Optimizing Tradeoff Between Learning Speed and Cost for Federated-Learning-Enabled Industrial IoT

Long Zhang, *Member, IEEE*, Suiyuan Wu, Haitao Xu, *Member, IEEE*, Qilie Liu, Choong Seon Hong, *Senior Member, IEEE*, and Zhu Han, *Fellow, IEEE*

***Abstract*—A combination of Industrial Internet of Things (IIoT) and federated learning (FL) is deemed as a promising solution to realize Industry 4.0 and beyond. However, scheduling more IIoT devices engaged in FL contributes to accelerated learning speed, but resulting in increased learning cost in terms of energy consumption and model accuracy reduction. In this article, we investigate the tradeoff between learning speed and cost in a three-layer FL-enabled IIoT system. Particularly, a weighted learning utility function is designed by capturing such a tradeoff. We aim to maximize the weighted learning utility in an FL training round by jointly optimizing the edge association as well as the allocations of resource block, computation capacity, and transmit power of the IIoT device. The resulting problem is a nonconvex and mixed-integer optimization problem, and consequently, it is difficult to solve. We thereby decompose the original problem into three subproblems, and then propose an overall alternating optimization algorithm to solve the subproblems iteratively until convergence. Via experimental results, it is demonstrated that the proposed scheme significantly improves the system-wide learning utility as compared to other baseline schemes. It is also shown that the proposed scheme can achieve the optimized tradeoff between learning speed and learning cost.**

***Index Terms*—Edge association, federated learning (FL), Industrial Internet of Things (IIoT), learning speed and cost tradeoff, resource allocation.**

Manuscript received 15 February 2023; revised 27 August 2023 and 16 October 2023; accepted 2 November 2023. Date of publication 7 November 2023; date of current version 26 March 2024. This work was supported in part by the Hebei Natural Science Foundation under Grant F2022402001 and Grant F2021402005; in part by the Open Fund of Chongqing Key Laboratory of Mobile Communications Technology under Grant cqupt-mct-202201; in part by the Central Fund Project for Guiding Local Science and Technology Development under Grant 236Z0401G; in part by NSF under Grant CNS-2107216, Grant CNS-2128368, Grant CMMI-2222810, and Grant ECCS-2302469; in part by the U.S. Department of Transportation; in part by Toyota; and in part by Amazon. *(Corresponding author: Long Zhang.)*

Long Zhang is with the School of Information and Electrical Engineering, Hebei University of Engineering, Handan 056038, China, and also with the Chongqing Key Laboratory of Mobile Communications Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: lzhang0310@gmail.com).

Suiyuan Wu is with the School of Information and Electrical Engineering, Hebei University of Engineering, Handan 056038, China (e-mail: suiyuan.wu@hotmail.com).

Haitao Xu is with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China (e-mail: alex_xuht@hotmail.com).

Qilie Liu is with the Chongqing Key Laboratory of Mobile Communications Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: liuql@cqupt.edu.cn).

Choong Seon Hong is with the Department of Computer Science and Engineering, Kyung Hee University, Yongin 17104, Gyeonggi, Republic of Korea (e-mail: cshong@khu.ac.kr).

Zhu Han is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 02447, South Korea (e-mail: hanzhu22@gmail.com).

Digital Object Identifier 10.1109/JIOT.2023.3330754

I. Introduction

THE INDUSTRIAL Internet of Things (IIoT) has emerged as a promising technology that drives greater efficiency and productivity in industries, thus attracting huge attention from academia, manufacturers, and plant designers [1], [2], [3]. A variety of new IIoT applications are developed in manufacturing and industrial processes, from digital factory and smart factory to Industry 4.0 and beyond. As an example of smart factory, a large number of IIoT devices are connected and synchronized to monitor, collect, and transmit industrial data on the field for meeting industrial expectations of security and reliability. Particularly, the industrial data is aggregated and transmitted to either the edge servers (ESs) for time-critical IIoT applications, or the central cloud server (CS) for further processing and analyzing [3], [4]. However, due to the communication resource constraints and the data privacy concerns, it may no longer be suitable for all IIoT devices to upload their industrial data to the CS for training data models. As a distributed learning paradigm, federated learning (FL) has been developed to allow multiple IIoT devices to cooperatively learn a global model from the locally trained model parameters without aggregating the raw data to the CS [5], [6], [7], [8], [9], [10]. This helps reduce communication overheads while alleviating the privacy concerns of industrial data sharing in IIoT systems.

A. Related Works and Motivation

Driven by the benefits of FL in industries, many potential issues in the realization of FL-enabled IIoT systems have been identified and discussed separately [3]. Among them, resource allocation is a critical issue that has gained significant interest in wireless scenarios. Yao and Ansari [11] proposed a joint CPU frequency and transmit power control problem to minimize the energy consumption of FL under the learning delay constraint. Yang et al. [12] studied the energy-efficient resource allocation problem, in which the time, bandwidth, power, computation frequency, and learning accuracy were iteratively optimized to minimize the total energy consumption of FL. An energy consumption minimization problem was formulated in [13] for an intelligent reflecting surface (IRS)-aided FL system, by jointly optimizing the communication

resources, IRS phase shifts, and local accuracy control. Feng et al. [14] applied the heterogeneous computing and wireless power transfer into an FL system, and formulated an energy consumption minimization problem to achieve the efficient FL. However, the above works have not considered the device scheduling in FL, which actually shows a direct impact on the convergence rate of the training process.

To further optimize the FL learning performance, recent progress has been made to explore the joint device selection and resource allocation problem. Chen et al. [9] proposed a joint user selection and wireless resource allocation framework to minimize the FL training loss over a cellular network. In [15], the joint device scheduling and resource allocation scheme was proposed to maximize the FL model accuracy within a given total training time budget. Chen et al. [16] presented the joint communication efficiency and resource optimization problem over an FL-enabled IoT network, and decoupled it into the client scheduling and resource allocation subproblems. Different from the above solutions [9], [15], [16] that considered only a single training round, Xu and Wang [17] formulated a stochastic optimization problem for joint client selection and bandwidth allocation under the long-term client energy constraint to improve the learning performance. Chen et al. [18] investigated the joint learning, wireless resource allocation, and user selection problem to minimize the FL convergence time. Zeng et al. [19] proposed an energy-efficient policy, which jointly considers the bandwidth allocation and device scheduling to reduce the sum device energy consumption while guaranteeing the learning speed. However, in the aforementioned works, only a two-layer FL framework has been considered to empower the wireless systems, which actually shows the limited performance gains in large-scale IIoT systems due to long transmission distance and higher communication overheads.

To tackle this challenge, some recent studies have focused on the three-layer FL framework for more performance gains. Liu et al. [20] designed a joint user association and wireless resource allocation scheme in a hierarchical FL to improve the learning performance. Zhang et al. [5] formulated the joint device selection and resource allocation problem in a three-layer FL-enabled IIoT network to minimize the training loss. And beyond that, the cost minimization problem of three-layer FL was investigated to tradeoff the delay and energy consumption in the wireless edge networks [21], [22] or to balance the delay, energy consumption, and model accuracy in the IIoT scenarios [23]. A network-aware FL optimization problem was formulated in [24], so as to strike the balance between the global loss and running time cost in a wireless fog–cloud system. However, the above works either focus on the learning performance enhancement or the cost-efficiency optimization, in terms of time and energy consumption, but the tradeoff between them has not been yet studied.

From the above review, although many researches have laid a solid foundation on joint device scheduling and resource allocation in the three-layer FL-supported wireless scenarios, the issue of optimization in the tradeoff between learning speed and learning cost in FL-enabled IIoT systems has not been addressed and remains an appealing study. It is generally expected to schedule as many devices as possible to engage in FL for speeding up the convergence, which is actually constrained by limited communication resources. This doing may result in more energy consumption and higher possibilities in learning loss, thus yielding the learning cost. Therefore, the learning speed and cost tradeoff design for achieving the joint optimization of IIoT device scheduling and resource allocation is a necessary and practical consideration, which also motivates us to start this work. Around this issue, we seek for maximizing the learning utility incorporating the learning speed and cost tradeoff in the FL-enabled IIoT systems by jointly optimizing edge association as well as communication and computing resource allocation, which forms the first research in this field, to the extent of our knowledge.

B. Contributions and Article Organization

The main contribution of this article is a novel joint optimization framework for achieving an optimized tradeoff between learning speed and learning cost in a three-layer FL-enabled IIoT system. Our specific contributions are shown as follows.

1) We identify the learning speed and cost tradeoff in the FL training process, and utilize the tradeoff to design a weighted learning utility function under one training round. The learning cost combines both the system-wide energy consumption of all scheduled IIoT devices and the effect of the total packet error rate on the FL model accuracy.
2) We formulate a joint edge association and resource allocation problem, aiming to maximize the weighted learning utility. In view of the challenges in solving this problem, we then decompose it into three subproblems: a) joint resource block (RB) assignment and edge association; b) computation capacity allocation; and c) transmit power allocation. A low-complexity overall iterative algorithm is proposed to solve the subproblems alternately.
3) We introduce a three-uniform weighted hypergraph to model the 3-D mapping relation for achieving the joint RB assignment and edge association. This subproblem is thus converted to find a 3-D hypergraph matching of a maximum-weight subset of vertex-disjoint hyperedges, which has been obtained by a local search (LS)-based iterative searching algorithm.
4) We compare the proposed joint optimization scheme with the baselines via numerical experiments. The results demonstrate that the proposed scheme has the enhanced learning performance in both the training loss and model accuracy, and can achieve the desired tradeoff between learning speed and learning cost.

The system model is described in Section II. In Section III, we formulate a joint optimization problem and decompose it into three subproblems. Section IV proposes a framework to alternately optimize the three decoupled subproblems. Experimental results are provided and discussed in Section V. Finally, we conclude in Section VI.

