Optimizing Federated Learning With Deep Reinforcement Learning for Digital Twin Empowered Industrial IoT

Wei Yang ^(D), Wei Xiang ^(D), Senior Member, IEEE, Yuan Yang ^(D), and Peng Cheng ^(D), Member, IEEE

Abstract—The accelerated development of the Industrial Internet of Things (IIoT) is catalyzing the digitalization of industrial production to achieve Industry 4.0. In this article, we propose a novel digital twin (DT) empowered IIoT (DTEI) architecture, in which DTs capture the properties of industrial devices for real-time processing and intelligent decision making. To alleviate data transmission burden and privacy leakage, we aim to optimize federated learning (FL) to construct the DTEI model. Specifically, to cope with the heterogeneity of IIoT devices, we develop the DTEI-assisted deep reinforcement learning method for the selection process of IIoT devices in FL, especially for selecting IIoT devices with high utility values. Furthermore, we propose an asynchronous FL scheme to address the discrete effects caused by heterogeneous IIoT devices. Experimental results show that our proposed scheme features faster convergence and higher training accuracy compared to the benchmark.

Index Terms—Deep reinforcement learning (DRL), digital twin (DT), federated learning (FL), Industrial Internet of Things (IIoT), learning efficiency, real time.

I. INTRODUCTION

D RIVEN by the next generation of information technologies, including the Internet of Things (IoT), digital twin (DT), 6G, etc., the Industrial Internet of Things (IIoT) enables the transformation and development of digital networks and intelligent enterprises [1], [2]. As an emerging application, the IIoT has brought disruptive changes and impacts to the

Wei Yang and Yuan Yang are with the School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China (e-mail: yangwei@stu.xaut.edu.cn; yangyuan@xaut.edu.cn).

Wei Xiang is with the School of Computing, Engineering and Mathematical Sciences, La Trobe University, Melbourne 3086, Australia, and also with the College of Science and Engineering, James Cook University, Cairns, QLD 4870, Australia (e-mail: w.xiang@latrobe.edu.au).

Peng Cheng is with the School of Computing, Engineering and Mathematical Sciences, La Trobe University, Melbourne, VIC 3086, Australia (e-mail: p.cheng@latrobe.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2022.3183465.

Digital Object Identifier 10.1109/TII.2022.3183465

traditional manufacturing industry by connecting machines, intelligent algorithms, and industries. In an intelligent factory, smart devices enable real-time collection and analysis of data to make intelligent decisions and optimize production. However, the IIoT requires distributed intelligent services to change in real time with the dynamic environment, which is a challenging task due to the complexity of the industrial environment and the heterogeneity of IIoT devices [3].

The DT concept was first proposed in [4] and then adopted by NASA in 2011 for fault diagnosis and maintenance of flight systems, which attracted great attention. Currently, DT has been extended to the military, smart cities, manufacturing, and so on [5], [6]. DT can provide feedback and reflect the bi-directional dynamic mapping process, which provides a feasible solution to capture the dynamic industrial environment [7]. However, DT modeling in IIoT still faces some difficulties. First, DTs need to be driven by massive data distributed across IIoT devices, but given privacy, competition, and security issues, integrating data scattered across various devices is nearly impossible. Second, the real-time interaction between the DTs and the entity object requires frequent communications among the devices [8].

Federated learning (FL), a new type of distributed machine learning paradigm, has great advantages in training private and heterogeneous data [9]. It has become an advanced paradigm for realizing distributed training of IIoT [10]. Specifically, FL trains a model using local computing capability and the device data and then aggregates the trained model parameters on the server side. The aggregated parameters serve as the initial parameters for the next round of local training. Because all client data is only used for local model training, FL avoids direct data leakage to protect client privacy and data security. Several advanced FL strategies were designed to improve model accuracy from training efficiency and privacy protection perspectives [11]–[14].

With the rapid development of IIoT, IIoT devices with wide geographic distribution vary significantly in their forms, incurring a series of problems such as the heterogeneous data and the complex network environment. In this article, we propose a new DT empowered IIoT (DTEI) architecture, where the virtual models of the physical objects in IIoT are constructed through capturing the real-time status of the base stations (BSs) and devices. Then, to improve the training model efficiency of the IIoT device, a deep reinforcement learning (DRL) assisted FL framework is proposed. DRL has a natural advantage in solving high-dimensional decision-making problems. Therefore, DRL

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/



Manuscript received 25 March 2022; revised 23 May 2022; accepted 7 June 2022. Date of publication 4 July 2022; date of current version 13 December 2022. The work of Wei Yang and Yuan Yang was supported in part by the Shaanxi Innovation Capability Support project under Grant 2021TD-25 and in part by the Natural Science Basic Research Program of Shaanxi under Grant 2021JQ-478. Paper no. TII-22-1246. (Corresponding authors: Wei Xiang; Yuan Yang.)

is used to select some high-efficiency IIoT devices for aggregation [15], [16]. It is also noted that the straggler effect of heterogeneous IIoT scenarios causes serious training delays in synchronous FL. To address these problems, we propose an asynchronous FL framework, which is a DRL-supported device clustering scheme. The major contributions of this article are summarized as follows.

- We propose a new architecture of DTEI to integrate DT with the IIoT network, which maps a device's real-time operating state and behavior to a virtual space. In particular, we adopt DTEI to capture the characteristics of IIoT devices for dynamic perception and intelligent decision.
- We exploit the optimized FL to construct the DTEI model. Specifically, we develop a DTEI-assisted DRL method for IIoT device selection to improve the efficiency and performance of FL.
- 3) We propose the asynchronous FL, a DRL-based algorithm, to avoid the straggler effect with device selection and clustering mechanism. Experimental results show that our proposed scheme significantly outperforms its benchmark counterparts.

