Spectrum and Computing Resource Management for Federated Learning in Distributed Industrial IoT

Weiting Zhang*, Dong Yang*, Wen Wu†, Haixia Peng†, Hongke Zhang*, and Xuemin (Sherman) Shen†
*School of Electronic and Information Engineering, Beijing Jiaotong University, China
†Department of Electrical and Computer Engineering, University of Waterloo, Canada
Email: {17111018, dyang, hkzhang}@bjtu.edu.cn, {w77wu, h27peng, sshen}@uwaterloo.ca

***Abstract*—Federated learning (FL) is a distributed paradigm to support deep neural network (DNN) training while preserving the data owners' privacy. In this paper, we investigate the resource management problem for FL in distributed industrial Internet of Things (IIoT) networks. Specifically, we introduce a three-layer collaborative architecture to support FL. DNNs are trained locally at the selected IIoT devices, and then the DNN model parameters are aggregated by edge servers every FL epoch or by a cloud server every a few FL epochs to update the global DNN model. To enable efficient FL in the resource-limited IIoT networks, judicious computing and spectrum resource allocation is required for training and transmitting the DNN model parameters. Thus, we formulate a joint device selection and resource allocation problem to minimize the FL evaluating loss while satisfying the strict FL epoch delay and devices' energy consumption requirements. Since the decisions of device selection and resource allocation are coupled, we transform the joint optimization problem into a Markov decision process and propose a dynamic resource management scheme based on deep reinforcement learning approaches to efficiently facilitate the FL. Simulation results demonstrate that the proposed scheme can effectively improve the FL performance comparing with benchmarks.**

I. Introduction

With the advent of the Internet of Things (IoT), tremendous data is generated at the network edge. According to the forecast from IDC, 41.6 billion connected IoT devices will generate 79.4 zettabytes of data by 2025 [1]. Endowed with voluminous data for training, deep neural networks (DNNs) are expected to greatly improve usability, especially for industrial IoT (IIoT) applications [2]. For example, for condition monitoring applications, through training a DNN model on a large dataset, the DNN can provide high-accuracy identification service for the health conditions of industrial facilities. However, traditional centralized DNN training approaches need to collect massive raw data from network nodes, which leads to critical information leakage issues [3].

To solve the above issue, Google has proposed the paradigm of federated learning (FL), in which the DNN models are distributed trained at local devices and then the local model parameters are periodically aggregated in a centralized node to update the global model by a federated averaging (FedAvg) algorithm [4]. After local training and parameter aggregation for multiple FL epochs, a global DNN model can be obtained which can provide high-accuracy performance. Within the FL paradigm, not only DNN models can be trained on the dispersive datasets held by the local devices, but also the device's local data does not need to be transmitted to the centralized node, thus preserving data privacy for data owners.

There are some prior research works on FL. Some works focus on device selection to accelerate the FL convergence. Generally, selecting more devices in each FL epoch can improve the FL convergence. To this end, Yang *et al*. [5] aimed at maximizing the number of participating devices in each FL epoch, while more participating devices may result in higher communication cost due to the excessive parameter transmission. Wang *et al*. [6] proposed a learning-based scheme to select devices to participate in the FL, thereby effectively speeding up FL convergence. In addition, many efforts have been devoted to efficient resource allocation to support the parameter aggregation of DNN models. In [7], a training delay minimization problem is studied in wireless networks, in which the spectrum resource is optimally allocated for the transmission of model parameters. In [8], Zeng *et al.* proposed an opportunistic spectrum access scheme to minimize the energy consumption of FL. The above works provide valuable insights to facilitate efficient FL within a single base station (BS) coverage. Different from these schemes, our work focuses on enabling FL in a distributed IIoT network, in which the required data is geographically dispersed among multiple BSs and FL needs to be efficiently operated with constrained network resources.

Enabling FL in distributed IIoT networks faces many challenges. Firstly, smart factories are generally deployed in different geographical areas. For example, an industrial company would possess three smart factories located in three different cities, respectively. To provide services for all factories, such as health condition identification of industrial facilities, the company should establish a global DNN model based on the data generated by these factories. However, due to the geographically distributed factories, directly applying single aggregation node based FL architecture (i.e., device-cloud) to support DNN training will lead to higher backbone communication overhead. Secondly, the IIoT device's energy is limited and cherishable, which should be properly allocated for each FL epoch to satisfy the energy consumption requirement. In the FL, the energy of IIoT devices is consumed in two steps: (1) local model training and (2) model parameter transmission. Satisfying energy consumption requirements desires judicious resource allocation for both spectrum and computing resources to enable more IIoT devices to participate in FL.