TABLE I
SUMMARY OF KEY NOTATIONS

Notation	Description	Notation	Description
Set and size		**Federated learning model**	
$\mathcal{N}$, $\mathcal{M}$, $\mathcal{K}$	IIoT devices, ESs, and RBs	$\mathcal{D}_n$, $\boldsymbol{\omega}_{n,i}^t$, $I_l(\theta)$	Local dataset, local model, and local iterations
$\mathcal{N}_m^{edge}$	IIoT devices associated with ES m	$\boldsymbol{\omega}^t$, $\boldsymbol{\omega}_{m,I_e(\epsilon,\theta)}^t$, $I_e(\epsilon,\theta)$	Global model, edge model, and edge iterations
$\mathcal{S}$	MN_e ESs and virtual ESs	$\mathcal{D}$, $\mathcal{D}_m$, η	Global dataset, edge dataset, and learning rate
N_e	Most number of IIoT devices served by ES	θ, ϵ	Local training accuracy, edge aggregation accuracy
Communication model		$F_n(\boldsymbol{\omega}_n)$, $F(\boldsymbol{\omega})$	Local loss function, global loss function
$h_{n,m}^k$, $R_{n,m}^k$	Channel gain and achievable rate	**Delay and energy consumption model**	
$g_{n,m}^k$, $\varpi_{n,m}^k$	Large-scale fading and shadowing standard deviation	T_n^{lt}, T_n^{ul}	Local computation delay and local uploading delay
$p_{n,m}^k$, I_k	Transmit power and total interference of cellular users	E_n^{lt}, E_n^{ul}	Local training and uploading energy consumption
B, σ_m^2	Bandwidth of RB and AWGN power at ES	T_n^{edge}, E_n	Delay and total energy consumption of IIoT device
$Q_{n,m}^{error}$, Q^{error}	Packet error rate of IIoT device and total packet error rate	E^{all}	Energy consumption for all involved IIoT devices
Problem formulation		f_n, C_n	CPU clock frequency and CPU cycles for 1-sample
a_n^k, $b_{n,m}$	RB assignment indicator and edge association indicator	ψ_n	Data size of local model and $\nabla F_n\left(\boldsymbol{\omega}_{n,i}^t\right)$
$P_n^{\max}$, ΔT	Maximum power and delay upper bound	**Hypergraph matching design**	
$f_n^{\min}$, $f_n^{\max}$	Lower and upper bound of computation capacity	$\mathcal{H}$, $\mathcal{V}$, $\mathcal{E}$	Hypergraph, vertex set, and hyperedge set
S^{fl}, C^{fl}	FL learning speed and learning cost	$\mathscr{G}$, V, E	Representative graph, vertex set, and edge set
U^{fl}, w	Weighted learning utility and weighting factor	n_ℓ, k_ℓ, s_ℓ	Element-involved combination of $\mathcal{N}$, $\mathcal{K}$, $\mathcal{S}$
Resource allocation optimization		$W(e_\ell)$, $W(v_\ell)$	Weight of hyperedge e_ℓ and weight of vertex v_ℓ
$\tilde{f}_n$, $\tilde{p}_{n,m}^k$	Lower bounds of computation capacity and transmit power	φ, $\mathscr{G}_\varphi$, $\mathcal{I}_{\mathscr{G}_\varphi}$	Talons, induced subgraph, and independent set
$f(p_n)$	Objective function of reformulated power allocation problem	$\mathcal{X}_{\mathscr{G}}$, $\mathcal{W}(\mathcal{X}_{\mathscr{G}})$	Initial independent set and sum-weight of vertices
$g\left(p_n, p_n^{[z-1]}\right)$	Convex approximation function of $f(p_n)$ at z-th iteration	$\mathcal{A}\left(\mathcal{I}_{\mathscr{G}_\varphi}, \mathcal{X}_{\mathscr{G}}\right)$	Set of adjacent vertices of independent set $\mathcal{I}_{\mathscr{G}_\varphi}$

II. SYSTEM MODEL

In this section, we first introduce the three-layer FL model, and then present the communication model for enabling the local model uploading, followed by the delay and energy consumption models. For convenience, some key notations used throughout the rest of this article are listed in Table I.

A. Federated Learning Model

Consider a hierarchical FL architecture over an IIoT system for the smart factory applications, as shown in Fig. 1, which consists of three layers, i.e., the end, edge, and cloud layers.

1) *End Layer:* A set $\mathcal{N} = \{1, 2, \ldots, N\}$ of IIoT devices are randomly distributed in a smart factory to collect real-time industrial data. The data may be used to train the model parameters locally at the IIoT devices for safety prewarning and fault diagnostic.
2) *Edge Layer:* A set of access points (APs), each with the limited coverage area, are uniformly attached on the room ceiling of the smart factory to provide the industrial services for IIoT devices via the wireless links. Through the dedicated wired link, each AP is connected to an ES, providing edge model parameter aggregation.[1]
3) *Cloud Layer:* A CS is deployed to provide global model aggregation for producing a new global model in each training round. Each AP is connected to the CS via the high-capacity fiber backhaul link.

Each IIoT device n is assumed to own a local data set $\mathcal{D}_n$ represented by a collection of input–output data samples $\{(\boldsymbol{x}_{nj}, y_{nj})\}_{j=1}^{|\mathcal{D}_n|}$, where $|\mathcal{D}_n|$ is the number of data samples collected by IIoT device n, $\boldsymbol{x}_{nj} \in \mathbb{R}^d$ is the jth input sample vector with $d > 1$ data features, and $y_{nj} \in \mathbb{R}$ is the ground-truth output label of $\boldsymbol{x}_{nj}$. The goal of IIoT device n is to find a local model parameter $\boldsymbol{\omega}_n$ that characterizes the output y_{nj} with a loss function $l_j(\boldsymbol{x}_{nj}, y_{nj}, \boldsymbol{\omega}_n)$ when the input is $\boldsymbol{x}_{nj}$. To capture the error of model parameter $\boldsymbol{\omega}_n$, the loss function on data set $\mathcal{D}_n$ of IIoT device n is defined as[2]

$$F_n\left(\boldsymbol{\omega}_n, \boldsymbol{x}_{n1}, y_{n1}, \ldots, \boldsymbol{x}_{n|\mathcal{D}_n|}, y_{n|\mathcal{D}_n|}\right) = \frac{1}{|\mathcal{D}_n|}\sum_{j=1}^{|\mathcal{D}_n|} l_j\left(\boldsymbol{x}_{nj}, y_{nj}, \boldsymbol{\omega}_n\right). \tag{1}$$

Define $F_n(\boldsymbol{\omega}_n) \triangleq F_n(\boldsymbol{\omega}_n, \boldsymbol{x}_{n1}, y_{n1}, \ldots, \boldsymbol{x}_{n|\mathcal{D}_n|}, y_{n|\mathcal{D}_n|})$ for notational simplicity. We denote the data set of all IIoT devices as $\mathcal{D} = \bigcup_{n=1}^N \mathcal{D}_n$. The training target of the FL algorithm is to find an optimal global model $\boldsymbol{\omega}^*$ that minimizes the global loss function $F(\boldsymbol{\omega})$ on data set $\mathcal{D}$ through the following optimization problem:

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}\in\mathbb{R}^d} F(\boldsymbol{\omega}) := \frac{\sum_{n=1}^N |\mathcal{D}_n| F_n(\boldsymbol{\omega}_n)}{|\mathcal{D}|} \tag{2a}$$

$$\text{s.t.} \quad \boldsymbol{\omega}_1 = \boldsymbol{\omega}_2 = \cdots = \boldsymbol{\omega}_N = \boldsymbol{\omega}. \tag{2b}$$

The optimal global model can be obtained by solving (2), and the training is conducted in a distributed manner through an iterative update process at the CS, the ESs, and the IIoT devices engaging in FL. Generally, in round t (i.e., global round t) of FL, the main training process is detailed as follows.

1) Global Model Downloading: At the beginning of round t, the CS broadcasts an updated global model $\boldsymbol{\omega}^t$ in (2a) to a set of $\mathcal{M} = \{1, 2, \ldots, M\}$ ESs. Afterward, ES m broadcasts $\boldsymbol{\omega}^t$ to all the associated IIoT devices engaged in FL, denoted by a set $\mathcal{N}_m^{\text{edge}}$ $\forall m \in \mathcal{M}$ and $\mathcal{N}_m^{\text{edge}} \subset \mathcal{N}$.

2) Local Training: After receiving global model $\boldsymbol{\omega}^t$, IIoT device n sets $\boldsymbol{\omega}_{n,0}^t := \boldsymbol{\omega}^t$ as the initial model parameter $\forall n \in \mathcal{N}_m^{\text{edge}}$. At each local iteration i, IIoT device n trains the local model $\boldsymbol{\omega}_{n,i}^t$ on local data set $\mathcal{D}_n$ via the stochastic gradient descent (SGD) by

$$\boldsymbol{\omega}_{n,i+1}^t = \boldsymbol{\omega}_{n,i}^t - \eta\nabla F_n\left(\boldsymbol{\omega}_{n,i}^t\right), \; i = 0, 1, \ldots, I_l(\theta) - 1 \tag{3}$$

[1]Due to the dedicated wired connection between AP and its integrated ES, we ignore the effect of wired transmission on the latency or energy consumption in the FL training process.

[2]Note that the loss function of the local model may vary over different learning tasks. Taking the linear regression learning as an example, the loss function of the jth data sample at IIoT device n can be denoted by $l_j(\boldsymbol{x}_{nj}, y_{nj}, \boldsymbol{\omega}_n) = (1/2)\|y_{nj} - \boldsymbol{x}_{nj}^T\boldsymbol{\omega}_n\|^2$ [25].

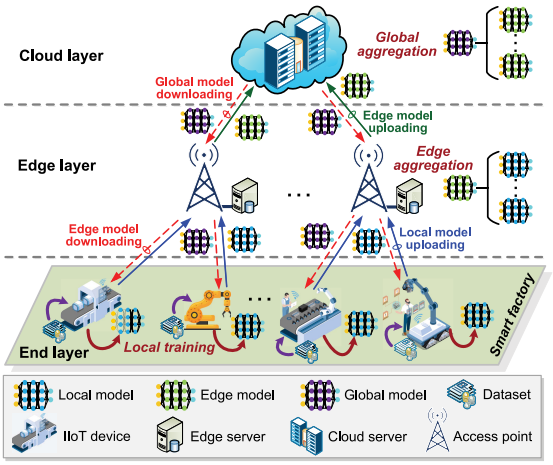


Fig. 1. Architecture of a three-layer FL-enabled IIoT system.

until $\|\nabla F_n(\boldsymbol{\omega}^t_{n,i+1})\| \leq \theta \|\nabla F_n(\boldsymbol{\omega}^t_{n,i})\|$. Here, η is a predefined learning rate, which is often decreased over time. Note that the number of local iterations for local model updating can be derived as $I_l(\theta) = \kappa \log(1/\theta)$, where κ is a constant associated with the data size and the local learning models [26], and $\theta \in (0, 1)$ is a local training accuracy that is equal to each IIoT device for a same model.