The rest of this article is organized as follows. We discuss the related work in Section II. Then, Section III presents the system model and the problem formulation. Next, Section IV introduces our proposed DRL-assisted asynchronous FL algorithm. In Section V, we conduct experimental evaluations. Finally, Section VI concludes this article.

II. RELATED WORK

A. Federated Learning for Digital Twin

Constructing a DT model requires synchronizing a massive amount of data, but limited computing resources and communication capabilities hinder the digitization of the IIoT. In addition, people's increasing attention to data security and privacy has also brought new challenges to DT modeling. Due to the unique advantages of FL in terms of efficiency and security, some authors have exploited FL to construct DT models. Lu et al. [17] proposed the DT edge network, which utilizes FL to construct the DT model of IIoT devices based on the operating status of IIoT devices. Sun et al. [18] applied DT to the IIoT architecture, where DT reflects the dynamic properties of the industrial device to assist FL. Lu et al. [19] proposed the DT wireless network (DTWN) architecture, which transmits the real-time data that is processed and calculated at the edge servers, and the blockchain-empowered FL framework, which runs in DTWN for cooperative computing and improves the system's efficiency and security. However, these works do not consider the influence of heterogeneous devices and complex network environments on the accuracy of the training model. We balance data diversity and global model performance with the DRL-supported device selection clustering algorithm to address the heterogeneity challenge in IIoT.

B. Deep Reinforcement Learning for Industrial IoT

The DRL technology has been widely used in IIoT scenarios for computation offloading decision-making and dynamic resource management due to its advantages in solving problems with large-scale time-varying features. Dai et al. [8] formulated the problem of stochastic computation offloading and energy management as an optimization problem. In order to solve this optimization problem, the authors transformed the stochastic programming problem into a deterministic time slot problem by exploiting the Lyapunov optimization strategy and developed an asynchronous DRL algorithm to explore the optimal resource allocation strategy. Guo et al. [15] proposed an FL-based DRL algorithm to adjust the critical parameters of the IIoT system for achieving efficient and flexible resource management. Chen et al. [16] transformed the optimization problem of resource allocation into a Markov decision process (MDP) to minimize the average delay of the task and proposed a dynamic resource management scheme based on DRL to solve the MDP problem. In this article, we propose the DTEI-assisted DRL scheme for the selection process of IIoT devices to improve the efficiency and performance of FL.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. Digital Twin Empowered Industrial IoT Model

We introduce DT in actual IIoT scenarios, such as intelligent factory and intelligent transportation. For instance, in the intelligent transportation system, DT can assist a vehicle in perceiving vehicle status and real-time road condition information and provide users with emergency avoidance and navigation information. The DTEI architecture is illustrated in Fig. 1. We propose a two-layer heterogeneous network in the IIoT, which is composed of the physical layer and the DT layer. The physical layer consists of BSs and client devices such as intelligent machines, vehicles, and sensors in IIoT environments. In reality, there are plenty of BSs as shown in Fig. 1, indexed by $\mathcal{B} = \{1, 2, \dots, B\}$. The BS is equipped with edge servers and DRL agents with sufficient communication, calculation, and artificial intelligence processing capabilities. The client devices are denoted by $\mathcal{N} = \{1, 2, \dots, N\}$, which collect data from sensors and applications on the device and save it locally, denoted by \mathcal{D}_i with dimension $D_i = |\mathcal{D}_i|$. The BSs are connected with the IIoT devices within their coverage through wireless communications. After training the model locally, the IIoT devices send the trained local model parameters to the edge server for global update. The DT of the client device is served by its corresponding BS, which collects the physical status of the device in real time and dynamically presents the current training status of the device in digital form. In time slot t, the DTs of device i and BS b are denoted by $DT_i(t) = \{M_i(t), f_i(t), P_i(t), D_i(t)\}$ and $DT_b(t) =$ $\{M_b(t), f_b(t), P_b(t), D_b(t)\}$, respectively, where $M_i(t)$ and $M_b(t)$ refer to the current training statuses of device i and BS b, respectively. $f_i(t)$ and $f_b(t)$ denote the current computing capabilities of device i and BS b, respectively. $P_i(t)$ and $P_b(t)$ stand for communication resources of device *i* and BS *b*, respectively. And $D_i(t)$ and $D_b(t)$ represent the data of device *i* and BS *b*, respectively, which are processed in time slot t.

The proposed DTEI architecture connects IIoT devices at the physical layer with virtual systems at the DT layer. The DTEI can not only reflect the characteristics of physical entities in real time but also simulate and predict the system, which plays a



Fig. 1. Federated learning for digital twin empowered IIoT.

vital role in the optimization of service quality. In this article, we exploit DTEI to monitor the dynamic network environment to provide assistance for the system's intelligent decision-making.

B. Federated Learning for Digital Twin Empowered IIoT

As illustrated in Fig. 1, we exploit FL to construct the DT model in DTEI, which can respond according to the status and rules of the actual device. The client devices collect large amounts of running data from monitoring the environment in real time, which are used to train local models. Then, the local devices upload their model to the edge server, which aggregates and updates the model parameters and returns them to the local devices. The loss function of the error between the quantified estimated value and the true value is denoted by l(w). The loss function $L(w_i)$ of the client device i on dataset \mathcal{D}_i can be expressed as

$$L(w_i) = \frac{1}{D_i} \sum_{X_j \in \mathcal{D}_i} l(w_i, X_j) \tag{1}$$

where X_j refers to the sample point of local data \mathcal{D}_i . The global loss function is

$$L(w_g) = \frac{1}{N} \sum_{i \in \mathcal{N}} L(w_i)$$
⁽²⁾

where $N \subseteq \mathcal{N}$ indicates that the collection of the selected N client devices participates in FL, and w_g denotes the aggregation model parameter. The goal of FL is to minimize the following global loss:

$$\min_{w \in \mathbb{R}_d} L(w_g). \tag{3}$$

In practical applications, we need to select efficient devices for some specific applications to construct the DTs because of the client devices' heterogeneity and the complex dynamic network environment.