In this paper, we investigate the resource management problem for efficient FL in distributed IIoT networks. The main contributions of this paper are three-fold:

- To enable efficient FL over geographically dispersed data, we introduce a three-layer collaborative FL architecture in the distributed IIoT networks. The model parameters that are locally trained at IIoT devices are aggregated by edge servers every FL epoch or by a cloud server every a few FL epochs. As such, backbone network resources for parameter transmission can be greatly saved.
- We formulate the three-layer FL as a stochastic optimization problem. The goal is to optimally make device selection and resource allocation decisions to minimize the FL evaluating loss while satisfying the FL epoch delay and devices' energy consumption requirements.
- To solve the coupled optimization problem, we transform the formulated problem into a Markov decision process (MDP), and propose a deep reinforcement learning (RL) based scheme to learn the optimal device selection and resource allocation policy, thereby effectively facilitating FL in distributed IIoT networks.

The remainder of this paper is organized as follows. In Section II, the system model and problem formulation are described. In Section III, the formulated problem is transformed into an MDP, and a deep RL-based algorithm is proposed to solve the MDP. Simulation results are presented in Section IV and concluding remarks are provided in Section V.

II. System Model and Problem Formulation

In this section, a collaborative FL architecture is presented to support efficient FL in distributed IIoT networks, followed by the FL epoch delay and energy consumption models.

A. Collaborative FL Architecture for IIoT

As shown in Fig. 1, we consider a three-layer collaborative architecture to support FL in large-scale distributed IIoT networks [9]. The architecture is composed of three layers, namely, device, edge, and cloud layers. Each layer has the specific functionalities: (1) In the device layer, energy-limited IIoT devices, such as industrial gateways that are endowed with computing power to support local model training, are distributed among smart factories located in different cities [2]. DNN models are locally trained at IIoT devices to complete the parameter updating while preserving the devices' privacy; (2) In the edge layer, edge servers deployed on the BSs are utilized to perform edge parameter aggregations after each local training; and (3) In the cloud layer, a cloud server is deployed to conduct global parameter aggregations after a certain number of edge aggregations. In addition, a central controller is deployed at the cloud server to coordinate the FL operation in the considered IIoT networks.

We consider a set of edge servers, denoted by $\mathcal{M} = \{1, 2, \cdots, M\}$. The set of IIoT devices under the coverage of edge server $i \in \mathcal{M}$ is given by $\mathcal{N}_i = \{1, 2, \cdots, |\mathcal{N}_i|\}$. The local dataset held by the IIoT device j under edge server i, denoted by $\mathcal{D}_{i,j} = \{(\boldsymbol{x}_d, y_d)\}_d$, cannot be shared with

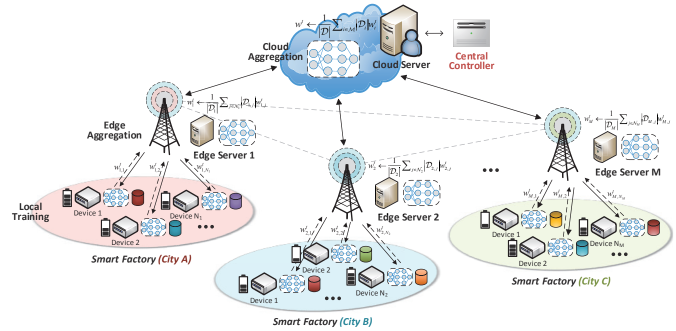


Fig. 1. The three-layer collaborative FL architecture for the distributed IIoT networks.

other network nodes, where $\boldsymbol{x}_d$ and y_d are the input and the label, respectively, d is the sample index and $|\mathcal{D}_{i,j}|$ is the sample volume of the IIoT device. In addition, $\mathcal{D}_{i,j}^T$ and $\mathcal{D}_{i,j}^E$ are the training and evaluating sample sets of the IIoT device, respectively, where $\mathcal{D}_{i,j}^T \cup \mathcal{D}_{i,j}^E = \mathcal{D}_{i,j}$. In the three-layer collaborative architecture, the FL operates in terms of FL epoch as follows.

1) Task Initialization: In the cloud server, a training task is initialized, and a global DNN model with a certain number of parameters is instantiated, such as a diagnostic DNN that is used to identify the health conditions of industrial facilities.

2) Device Selection and Resource Allocation: Once the training task is initialized, the central controller will make the device selection and resource allocation decisions. The decisions include: (1) which IIoT devices should be selected to participate in the current FL epoch; and (2) how much spectrum and computing resources should be allocated for model parameter transmission and local model training, respectively.

3) Global Model Parameter Distribution: With the decisions made by the central controller, global model parameters can be distributed to the selected IIoT devices from the cloud according to the available downlink spectrum resources.

4) Local Model Training: The model parameters are respectively trained on the local training samples of the participating IIoT devices for one FL epoch with the allocated computing resources.