3) Local Model Uploading: IIoT device n uploads its locally updated model $\boldsymbol{\omega}^t_{n,I_l(\theta)}$ to the associated ES m in a decentralized way $\forall m \in \mathcal{M}$.

4) Edge Aggregation: ES m synchronously aggregates the model parameters $\boldsymbol{\omega}^t_{n,I_l(\theta)}$ from the set $\mathcal{N}^{\text{edge}}_m$ of IIoT devices, and generates the updated edge model as

$$\boldsymbol{\omega}^t_m = \frac{1}{|\mathcal{D}_m|} \sum_{n \in \mathcal{N}^{\text{edge}}_m} |\mathcal{D}_n| \boldsymbol{\omega}^t_{n,I_l(\theta)} \tag{4}$$

where $\mathcal{D}_m = \bigcup_{n \in \mathcal{N}^{\text{edge}}_m} \mathcal{D}_n$ is the data set of ES m. ES m then sends the updated edge model $\boldsymbol{\omega}^t_m$ back to the set $\mathcal{N}^{\text{edge}}_m$ of IIoT devices for local training. After that, ES m performs edge aggregation for $I_e(\epsilon, \theta)$ edge iterations until reaching an edge accuracy $\epsilon \in (0, 1)$,[3] such that [27]

$$I_e(\epsilon, \theta) = \frac{\mathcal{O}\left(\log\left(\frac{1}{\epsilon}\right)\right)}{1 - \theta}. \tag{5}$$

5) Global Aggregation: This stage consists of two steps: 1) edge model uploading and 2) global model aggregation. First, ES m forwards the updated edge model $\boldsymbol{\omega}^t_{m,I_e(\epsilon,\theta)}$ to the CS for averaging. Then, the CS receives the distributed model parameters from all ESs, and then aggregates them as follows:

$$\boldsymbol{\omega}^{t+1} = \frac{1}{|\mathcal{D}|} \sum_{m \in \mathcal{M}} |\mathcal{D}_m| \boldsymbol{\omega}^t_{m,I_e(\epsilon,\theta)}. \tag{6}$$

Then, the updated global model $\boldsymbol{\omega}^{t+1}$ is generated for round $t+1$ of local training. Note that during the FL training process, the global model parameter $\boldsymbol{\omega}$ in (6) can be iteratively updated for each round until convergence.

To facilitate analysis, we also make some assumptions on global loss function $F(\boldsymbol{\omega})$ as follows [9], [28], [29].

Assumption 1: The gradient $\nabla F(\boldsymbol{\omega})$ of $F(\boldsymbol{\omega})$ is uniformly Lipschitz continuous with respect to $\boldsymbol{\omega}$, so we have

$$\|\nabla F(\boldsymbol{\omega}_{t+1}) - F(\boldsymbol{\omega}_t)\| \leq \nu \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\| \tag{7}$$

where $\nu > 0$ is a predefined positive constant, and $\|\cdot\|$ is the Euclidean norm.

Assumption 2: For a given positive constant $\chi > 0$, $F(\boldsymbol{\omega})$ is strongly convex, satisfying the following inequality:

$$F(\boldsymbol{\omega}_{t+1}) \geq F(\boldsymbol{\omega}_t) + (\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t)^T \nabla F(\boldsymbol{\omega}_t) + \frac{\chi}{2} \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\|^2. \tag{8}$$

Assumption 3: $F(\boldsymbol{\omega})$ is twice-continuously differentiable. Making use of (7) and (8), we thus determine that

$$\chi \mathbf{I} \preceq \nabla^2 F(\boldsymbol{\omega}) \preceq \nu \mathbf{I} \tag{9}$$

where $\mathbf{I}$ is the identity matrix.

Assumption 4: For two given nonnegative constants $\xi_1, \xi_2 \geq 0$, we obtain $\|\nabla l_j(\boldsymbol{x}_{nj}, y_{nj}, \boldsymbol{\omega}_t)\|^2 \leq \xi_1 + \xi_2 \|\nabla F(\boldsymbol{\omega}_t)\|^2$.

B. Communication Model

We focus on the uplink frequency-division multiple access (FDMA) transmission, where each participating IIoT device communicates with the AP for local model uploading through an orthogonal RB.[4] The overall available bandwidth is divided to a set $\mathcal{K} = \{1, 2, \ldots, K\}$ of RBs, each with an equally sized bandwidth of B. We define a RB assignment indicator a^k_n for IIoT device n and RB k, such that if RB k is assigned to IIoT device n then $a^k_n = 1$; otherwise, $a^k_n = 0$. To show the association relationship between participating IIoT device n and ES m, we denote by $b_{n,m}$ an edge association indicator, where $b_{n,m} = 1$ indicates that IIoT device n is associated with ES m, and $b_{n,m} = 0$ otherwise.[5]

The uplink transmission is assumed to follow the quasi-static Rayleigh block fading, where the channel gain between IIoT device and ES remains to be constant during the local model uploading. To better reflect the noisy factory environments, we adopt the Rayleigh fading with the shadowing standard deviation [30]. Therefore, the channel gain between IIoT device n and ES m on RB k is written by $h^k_{n,m} = g^k_{n,m} \varpi^k_{n,m}$, where $g^k_{n,m}$ is the frequency and distance-dependent large-scale fading effect that consists of path loss and shadowing [31], and $\varpi^k_{n,m}$ is the shadowing standard deviation.

[3]Here, edge accuracy ϵ is treated as the equal value to all ESs. Note that there will always be on the order of $\log(1/\epsilon)$ outer iterations required in the determination of edge iterations for achieving edge accuracy of ϵ [27].

[4]Note that, similar to [32], we do not consider the downlink transmission, since the time overhead caused by the edge model aggregation and the global/edge model parameter distribution at the ES can be negligible compared to the uplink one. The reason is that the downlink typically has a larger bandwidth than the uplink, and, more particularly, the AP has much higher transmit power and processing capability than the IIoT device.

[5]Note that IIoT device n is selected to participate in the FL, if and only if IIoT device n is assigned to RB k, i.e., $a^k_n = 1$, and at the same time, IIoT device n is associated with ES m, i.e., $b_{n,m} = 1$.

Hence, the achievable rate of IIoT device n to upload the locally updated model to ES m on RB k can be given as

$$R_{n,m}^{k} = \sum_{m=1}^{M}\sum_{k=1}^{K} a_n^k b_{n,m} B \log_2\left(1 + \frac{p_{n,m}^k h_{n,m}^k}{I_k + \sigma_m^2}\right) \tag{10}$$

where $p_{n,m}^k$ is the transmit power of IIoT device n to ES m on RB k, I_k is the total interference caused by the cellular users that are located in other service areas on RB k, and σ_m^2 is the additive white Gaussian noise (AWGN) power at ES m.

During the FL training process, the ES only uses the available local model parameters without data errors in the model packets for edge aggregation. Since the iterative exchange of the local model between IIoT device and ES is performed over the wireless transmission, the random channel variations will cause the packet errors, which have showed an adverse effect on the global model accuracy [9], [28]. Therefore, we incorporate the transmission errors into the exchange of local model for edge aggregation. As analyzed in [33], the packet error rate of IIoT device n sending local model $\boldsymbol{\omega}_{n,I_l(\theta)}^t$ to ES m in each FL training round can be determined by

$$Q_{n,m}^{\text{error}} = \sum_{m=1}^{M}\sum_{k=1}^{K} a_n^k b_{n,m}\left(1 - \exp\left(\frac{-\lambda\left(I_k + \sigma_m^2\right)}{p_{n,m}^k h_{n,m}^k}\right)\right) \tag{11}$$

where λ is a suitable waterfall threshold [33].

As a consequence of Assumptions 1−4 that the global loss function $F(\boldsymbol{\omega})$ obeys, the effect of total packet error rate on the FL model accuracy can be written as

$$Q^{\text{error}} = \sum_{n=1}^{N} |\mathcal{D}_n| Q_{n,m}^{\text{error}}. \tag{12}$$

C. Delay and Energy Consumption Model

1) Local Training: We denote by f_n the CPU clock frequency to describe the computation capacity of IIoT device n. Let C_n be the number of CPU cycles required for executing one data sample at IIoT device n. Therefore, the computation delay for local training at IIoT device n over $I_l(\theta)$ local iterations in a round can be given by

$$T_n^{lt} = \frac{I_l(\theta) C_n |\mathcal{D}_n|}{f_n}. \tag{13}$$

The energy consumption of local training at IIoT device n over $I_l(\theta)$ local iterations can be accordingly derived as

$$E_n^{lt} = I_l(\theta) C_n |\mathcal{D}_n| \varrho f_n^2 \tag{14}$$

where ϱ is an effective switched capacitance associated with the CPU chip architecture of the IIoT device.

2) Local Model Uploading: Since the dimension of updated local model $\boldsymbol{\omega}_{n,I_l(\theta)}^t$ for $I_l(\theta)$ local iterations and $\nabla F_n(\boldsymbol{\omega}_{n,i}^t)$ under each FL round is fixed, we denote their data size (in bits) for IIoT device n by ψ_n, which is assumed to be constant throughout the training process. As a result, the delay caused by IIoT device n to upload the locally updated model to ES m on RB k at an edge iteration can be expressed as

$$T_n^{ul} = \frac{\psi_n}{R_{n,m}^k}. \tag{15}$$

Thereby, the energy consumption of IIoT device n for local model uploading at an edge iteration can be obtained by

$$E_n^{ul} = p_{n,m}^k T_n^{ul} = \frac{p_{n,m}^k \psi_n}{R_{n,m}^k}. \tag{16}$$

3) Edge and Global Aggregation: Since the downlink transmission always has a larger bandwidth than the uplink one, we ignore the delay of distributing the edge model parameters from ES to IIoT device. The delay and energy consumption of global model downloading from the CS to the AP will be also not considered in this work due to the high-capacity fiber backhaul link between them. In addition, the time and energy overhead of edge aggregation to update the model parameters for a few edge iterations can be further ignored owing to the superior processing capability of the ES.

