to solve the computational resource allocation and device access scheduling joint optimization problem.

The main contributions of this work can be summarized as follows.

- *Low-Latency and Accurate DT:* We jointly optimize the total iteration delay and the loss function of DT to achieve a well tradeoff between the delay and accuracy performances.
- *Access Priority and Energy Consumption Awareness:* We convert the long-term access priority and energy consumption constraints into queue stability constraints. The access priority and energy consumption awareness are achieved by optimizing resource management in according with access priority and energy consumption queue backlogs.

II. System Model

We propose a FL-based DT framework for low-carbon smart park as shown in Fig. 1. The objective is to set up a digital model $\mathcal{M}$ for the physical networks by FL. FL enables each device to train local models by using local limited knowledge, and integrates distributed knowledge into the global network through federated aggregation. The proposed framework mainly consists of three layers, i.e., the device layer, the edge layer, and the DT layer. In the device layer, each device performs local model training and transmits model parameters to the edge layer. The edge layer consists of a base station (BS) and a co-located edge server to provide access and computing services. Based on the collected local model parameters, the edge server performs edge aggregation and obtains a global model. The DT layer is constructed by the edge server, which achieves real-time interaction with devices through data collection. Meanwhile, DT can assist the edge server to optimize network resource management. The acronyms are summarized in Appendix A.

A. Local Training

There exist I devices in the considered scenario, the set of which is denoted as $\mathcal{U} = \{u_1, \cdots, u_i, \cdots, u_I\}$. We partition the total duration into T iterations, the set of which is defined as $\mathcal{T} = \{1, \cdots, t, \cdots, T\}$. We assume that the channel state information (CSI) remains unchanged within an iteration, but varies across different iterations [10]. At the t-th iteration, u_i downloads the global model of the $(t-1)$-th iteration as the local model $\mathcal{M}_i$, and sets the parameters of $\mathcal{M}_i$ as that of the global model, i.e., $\boldsymbol{\omega}_i(t-1) = \boldsymbol{\omega}_g(t-1)$. A local dataset $\mathcal{D}_i$ is utilized to train $\mathcal{M}_i$ [11]. We denote the required CPU cycles for executing a data sample as ξ_i and the CPU-cycle

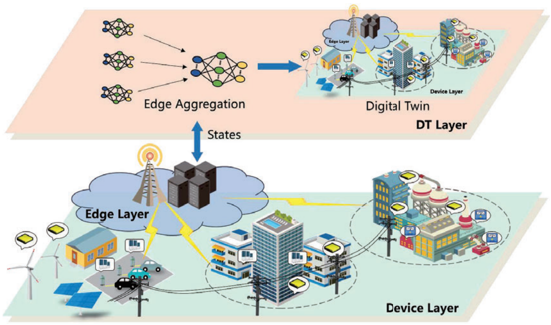

Fig. 1. FL-based DT framework for low-carbon smart park.

frequencies allocated for local training as $f_i(t)$. The energy consumption and delay of local training are calculated as

$$E_i^L(t) = \alpha \xi_i D_i f_i^2(t),\ \tau_i^L(t) = \frac{\xi_i D_i}{f_i(t)}, \tag{1}$$

where α is the energy consumption coefficient, and D_i represents the number of data samples in $\mathcal{D}_i$.

Let ζ_j and φ_j refer to the input and target output of a sample in dataset $\mathcal{D}_i$. The one-sample loss function $f(\boldsymbol{\omega}_i(t-1), \zeta_j, \varphi_j)$ is defined as the deviation of the real output from the target output. Therefore, the loss function of u_i on dataset $\mathcal{D}_i$ is given by

$$F_i(\boldsymbol{\omega}_i(t-1), t) = \frac{1}{D_i} \sum_{\zeta_j, \varphi_j \in \mathcal{D}_i} f(\boldsymbol{\omega}_i(t-1), \zeta_j, \varphi_j). \tag{2}$$

The loss function can be used to measure the accuracy of the local model and guide the updating of $\boldsymbol{\omega}_i(t)$ by utilizing the gradient descent method [12], i.e.,

$$\boldsymbol{\omega}_i(t) = \boldsymbol{\omega}_i(t-1) - \eta \bigtriangledown F_i(\boldsymbol{\omega}_i(t-1), t), \tag{3}$$

where $\eta > 0$ is the learning step.