5) Edge Aggregation: After each local update on IIoT devices, each edge server $i \in \mathcal{M}$ aggregates the devices' model parameters using the allocated uplink spectrum resources. With the aggregated parameters $\boldsymbol{w}_i$, we can obtain the evaluating loss $f_d(\boldsymbol{w}_i)$ on each device's evaluating sample of edge server i. Then, the evaluating loss $F_i(\boldsymbol{w}_i)$ for edge server i can be given by $F_i(\boldsymbol{w}_i) = 1/\left|\mathcal{D}_i^E\right| \sum_{d \in \mathcal{D}_i^E} f_d(\boldsymbol{w}_i)$, where $\left|\mathcal{D}_i^E\right|$ is the evaluating sample volume held by the participating IIoT devices of edge server i. Thus, the evaluating loss $F_e(\boldsymbol{w}_i)$ for the edge aggregation can be defined as:

$$F_e(\boldsymbol{w}_i) = \frac{1}{M} \sum_{i \in \mathcal{M}} F_i(\boldsymbol{w}_i), \tag{1}$$

which is a mean value of $F_i(\boldsymbol{w}_i)$ in terms of the edge server's number. In the next FL epoch, the aggregated model parameters $\boldsymbol{w}_i$ can be distributed to the participating IIoT devices from the edge servers for further updating.

6) Cloud Aggregation: To reduce the communication overhead between the edge and cloud, the cloud server aggregates all the edge servers' model parameters only after every t_1 edge aggregations. Thus, the evaluating loss can be described as:

$$F_c(\boldsymbol{w}) = \frac{1}{|\mathcal{D}|}\sum_{d=1}^{|\mathcal{D}|} f(\boldsymbol{w}), \tag{2}$$

where $\boldsymbol{w}$ denotes the model parameters aggregated by the cloud server and $f(\boldsymbol{w})$ denotes the evaluating loss on each evaluating sample of the participating IIoT devices with parameters $\boldsymbol{w}$, and $|\mathcal{D}|$ denotes the evaluating sample volume held by all participating IIoT devices in current FL epoch.

B. FL Epoch Delay Model

During the FL process, each FL epoch is constrained by an epoch delay $T_{i,j}^{\max}$, which is composed of three parts defined as follows.

1) Delay of Model Parameter Distribution: In each FL epoch $t \in \mathcal{R}$, a set of IIoT devices are selected to participate in the DNN training. Let $o_{i,j,t} \in \{0,1\}, \forall i \in \mathcal{M}, j \in \mathcal{N}_i, t \in \mathcal{R}$ denote the binary decision variable of device selection. Here, if $o_{i,j,t}$ is 1, IIoT device j under edge server i is selected in FL epoch t, otherwise is not. We consider orthogonal frequency allocation for model parameter distribution from the edge or the cloud to the IIoT device (i.e., downlink) and aggregation (i.e., uplink). Bandwidth $B^{\downarrow}$ and $B^{\uparrow}$ are available for the downlink and uplink, respectively. Hence, the downlink spectrum efficiency for BS i is described as $\gamma_{i,j,t}^{\downarrow} = \log_2\left(1 + p_{i,t}|g_{i,j,t}|^2/\nu^2\right), \forall i \in \mathcal{M}, j \in \mathcal{N}_i$, where $p_{i,t}$ denotes the transmission power of BS i, $|g_{i,j,t}|^2$ denotes the channel gain, and ν^2 denotes the Gaussian white noise power [10]. Here, BS i allocates a fraction $\xi_{i,j,t}^{\downarrow}$ of bandwidth $B^{\downarrow}$ to device j, where $0 \leq \xi_{i,j,t}^{\downarrow} \leq 1$, and we have $\sum_{i\in\mathcal{M},j\in\mathcal{N}_i} o_{i,j,t}\xi_{i,j,t}^{\downarrow} = 1$. Thus, the downlink transmission rate for BS i is described as

$$R_{i,j,t}^{\downarrow} = \xi_{i,j,t}^{\downarrow} B^{\downarrow} \gamma_{i,j,t}^{\downarrow}, \forall i \in \mathcal{M}, j \in \mathcal{N}_i. \tag{3}$$

The parameter distribution delay $d_{i,j,t}^{\downarrow}$ has two cases. For the cloud aggregation, the global model parameters need to be distributed from the cloud server to IIoT devices, then $d_{m,n,t}^{\downarrow} = S_g/R_{m,n,t}^{\downarrow} + S_g/R^c$, where S_g (in bits) denotes the size of the global model parameters and R^c is the backhaul transmission rate between the cloud and the BS. For the edge aggregation, the edge servers respectively distribute its aggregated model parameters to IIoT devices, and thus $d_{m,n,t}^{\downarrow} = S_g/R_{m,n,t}^{\downarrow}$.