Therefore, after $I_e(\epsilon, \theta)$ edge iterations for achieving edge accuracy of ϵ, the total energy consumption of IIoT device n in a round can be interpreted as follows:

$$\begin{aligned} E_n &= \sum_{m=1}^{M}\sum_{k=1}^{K} I_e(\epsilon,\theta) a_n^k b_{n,m}\left(E_n^{lt} + E_n^{ul}\right) \\ &= \sum_{m=1}^{M}\sum_{k=1}^{K} I_e(\epsilon,\theta) a_n^k b_{n,m}\left(I_l(\theta) C_n |\mathcal{D}_n| \varrho f_n^2 + \frac{p_{n,m}^k \psi_n}{R_{n,m}^k}\right). \end{aligned} \tag{17}$$

Accordingly, the *system-wide* energy consumption for all the participating IIoT devices in a round can be given by

$$E^{\text{all}} = \sum_{n=1}^{N} E_n. \tag{18}$$

Taking the computation delay and the uploading delay into account, the *edge-level* delay of IIoT device n under an edge iteration of each FL training round can be written as

$$\begin{aligned} T_n^{\text{edge}} &= \sum_{m=1}^{M}\sum_{k=1}^{K} a_n^k b_{n,m}\left(T_n^{lt} + T_n^{ul}\right) \\ &= \sum_{m=1}^{M}\sum_{k=1}^{K} a_n^k b_{n,m}\left(\frac{I_l(\theta) C_n |\mathcal{D}_n|}{f_n} + \frac{\psi_n}{R_{n,m}^k}\right). \end{aligned} \tag{19}$$

III. Problem Formulation and Decomposition

In this section, we design the weighted learning utility function via identifying the learning speed and cost tradeoff and use the utility function to formulate a joint edge association and resource allocation problem by taking into account the necessary constraints. Due to the challenges to solve the problem efficiently, we decompose it into three subproblems and elaborate them separately.

A. System Constraints

1) Edge Association Constraints: We consider that each IIoT device can associate at most one ES to participate the FL, and each ES can serve at most N_e IIoT devices due to

the hardware and cost limitation, for $N_e \geq \max_{m\in\mathcal{M}}\{|\mathcal{N}_m^{\text{edge}}|\}$. Thereby, we give the following edge association constraints:

$$\sum_{m=1}^{M} b_{n,m} \leq 1 \ \forall n \in \mathcal{N} \tag{20}$$

$$\sum_{n=1}^{N} b_{n,m} \leq N_e \ \forall m \in \mathcal{M}. \tag{21}$$

2) Communication Resource Constraints: Note that each RB is exclusively assigned to only one IIoT device, and each IIoT device is allocated to at most one RB. Accordingly, the RB assignment constraints can be, respectively, specified by

$$\sum_{n=1}^{N} a_n^k \leq 1 \ \forall k \in \mathcal{K} \tag{22}$$

$$\sum_{k=1}^{K} a_n^k \leq 1 \ \forall n \in \mathcal{N}. \tag{23}$$

We note that transmit power $p_{n,m}^k$ of IIoT device n to ES m on RB k should be adjusted in a continuous way but must be also limited by the maximum power $P_n^{\max}$. Then, the power constraint for IIoT device n is obtained by

$$0 \leq p_{n,m}^k \leq P_n^{\max} \ \forall n \in \mathcal{N}, m \in \mathcal{M}, k \in \mathcal{K}. \tag{24}$$

3) Computation Resource Constraint: To enable the efficient local training, the computation capacity for IIoT device n needs to be chosen within a given range of $[f_n^{\min}, f_n^{\max}]$. Thus, the computing resource constraint is formulated as

$$f_n^{\min} \leq f_n \leq f_n^{\max} \ \forall n \in \mathcal{N}. \tag{25}$$

4) Delay Constraint: We adopt a synchronous FL mechanism to aggregate the model parameters at the ES [34]. Obviously, the practical edge-level delay of the participating IIoT device under an edge iteration may be different throughout the training process, and thus, we set an upper bound $\Delta T > 0$ for each IIoT device to ensure the effectiveness of FL operation. The upper bound constraint of the edge-level delay for IIoT device n under an edge iteration can be given by

$$\sum_{m=1}^{M}\sum_{k=1}^{K} a_n^k b_{n,m}\left(\frac{I_l(\theta)C_n|\mathcal{D}_n|}{f_n} + \frac{\psi_n}{R_{n,m}^k}\right) \leq \Delta T \ \forall n \in \mathcal{N}. \tag{26}$$

B. Utility Function Design

To accelerate the convergence rate, the FL training process typically schedules as many IIoT devices as possible to participate in each round. However, uploading local models over wireless medium with limited communication resources becomes a performance bottleneck that allows more IIoT devices scheduled for engaging in FL. Thereby, to identify the learning performance, we employ the *learning speed* to characterize the total number of IIoT devices scheduled to participate in each round [16], [19]. Utilizing binary indicators a_n^k and $b_{n,m}$, the FL learning speed follows that:

$$S^{\text{fl}} = \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K} a_n^k b_{n,m}. \tag{27}$$

Although it is desirable to schedule more IIoT devices engaged in FL to achieve fast learning, more scheduled devices not only consume more system-wide energy but also impact the packet error rate on the model accuracy. Therefore, two categories of costs should be incorporated into the FL training process, which are the system-wide energy consumption and packet error rate costs for all scheduled IIoT devices, respectively. As such, the FL *learning cost* can be defined by

$$C^{\text{fl}} = E^{\text{all}} + \alpha \cdot Q^{\text{error}} \tag{28}$$

where α is the normalizing factor used to ensure the unitless combination of energy consumption and packet error rate. To improve the overall learning performance, both the learning speed and cost are critical to the participating IIoT devices, and particularly, the tradeoff between them needs to be well identified to increase the total learning utility for each FL training round. Combining (27) and (28), the weighted learning utility function in a round can be defined as

$$U^{\text{fl}} = w \cdot S^{\text{fl}} - (1-w) \cdot C^{\text{fl}} \tag{29}$$

where $w \in (0, 1]$ is a weighting factor that controls the tradeoff between learning speed and learning cost.

C. Problem Formulation

Define $\mathbf{a} = \{a_n^k \ \forall n, k\}$, $\mathbf{b} = \{b_{n,m} \ \forall n, m\}$, $\mathbf{p} = \{p_{n,m}^k \ \forall n, m, k\}$, and $\mathbf{f} = \{f_n \ \forall n\}$. Our objective is to investigate the tradeoff between the learning speed and cost from the perspective of scheduled IIoT devices by jointly optimizing edge association as well as communication and computation resource allocation. Specifically, we aim to maximize the weighted learning utility by jointly optimizing the edge association in $\mathbf{b}$, RB assignment in $\mathbf{a}$, transmit power allocation in $\mathbf{p}$, and computation capacity allocation in $\mathbf{f}$.[6] Therefore, the problem of our interest is formulated as

$$\max_{\mathbf{a},\mathbf{b},\mathbf{p},\mathbf{f}} \quad U^{\text{fl}}(\mathbf{a}, \mathbf{b}, \mathbf{p}, \mathbf{f}) \tag{30a}$$

$$\text{s.t.} \quad (20)-(26) \tag{30b}$$

$$a_n^k \in \{0, 1\} \ \forall n \in \mathcal{N}, k \in \mathcal{K} \tag{30c}$$

$$b_{n,m} \in \{0, 1\} \ \forall n \in \mathcal{N}, m \in \mathcal{M}. \tag{30d}$$

D. Problem Decomposition

We note that the challenges of solving problem (30) lie in the following reasons. First, the feasible set of problem (30)

[6]Note that in our proposed joint optimization framework, we consider the fixed number of data samples $|\mathcal{D}_n|$ collected by IIoT device for local training. In practice, the number of data samples affects not only the computation delay and energy consumption of local training but also the total packet error rate for the FL model accuracy. Thereby, the FL learning cost of interest is highly related with the amount of data at local training. To this end, the number of data samples at the IIoT device can be optimized to improve the FL learning performance as well. By letting $D_n \triangleq |\mathcal{D}_n|$ for notational simplicity, a bound constraint of $D_n^{\min} \leq D_n \leq D_n^{\max}$ w.r.t. the number of data samples can be incorporated into the formulated problem in (30) with the same objective function. To solve this problem via the alternating optimization framework as we will discuss in Section III-D, a subproblem can be designed to optimize the allocation of D_n by treating edge association $\mathbf{b}$, RB assignment $\mathbf{a}$, transmit power $\mathbf{p}$, and computation capacity $\mathbf{f}$ to be fixed. Note that the subproblem is shown as an integer programming problem, which can be efficiently solved by the branch-and-bound method [35].

is nonconvex since optimization variables a_n^k and $b_{n,m}$ are binary variables. Second, optimization variables a_n^k, $b_{n,m}$, $p_{n,m}^k$, and f_n are highly coupled in both the objective function and constraint in (26), which results in nonconvexity of (30). The optimal edge association and resource allocation problem itself is of high computational complexity, particularly when the number of IIoT devices and ESs becomes large. Therefore, problem (30) is a mixed-integer nonconvex optimization problem, and it is generally difficult to solve it efficiently with standard optimization methods.

To tackle the above challenges, we decompose the original problem in (30) into three subproblems: 1) joint RB assignment and edge association; 2) computation capacity allocation; and 3) transmit power allocation, which are specified as follows.