B. Transmission Model

We assume that the BS can only select $N(t) \ll I$ devices for model uploading in each iteration. A device is unavailable for local training and parameter transmission if it runs out of battery. We assume that the number of available devices is larger than $N(t)$ due to $N(t) \ll I$, the set of which is denoted as $\mathcal{S}(t) \subseteq \mathcal{I}$. Denote the device access scheduling variable as $x_i(t) \in \{0, 1\}$, where $x_i(t) = 1$ indicates that u_i is scheduled to upload its model parameters in iteration t, and $x_i(t) = 0$ otherwise. When $x_i(t) = 1$, u_i transmits the model parameters $\boldsymbol{\omega}_i(t)$ and the state information s_i to the edge server for the construction of its DT, which is given by

$$DT_i = \Gamma(\mathcal{M}_i, \mathcal{D}_i, s_i). \tag{4}$$

Denote $|\boldsymbol{\omega}_i(t)|$ as the size of $\boldsymbol{\omega}_i(t)$, and the transmission delay of s_i is ignored since the size of s_i is very small. Denoting B_i as the subchannel bandwidth, and $P_i(t)$ as the transmission power, the transmission rate is given by

$$r_i(t) = B_i \log_2 \left(1 + \frac{h_i(t) P_i(t)}{I_i(t) + N_0}\right), \tag{5}$$

where $h_i(t)$ represents the channel gain. N_0 and $I_i(t)$ denote the noise power and the electromagnetic interference power, respectively. The energy consumption and delay of parameter transmission are calculated as

$$E_i^U(t) = \frac{P_i(t)|\boldsymbol{\omega}_i(t)|}{r_i(t)},\ \tau_i^U(t) = \frac{|\boldsymbol{\omega}_i(t)|}{r_i(t)}. \tag{6}$$

C. Packet Error Rate Model

The edge server adopts cyclic redundancy check (CRC) to check packet error of the received model parameters. The packet error rate of CRC is expressed as

$$q_i(t) = 1 - \exp\Big(-\frac{k(I_i(t)+N_0)}{P_i(t)h_i(t)}\Big), \tag{7}$$

where k represents the waterfall threshold [13].

Through CRC, the existence of parameter transmission error can be detected. Denote a binary indicator as $a_i(t) \in \{0,1\}$, where $a_i(t) = 1$ represents no error occurs, and $a_i(t) = 0$ otherwise. $a_i(t)$ is obtained by

$$a_i(t) = \begin{cases} 1, \text{with probability } 1 - q_i(t), \\ 0, \text{with probability } q_i(t). \end{cases} \tag{8}$$

D. Edge Aggregation

When all the scheduled devices have completed the parameter transmission, the edge server executes edge aggregation, which are introduced below.

Denote $f_a^s(t)$ as the computational resources allocated for edge aggregation, respectively. The global model $\boldsymbol{\omega}_g(t)$ is aggregated as

$$\boldsymbol{\omega}_g(t) = \sum_{i=1}^{I} \frac{D_i x_i(t) a_i(t) \boldsymbol{\omega}_i(t)}{\sum_{i=1}^{I} D_i x_i(t) a_i(t)}. \tag{9}$$

The delay for edge aggregation is calculated as

$$\tau_a(t) = \frac{C_0 \sum_{i=1}^{I} x_i(t) a_i(t) |\boldsymbol{\omega}_i(t)|}{f_a^s(t)}, \tag{10}$$

where C_0 denotes the CPU cycles per bit processing.

The loss function of $\boldsymbol{\omega}_g(t)$ is employed to measure the accuracy of the global model, which is calculated as

$$F_g(\boldsymbol{\omega}_g(t), t) = \frac{1}{D_g} \sum_{i=1}^{I} x_i(t) a_i(t) F_i(\boldsymbol{\omega}_i(t-1), t), \tag{11}$$

where $D_g = \sum_{i=1}^{I} x_i(t) D_i$ is the total data samples of the scheduled devices in the t-th iteration.

E. Energy Consumption and Delay Model

The total energy consumption of u_i includes the energy consumption of local training and that of parameter transmission, which is calculated as

$$E_i(t) = x_i(t)(E_i^L(t) + E_i^U(t)). \tag{12}$$

The total delay of each iteration includes the largest sum of the local training delay $\tau_i^L(t)$ and the parameter transmission delay $\tau_i^U(t)$ among all devices, and the edge aggregation delay $\tau_a(t)$ at the server side, and the global model broadcasting delay τ_b, which can be depicted as

$$\tau(t) = \max_i \{x_i(t)\tau_i(t)\} + \tau_a(t) + \tau_b, \tag{13}$$

where $\tau_i(t) = \tau_i^L(t) + \tau_i^U(t)$.