C. Data Utility Model

In order to select the efficient device, we first evaluate the utility of device data. When performing FL tasks, the higher

utility of the device's training data can result in the higher accuracy of the local model and the better prediction performance of the aggregated global model. Therefore, a metric is required to quantify the device's potential contribution to task completion. To quantify the contribution and utility of device data in the task, we define the prediction accuracy of the local model as the evaluation metric. Considering the unique characteristics of IIoT device data in FL, we focus on three crucial factors of device data, i.e., the data quality, the data size, and the data distribution. Based on the experimental validation in [20], a large training data size or high data quality usually contributes to improving the model's prediction performance. Let D_i^{\max} be the maximum training data size that device *i* can contribute to task λ_j , and we have $0 \le D_i \le D_i^{\max}$. Here, $D_i = 0$ indicates that device *i* fails to participate in task λ_i . Let q_i represent the data quality of the training sample of device i in task λ_i , and the constraint conditions are met

$$q_i = \begin{cases} 0 & \text{Device } i \text{ is invalid} \\ (0,1] & \text{otherwise} \end{cases}$$
(4)

where $q_i = 1$ denotes the highest data quality of device *i*'s training sample; $q_i = 0$ indicates that device *i* is "free-riding" or malicious, which tricks BS *j* by providing redundant and fake local training samples.

In terms of data distribution, it is frequently assumed that data is independent and identically distributed (i.i.d.) in traditional centralized learning, but in FL scenarios, data is usually non-i.i.d. The weight divergence, measured by the earth mover's distance (EMD) metric, is the main reason for the decrease in FL accuracy [21]. A larger EMD value leads to a larger weight divergence, which will have an adverse effect on the model's training. We consider an *L*-classification task, where the data sample $\mathcal{D}_i = \{x_i, y_i\}$ of device *i* is distributed on $\mathcal{X} \times \mathcal{Y}$ and follows the distribution \mathbb{F}_i , in which \mathcal{X} and \mathcal{Y} indicate the compact space and the label space, respectively. Here, ϕ_i denotes the EMD of \mathcal{D}_i . Considering the overall distribution \mathbb{F}_a in task λ_j , the EMD of device *i* is defined as

$$\phi_i = \sum_{b \in \mathcal{Y}} \|\mathbb{F}_i(y=b) - \mathbb{F}_a(y=b)\|$$
(5)

where $\mathbb{F}_i(y = b)$ and $\mathbb{F}_a(y = b)$ represent the proportion of the data sample labeled *b* in the local data sample of device *i* and the proportion of all devices involved in task λ_j , respectively. According to the experimental results in [22], the local model prediction accuracy of device *i* in task λ_j is expressed as $\rho_i \in [0, 1]$, which is given by

$$\rho_i = \mu(D_i, q_i, \phi_i)$$

= $\nu(\phi_i) - \alpha_1 e^{-\alpha_2(\alpha_3 q_i D_i)^{\nu(\phi_i)}}.$ (6)

Here, $\nu(\phi_i) = \alpha_4 e^{-\left(\alpha_5 + \frac{\phi_i}{\alpha_6}\right)^2} < 1$, where $\alpha_k > 0$ $(1 \le k \le 6)$ is the curve fitting coefficient. The first term $\nu(\phi_i)$ reflects the deterioration of the model performance with the increase of the EMD ϕ_i . The exponential term $\alpha_1 e^{-\alpha_2(\alpha_3 q_i D_i)^{\nu(\phi_i)}}$ indicates that the larger data size and the higher data quality contribute to the better performance of the trained model.

D. Energy Consumption Model

In the process of FL, the client device's energy consumption includes the consumption of local data, the computation of local training, and the communication of the global aggregation [23]. The energy consumption of local data arises from the deployment of smart devices and preprocessing of device data, while data annotation and cleansing require expensive human efforts. We denote the number of central processing unit (CPU) cycles required for device *i* as ε_i to execute a unit of the data sample, and f_i indicates the CPU cycle frequency of device *i*. The local data consumption C_i^{data} is expressed as

$$C_i^{\text{data}} = D_i \varepsilon_i q_i f_i^2 \tag{7}$$

where $D_i > 0$ represents the size of device *i*'s local data. The computational energy consumed by training device *i* can be written as

$$C_i^{\rm cmp} = D_i \varepsilon_i E_i^l m_i f_i^2 \tag{8}$$

where E_i^l is the local training epoches' number, and m_i is the model size. In order to synchronize the model parameters to the BS, the local device shares the U uplink subchannel on the basis of orthogonal frequency division multiple access, denoted as a set $\mathcal{H} = \{1, 2, ..., H\}$. Let B be the number of bits of model parameters. The communication energy consumption of device *i* for model aggregation is

$$C_i^{\text{com}} = \frac{\delta B}{\sum_{h=1}^H T_{i,h} G \log_2(1 + \frac{P_{i,h}\xi_{i,h}}{L})}$$
(9)

where δ represents the normalization factor of communication energy consumption, G is the subchannel bandwidth, $T_{i,h}$ denotes the device *i*'s time fraction allocated on subchannel u, $P_{i,h}$ indicates the device *i*'s transmission power, L represents the noise power, and ξ refers to the channel power gain.

E. Problem Formulation

The clients' requirements and geographic distribution of IIoT devices are usually diverse, resulting in heterogeneous data and uneven data quality. The main challenges of FL in the field of IIoT are the inefficiency of data training, the high cost of wireless communication, and the long time of model aggregation due to data features. Hence, to enhance the efficiency and accuracy of model aggregation, the DRL algorithm is adopted to select IIoT devices with high utility values for training.