2) Delay of Local Model Training: The IIoT devices are endowed with the same computing capabilities (CPU cycles per second), denoted by f. Considering the long-term energy consumption constraints of IIoT devices, the computing resource should be judiciously allocated in each FL epoch, thereby satisfying the long-term energy requirement of IIoT devices. Let $\eta_{i,j,t}\,(0 \leq \eta_{i,j,t} \leq 1)$ denote the fraction of computing resource used in the FL epoch. Thus, the delay for DNN training at IIoT device j under edge server i's coverage can be given by

$$\tau_{i,j,t} = \frac{a\left|\mathcal{D}_{i,j}^T\right|}{\eta_{i,j,t} f}, \forall i \in \mathcal{M}, j \in \mathcal{N}_i, \tag{4}$$

where a is the number of CPU cycles required for computing per bit data of IIoT device j under BS i, and $\left|\mathcal{D}_{i,j}^T\right|$ denotes the size of training samples held by the participating IIoT devices.

3) Delay of Model Parameter Uploading: For model parameter aggregations, the uplink spectrum efficiency is described as $\gamma_{i,j,t}^{\uparrow} = \log_2\left(1 + p_{i,j,t}|g_{i,j,t}|^2/\nu^2\right), \forall i \in \mathcal{M}, j \in \mathcal{N}_i$, where $p_{i,j,t}$ denotes the transmission power from IIoT device j to edge server i. Thus, the uplink transmission rate for IIoT device j is given by $R_{i,j,t}^{\uparrow} = \xi_{i,j,t}^{\uparrow} B^{\uparrow} \gamma_{i,j,t}^{\uparrow}, \forall i \in \mathcal{M}, j \in \mathcal{N}_i$. Here, IIoT device j is allocated with a fraction $\xi_{i,j,t}^{\uparrow}$ of bandwidth $B^{\uparrow}$, where $0 \leq \xi_{i,j,t}^{\uparrow} \leq 1$, and we have $\sum_{i\in\mathcal{M},j\in\mathcal{N}_i} o_{i,j,t}\xi_{i,j,t}^{\uparrow} = 1$. Then, for the cloud aggregation, the parameter uploading delay $d_{i,j,t}^{\uparrow}$ is $S_l/R_{i,j,t}^{\uparrow} + S_l/R^c, \forall i \in \mathcal{M}, j \in \mathcal{N}_i$. Otherwise, the $d_{i,j,t}^{\uparrow}$ is $S_l/R_{i,j,t}^{\uparrow}$. Here, S_l (in bits) denotes the size of the local model parameters. Since the size of DNN model parameters is the same during the FL process, we have $S_l = S_g$.

Given the transmission- and training-related delays, the total delay for each FL epoch of each IIoT device can be given by

$$T_{i,j,t} = d_{i,j,t}^{\downarrow} + \tau_{i,j,t} + d_{i,j,t}^{\uparrow}, \forall i \in \mathcal{M}, j \in \mathcal{N}_i. \tag{5}$$

Here, the total delay is leveraged to evaluate whether the IIoT device satisfies the FL epoch delay requirement or not.

C. Energy Consumption Model

In the long-term FL process, each IIoT device is constrained by its own energy capacity $E_{i,j}^{\max}$, which should be satisfied to ensure the IIoT devices can efficiently participate in the DNN training. In the model parameter uploading stage, the energy consumption of IIoT device j under edge server i's coverage can be calculated by

$$E_{i,j,t,com} = p_{i,j,t} d_{i,j,t}^{\uparrow}, \forall i \in \mathcal{M}, j \in \mathcal{N}_i. \tag{6}$$

In addition, performing DNN training on IIoT devices also consumes energy, which is related to the computing workload of the local training process and the allocated computing resource of the IIoT devices. Thus, the DNN training energy consumption of each IIoT device can be given by

$$E_{i,j,t,cmp} = \frac{\kappa}{2} a\left|\mathcal{D}_{i,j}^T\right| (\eta_{i,j,t} f)^2, \forall i \in \mathcal{M}, j \in \mathcal{N}_i, \tag{7}$$

where κ denotes the effective capacitance coefficient of the IIoT devices' computing chipset [9].

Taking $E_{i,j,t,com}$ and $E_{i,j,t,cmp}$ into consideration, the cumulative energy consumption can be given by

$$E_{i,j,t} = E_{i,j,t-1} + E_{i,j,t,com} + E_{i,j,t,cmp}, \forall i \in \mathcal{M}, j \in \mathcal{N}_i. \tag{8}$$

Here, the cumulative energy consumption is used to calculate the remaining energy capacity, thereby evaluating whether the IIoT device satisfies the energy consumption requirement or not.