III. Problem Formulation and Transformation

In this section, we firstly introduce the long-term constraints and optimization problem. Then, we elaborate the Lyapunov optimization-based problem transformation.

A. Access Priority Constraint

Based on the service importance and QoS requirements, devices have different access priorities. The devices with higher access priority are required to complete sufficient times of local training and parameter transmission, which ensures the consistency between their local models and the DT. Therefore, the long-term access priority constraint [14] can be depicted as

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[x_i(t)] \geq e_i, \tag{14}$$

where e_i is the minimum probability of times that u_i should be selected for local training and parameter transmission.

B. Energy Consumption Constraint

Considering the limited battery capacity of devices, a long-term energy consumption constraint is denoted as

$$\sum_{t=1}^{T} E_i(t) \leq E_{i,max}, \tag{15}$$

where $E_{i,max}$ represents the energy budget of u_i.

C. Problem Formulation

The objective is to minimize the total iteration delay by jointly optimizing computational resource allocation and access scheduling under the long-term constraints of access priority and energy consumption. The joint optimization problem is formulated as

$$\begin{aligned}
&\mathbf{P1}: \min_{\boldsymbol{f},\boldsymbol{x}} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} V_\tau \tau(t), \\
&\text{s.t. } C_1: x_i(t) \in \{0,1\}, \forall u_i \in \mathcal{S}(t), \forall t \in \mathcal{T}, \\
&\quad C_2: \sum_{i=1}^{I} x_i(t) = N(t), \forall t \in \mathcal{T}, \\
&\quad C_3: 0 \leq f_i(t) \leq f_{i,max}(t), \forall u_i \in \mathcal{S}(t), \forall t \in \mathcal{T}, \\
&\quad C_4: \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[x_i(t)] \geq e_i, \forall u_i \in \mathcal{S}(t), \\
&\quad C_5: \sum_{t=1}^{T} E_i(t) \leq E_{i,max}, \forall u_i \in \mathcal{U},
\end{aligned} \tag{16}$$

where V_τ is a non-negative weight. $\boldsymbol{f} = (f_i(t) : \forall u_i \in \mathcal{S}(t), t \in \mathcal{T})$ denotes the vector of device computational resource allocation variables. $\boldsymbol{x} = (x_i(t) : \forall u_i \in \mathcal{S}(t), t \in \mathcal{T})$ denotes the vector of device access scheduling. C_1 and C_2 indicate that the BS can schedule $N(t)$ devices simultaneously for uploading model parameters at most. C_3 represents the device computational resource allocation constraint, where $f_{i,max}(t)$ is the maximum available computational resources of u_i. C_4 and C_5 are the long-term constraints of access priority and energy consumption, respectively.

D. Problem Transformation

The long-term constraint and the short-term resource management are decoupled by the concept of virtual queues. Specifically, virtual queues $F_i(t)$ and $Y_i(t)$ corresponding to C_4 and C_5 are evolved as

$$F_i(t+1) = \max\{F_i(t) + e_i - x_i(t), 0\}, \tag{17}$$

$$Y_i(t+1) = \max\left\{Y_i(t) + E_i(t) - \frac{E_{i,max}}{T}, 0\right\}. \tag{18}$$

If $F_i(t)$ and $Y_i(t)$ are mean rate stable, C_4 and C_5 hold automatically [15]. Therefore, $\mathbf{P1}$ can be equivalently rewritten as

$$\begin{aligned} \mathbf{P2} : &\min_{\boldsymbol{f},\boldsymbol{x}} \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\Big(\Phi(t)\Big) \\ \text{s.t.}\ \ & C_1 \sim C_3, \\ & C_4 : F_i(t) \text{ and } Y_i(t) \text{ are mean rate stable.} \end{aligned} \tag{19}$$

By employing Lyapunov optimization [15], $\mathbf{P2}$ can be transformed to minimize $\Lambda(t)$, which is depicted as

$$\begin{aligned} \mathbf{P3} : &\min_{\boldsymbol{f},\boldsymbol{x}} \Lambda(t) \\ \text{s.t.}\quad & C_1 \sim C_3. \end{aligned} \tag{20}$$

where $\Lambda(t)$ is given as (21), and V is a positive weight.