In order to formulate the device selection problem, we introduce $\mathbf{k}^t = [\kappa_i^t]$ as the indicator vector of the device selection state in time slot t. $\kappa_i^t = 1$ means that device i has been selected/activated, while $\kappa_i^t = 0$ indicates that it has not been selected/activated. The total energy cost of device i is equal to the sum of the costs in (7), (8), and (9), expressed as

$$C_i = C_i^{\text{data}} + C_i^{\text{cmp}} + C_i^{\text{com}}$$
$$= D_i \varepsilon_i q_i f_i^2 + D_i \varepsilon_i E_i^l m_i f_i^2 + \frac{\mu B}{\sum_{h=1}^H T_{i,h} G \log_2(1 + \frac{P_{i,h} \xi_{i,h}}{L})}.$$

We define the utility function of device i in time slot t as

$$u^{t}(\kappa_{i}^{t}) = \omega \sigma \rho_{i}^{t} - (1 - \omega)C_{i}^{t}$$
(11)

where $\omega \in (0, 1]$ is the weight coefficient that balances costs and benefits, and σ represents the adjustment parameter.

We employ the MDP $\mathcal{M} = (S, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ to describe the combinatorial optimization problem of device selection, where S denotes the state space, \mathcal{A} indicates the action space, \mathcal{P} is the state transition probability, \mathcal{R} represents the reward function, and $\gamma \in (0, 1]$ is the reward discount factor. The devices selection problem is expressed as follows:

$$\max_{\mathbf{k},t} u^t \left(\mathbf{k}^t \right) \tag{12}$$

s.t.
$$\kappa_i^t \in \{0, 1\} \ \forall i$$
 (12a)

$$0 \le D_i \le D_i^{\max} \ \forall i \tag{12b}$$

$$f_i^{\min} \le f_i(t) \le f_i^{\max} \,\forall i \tag{12c}$$

$$p_i^{\min} \le p_i(t) \le p_i^{\max} \ \forall i \tag{12d}$$

$$C_i(t) \le C_{\text{thd}} \ \forall i. \tag{12e}$$

Constraint (12a) is the selection status of device i in time slot t. Constraint (12b) denotes the amount of training data that device i can contribute to task λ_j . Constraints (12c) and (12d) represent the computation resource and transmission power constraints, respectively. Constraint (12e) indicates the client's energy consumption limit, where C_{thd} is determined by the device power supply.

IV. DRL FOR DEVICE SELECTION BASED ON DT

A. MDP-Based Device Selection Problem

For the sake of addressing the device selection problem in (12), the system first constructs MDP $\mathcal{M} = (S, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ and adopts the DRL scheme to explore actions. We exploit DT to monitor the model's training state and complex network environment, where the system state $s(t) \in S$ is created by the DT and transmitted to the DRL agent. Specifically, in the workflow design of the DTEI-assisted DRL system, as shown in Fig. 2, the agent interacts with the DT in the DRL setting, in which the agent is the decision-maker for device selection, and



Fig. 2. Workflow of using DTEI-assisted DRL for devices selection design.

the DT sets the restrictions, rules, and reward mechanism. In time slot t, the agent selects the action $\kappa(t) \in \mathcal{A}$ when perceiving the state s(t). After performing this action, the current state is transferred to the next state s(t + 1) with the consequence of the agent obtaining the reward r(t). The DRL aims to maximize the expected discounted cumulative reward through searching for the optimal strategy π , which maps state s(t) to action $\kappa(t)$. The parameters of the MDP model we defined are described as follows.

1) *State Space:* In time slot t, the system states consist of transmission power $\mathbf{p}(t) = \{p_1^t, p_2^t, \dots, p_i^t\}$, available computing resource of devices $\mathbf{f}(t) = \{f_1^t, f_2^t, \dots, f_i^t\}$, state of the model $\mathbf{q}(t) = \{q_1^t, q_2^t, \dots, q_i^t\}$, and the selection state of devices $\mathbf{k}(t-1) = \{\kappa_1^{t-1}, \kappa_2^{t-1}, \dots, \kappa_i^{t-1}\}$. The current state $s(t) \in S$ is expressed as

$$s(t) = \{\mathbf{p}(t), \mathbf{f}(t), \mathbf{q}(t), \mathbf{k}(t-1)\}.$$
 (13)

2) Action Space: The agent's action is the device selection decision in round t. The action $\kappa(t) \in \mathcal{A}$ is defined as

$$\mathbf{k}(t) = (\kappa_1^t, \kappa_2^t, \dots, \kappa_i^t) \tag{14}$$

where $\kappa_i^t = 1$ indicates that device *i* is selected as the device participating in the training, and $\kappa_i^t = 0$ implies that device *i* has not been selected during the training process.

- Policy: The policy π : S → A denotes the mapping between the state space and the action space. In round t, the executed action can be obtained through strategy κ(t) = π(s(t)). The DT states transition based on the device selection actions.
- 4) *Reward:* The system utilizes the reward function r to evaluate the action. In round t, the agent implementing the decision of device selecting adopts action $\kappa(t)$ in state

s(t). The reward function of action evaluation is

$$r(s(t),\kappa(t)) = \frac{1}{|\sum_{i=1}^{n} \kappa_i|} \sum_{i=1}^{n} u_i^t \kappa_i^t$$

$$= \frac{1}{|\sum_{i=1}^{n} \kappa_i|} \sum_{i=1}^{n} \kappa_i^t \left[\omega \sigma \rho_i^t - (1-\omega) C_i^t \right].$$
(15)

The effect of performing action $\kappa(t)$ in round t is evaluated by the reward function $r(s(t), \kappa(t))$. The total cumulative reward is expressed as

$$R = \sum_{t=1}^{T} \gamma r(s(t), \kappa(t))$$
(16)

where $\gamma \in (0, 1]$ is the reward discount factor.

Next State: When finishing performing action κ(t) in state s(t), the next state s(t + 1) ⇐ s(t) + π(s(t)) is the prediction obtained through DT operating deep Q-network (DQN). The new updated state contains p(t + 1), f(t + 1), q(t + 1), k(t).