D. FL Evaluating Loss

The FL operates epoch-by-epoch with the selected IIoT devices which can satisfy the FL epoch delay and energy consumption requirements. The goal of the iteration process is to find the optimal DNN model parameters $\boldsymbol{w}^*$ which can minimize the evaluating loss on the evaluating samples of the IIoT devices. Considering both the edge and cloud aggregation cases, the FL evaluating loss of the DNN model can be expressed as

$$F(\boldsymbol{w}) = \begin{cases} F_e(\boldsymbol{w}_i), & \text{if } mod(t, t_1) \neq 0, \\ F_c(\boldsymbol{w}), & \text{if } mod(t, t_1) = 0. \end{cases} \tag{9}$$

Here, if $mod(t, t_1) = 0$, the cloud aggregates all DNN model parameters and obtains the evaluating loss with the global parameters $\boldsymbol{w}$, otherwise the edge servers respectively aggregate its covered model parameters of IIoT devices and obtain the evaluating loss with the aggregated parameters $\boldsymbol{w}_i$.

E. Problem Formulation

In this work, we focus on enabling efficient FL in distributed IIoT networks via flexible device selection and resource allocation. As described in the previous subsection, the federated evaluating loss is utilized to evaluate the convergence performance of FL. Thus, aiming at minimizing the federated evaluating loss of DNN models while satisfying the FL epoch delay and long-term energy consumption requirements of IIoT devices, we formulate the problem as follows:

$$\text{P1}: \min_{\boldsymbol{o}, \boldsymbol{\xi}^{\downarrow}, \boldsymbol{\xi}^{\uparrow}, \boldsymbol{\eta}} F(\boldsymbol{w})$$

$$\begin{aligned}
\text{s.t.} \quad & E_{i,j,t} \leq E_{i,j}^{\max}, \forall i \in \mathcal{M}, j \in \mathcal{N}_i, t \in \mathcal{R}, & \text{(10a)} \\
& T_{i,j,t} \leq T_{i,j}^{\max}, \forall i \in \mathcal{M}, j \in \mathcal{N}_i, t \in \mathcal{R}, & \text{(10b)} \\
& o_{i,j,t} \in \{0,1\}, \forall i \in \mathcal{M}, j \in \mathcal{N}_i, t \in \mathcal{R}, & \text{(10c)} \\
& \sum_{i \in \mathcal{M}, j \in \mathcal{N}_i} o_{i,j,t} \xi_{i,j,t}^{\downarrow} = 1, \xi_{i,j,t}^{\downarrow} \in [0,1], \forall t \in \mathcal{R}, & \text{(10d)} \\
& \sum_{i \in \mathcal{M}, j \in \mathcal{N}_i} o_{i,j,t} \xi_{i,j,t}^{\uparrow} = 1, \xi_{i,j,t}^{\uparrow} \in [0,1], \forall t \in \mathcal{R}, & \text{(10e)} \\
& \eta_{i,j,t} \in [0,1], \forall i \in \mathcal{M}, j \in \mathcal{N}_i, t \in \mathcal{R}. & \text{(10f)}
\end{aligned}$$

The optimization variables include four components: device selection decision $\boldsymbol{o} = \{o_{1,1,t}, \cdots, o_{i,j,t}, \cdots, o_{M,|\mathcal{N}_i|,t}\}, \forall t \in \mathcal{R}$, downlink and uplink spectrum resource allocation decisions $\boldsymbol{\xi}^{\downarrow} = \{\xi_{1,1,t}^{\downarrow}, \cdots, \xi_{i,j,t}^{\downarrow}, \cdots, \xi_{M,|\mathcal{N}_i|,t}^{\downarrow}\}, \forall t \in \mathcal{R}$ and $\boldsymbol{\xi}^{\uparrow} = \{\xi_{1,1,t}^{\uparrow}, \cdots, \xi_{i,j,t}^{\uparrow}, \cdots, \xi_{M,|\mathcal{N}_i|,t}^{\uparrow}\}, \forall t \in \mathcal{R}$, and computing resource allocation decision $\boldsymbol{\eta} = \{\eta_{1,1,t}, \cdots, \eta_{i,j,t}, \cdots, \eta_{M,|\mathcal{N}_i|,t}\}, \forall t \in \mathcal{R}$. Constraints (10a) and (10b) restrict the cumulative energy consumption and the FL epoch delay should be within $E_{i,j}^{\max}$ and $T_{i,j}^{\max}$, respectively. Constraint (10c) indicates whether select an IIoT device to participate in the current FL epoch. Constraints (10d)-(10f) ensure that the resource allocation decisions should be made with the available bandwidth $B^{\downarrow}$ and $B^{\uparrow}$, and computing f resources, respectively. Through solving the formulated problem, FL can be efficiently operated in the considered scenario with satisfied FL epoch delay and long-term energy consumption requirements. As problem P1 is a mix-integer nonlinear optimization problem and the considered decision variables are coupled with each other, it is difficult to directly solve it via conventional optimization methods [11].

III. Deep RL for Efficient Federated Learning

A. Problem Transformation

To enable the efficient FL, we transform the formulated problem P1 into an MDP and solve it by a deep RL-based method to achieve flexible device selection and resource allocation, thereby minimizing the FL evaluating loss. The mechanism of the deep RL is that the agent iteratively captures the dynamics of the environment to learn the optimal policy π^*. The policy can accurately map a state $s_t \in \mathcal{S}$ to an action $a_t \in \mathcal{A}$, which is evaluated by the reward function r_t. In the considered distributed IIoT scenario, the central controller is modelled as an agent, and everything beyond that is regarded as environment. The state, action, and reward can be described as follows.