We first consider the joint RB assignment and edge association subproblem by fixing transmit power $\mathbf{p}$ and computation capacity $\mathbf{f}$. The first subproblem can be thus written by

$$\max_{\mathbf{a},\mathbf{b}} \quad U_1^{\mathrm{fl}}(\mathbf{a},\mathbf{b}) \tag{31a}$$

$$\text{s.t.} \quad (20)-(23),\ (26), \tag{31b}$$

$$a_n^k \in \{0,1\}\ \forall n \in \mathcal{N}, k \in \mathcal{K} \tag{31c}$$

$$b_{n,m} \in \{0,1\}\ \forall n \in \mathcal{N}, m \in \mathcal{M}. \tag{31d}$$

In the second subproblem of computation capacity allocation, the RB assignment $\mathbf{a}$, edge association $\mathbf{b}$, and transmit power $\mathbf{p}$ are considered to be fixed. Accordingly, the second subproblem can be formulated as

$$\max_{\mathbf{f}} \quad \hat{U}_2^{\mathrm{fl}}(\mathbf{f}) = -(1-w)\sum_{n=1}^{N} I_e(\epsilon,\theta) I_l(\theta) C_n |\mathcal{D}_n| \varrho f_n^2 \tag{32a}$$

$$\text{s.t.} \quad (26),\ f_n^{\min} \le f_n \le f_n^{\max}\ \forall n \in \mathcal{N}. \tag{32b}$$

We consider the third subproblem of transmit power allocation by setting RB assignment $\mathbf{a}$, edge association $\mathbf{b}$, and computation capacity $\mathbf{f}$ to be fixed. Then, we show the third subproblem as follows:

$$\max_{\mathbf{p}} \quad \hat{U}_3^{\mathrm{fl}}(\mathbf{p}) = \sum_{n=1}^{N}\left[-(1-w) I_e(\epsilon,\theta)\frac{p_{n,m}^k \psi_n}{B\log_2\left(1+\frac{p_{n,m}^k h_{n,m}^k}{I_k+\sigma_m^2}\right)}\right.$$

$$\left. -(1-w)\alpha|\mathcal{D}_n|\left(1-\exp\left(\frac{-\lambda\left(I_k+\sigma_m^2\right)}{p_{n,m}^k h_{n,m}^k}\right)\right)\right] \tag{33a}$$

$$\text{s.t.} \quad (26), 0 \le p_{n,m}^k \le P_n^{\max}\ \forall n \in \mathcal{N}, m \in \mathcal{M}, k \in \mathcal{K}. \tag{33b}$$

The three subproblems are alternatively optimized with multiple iterations, and in the $(r+1)$th iteration ($r = 0, 1, 2, \ldots,$), three steps are performed: 1) we first optimize $\mathbf{a}$ and $\mathbf{b}$ jointly in problem (31) with $\mathbf{p}_r$ and $\mathbf{f}_r$ fixed as the optimized values obtained in the rth iteration, and obtain the solution denoted by $\mathbf{a}_{r+1}^*$, $\mathbf{b}_{r+1}^*$; 2) we then optimize $\mathbf{f}_r$ in problem (32) with given $\mathbf{a}_{r+1}^*$, $\mathbf{b}_{r+1}^*$, $\mathbf{p}_r$, and obtain the solution denoted as $\mathbf{f}_{r+1}^*$; and 3) we optimize $\mathbf{p}_r$ in problem (33) by fixing $\mathbf{a}_{r+1}^*$, $\mathbf{b}_{r+1}^*$, $\mathbf{f}_{r+1}^*$, and obtain the solution denoted by $\{\mathbf{p}_{r+1}^*\}$. The iterations will continue until

$$U^{\mathrm{fl}}\left(\mathbf{a}_{r+1}^*, \mathbf{b}_{r+1}^*, \mathbf{p}_{r+1}^*, \mathbf{f}_{r+1}^*\right) - U^{\mathrm{fl}}\left(\mathbf{a}_r^*, \mathbf{b}_r^*, \mathbf{p}_r^*, \mathbf{f}_r^*\right) < \varepsilon \tag{34}$$

where $U^{\mathrm{fl}}(\mathbf{a}_r^*, \mathbf{b}_r^*, \mathbf{p}_r^*, \mathbf{f}_r^*)$ is the objective function value of problem (30) after the rth iteration, and $\varepsilon > 0$ is a tolerance threshold of convergence.

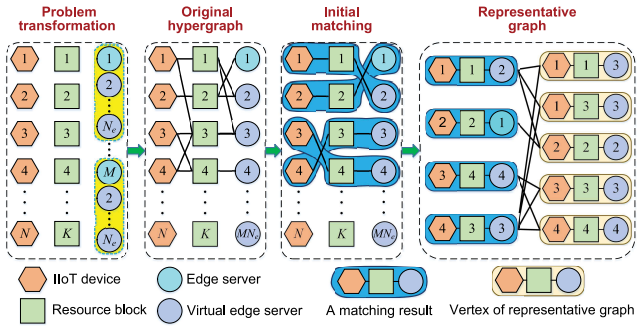


Fig. 2. Illustration of problem transformation, original hypergraph, initial matching, and representative graph.

IV. Proposed Solution

In this section, we will propose the solution to the above three subproblems, and then present the overall algorithm via alternately optimizing the subproblems in an iterative way.

A. Joint RB Assignment and Edge Association

Problem (31) is a 0-1 integer programming problem with a nonconvex objective function, aiming to find an assignment of either 0 or 1 to binary variables $\{\mathbf{a}, \mathbf{b}\}$, such that the weighted learning utility is maximized while satisfying binary constraints (31b)–(31d). Basically, problem (31) is NP-complete, meaning that no general convex optimization methods exist to solve it optimally. The computational cost for the traditional exhaustive search algorithms grows exponentially with the system scale, e.g., N and M, making it inefficient in practical FL system designs. To solve this problem efficiently, we recognize that the joint RB assignment and edge association can be regarded as a three-uniform weighted hypergraph matching process, by identifying the 3-D mapping relation among the IIoT devices, RBs, and ESs. Specifically, the IIoT devices, RBs, and ESs are set as the vertices of hypergraph, and the combinations by choosing an IIoT device, an RB, and an ES are denoted by the hyperedges of hypergraph. The weight of hyperedge can be defined as the learning utility of an IIoT device in a round. Therefore, the IIoT devices, RBs, and ESs act as three disjoint sets to be mutually matched with each other via the hypergraph matching algorithm to maximize the weighted learning utility, and thus the solution can be obtained.

1) Problem Transformation via Hypergraph: To identify the 3-D mapping relations among the IIoT devices, RBs, and ESs, we first develop some virtual elements for the ESs in the edge layer. Since each ES can serve at most N_e IIoT devices, $N_e - 1$ virtual ESs are devised for ES m. Taking ES M as an example, we create $N_e - 1$ virtual ESs, labeled as "2," "3," ..., and "N_e," as shown in Fig. 2. Such a virtualized design

ensures the edge association constraints in (20) and (21) of problem (31). Hence, there are $M \cdot N_e$ ESs along with the virtual ones in the edge layer, which can be used to associate with the participating IIoT devices.

Essentially, problem (31) can be regarded as the determination of the optimized 3-D mapping relation among the IIoT devices, RBs, and ESs, aiming to achieve the weighted learning utility maximization. For the converted 3-D mapping, the conventional ordinary graph model, generally used to describe pairwise relation between two objects, cannot exactly analyze the 3-D relation among three objects. However, the hypergraph model wherein every hyperedge can be defined as a subset of the vertex set is a natural fit for characterizing the 3-D mapping relation among multiple objects [36]. This enables us to transform problem (31) into a three-uniform weighted hypergraph model by leveraging hypergraph theory.

Definition 1: A hypergraph $\mathcal{H}$ is an ordered pair $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are disjoint finite sets such that $\mathcal{V} \neq \emptyset$, together with an incidence function $\Psi : \mathcal{E} \to 2^{\mathcal{V}}$. The elements of $\mathcal{V}$ are called vertices, and the elements of $\mathcal{E}$ are called hyperedges. In a weighted hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E}, W(e))$, each hyperedge $e \in \mathcal{E}$ is associated with a weight function $W(e) : \mathcal{E} \to \mathbb{R}^+$.

Denote $\mathcal{S} = \{1, 2, \ldots, \mathrm{MN}_e\}$ as the set of MN_e ESs and virtual ESs. Combining the sets of IIoT devices and RBs, an element in $\mathcal{N}$, $\mathcal{K}$, and $\mathcal{S}$ stands for a vertex in hypergraph $\mathcal{H}$. The union of $\mathcal{N}$, $\mathcal{K}$, and $\mathcal{S}$ refers to the nonempty finite set of vertices in hypergraph $\mathcal{H}$, such that $\mathcal{V} = \mathcal{N} \cup \mathcal{K} \cup \mathcal{S}$. With this in mind, we employ an element-involved combination $c_\ell \triangleq (n_\ell, k_\ell, s_\ell)$ to denote a hyperedge $e_\ell \in \mathcal{E}$ in hypergraph $\mathcal{H}$, while satisfying constraints (31b)–(31d) simultaneously, for $n_\ell \in \mathcal{N}$, $k_\ell \in \mathcal{K}$, $s_\ell \in \mathcal{S}$, and $\ell = 1, 2, \ldots, |\mathcal{E}|$. Therefore, we have hyperedge $e_\ell \triangleq \{n_\ell, k_\ell, s_\ell\}$. Since each hyperedge e_ℓ consists of only three vertices n_ℓ, k_ℓ, and s_ℓ, we thus refer to $\mathcal{H}$ as a three-uniform hypergraph. Note that such a combination c_ℓ obviously captures the 3-D mapping relation for the RB, IIoT device, and ES in the considered system.

Taking objective function (31a) into account, the weight of hyperedge $e_\ell \triangleq \{n_\ell, k_\ell, s_\ell\}$ can be defined as the learning utility of IIoT device n_ℓ in an FL round, i.e.,

$$W(e_\ell) = w a_{n_\ell}^{k_\ell} b_{n_\ell, s_\ell} - (1 - w)\left(E_{n_\ell} + \alpha |\mathcal{D}_{n_\ell}| Q_{n_\ell, s_\ell}^{\text{error}}\right) \ \forall n_\ell, k_\ell, s_\ell. \quad (35)$$

From (35), problem (31) is transformed into a three-uniform weighted hypergraph model. With original hypergraph $\mathcal{H}$, our target for the weighted learning utility maximization problem is to find a 3-D hypergraph matching that is a collection of N^* vertex-disjoint hyperedges with the maximum total weights.