The goal of device selection is to minimize energy consumption and maximize model accuracy in FL. The DRL agent aims to maximize the cumulative reward by exploring κ as follows:

$$\kappa = \arg \max \mathbb{E}\left[\sum_{t=1}^{T} \gamma r(s(t), \kappa(t))\right].$$
 (17)

B. DRL-Based Device Selection Algorithm

Currently, RL is one of the widely adopted approaches to address dynamic programming problems [24]. The efficiency of the traditional RL-based methods is relatively low because they require calculating the value functions of all possible state and action space pairs. The DRL explores policy and value functions through deep neural networks (DNN), which is considered as the most effective method to solve complex MDP models [25]. In our constructed MDP model, the state space and the action space are continuous and high-dimensional. We address this MDP problem through the deep deterministic policy gradient (DDPG), a DRL framework based on actor-critic. The DDPG contains the actor policy network, critic value function network, and target network. In addition, the DDPG employs the replay memory buffer \mathcal{B} to store the experience transinformation, including the system state s(t), the action $\kappa(t)$, the corresponding reward $r(s(t), \kappa(t))$, and the next state s(t+1), for training the network.

1) Actor Network: The optimal device selection action can be provided by the actor network which takes system state s(t) as input and action κ as output. In order to generate different actions to explore potential superior policies, random noise is added to the decision-making mechanism as follows:

$$\kappa(t) = \pi \left(s(t) | w^{\pi} \right) + \vartheta(t) \tag{18}$$

where w^{π} is a parameter of the actor network and $\vartheta(t)$ is the random noise. The actor network update adopts the policy

Algorithm 1: DRL-Based Device Selection Algorithm.
Require: The actor network parameters w^{π} , the critic
network parameters w^Q , the target actor network
parameters $w^{\pi'}$, the target critic network parameters $w^{Q'}$;
Ensure: Optimized neural network parameters w^{π} and w^{Q} ;
1: Init : Initialize the network parameters w^{π} , w^{Q} ,
$w^{\pi'} \leftarrow w^{\pi}, w^{Q'} \leftarrow w^Q$; Initialize replay buffer \mathcal{B} ;
2: for each episode do
3: Initialize the IIoT environment setup and receive the
initial state $s(1)$; Initial random noise $\vartheta(t)$;
4: for $t = 1,, T$ do
5: Choose and execute action $\kappa(t)$, calculate
$r(s(t), \kappa(t))$ with (15) and receive $s(t+1)$;
6: Store transition $(s(t), \kappa(t), r(t), s(t+1))$ in \mathcal{B} ;
7: Sample M experiences $(s(i), \kappa(i), r(i), s(i+1))$
from \mathcal{B} ;
8: Calculate the target value $y(i)$ based on (24);
9: Update the actor network $\pi(s w^{\pi})$ based on (21);
10: Update the critic network $Q(s, \kappa w^Q)$ by (26);
11: Update the target network parameters $w^{\pi'}$ and $w^{Q'}$
based on (27);
12: end for

13: **end for**

gradient descent, which is defined as

$$\nabla_{w^{\pi}}J = \frac{1}{M} \sum_{i=1}^{M} \left[\nabla_{\kappa}Q(s_i,\kappa|w^Q) |_{\kappa=\pi(s_i|w^{\pi})} \nabla_{w^{\pi}}\pi(s_i) \right]$$
(19)

where M is the number of samples of experience data $(s(t), \kappa(t), r(t), s(t+1))$.

It is verified that the deterministic policy gradient is equivalent to the stochastic policy gradient $\nabla \pi(\kappa | s, w^{\pi})$ [26]. Therefore, the deterministic strategy gradient is shown as

$$\nabla \pi \left(\kappa | s, w^{\pi} \right) \approx \mathbb{E}_{\pi} \left[\nabla_{\kappa} Q(s, \kappa | w^{Q}) |_{\kappa = \pi(s_{i} | w^{\pi})} \nabla_{w^{\pi}} \pi(s) \right].$$
(20)

In each training iteration, a mini-batch of experiences $(s(t), \kappa(t), r(t), s(t+1))$ from replay memory buffer \mathcal{B} is randomly sampled to update network parameters w^{π} , in which the update formula is

$$w^{\pi} = w^{\pi} + \eta^{\pi} \cdot \mathbb{E}\left[\nabla_{\kappa}Q(s_i,\kappa|w^Q)|_{\kappa=\pi(s_i|w^{\pi})}\nabla_{w^{\pi}}\pi(s_i)\right]$$
(21)

where η^{π} is the actor network's learning rate.

2) Critic Network: For policy $\pi(s(t)|w^{\pi})$, we define an stateaction pair value function Q-value $Q^{\pi}(s(t), \kappa(t)|w^{Q})$, which represents the expected return of $\kappa(t)$ taken in s(t). Based on the Bellman optimality equation, the value function can be represented as

$$Q^{\pi}\left(s(t),\kappa(t)|w^{Q}\right) = \mathbb{E}\left[r\left(s(t),\kappa(t)\right) + \gamma Q\left(s(t+1),\pi\left(s(t+1)|w^{Q}\right)\right)\right].$$
⁽²²⁾

The critic network evaluates the taken action whose results are compared with the target value of the target network to ensure that the training parameter w^Q can be updated in the correct direction. The loss function of the critic network's training network parameters is defined as

$$L(w^Q) = \mathbb{E}\left[\left(Q^{\pi}\left(s(t),\kappa(t)|w^Q\right) - y\right)^2\right]$$
(23)

where $Q^{\pi}(s(t), \kappa(t)|w^Q)$ refers to the return value of action $\kappa(t)$ and y(t) denotes the objective value generated from the target network through

$$y(t) = r(s(t), \kappa(t)) + \gamma Q' \left(s(t+1), \pi' \left(s(t+1) | w^{\pi'} \right) | w^{Q'} \right)$$
(24)

where $w^{\pi'}$ and $w^{Q'}$ are the target network's parameters. Then the loss function's gradient can be expressed as

$$\nabla L(w^Q) = \mathbb{E}\left[2\left(y - Q\left(s(t), \kappa(t) | w^Q\right)\right) \nabla Q\left(s(t), \kappa(t)\right)\right].$$
(25)