1) State: In FL epoch t, the central controller collects the state information, $s_t = \{E_{i,j}^{\max} - E_{i,j,t-1}\}_{\forall i \in \mathcal{M}, j \in \mathcal{N}_i}$, which indicates the remaining energy of IIoT devices.

2) Action: Based on the observed states, the agent makes decisions, $a_t = \{o_{i,j,t}, \xi_{i,j,t}^{\downarrow}, \xi_{i,j,t}^{\uparrow}, \eta_{i,j,t}\}_{\forall i \in \mathcal{M}, j \in \mathcal{N}_i}$, to select the IIoT devices to participate in the current FL epoch, and allocate the spectrum and computing resources to the participating devices.

3) Reward: The agent can obtain a reward after the action a_t has been taken based on the observed state s_t. To minimize the FL evaluating loss of DNN models, we therefore define the reward function as

$$r_t = \begin{cases} -F_e(\boldsymbol{w}), & \text{if } mod(t, t_1) \neq 0, \\ -F_c(\boldsymbol{w}), & \text{if } mod(t, t_1) = 0, \\ -H, & \textit{no aggregation}, \end{cases} \tag{11}$$

where $F_e(\boldsymbol{w})$ and $F_c(\boldsymbol{w})$ denote the evaluating losses of DNN models after edge and cloud aggregations, respectively. Here, H is a relatively large constant that penalizes the decisions that cannot aggregate any model parameters given the delay and energy constraints.

B. DDPG-Based Algorithm for Efficient FL

As previously mentioned, this paper focuses on ensuring the efficient FL in distributed IIoT networks, thus the device selection and resource allocation (DSRA) decisions should be made in real time. To this end, instead of conventional optimization approaches, we leverage a deep RL method, i.e., deep deterministic policy gradient (DDPG) [12], to solve the transformed MDP problem.

In the proposed DDPG-based resource management algorithm, the objective is to select a DSRA action with the optimal policy π^* to minimize the FL evaluating loss of DNN models. Specifically, the algorithm is mainly supported by two components: (1) replay memory technique and (2) "current-

TABLE I
PARAMETERS OF DDPG-BASED ALGORITHM.

Parameter	Value	Parameter	Value
Actor	(180, 300)	Critic	(450, 100)
Replay buffer size	10,000	Batch size	64
Episode number	300	Epoch number	200
Discount factor	0.95	Update factor	0.01
Actor learning rate	10^{-4}	Critic learning rate	5×10^{-4}

target" network pairs. The former is utilized to store the experiences of the learning agent during iterative interaction with the environment, thereby allowing the learning algorithm to benefit from the uncorrelated transitions. The latter is designed for achieving effective convergence of classical actor-critic based algorithms, in which the current network parameters are slowly copied from the target networks via a "soft" update strategy, such that greatly stabilizing the learning process.

The details of the algorithm is given as follows: Firstly, in each learning step, the agent randomly samples a minibatch experience (s_k, a_k, r_k, s_{k+1}) from $\mathcal{K}_b$. Secondly, the actor makes DSRA actions $a = \pi_{\boldsymbol{\theta}}(s_k)$ and $a' = \pi_{\boldsymbol{\theta}'}(s_{k+1})$ via the current and target networks with parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. The critic evaluates a and a_k via its current network with parameter $\boldsymbol{\phi}$ to calculate $Q_{\boldsymbol{\phi}}(s_k, a_k)$, and the target critic evaluates a' with parameter $\boldsymbol{\phi}'$ to calculate $Q_{\boldsymbol{\phi}'}(s_{k+1}, \pi_{\boldsymbol{\theta}'}(s_{k+1}))$. Thirdly, the target Q-value can be calculated by $y_k = r_k + \gamma Q_{\boldsymbol{\phi}'}(s_{k+1}, \pi_{\boldsymbol{\theta}'}(s_{k+1}))$, where $\gamma \in [0, 1]$ denotes the discount factor. Thus, the loss function can be described as

$$L(\boldsymbol{\phi}) = \frac{1}{2K_b} \sum_{k \in \mathcal{K}_b} \left(y_k - Q_{\boldsymbol{\phi}}(s_k, a_k)\right)^2, \tag{12}$$

where K_b denotes the size of minibatch experiences. After each iteration, the parameter $\boldsymbol{\phi}$ can be updated via minimizing the loss function.