2) Algorithm Design: For a 3-D hypergraph matching, to seek a maximum-weight subset of vertex-disjoint hyperedges is NP-hard. Inspired by the LS algorithm with an increasing approximation ratio by $[\beta/(\beta - 1)]$ factor (integer $\beta \geq 2$) in polynomial time [36], [37], we employ the LS to design an efficient iterative algorithm for obtaining the solution to (31).

Definition 2: A representative graph $\mathcal{G}$ is an ordinary graph $\mathcal{G} = (\mathsf{V}, \mathsf{E})$, where a vertex $v_\ell = (n_\ell, k_\ell, s_\ell) \in \mathsf{V}$ in $\mathcal{G}$ denotes a hyperedge $e_\ell = \{n_\ell, k_\ell, s_\ell\} \in \mathcal{E}$ of hypergraph $\mathcal{H}$, and an ordinary edge in E denotes the adjacent hyperedges in $\mathcal{H}$ if they intersect with each other at least one vertex of $\mathcal{H}$. The weight of vertex v_ℓ in $\mathcal{G}$ equals the weight of hyperedge e_ℓ of $\mathcal{H}$, i.e., $W(v_\ell) = W(e_\ell)$.

Algorithm 1 Greedy Algorithm

Input: Original weighted hypergraph $\mathcal{H}$;

Output: An initial independent set $\mathcal{X}_\mathcal{G}$ of representative graph $\mathcal{G}$;

1: Generate representative graph $\mathcal{G}$ with vertex set V according to original weighted hypergraph $\mathcal{H}$;
2: Set $\mathcal{X}_\mathcal{G} = \emptyset$;
3: **while** $\mathsf{V} \neq \emptyset$ **do**
4: Select a vertex $v^* = \arg\max_{v \in \mathsf{V}}\{W(v)\}$ in $\mathcal{G}$;
5: Set $\mathcal{A}_{v^*} = \emptyset$. Find all the adjacent vertices of v^*, and let $\mathcal{A}_{v^*}$ be the set of the adjacent vertices of v^*;
6: Set $\mathcal{X}_\mathcal{G} = \mathcal{X}_\mathcal{G} \cup \{v^*\}$ and $\mathsf{V} = \mathsf{V} - \{v^*\} \cup \mathcal{A}_{v^*}$;
7: **end while**

Based on Definition 2, vertex $(1, 1, 2)$ of representative graph $\mathcal{G}$ denotes hyperedge $\{1, 1, 2\}$ of hypergraph $\mathcal{H}$, and the edge between vertices $(1, 1, 2)$ and $(1, 3, 3)$ of representative graph $\mathcal{G}$ implies that hyperedges $\{1, 1, 2\}$ and $\{1, 3, 3\}$ intersect at vertex "1" of hypergraph $\mathcal{H}$, as shown in Fig. 2.

Definition 3: For the converted representative graph $\mathcal{G}$, a φ-claw is an induced subgraph $\mathcal{G}_\varphi$, whose center vertex connects to φ independent vertices, called the talons, which forms an independent set $\mathcal{I}_{\mathcal{G}_\varphi}$ with φ vertices.

Based on Definition 3, we find that there exist 1-claw and 2-claw for center vertex $(1, 1, 2)$ of representative graph $\mathcal{G}$, as shown in Fig. 2. However, we cannot search for a 4-claw for any center vertex of $\mathcal{G}$, indicating that $\mathcal{G}$ is 4-claw free for the 3-D hypergraph matching.

For brevity, we denote by $\mathcal{X}_\mathcal{G}$ an initial independent set of representative graph $\mathcal{G}$, and let $\mathcal{W}(\mathcal{X}_\mathcal{G})$ be the sum of weights for all the vertices in $\mathcal{X}_\mathcal{G}$. Note that an initial independent set refers to an initial matching result of the original hypergraph. To find an initial independent set $\mathcal{X}_\mathcal{G}$, we approximately derive a weight-maximum feasible subset of edge-disjoint vertices in $\mathcal{G}$ by using a greedy algorithm, as given in Algorithm 1.

To improve the overall performance for finding a 3-D hypergraph matching, we propose an LS-based algorithm to search for φ-claw repeatedly, for increasing the sum of weights of initial matching [36], [37]. Through the iterative searching, we thus obtain the weight-maximum 3-D hypergraph matching result, including N^* vertex-disjoint hyperedges. Denote $\mathcal{A}(\mathcal{I}_{\mathcal{G}_\varphi}, \mathcal{X}_\mathcal{G})$ as the set of adjacent vertices of independent set $\mathcal{I}_{\mathcal{G}_\varphi}$ with φ vertices in an initial independent set $\mathcal{X}_\mathcal{G}$. We then present Algorithm 2, which gives the procedure of the proposed algorithm. Based on Algorithm 2, we obtain a weight-maximum subset $\mathcal{E}^*(\mathbf{a}^*, \mathbf{b}^*)$ as the 3-D hypergraph matching result, which has N^* vertex-disjoint hyperedges. Thereby, the suboptimal solution to (31) is thus achieved, and the weighted learning utility can be given by

$$U_1^{\text{fl}}(\mathbf{a}^*, \mathbf{b}^*) = \mathcal{W}(\mathcal{X}_\mathcal{G}^*). \quad (36)$$

Algorithm 2 LS-Based 3-D Hypergraph Matching Algorithm for Solving Problem (31)

Input: Original weighted hypergraph $\mathcal{H}$;
Output: 3-D hypergraph matching $\mathcal{E}^*(\mathbf{a}^*, \mathbf{b}^*)$;
1: Transform $\mathcal{H}$ into a representative graph $\mathcal{G}$;
2: Obtain an initial independent set $\mathcal{X}_{\mathcal{G}}$ of $\mathcal{G}$ according to Algorithm 1;
3: Sort all the vertices within $\mathcal{X}_{\mathcal{G}}$ in an ascending order by the weight of vertex. Set $\zeta = 1$;
4: **repeat**
5: Find all the adjacent vertices of the ζ-th vertex $\mathcal{X}_{\mathcal{G}}[\zeta]$ from $\mathcal{X}_{\mathcal{G}}$. Let $\mathcal{A}_{\mathcal{X}_{\mathcal{G}}[\zeta]}$ be the set of adjacent vertices of $\mathcal{X}_{\mathcal{G}}[\zeta]$;
6: Sort all the vertices within $\mathcal{A}_{\mathcal{X}_{\mathcal{G}}[\zeta]}$ in a descending order by the weight of vertex;
7: Set $\mathcal{X}_{\mathcal{G}}[\zeta]$ as the center vertex of φ-claw $\mathcal{G}_{\varphi}$;
8: **for** $\varphi = 1$ to 3 **do**
9: Search for φ-claw $\mathcal{G}_{\varphi}$ from $\mathcal{A}_{\mathcal{X}_{\mathcal{G}}[\zeta]}$ in $\mathcal{G}$;
10: **if** there exists φ-claw $\mathcal{G}_{\varphi}$ satisfying $\mathcal{W}\left(\left(\mathcal{X}_{\mathcal{G}} - \mathcal{A}\left(\mathcal{I}_{\mathcal{G}_{\varphi}}, \mathcal{X}_{\mathcal{G}}\right)\right) \cup \mathcal{I}_{\mathcal{G}_{\varphi}}\right) > \mathcal{W}(\mathcal{X}_{\mathcal{G}})$ **then**
11: Update $\mathcal{X}_{\mathcal{G}} = \left(\mathcal{X}_{\mathcal{G}} - \mathcal{A}\left(\mathcal{I}_{\mathcal{G}_{\varphi}}, \mathcal{X}_{\mathcal{G}}\right)\right) \cup \mathcal{I}_{\mathcal{G}_{\varphi}}$;
12: **go to** step 3;
13: **end if**
14: **end for**
15: Update $\zeta \leftarrow \zeta + 1$;
16: **until** $\zeta \geq |\mathcal{X}_{\mathcal{G}}|$
17: **return** $\mathcal{E}^*(\mathbf{a}^*, \mathbf{b}^*) = \mathcal{X}^*_{\mathcal{G}} = \{e^*_1, e^*_2, \ldots, e^*_{N^*}\}$;

B. Computation Capacity Allocation

From (26) in problem (32), one can readily see that, with the upper bound constraint of the edge-level delay, the computation capacity of IIoT device accordingly has a lower bound constraint in closed form, which is given below.

Theorem 1: The computation capacity f_n of IIoT device n is lower bounded as

$$f_n \geq \tilde{f}_n = \frac{I_l(\theta) C_n |\mathcal{D}_n|}{\Delta T - \frac{\psi_n}{R^k_{n,m}}}. \tag{37}$$

Proof: With given a^k_n, $b_{n,m}$, and $p^k_{n,m}$, the term $\psi_n / R^k_{n,m}$ in (26) can be determined due to the fixed achievable rate $R^k_{n,m}$. Given the upper bound ΔT, each IIoT device needs to complete the local training within a certain delay constraint, i.e., $\Delta T - (\psi_n / R^k_{n,m})$. Therefore, with the whole computation task $I_l(\theta) C_n |\mathcal{D}_n|$, to complete the local training, IIoT device n needs the minimum computation capacity as $\tilde{f}_n$, which can be further derived by

$$\tilde{f}_n \triangleq \frac{I_l(\theta) C_n |\mathcal{D}_n|}{\Delta T - \frac{\psi_n}{R^k_{n,m}}} \tag{38}$$

which yields the final closed-form bound in (37). ∎

Based on Theorem 1, we can derive the feasible set of computation capacity f_n by substituting (37) into (25), i.e.,

$$\max\left\{f^{\min}_n, \tilde{f}_n\right\} \leq f_n \leq f^{\max}_n. \tag{39}$$

With the feasible set of f_n, the final result is summarized in the following theorem.