The critic network's training method is similar to that of the actor network where a mini-batch of experience data is randomly sampled from the replay memory buffer \mathcal{B} to update network parameters, in which the update formula is

$$w^{Q} = w^{Q} + \eta \mathbb{E} \left[2 \left(y_{i} - Q \left(s_{i}, \kappa_{i} | w^{Q} \right) \right) \nabla Q \left(s_{i}, \kappa_{i} \right) \right]$$
(26)

where η is the critic network's learning rate.

3) Target Network and Experience Replay: To improve the robustness of network training, we introduce the target network, $Q'(s(t), \kappa(t)|w^{Q'})$ and $\pi'(s(t)|w^{\pi'})$, which are the copy network of the critic network and the actor network. The parameters of the target network are updated as follows:

$$\begin{cases} w^{Q'} = \tau w^Q + (1 - \tau) w^{Q'} \\ w^{\pi'} = \tau w^{\pi} + (1 - \tau) w^{\pi'} \end{cases}$$
(27)

where $\tau \in (0, 1]$ constrains the change of the target value.

Our scheme exploits the replay buffer mechanism in each training step to ensure that the training data is independently distributed. The device selection algorithm for the DRL-assisted FL is presented in Algorithm 1.

C. DRL-Based Asynchronous Federated Learning

Devices in the IIoT application scenario are highly heterogeneous, and the slowest device will limit the training speed of the synchronous learning solution, causing the so-called straggler effect. Therefore, we propose an asynchronous FL framework to address this issue. The main idea is to select the optimally participating devices, classify the devices with different utility values by cluster, and configure the corresponding aggregator for each cluster to realize asynchronous learning. In this case, each cluster can be trained at different local aggregation frequencies. In addition, we adopt Algorithm 1, based on the actor-critic DRL framework, to select devices that participate in asynchronous FL. Our proposed asynchronous FL framework mainly includes the following four steps.

1) *Device selection:* For the sake of improving the convergence rate and the model accuracy, a device with higher utility is selected to participate in FL within the given communication time. In the beginning, the server initializes the FL process through broadcasting the global model and the initialization parameter w_{ini} . The server

then selects the optimal subset of devices $N_i \in \mathcal{N}$ through the DRL-based algorithm.

- 2) Device clustering: We classify devices based on the data size and computing power by clustering algorithm Kmeans [27] and then assign corresponding aggregators to constitute the local training cluster. Hence, in the identical cluster of the local model, the training time of the device is similar which eliminates the straggler effect.
- 3) Local training: The distributed stochastic gradient descent is used for local training. In round t, the model $w_i(t)$ is trained by the local device i on its data D_i through calculating the local gradient descent $\nabla F_i(w_{t-1})$ according to w_{t-1} , as shown in (28). Then, device i transmits the updated local parameter $w_i(t)$ to the corresponding aggregators for aggregation

$$w_i(t) = w(t) - \tau \nabla F_i(w_{t-1}) \tag{28}$$

where τ represents the learning rate.

4) Global aggregation: The aggregator obtains the model trained by the local devices and performs global aggregation by aggregating the local model $w_i(t)$ into the weighted global model w(t) as follows:

$$w(t) = \frac{\sum_{i=1}^{N} \beta_i w_i(t)}{\sum_{i=1}^{N} \beta_i}$$
(29)

where N is the number of training devices and β_i denotes device *i*'s contribution capability factor to the global model in iteration t, which is determined by the data utility model, and $\sum_i \beta_i = 1$.

The synchronous weighted average training strategy has its drawbacks. For instance, it ignores the influence of differences between training data and is prone to overfitting and other problems. Moreover, the decision-making process usually merely considers the training accuracy without optimizing the problem from multidimensions. The proposed asynchronous framework with the device selection and clustering mechanism eliminates the straggler effect, effectively avoids inefficient devices and even malicious attacks, and improves the convergence rate and learning quality. Although the DRL-based method requires massive samples for training, it can improve the training effect and maintain the practical significance through addressing the problems like resource consumption and sample distribution.

V. EXPERIMENTS

The evaluation of the performance of the proposed asynchronous FL protocol is conducted on the MNIST dataset, which contains 60 000 training examples and 10 000 testing examples [28]. We simulate real IIoT applications, such as intelligent factory instrument recognition, traffic flow monitoring, robot path exploration, etc., through learning on the image dataset. To simulate the IIoT settings, we assume that there are 100 smart devices in the system and consider a scenario where a single BS is used as an aggregation server. The dataset is divided into 100 pieces, which are allocated to 100 smart devices. The convolutional neural network model, which includes two convolutional layers, two fully connected layers, and an average pooling layer,



Fig. 3. Evaluation of the data utility function.

is utilized as the local training model. We adopt the state-ofthe-art asynchronous and synchronous FL schemes, i.e., asynchronous federated stochastic gradient descent for vertically partitioned (AFSGD-VP) [13] and communication-efficient for federated averaging (CE-FedAvg) [14], as the baselines to evaluate the effect of the proposed scheme. In addition, we add two common baselines, namely centralized training and stand-alone training. The former is that the centralized dataset training has the optimal model accuracy, and the latter is that the training model on the local dataset has poor model accuracy.