In addition, the policy objective function of the actor can be described as

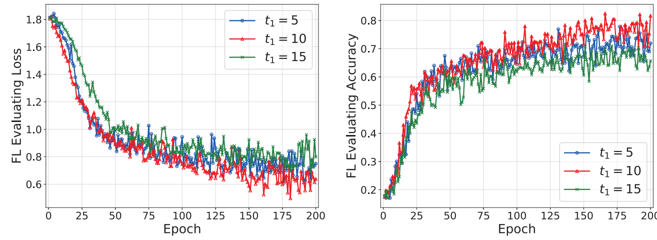
$$J(\boldsymbol{\theta}) = \frac{1}{K_b} \sum_{k \in \mathcal{K}_b} Q_{\boldsymbol{\phi}}(s_k, a_k). \tag{13}$$

Here, the parameter $\boldsymbol{\theta}$ of the actor can be updated via a policy gradient method. Then, the soft update strategy is adopted for the parameter update of target networks to realize stable convergence.

IV. SIMULATION RESULTS

A. Simulation Setting

We consider a distributed IIoT scenario, in which three smart factories are operated in different cities, and 30 IIoT devices are deployed in each factory that is covered by a BS. We deploy an edge server on the BS and a cloud server on the cloud. The transmission power of an IIoT device is set to be 30 dBm, and the noise power is set to be -104 dBm [13]. The amounts of uplink and downlink spectrum resources are 20 MHz and 40 MHz, respectively, unless specified. We adopt Raspberry Pi 4B [14] as an IIoT device. The on-board



(a) Loss (b) Accuracy

Fig. 2. FL evaluating loss and accuracy with respect to different cloud aggregation frequencies.

computing capability is set to be 1.5 GC/s (GC = 10^9 cycles), the effective capacitance is set to be 2×10^{-28}, and the number of CPU cycles to execute one training sample is set to be 20 cycles/bit [9]. The maximum FL epoch delay tolerance is set to be 5 s, and the energy capacity of each IIoT device follows a uniform distribution within [800, 1500] Joule.

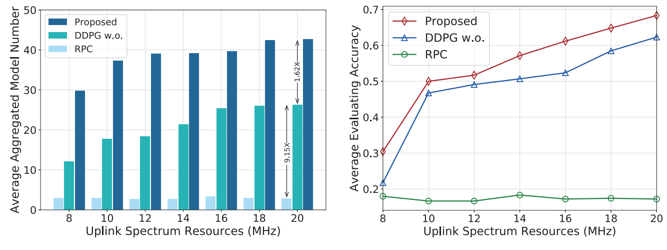
For the FL DNN models, we design a LeNet [15] with two 5×5 convolutional layers, two fully-connected layers, and a softmax output layer (61,366 total trainable model parameters), to complete a classification task. Hence, the size of model parameters is 240 KB. The LeNet is trained on the AWE dataset, which is a vibration-based industrial sensor dataset contains six sample classes of health conditions of industrial facilities [2]. The sample volume of the AWE dataset is 23,436, and each sample contains 1,024 sampling points. The ratio of the training set and evaluating set is 9:1. The FL is operated in an non-IID case, in which each IIoT device only holds a few classes of samples. Specifically, we first sort the samples by the class label, partition them into 180 blocks with the same size, and assign 2 blocks to each IIoT device [4]. The learning rate and batch size of the LeNet are set to be 10^{-3} and 64, respectively. The LeNet model is trained to perform health condition identification of industrial facilities, and thus the loss function of the LeNet is defined as $f(\boldsymbol{w}) = \sum_d y_d \log \hat{y_d}$ [6], where y_d is the sample label and $\hat{y_d}$ is the predicted label.

For the DDPG-based resource management algorithm, we adopt two-layer fully-connected neural networks as the actor and critic of the learning agent, respectively. The constant H in the reward function is set to be 10. Other parameters are listed in Table I. To show the efficiency of the proposed scheme, the following benchmarks are used for performance comparison.

- The DDPG without edge aggregation (DDPG w.o.): This scheme is designed for a two-layer FL aggregation architecture, in which the FL model parameters are only aggregated by the cloud server.
- Random probabilistic configuration (RPC): In this scheme, the central controller randomly makes the DSRA decisions. All available actions are selected with the same probability.

B. Performance Evaluation

First, we evaluate the convergence performance of the three-layer collaborative FL. Here, the cloud aggregation is performed every t_1 FL epochs. The evaluating loss and accuracy



(a) (b)

Fig. 3. Average aggregated model number and evaluating accuracy of different algorithms with respect to uplink spectrum resources.

are the metrics for the evaluation of FL performance. As shown in Fig. 2, in the considered three-layer collaborative architecture, the LeNet model trained on the AWE dataset can achieve a low evaluating loss and a high accuracy within 200 FL epochs. In particular, lower evaluating loss and higher accuracy can be reached if the cloud aggregation frequency is set properly. It can be seen that the lowest loss and the highest accuracy are achieved when the cloud aggregation frequency t_1 is set to be 10. In addition, due to the non-IID data distribution and the limited energy capacities of IIoT devices, the FL may not benefit from a large cloud aggregation frequency. This is because some IIoT devices may not be able to participate in the subsequent FL epochs, such that the model parameters trained on these devices cannot be globally aggregated, which may slow down FL convergence.