IV. Low-Latency Digital Twin Assisted Network Resource Management

In this section, we propose LEGION to solve $\mathbf{P3}$.

A. Device Access Scheduling and Computational Resource Allocation Joint Optimization

We decompose $\mathbf{P3}$ into two subproblems, i.e., access scheduling and computational resource allocation.

The access scheduling subproblem can be modeled as an Markov decision process (MDP), and solved by DT assisted DQN. The state space is denoted as $\boldsymbol{G}(t) = \{\boldsymbol{F}(t), \boldsymbol{Y}(t)\}$, which can be obtained from the DT. The action space is denoted as $\boldsymbol{A}(t) = \{x_1(t), x_2(t), \cdots, x_I(t)\}$, and the cost function is defined as the optimization objective of $\mathbf{P3}$. DQN can learn the optimal strategy based on the estimated Q values $\boldsymbol{Q}(\boldsymbol{G}(t), \boldsymbol{\varpi}(t))$, where $\boldsymbol{\varpi}(t)$ is the parameter set of the deep neural network.

The computational resource allocation subproblem is denoted as $\mathbf{P3'}$. DT assists network resource management by providing the estimates of key state information. Specifically, since $f_{i,max}(t)$, $h_i(t)$, and $I_i(t)$ are not available to edge server, we employ the empirical values estimated by DT, i.e., $f'_{i,max}(t)$, $h'_i(t)$, and $I'_i(t)$. Then, $\mathbf{P3'}$ is given by

$$\begin{aligned} \mathbf{P3'} : &\min_{\boldsymbol{f}} \Psi(\boldsymbol{f}) \\ &+ V\Big\{\max_{u_i\in\boldsymbol{\pi}(t)}\Big\{\frac{\xi_i D_i}{f_i(t)} + \frac{|\boldsymbol{\omega}_i(t)|}{B_i\log_2(1+\frac{h'_i(t)P_i(t)}{I'_i(t)+N_0})}\Big\}\Big\} \\ \text{s.t.}\ \ & C'_3 : 0 \le f_i(t) \le f'_{i,max}(t), \forall u_i \in \boldsymbol{\pi}(t), \forall t \in \mathcal{T}, \end{aligned} \tag{22}$$

where

$$\begin{aligned} \Psi(\boldsymbol{f}) = &\sum_{u_i\in\boldsymbol{\pi}(t)}\Big[-F_i(t) - e_i + \Big(Y_i(t) \\ &- \frac{E_{i,max}}{T}\Big)\Big(\alpha\xi_i D_i f_i^2(t) + \frac{P_i(t)|\boldsymbol{\omega}_i(t)|}{B_i\log_2(1+\frac{h'_i(t)P_i(t)}{I'_i(t)+N_0})}\Big). \end{aligned} \tag{23}$$

Here, $\boldsymbol{\pi}(t) \subseteq \mathcal{S}(t)$ represents the set of the selected devices.

Based on $\max\{x_1, \cdots, x_M\} = \lim_{k\to\infty}\frac{1}{k}\ln(e^{kx_1} + \cdots + e^{kx_M})$ [16], and defining $\gamma_i = \frac{\xi_i D_i}{f_i(t)} + \frac{|w_i(t)|}{B_i\log_2(1+\frac{h'_i(t)P_i(t)}{I'_i(t)+N_0})}$, $\mathbf{P3'}$ can be approximated as

$$\begin{aligned} \widetilde{\mathbf{P3'}} : &\min_{\boldsymbol{f}} \Upsilon(t) = \Psi(\boldsymbol{f}) + \lim_{k\to\infty}\frac{V}{k}\ln(e^{k\gamma_1} + \cdots + e^{k\gamma_I}) \\ \text{s.t.}\ \ & C'_3. \end{aligned} \tag{24}$$

$\widetilde{\mathbf{P3'}}$ is convex and can be solved easily.