We set four different EMD values $\phi = 0$, $\phi = 0.2$, $\phi = 0.4$, and $\phi = 0.8$ for evaluating the data utility function in (6). Different EMD values can be obtained by varying the number of labels and the data size on the local device. The larger EMD value results in the higher data similarity, which provides low-quality local model parameters for global aggregation, thereby reducing the global model accuracy. Here, the data utility is measured by the accuracy of the model prediction. Fig. 3 shows that the data utility function in (6) can fit the experimental results well. When $\phi = 0$ and $\phi = 0.8$, the training model accuracy is the highest and lowest, respectively. We refer to low-quality devices as inefficient devices, which have low data utility, poor communication, and computing capabilities, and an adverse effect on global aggregation.

In order to evaluate the impact of inefficient devices on FL, we compare the performance of the proposed schemes with different numbers of inefficient devices under the condition of no device selection. Among the 30 devices participating in the training, we set up 4, 8, and 12 inefficient devices, respectively. Figs. 4 and 5 show that the performance drops significantly with the increase of inefficient devices. In particular, when there are 12 inefficient devices in training, the model fails to converge due to the large proportion of inefficient devices. In addition to low-quality data, the reason for the rapid deterioration of performance is that devices with poor communication capabilities quit in the training process. The experimental results show that optimizing the selection of training devices plays an important role in improving the system's performance.

We evaluate the proposed scheme's performance on different numbers of training devices. To verify the developed device selection scheme's performance, we established three training device groups, 30, 50, and 70, with 10 inefficient devices in



Fig. 4. Accuracy of different numbers of inefficient devices.



Fig. 5. Loss of different numbers of inefficient devices.



Fig. 6. Accuracy of different numbers of training devices.

each group. Then, the proposed device selection algorithm is evaluated on the three groups of training devices and compared with the first group of devices without using the device selection algorithm. Figs. 6 and 7 show the prediction accuracy and the loss of the training model, respectively. The experimental results demonstrate that the scheme has excellent convergence and accuracy. As the number of devices involved increases from 30 to 50 to 70, the model's accuracy increases slightly. This is because a larger number of utility devices involved in the training results in a higher quality model. The comparison result of whether to adopt device selection or not shows that the device selection scheme can eliminate the adverse effect of the inefficient device on the training results.



Fig. 7. Loss of different numbers of training devices.



Fig. 8. Global accuracy of different benchmark schemes.



Fig. 9. Training time of different benchmark schemes.

The proposed scheme is compared with the baseline methods, and the accurate comparison of the resulting model is shown in Fig. 8. The accuracy of the centralized model is the highest, while the model accuracy of the stand-alone training methods is the lowest. Due to the insufficient number of samples and the single type of samples used for local training, the performance of the local optimal solution is lower than that of the global optimal solution. Our proposed scheme's performance is close to that of centralized training and better than AFSGD-VP and CE-FedAvg. In addition, our scheme requires fewer iterations than AFSGD-VP and CE-FedAvg to reach its optimum. However, the superior performance of concentrated training comes at the expense of security. The asynchronous training method AFSGD-VP ignores the influence of the inefficient device, while the synchronous training method CE-FedAvg is affected by the straggler effect. Our method takes these two aspects into consideration, resulting in a better performance. In addition, we evaluate our proposed scheme in terms of time cost and a comparison with the baseline scheme, as shown in Fig. 9. As seen from the figure, our proposed scheme is superior to other methods in the case of minimal training time. This is because the device selection and clustering mechanism in our scheme eliminate the straggler effect and effectively avoid inefficient devices, through which the convergence speed and learning quality are improved.

VI. CONCLUSION

This article proposed the DTEI architecture which employs DTs in IIoT for real-time perception and intelligent decisionmaking. We exploited FL to construct DTs model on the basis of the operating state and the behavior of devices. With the objective of improving the model training efficiency and accuracy of the IIoT device, we developed a DRL-assisted FL framework. The DRL is used to select some high-efficiency IIoT devices for aggregation. In addition, the straggler effects of heterogeneous IIoT scenarios can be eliminated by our proposed asynchronous FL framework. Experimental results were presented to show that the proposed scheme performs better than the benchmark scheme in aspects of convergence rate and training accuracy.

REFERENCES

- N. Zhang *et al.*, "Physical-layer authentication for Internet of Things via WFRFT-based gaussian tag embedding," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 9001–9010, Sep. 2020.
- [2] Y. Qu, S. R. Pokhrel, S. Garg, L. Gao, and Y. Xiang, "A blockchained federated learning framework for cognitive computing in industry 4.0 networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2964–2973, Apr. 2021.
- [3] T. Wang, H. Luo, W. Jia, A. Liu, and M. Xie, "MTES: An intelligent trust evaluation scheme in sensor-cloud-enabled industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2054–2062, Mar. 2020.
- [4] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and US air force vehicles," in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Structures*, Honolulu, HI, USA, 2012, Art. no. 1818.
- [5] O. El Marai, T. Taleb, and J. Song, "Roads infrastructure digital twin: A step toward smarter cities realization," *IEEE Netw.*, vol. 35, no. 2, pp. 136–143, Mar. 2021.
- [6] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: Stateof-the-art," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2018.
- [7] 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. 2019.
- [8] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for stochastic computation offloading in digital twin networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4968–4977, Jul. 2021.
- [9] B. McMahan, E. Moore, D. Ramage, and S. Hampson, "Communicationefficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, Fort Lauderdale, FL, USA, 2017, pp. 1273– 1282.
- [10] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive IoT networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4641–4654, May 2020.
- [11] H. Sun, S. Li, F. R. Yu, Q. Qi, J. Wang, and J. Liao, "Toward communication-efficient federated learning in the Internet of Things with edge computing," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 11053–11067, Nov. 2020.