To demonstrate the effectiveness of the resource management scheme for FL, we compare the proposed algorithm with the DDPG w.o. and the RPC algorithms in terms of average aggregated model number and evaluating accuracy. As shown in Fig. 3(a), more model parameters of IIoT devices can be aggregated (i.e., edge or cloud aggregation) with more available uplink spectrum resources. Specifically, when the amount of uplink spectrum resource is 20 MHz, the proposed algorithm achieves $1.62\times$ and $13.85\times$ average aggregated model number improvement compared with the DDPG w.o. and RPC benchmarks. As shown in Fig. 3(b), with the increasing of the available uplink spectrum resources, the average evaluating accuracy achieved by the proposed algorithm increases. Particularly, the proposed algorithm can achieve a higher average evaluating accuracy than the benchmarks. When the amount of uplink spectrum resource is 20 MHz, the proposed algorithm can increase the average evaluating accuracy by 9.7%, as compared to the DDPG w.o. algorithm. The reason is that two-layer (i.e., device-cloud) FL aggregation leads to higher parameter transmission delay between the device and the cloud layers, which prevents the cloud server from aggregating more parameters within strict epoch delay requirements, and thus against FL performance improvement. The above observations indicate that the proposed resource management scheme greatly improves the FL performance over a three-layer aggregation architecture.

V. Conclusion

This paper has studied the resource management problem for FL in the distributed and resource-limited IIoT networks. A DDPG-based learning scheme has been proposed to make the optimal device selection and resource allocation decisions, thereby improving the FL performance while satisfying the strict epoch delay and the long-term energy consumption requirements of devices. The proposed scheme can not only select appropriate IIoT devices to participate in the FL, but also judiciously allocate spectrum and computing resources to support model parameter transmission and local training with preserved data privacy. For the future work, we will investigate a decentralized learning-based resource management scheme for FL in large-scale IIoT networks.

Acknowledgment

This work was supported in part by the National Key R&D Program of China (2018YFB1702000), in part by the National Natural Science Foundation of China (61771040), and in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

[1] “Worldwide global datasphere IoT device and data forecast, 2020-2024,” [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=US45066919, Accessed on: June 18, 2019.

[2] W. Zhang, D. Yang, Y. Xu, X. Huang, J. Zhang, and M. Gidlund, “DeepHealth: A self-attention based method for instant intelligent predictive maintenance in industrial Internet of things,” *IEEE Trans. Ind. Informat.*, Early Access, 2020, doi: 10.1109/TII.2020.3029551.

[3] W. Lim, N. Luong, D. Hoang, Y. Jiao, Y. Liang, Q. Yang, D. Niyato, and C. Miao, “Federated learning in mobile edge networks: A comprehensive survey,” *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, Apr. 2020.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. PMLR AIS,* Florida, USA, pp. 1273–1282, 2017.

[5] K. Yang, T. Jiang, Y. Shi, and Z. Ding, “Federated learning via over-the-air computation,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[6] H. Wang, Z. Kaplan, D. Niu, and B. Li, “Optimizing federated learning on non-IID data with reinforcement learning,” in *Proc. IEEE INFOCOM,* Beijing, China, pp. 1698–1707, 2020.

[7] M. Chen, H. Poor, W. Saad, and S. Cui, “Convergence time minimization of federated learning over wireless networks,” in *Proc. IEEE ICC,* Dublin, Ireland, 2020.

[8] Q. Zeng, Y. Du, K. Huang, and K. Leung, “Energy-efficient radio resource allocation for federated edge learning,” in *Proc. IEEE ICC,* Dublin, Ireland, 2020.

[9] L. Liu, J. Zhang, S. Song, and K. Letaief, “Client-edge-cloud hierarchical federated learning,” in *Proc. IEEE ICC,* Dublin, Ireland, 2020.

[10] W. Wu, N. Cheng, N. Zhang, P. Yang, W. Zhuang, and X. Shen, “Fast mmWave beam alignment via correlated bandit learning,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 12, pp. 5894–5908, Dec. 2019.

[11] X. Shen, J. Gao, W. Wu, K. Lyu, M. Li, W. Zhuang, X. Li, and J. Rao, “AI-assisted network-slicing based next-generation wireless networks,” *IEEE Open J. Veh. Technol.*, vol. 1, no. 1, pp. 45–66, Jan. 2020.

[12] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” in *Proc. ICLR,* San Juan, Puerto Rico, 2016.

[13] H. Peng and X. Shen, “Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks,” *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2416–2428, Mar. 2020.

[14] Raspberry Pi 4 Model B. [Online]. Available: https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf, Accessed on: Jun. 21, 2019.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.