B. LEGION

The procedures of LEGION are summarized in Algorithm 1. First, the virtual queue backlogs and the access scheduling indicators are initialized as zero. At the beginning of each iteration, the server selects the devices for parameter transmission based on the estimated Q values and obtains $\boldsymbol{\pi}(t)$. Then, based on the empirical values estimated by DT, i.e., $f'_{i,max}(t)$, $h'_i(t)$, and $I'_i(t)$, the server decides the optimal computational resource allocation and transmission power for $u_i \in \boldsymbol{\pi}(t)$ by solving $\widetilde{\mathbf{P3'}}$. Next, $u_i \in \boldsymbol{\pi}(t)$ downloads the global model $\boldsymbol{\omega}_g(t-1)$, performs local training, transmits model parameters, and updates the virtual queues $F_i(t)$ and $Y_i(t)$ according to (17) and (18). Afterwards, the server obtains the optimal computational resource allocation strategy and executes edge aggregation based on (9). Next, the server calculates the cost $\theta(t)$ as (20) and transfers the DQN to the next state $\boldsymbol{G}(t+1)$. Finally, the server updates the parameter set $\boldsymbol{\varpi}(t)$ of DQN based on gradient descent method.

$$\Lambda(t) = \sum_{i=1}^{I}\Big[-x_i(t)(F_i(t)+e_i) + x_i(t)(Y_i(t) - \frac{E_{i,max}}{T})\Big(E_i^L(t) + E_i^U(t)\Big) + V\Big\{\max_i\Big\{x_i(t)(\tau_i^L(t) + \tau_i^U(t))\Big\}\Big\}. \tag{21}$$

Algorithm 1 LEGION

Input: I, $N(t)$, $\{E_{i,max}\}$, $\{f_{i,max}(t)\}$, $\{F_i(t)\}$, $\{Y_i(t)\}$, $f^s_{max}(t)$, $\boldsymbol{G}(t)$.
Output: $\boldsymbol{f}, \boldsymbol{x}$.
Initialize $F_i(0) = 0, Y_i(0) = 0$, and set $x_i(t) = 0$, $\forall u_i \in \mathcal{U}, \forall t \in \mathcal{T}$.
For iteration $t = 1, \cdots, T$ **do**
  Server side:
  Draw the access scheduling strategy $\boldsymbol{\pi}(t)$ based on $\mathbf{Q}(\boldsymbol{G}(t), \boldsymbol{\varpi}(t))$.
  Obtain $\boldsymbol{f}$ with the assistance of the DT by solving (24).
  Device side:
  For $u_i \in \boldsymbol{\pi}(t)$
    Download the global model $\boldsymbol{\omega}_g(t-1)$ and perform local model training and transmission.
    Update $F_i(t)$ and $Y_i(t)$ according to (17) and (18).
  end for
  Server side:
  Perform edge aggregation as (9).
  Calculate $\theta(t)$ as (20) and transfer to the next state $\boldsymbol{G}(t+1)$.
  Update $\boldsymbol{\varpi}(t)$ based on gradient descent.
end for

V. Simulations Results

We consider a low-carbon smart park scenario, which includes 1000 devices and 100 subchannels. The devices are randomly distributed among the smart park, and the BS co-located with the edge server is located at the center. The classic data set MNIST is employed to train DT, which consists of 6×10^4 training samples and 10^4 test samples [17]. Each device is randomly allocated with serval training samples, e.g., $100 \sim 400$, from MINST. The loss function of u_i is defined as the mean-square error between the target output φ_j and the real output $\tilde{\varphi}_j$ [17]. The simulation parameters are summarized in Table I [18]. Two state-of-the-art algorithms are utilized for comparison. The first one is the upper confidence bound policy-based client scheduling algorithm (CS-UCB) [19], which minimizes the total iteration delay through device access scheduling without considering the energy consumption constraint and computational resource allocation. The second one is a deep neural networks-based user association and resource management algorithm (RS-DNN) [20], which is implemented by optimizing device computational resource allocation without considering the access priority constraint and the devices are randomly scheduled.

TABLE I
Simulation Parameters

Parameter	Value	Parameter	Value
T, I	500,1000	τ_b	10 ms
$\lvert\boldsymbol{\omega}_i\rvert$	5 Kbits	k	0.023dB
ξ_i	10^6 cycles	ξ_0	10^5 cycles
N_0	-114 dBm	α	10^{-27} Watt·s^3/cycle^3
B_i	0.1 MHz	C_0	10^4 cycles
$E_{i,max}$	$25 \sim 125$ J	$f_{i,max}$	1.5 GHz
V	3	e_i	[0.5, 0.3, 0.1, 0.05]
φ	0.2	ι	0.5
D_i	$100 \sim 400$	D_{test}	300

Fig. 2 shows the average iteration delay versus iterations. Compared with RS-DNN, LEGION reduces the average iteration delay by 10.97%. CS-UCB outperforms LEGION in average iteration delay by 5.19% because all available computational resources are used for local training with the cost of substantial energy consumption increment.