Theorem 2: The optimal solution to (32) is given as follows:

$$f^*_n = \max\left\{f^{\min}_n, \tilde{f}_n\right\}. \tag{40}$$

Proof: The first-order derivative of objective function (32a) with respect to f_n can be written as

$$\frac{\partial \hat{U}^{\text{fl}}_2(\mathbf{f})}{\partial f_n} = -2(1-w) \sum_{n=1}^{N} I_e(\epsilon, \theta) I_l(\theta) C_n |\mathcal{D}_n| \varrho f_n < 0. \tag{41}$$

Note that objective function (32a) is monotonically decreasing with respect to f_n. Therefore, the optimal solution to (32) can be easily obtained when taking f_n as the minimum value $\max\{f^{\min}_n, \tilde{f}_n\}$ of the feasible set in (39). Then, the result in (40) directly follows, which completes the proof. ∎

C. Transmit Power Allocation

To make problem (33) tractable, we will use (26) to derive the lower bound on transmit power in closed form.

Theorem 3: The transmit power $p^k_{n,m}$ of IIoT device n is lower bounded by

$$p^k_{n,m} \geq \tilde{p}^k_{n,m} = \frac{I_k + \sigma^2_m}{h^k_{n,m}} \left(2^{\frac{\psi_n}{B\left(\Delta T - \frac{I_l(\theta) C_n |\mathcal{D}_n|}{f_n}\right)}} - 1\right). \tag{42}$$

Proof: From (26), given a^k_n, $b_{n,m}$, and f_n, we can directly obtain the fixed computation delay of local training for IIoT device n, i.e., $([I_l(\theta) C_n |\mathcal{D}_n|]/f_n)$. Given the upper bound ΔT, IIoT device n needs to upload its local model to the ES within a specific maximum latency, i.e., $\Delta T - ([I_l(\theta) C_n |\mathcal{D}_n|]/f_n)$. Accordingly, the lower bound of the achievable rate of IIoT device n to complete the model uploading can be expressed as

$$R^k_{n,m} \geq \frac{\psi_n}{\Delta T - \frac{I_l(\theta) C_n |\mathcal{D}_n|}{f_n}}. \tag{43}$$

Then, we can combine (10) and (43) to obtain the desired closed-form lower bound $\tilde{p}^k_{n,m}$ as

$$p^k_{n,m} \geq \frac{I_k + \sigma^2_m}{h^k_{n,m}} \left(2^{\frac{\psi_n}{B\left(\Delta T - \frac{I_l(\theta) C_n |\mathcal{D}_n|}{f_n}\right)}} - 1\right) \triangleq \tilde{p}^k_{n,m}. \tag{44}$$

This concludes the proof. ∎

Substituting (42) into (33b), problem (33) can be reformulated as follows:

$$\min_{\mathbf{p}} \quad \sum_{n=1}^{N} \left[(1-w) I_e(\epsilon, \theta) \frac{p^k_{n,m} \psi_n}{B \log_2\left(1 + \frac{p^k_{n,m} h^k_{n,m}}{I_k + \sigma^2_m}\right)} + (1-w) \alpha |\mathcal{D}_n| \left(1 - \exp\left(\frac{-\lambda\left(I_k + \sigma^2_m\right)}{p^k_{n,m} h^k_{n,m}}\right)\right)\right] \tag{45a}$$

$$\text{s.t.} \quad \tilde{p}^k_{n,m} \leq p^k_{n,m} \leq P^{\max}_n. \ \forall n \in \mathcal{N}, m \in \mathcal{M}, k \in \mathcal{K}. \tag{45b}$$

Problem (45) is a nonconvex optimization problem with respect to $\mathbf{p}$ due to the nonconvexity of objective function (45a). In general, there is no feasible method to obtain the optimal solution for this problem. A widely used method for solving it is the successive convex approximation (SCA) [38]. By using the SCA method, the nonconvex objective function and constraints are approximated by the convex objective function and constraints, respectively. Then, the original problem is transformed into a convex approximation problem, which can be solved iteratively to find a suboptimal solution.

Since (45b) is convex, we can develop an approximation function of optimization variable $p_{n,m}^k$ for IIoT device n by replacing the nonconvex objective function with a suitable convex approximant. Therefore, we solve the convex approximation of the original problem (45) at each iteration via the SCA. Since the optimization variables of $\mathbf{b}$ and $\mathbf{a}$ are given in (45), $p_{n,m}^k$ of IIoT device n is hereinafter written as p_n for notational simplicity. We then use $f(p_n)$ to denote objective function (45a) with respect to IIoT device n, and we can have

$$f(p_n) = (1-w)I_e(\epsilon,\theta)\frac{p_{n,m}^k \psi_n}{B\log_2\left(1+\frac{p_{n,m}^k h_{n,m}^k}{I_k+\sigma_m^2}\right)} + (1-w)\alpha|\mathcal{D}_n|\left(1-\exp\left(\frac{-\lambda\left(I_k+\sigma_m^2\right)}{p_{n,m}^k h_{n,m}^k}\right)\right). \quad (46)$$

Without losing generality, we denote by $g(p_n, p_n^{[z-1]})$ the convex approximation function of $f(p_n)$ in the zth iteration of the SCA method. In order to identify the approximation function of $f(p_n)$, we first consider to derive the first-order derivative of an objective function $f(p_n)$ as expressed in (47), shown at the bottom of the page.

Lemma 1 (Descent Lemma [39]): Consider a continuously differentiable function $f:\mathbb{R}^n \to \mathbb{R}$ with a Lipschitz continuous gradient and Lipschitz constant L, then for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, one can obtain

$$f(\mathbf{x}) \le f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) + \frac{L}{2}\|\mathbf{x}-\mathbf{y}\|^2. \quad (48)$$

With the help of Lemma 1, we can resort to the second-order Taylor expansion to construct the convex approximation function $g(p_n, p_n^{[z-1]})$, which can be written by

$$g\left(p_n, p_n^{[z-1]}\right) = f\left(p_n^{[z-1]}\right) + f'\left(p_n^{[z-1]}\right)\left(p_n - p_n^{[z-1]}\right) + \frac{L'}{2}\left(p_n - p_n^{[z-1]}\right)^2 \quad (49)$$

where $L' > 0$ is a positive constant.

From the above analysis, we design an efficient algorithm via the SCA to obtain the solution of (45). We first initialize $p_n^{[0]}$ to be a feasible solution, and then update $p_n^{[z]}$ iteratively by optimizing the convex approximation of (45) with convex approximation function $g(p_n, p_n^{[z-1]})$ until convergence. The detailed SCA method for solving (45) is implemented by Algorithm 3. As the iterations continue, the learning utility in an FL round will gradually increase.

Algorithm 3 SCA Method for Solving Problem (45)

1: Initialize iteration number $z = 0$, maximum iteration number Z, and tolerance δ;
2: Initialize $p_n^{[0]} \in \mathbf{p}$ as a feasible value to problem (45);
3: **repeat**
4: Set $z \leftarrow z+1$;
5: Let $p_n^{[z]} = \arg\min_{p_n \in \mathbf{p}} g\left(p_n, p_n^{[z-1]}\right)$;
6: Set $d_n^{[z]} = p_n^{[z]} - p_n^{[z-1]}$;
7: *Armijo step-size rule*: Choose $\phi^{init} > 0$, $\mu \in (0,1)$ and $\varsigma \in (0, 0.5)$. Let $\phi^{[z]} = \max\{\phi^{init}, \mu^j\}_{j=0,1,2,\ldots}$, while satisfying $f\left(p_n^{[z-1]}\right) - f\left(p_n^{[z-1]} + \phi^{[z]} d_n^{[z]}\right) \ge -\varsigma\phi^{[z]} f'\left(p_n^{[z-1]}\right) d_n^{[z]}$;
8: Set $p_n^{[z]} = p_n^{[z-1]} + \phi^{[z]}\left(p_n^{[z]} - p_n^{[z-1]}\right)$;
9: **until** $\left|p_n^{[z]} - p_n^{[z-1]}\right| < \delta$ or $z > Z$
10: **return** p_n^* or $p_n^{[z]}$;

Algorithm 4 Overall Alternating Optimization Algorithm for Solving Problem (30)

1: Initialize $\mathbf{p}_0$ and $\mathbf{f}_0$. Set iteration index $r = 1$;
2: **repeat**
3: Solve problem (31) via Algorithm 2 for given $\mathbf{p}_r$, $\mathbf{f}_r$, and denote the solution as $\mathbf{a}_{r+1}^*$, $\mathbf{b}_{r+1}^*$;
4: Solve problem (32) according to (40) for given $\mathbf{a}_{r+1}^*$, $\mathbf{b}_{r+1}^*$, $\mathbf{p}_r$, and denote the solution as $\mathbf{f}_{r+1}^*$;
5: Solve problem (45) via Algorithm 3 for given $\mathbf{a}_{r+1}^*$, $\mathbf{b}_{r+1}^*$, $\mathbf{f}_{r+1}^*$, and denote the solution as $\mathbf{p}_{r+1}^*$;
6: Set $r \leftarrow r+1$;
7: **until** $U^{\mathrm{fl}}\left(\mathbf{a}_{r+1}^*, \mathbf{b}_{r+1}^*, \mathbf{p}_{r+1}^*, \mathbf{f}_{r+1}^*\right) - U^{\mathrm{fl}}\left(\mathbf{a}_r^*, \mathbf{b}_r^*, \mathbf{p}_r^*, \mathbf{f}_r^*\right) < \varepsilon$

D. Overall Algorithm, Convergence, and Complexity Analysis

The three subproblems can be solved efficiently, which enables an effective solution to problem (30). We propose an overall iterative algorithm for alternately optimizing the three subproblems, through which we obtain a suboptimal solution of (30). The details of the proposed overall alternating optimization algorithm are outlined in Algorithm 4. We next analyze the convergence of Algorithm 4 as follows.