- [12] K. Wei *et al.*, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3454–3469, Apr. 2020.
- [13] B. Gu, A. Xu, Z. Huo, C. Deng, and H. Huang, "Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 1–13, Jun. 2021.
- [14] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5986–5994, Jul. 2020.
- [15] Y. Guo, Z. Zhao, K. He, S. Lai, J. Xia, and L. Fan, "Efficient and flexible management for industrial Internet of Things: A federated learning approach," *Comput. Netw.*, vol. 192, Jun. 2021, Art. no. 108122.
- [16] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen, and L. Zhao, "Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4925–4934, Jul. 2021.
- [17] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communicationefficient federated learning for digital twin edge networks in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5709–5718, Aug. 2020.
- [18] W. Sun, S. Lei, Z. Liu, and Y. Zhang, "Adaptive federated learning and digital twin for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5605–5614, Aug. 2020.
- [19] 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.
- [20] Y. Jiao, P. Wang, S. Feng, and D. Niyato, "Profit maximization mechanism and data management for data analytics services," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2001–2014, Jun. 2018.
- [21] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," 2018. [Online]. Available: https://arxiv.org/ abs/1806.00582
- [22] Y. Jiao, D. Wang, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," *IEEE Trans. Mob. Comput.*, vol. 20, no. 10, pp. 3034–3048, Oct. 2021.
- [23] 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. 12240–12251, Oct. 2020.
- [24] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.
- [25] M. Li, J. Gao, L. Zhao, and X. Shen, "Deep reinforcement learning for collaborative edge computing in vehicular networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 4, pp. 1122–1135, Dec. 2020.
- [26] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn.*, Beijing, China, 2014, pp. 387–395.
- [27] J. Choi and S. R. Pokhrel, "Federated learning with multichannel ALOHA," *IEEE Wireless Commun. Lett.*, vol. 9, no. 4, pp. 499–502, Apr. 2020.
- [28] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/



Wei Yang received the B.S. degree in communication engineering from Nanyang Normal University, Nanyang, China, in 2014, and the M.S. degree in signal and information processing from the School of Computer Science, Zhengzhou University of Light Industry, Zhengzhou, China, in 2018. He is currently working toward the Ph.D. degree in electronic science and technology with the School of Automation and Information Engineering, Xi'an University of Technology, Xi'an, China.

His research interests include federated learning, optimization, and wireless networks.



Wei Xiang (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees, both in electronic engineering, from the University of Electronic Science and Technology of China, Chengdu, China, in 1997 and 2000, respectively, and the Ph.D. degree in telecommunications engineering from the University of South Australia, Adelaide, Australia, in 2004. He is currently a Cisco Research Chair of Al and IoT and the Director of the Cisco-La Trobe Centre for Al and IoT, La Trobe University, Melbourne, Australia. Pre-

viously, he was Foundation Chair and Head of Discipline of IoT Engineering, James Cook University, Cairns, Australia. He has authored or coauthored more than 250 peer-reviewed papers including three books and 200 journal articles. He has served in a large number of international conferences in the capacity of General Co-Chair, TPC Co-Chair, Symposium Chair, etc. His research interest includes the Internet of Things, wireless communications, machine learning for IoT data analytics, and computer vision.

Prof. Xiang, due to his instrumental leadership in establishing Australia's first accredited Internet of Things Engineering degree program, was inducted into Pearcy Foundation's Hall of Fame in October 2018. He is an elected Fellow of the IET in U.K. and Engineers Australia. He was the recipient of the TNQ Innovation Award in 2016, and Pearcey Entrepreneurship Award in 2017, and Engineers Australia Cairns Engineer of the Year in 2017. He was a corecipient of four Best Paper Awards at WISATS'2019, WCSP'2015, IEEE WCNC'2011, and ICWMC'2009. He has been awarded several prestigious fellowship titles. He was named a Queensland International Fellow (2010-2011) by the Queensland Government of Australia, an Endeavour Research Fellow (2012-2013) by the Commonwealth Government of Australia, a Smart Futures Fellow (2012-2015) by the Queensland Government of Australia, and a JSPS Invitational Fellow jointly by the Australian Academy of Science and Japanese Society for Promotion of Science (2014-2015). He was the Vice Chair of the IEEE Northern Australia Section from 2016 to 2020. He was an Editor for IEEE COMMUNICATIONS LETTERS from 2015 to 2017 and is currently an Associate Editor for IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE INTERNET OF THINGS JOURNAL, IEEE ACCESS, and Nature journal of SCIENTIFIC REPORTS.



Yuan Yang received the B.S. degree in applied electronic technology, and the M.S., and Ph.D. degrees in microelectronics and solid state electronics from the Xi'an University of Technology, Xi'an, China, in 1997, 2000, and 2004, respectively.

From 2000 to 2004, she was a Lecturer, and from 2004 to 2009, she was an Assistant Professor. Since 2009, she has been a Professor with the Department of Electronics, Xi'an University of Technology. In 2005, she was a Visit-

ing Scholar with Kyushu University, Fukuoka, Japan. Her research interests include digital-analog mixed integrated circuit designs, driver and protection technology of power semiconductor device, deep learning algorithm, and hardware acceleration.



Peng Cheng (Member, IEEE) received the B.S. and M.S. (hons.) degrees in communication and information systems from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2006 and 2009, respectively, and the Ph.D. degree in communication and information systems from Shanghai Jiao Tong University, Shanghai, China, in 2013. From 2014 to 2017, he was a Postdoctoral Research Scientist with CSIRO, Sydney, Australia. From 2017 to 2020, he was an ARC DE-

CRA Fellow/Lecturer with the University of Sydney, Sydney, Australia. He is currently an ARC DECRA Fellow and a Senior Lecturer (Tenured Associate Professor in U.S. systems) with the Department of Computer Science and Information Technology, La Trobe University, Melbourne, Australia, and is affiliated with the University of Sydney. He has authored or coauthored more than 70 peer-reviewed research papers in leading international journals and conferences. His current research interests include wireless AI, machine learning, IoT, millimeter-wave communications, and compressive sensing theory.