Fig. 3 shows the DT loss function F_g versus iterations. When $t = 100$, LEGION outperforms CS-UCB, and RS-DNN in DT loss function by 25.73% and 54.21%, respectively. The reason is that LEGION leverages CRC to mitigate the adverse impacts of packet error on DT loss function minimization.

Fig. 4 shows the cumulative energy consumption of all devices versus iteration. When $t = 100$, LEGION outperforms CS-UCB and RS-DNN by 33.85% and 14.16%, respectively. Due to the ignorance of device computational resource allocation, CS-UCB performs worst in energy consumption. In combination with Fig. 2, CS-UCB reduces the average iteration delay at the expense of higher energy consumption.

Fig. 5 shows the box plots of access priority deficit. The access priority deficit reflects the deviation of the current performance from the specified constraint. Without the access priority awareness, RS-DNN has the largest average access priority deficit. Compared with CS-UCB and RS-DNN, LEGION reduces the access priority deficit fluctuation by 66.32% and 83.03%, respectively. Although access priority awareness

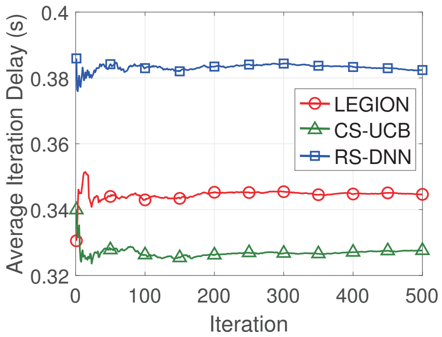

Fig. 2. Average iteration delay.

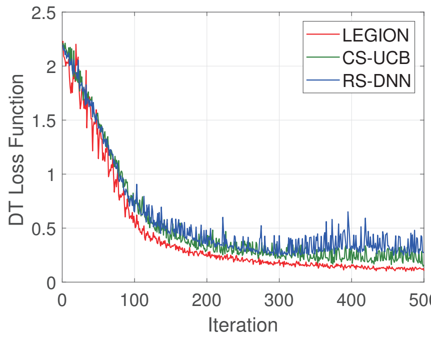

Fig. 3. DT loss function versus iterations.

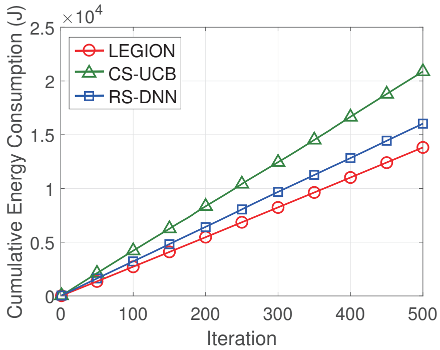

Fig. 4. Cumulative network energy consumption.

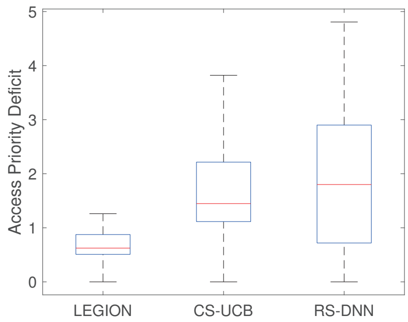

Fig. 5. Access priority deficit.

is also considered in CS-UCB, LEGION performs much better than CS-UCB with the assistance of DT.

VI. Conclusions

In this paper, we addressed the DT-empowered communication network resource management in the low-carbon smart park. We proposed LEGION to achieve a low-latency and accurate DT. Compared with CS-UCB and RS-DNN, LEGION reduces the DT loss function by 25.73% and 54.21%, and the access priority deficit fluctuation by 66.32% and 83.03%. LEGION also achieves the least access priority deficit and cumulative energy consumption due to the access priority and energy consumption awareness. In the future, we will consider security threats faced by energy management in low-carbon smart park.

Appendix A

TABLE II
Table of Acronyms

Acronyms	Interpretation	Acronyms	Interpretation
DT	Digital twin	DQN	Deep Q-network
BS	Base station	IoT	Internet of things
CSI	Channel state information	FL	Federated learning
CRC	Cyclic redundancy check	QoS	Quality of service
MDP	Markov decision process		

Acknowledgments

This work was supported by Science and Technology Project of State Grid Corporation of China under grant number 52094021N010 (5400-202199534A-0-5-ZN).

References

[1] Y. Cheng, M. Li, Y. Li, and Y. Ma, "Research on the mode of energy distribution network for smart low-carbon city," *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*, pp. 3445–3450, 2020.

[2] S. R. Pokhrel, S. Verma, S. Garg, A. K. Sharma, and J. Choi, "An efficient clustering framework for massive sensor networking in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4917–4924, Jul. 2021.

[3] C. Gehrmann and M. Gunnarsson, "A digital twin based industrial automation and control system security architecture," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 669–680, Jan. 2020.

[4] A. Saad, S. Faddel, T. Youssef, and O. Mohammed, "On the implementation of IoT-based digital twin for networked microgrids resiliency against cyber attacks," *IEEE Trans. Smart Grid*, vol. 11, no. 6, pp. 5138–5150, Nov. 2020.

[5] W. Shi, S. Zhou, Z. Niu, M. Jiang and L. Geng, "Joint Device Scheduling and Resource Allocation for Latency Constrained Wireless Federated Learning," *IEEE Trans. Wirel.* vol. 20, no. 1, pp. 453-467, Jan. 2021.

[6] R. Saha, S. Misra and P. K. Deb, "FogFL: Fog-Assisted Federated Learning for Resource-Constrained IoT Devices," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8456-8463, May 2021.

[7] W. Sun, N. Xu, L. Wang, H. Zhang, and Y. Zhang, "Dynamic digital twin and federated learning with incentives for air-ground networks," *IEEE Trans. Netw. Sci. Eng.*, vol. PP, no. 99, pp. 1–1, Dec. 2020.

[8] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 5098–5107, Jul. 2021.

[9] C. Wang, D. Deng, L. Xu, W. Wang, and F. Gao, "Joint interference alignment and power control for dense networks via deep reinforcement learning," *IEEE Wireless Commun. Lett.*, vol. 10, no. 5, pp. 966–970, May 2021.

[10] H. Liao, Z. Zhou, X. Zhao, and *et al.*, "Learning-based context-aware resource allocation for edge-computing-empowered industrial IoT," *IEEE Internet Things J,*, vol. 7, no. 5, pp. 4260-4277, Dec. 2020.

[11] Z. Zhou, *et al.*, "Secure and latency-aware digital twin assisted resource scheduling for 5G edge computing-empowered distribution grids," *IEEE Trans. Industr. Inform.*, vol. 34, no. 4, pp. 49–55, Jul. 2020.

[12] Z. Chang, L. Liu, X. Guo, and Q. Sheng, "Dynamic resource allocation and computation offloading for IoT fog computing system," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3348–3357, May 2021.

[13] Y. Xi, A. Burr, J. Wei, and D. Grace, "A general upper bound to evaluate packet error rate over quasi-static fading channels," *IEEE Trans. Wireless Commun.*, vol. 10, no. 5, pp. 1373–1377, May 2011.

[14] Z. Zhou, *et al.*, "Learning-based URLLC-aware task offloading for internet of health things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 396–410, Sept. 2021.

[15] Z. Zhou, H. Liao, X. Zhao, B. Ai and M. Guizani, "Reliable task offloading for vehicular fog computing under information asymmetry and information uncertainty," *IEEE Trans. Veh. Technol.*, pp. 1–6, 2021.

[16] G. Kreisselmeier and R. Steinhauser, "Application of vector performance optimization to a robust control loop design for a fighter aircraft," *Int. J. Contr.*, vol. 37, no. 2, pp. 251–284, Feb. 1983.

[17] V.-D. Nguyen, S. K. Sharma, *et al.*, "Efficient federated learning algorithm for resource allocation in wireless IoT networks," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3394–3409, Mar. 2021.

[18] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6G," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12 240–12 251, Oct. 2020.

[19] W. Xia *et al.*, "Multi-armed bandit-based client scheduling for federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7108–7123, Nov. 2020.

[20] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning and permissioned blockchain for digital twin edge networks," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2276–2288, Feb. 2021.