Theorem 4: Algorithm 4 is guaranteed to converge.

$$\frac{\partial f(p_n)}{\partial p_n} = \frac{(1-w)I_e(\epsilon,\theta)\psi_n \log_2\left(1+\frac{p_n h_{n,m}^k}{I_k+\sigma_m^2}\right) - \frac{(1-w)I_e(\epsilon,\theta)p_n\psi_n h_{n,m}^k}{\left(I_k+\sigma_m^2\right)\left(1+\frac{p_n h_{n,m}^k}{I_k+\sigma_m^2}\right)\ln 2}}{B\left(\log_2\left(1+\frac{p_n h_{n,m}^k}{I_k+\sigma_m^2}\right)\right)^2} - \frac{(1-w)\alpha|\mathcal{D}_n|\lambda\left(I_k+\sigma_m^2\right)}{h_{n,m}^k p_n^2}\exp\left(\frac{-\lambda\left(I_k+\sigma_m^2\right)}{p_n h_{n,m}^k}\right) \quad (47)$$

Proof: Denote $U_1^{\mathrm{fl}}(\mathbf{a}_r, \mathbf{b}_r, \mathbf{p}_r, \mathbf{f}_r)$, $\hat{U}_2^{\mathrm{fl}}(\mathbf{a}_r, \mathbf{b}_r, \mathbf{p}_r, \mathbf{f}_r)$, and $\hat{U}_3^{\mathrm{fl}}(\mathbf{a}_r, \mathbf{b}_r, \mathbf{p}_r, \mathbf{f}_r)$ as the objective function values of (31)–(33) in the rth iteration, respectively. In the $(r+1)$th iteration, in line 3 of Algorithm 4, for given $\{\mathbf{p}_r, \mathbf{f}_r\}$, we can obtain

$$U^{\mathrm{fl}}(\mathbf{a}_r, \mathbf{b}_r, \mathbf{p}_r, \mathbf{f}_r) \overset{(i)}{\leq} U_1^{\mathrm{fl}}\left(\mathbf{a}_{r+1}^*, \mathbf{b}_{r+1}^*, \mathbf{p}_r, \mathbf{f}_r\right) = U^{\mathrm{fl}}\left(\mathbf{a}_{r+1}^*, \mathbf{b}_{r+1}^*, \mathbf{p}_r, \mathbf{f}_r\right) \quad (50)$$

where inequality (i) holds since $\mathbf{a}_{r+1}^*$, $\mathbf{b}_{r+1}^*$ is the obtained solution of (31). Next, in line 4 of Algorithm 4, for given $\mathbf{a}_{r+1}^*$, $\mathbf{b}_{r+1}^*$, $\mathbf{p}_r$, we can have

$$U^{\mathrm{fl}}\left(\mathbf{a}_{r+1}^*, \mathbf{b}_{r+1}^*, \mathbf{p}_r, \mathbf{f}_r\right) \overset{(ii)}{\leq} \hat{U}_2^{\mathrm{fl}}\left(\mathbf{a}_{r+1}^*, \mathbf{b}_{r+1}^*, \mathbf{p}_r, \mathbf{f}_{r+1}^*\right) \overset{(iii)}{\leq} U^{\mathrm{fl}}\left(\mathbf{a}_{r+1}^*, \mathbf{b}_{r+1}^*, \mathbf{p}_r, \mathbf{f}_{r+1}^*\right) \quad (51)$$

where inequality (ii) holds due to the derived optimal solution $\mathbf{f}_{r+1}^*$ of (32), and inequality (iii) holds since the objective value of (30) is upper bounded by that of (32). Finally, in line 5 of Algorithm 4, under given $\mathbf{a}_{r+1}^*$, $\mathbf{b}_{r+1}^*$, $\mathbf{f}_{r+1}^*$, we can accordingly obtain

$$U^{\mathrm{fl}}\left(\mathbf{a}_{r+1}^*, \mathbf{b}_{r+1}^*, \mathbf{p}_r, \mathbf{f}_{r+1}^*\right) \leq \hat{U}_3^{\mathrm{fl}}\left(\mathbf{a}_{r+1}^*, \mathbf{b}_{r+1}^*, \mathbf{p}_{r+1}^*, \mathbf{f}_{r+1}^*\right) \leq U^{\mathrm{fl}}\left(\mathbf{a}_{r+1}^*, \mathbf{b}_{r+1}^*, \mathbf{p}_{r+1}^*, \mathbf{f}_{r+1}^*\right). \quad (52)$$

Combining (50)–(52), one can easily identify

$$U^{\mathrm{fl}}(\mathbf{a}_r, \mathbf{b}_r, \mathbf{p}_r, \mathbf{f}_r) \leq U^{\mathrm{fl}}\left(\mathbf{a}_{r+1}^*, \mathbf{b}_{r+1}^*, \mathbf{p}_{r+1}^*, \mathbf{f}_{r+1}^*\right) \quad (53)$$

which indicates that the value of objective function in problem (30) is nondecreasing at each iteration of Algorithm 4. Moreover, the objective value of problem (30) is upper bounded by a finite value under a given tolerance threshold of convergence, such that Algorithm 4 is guaranteed to converge. ■

The complexity of Algorithm 4 is analyzed as follows.

Theorem 5: The overall computational complexity of Algorithm 4 is given by $\mathcal{O}(I_{ls} \cdot A^3 + N + N \cdot I_{\mathrm{sca}})$.

Proof: In Algorithm 4, the computationally most expensive part is mainly incurred by solving the subproblems in (31) (line 3), (32) (line 4), and (45) (line 5), respectively.

In line 3 of Algorithm 4, subproblem (31) is solved by Algorithm 2. We then first analyze the computational complexity of Algorithm 2. In line 2 of Algorithm 2, Algorithm 1 is used to obtain the initial independent set $\mathcal{X}_{\mathcal{G}}$, and its respective complexity is $\mathcal{O}(\rho^2)$, where $\rho = \min\{N, M, K\}$. In line 3 of Algorithm 2, all vertices of $\mathcal{X}_{\mathcal{G}}$ are sorted in an ascending order, thus it has a complexity of $\mathcal{O}(\rho \log \rho)$. Denote the maximum number of elements in $\mathcal{X}_{\mathcal{G}}$ as A, thus in line 6, the complexity of sorting all the elements of $\mathcal{X}_{\mathcal{G}}$ in a descending order is $\mathcal{O}(A \log A)$. In lines 7–16 of Algorithm 2, φ-claw is iteratively searched to update $\mathcal{X}_{\mathcal{G}}$, which has a complexity of $\mathcal{O}(A^3)$. Therefore, the complexity of one-time LS algorithm execution would be $\mathcal{O}(\rho \log \rho + A \log A + A^3) = \mathcal{O}(A^3)$. Supposing I_{ls} is the number of iterations needed for the LS algorithm, the sum complexity of Algorithm 2 is then derived as $\mathcal{O}(I_{ls} \cdot A^3)$.

In line 4 of Algorithm 4, the optimal computation capacity can be directly obtained by solving subproblem (32) via (40), which has a complexity of $\mathcal{O}(N)$.

Finally, in line 5 of Algorithm 4, subproblem (45) is solved by Algorithm 3 for each IIoT device. Let I_{sca} be the number of iterations needed for algorithm convergence. Then, the sum complexity of Algorithm 3 is achieved by $\mathcal{O}(N \cdot I_{\mathrm{sca}})$.

To summarize, the overall computational complexity of Algorithm 4 is calculated as $\mathcal{O}(I_{ls} \cdot A^3 + N + N \cdot I_{\mathrm{sca}})$, which is linear in the number of IIoT devices. ■

It is noted that Algorithm 4 is performed at the CS before conducting the FL training process.[7] For the real implementation of Algorithm 4, the CS and each ES need to first collect the information of the system model parameters for each IIoT device, including the channel state information (CSI) as well as the information of C_n, $|\mathcal{D}_n|$, $f_n^{\max}$, $P_n^{\max}$, ψ_n, N_e, etc.[8] All of these information can be uploaded by the IIoT device before executing Algorithm 4, which, thereby, will not affect the latency or energy consumption of the FL training process.

Remark 1: We highlight that all complete knowledge about the IIoT system (e.g., CSI of the uplink transmission) and the local devices (e.g., IIoT devices, ESs, and CS) is available to enable the real implementation of Algorithm 4 before executing the FL process. However, perfect knowledge w.r.t. the CSI and the system parameter information may be difficult to obtain in reality, especially in case of fast fading channels in practical factory environments as well as hardware limitations of local devices [41]. To tackle the challenges imposed by the imperfect knowledge, a potential solution is to employ the deep reinforcement learning (DRL) approaches, which have shown to be highly beneficial in real-time decision making and ease of implementation for practical system deployment. In this regard, both the CS and ES can manage the information of the system model parameters w.r.t. the IIoT devices, without actually knowing the system states and channel models, thereby eliminating the need for precise knowledge. Particularly, the high-dimensional mixed-integer nonconvex problem in (30) can be reformulated as a constrained Markov Decision Process, and the learning process can be designed through the DRL-based framework, e.g., deep deterministic policy gradient (DDPG) algorithm and soft actor–critic (SAC) algorithm.

V. Experimental Results

In this section, we provide the experimental results to illustrate the performance of the proposed joint edge association and resource allocation scheme.

[7]In practice, the CS enabled by our FL framework will well support the running of optimization solution via Algorithm 4 under different layout conditions w.r.t. the end and edge layers. Since the industrial manufacturing sites are always constrained by the factory's geographical location in both size and space, where there is not too much IIoT devices, the proposed optimized solution thereby is practical to be deployed.

[8]To acquire the CSI for all IIoT devices, the channel coefficients during the uplink transmission needs to be estimated at the ES through training the pilot sequences transmitted from the IIoT devices. Roughly speaking, the ES can gather the CSI of the IIoT device via the dedicated feedback channel [40]. The gathered CSI at the ES can be then shared to the CS through the high-capacity fiber backhaul link. A detailed discussion on the channel estimation in the considered IIoT system is left for future work.

A. Experimental Setup

We simulate an IIoT system that $M = 5$ ESs are uniformly positioned inside a 600 m $\times$ 600 m smart factory area, of which $N = 30$ IIoT devices are randomly distributed with $K = 30$ equally sized RBs. For each IIoT device, the number of CPU cycles for executing one data sample is set to $C_n = 4 \times 10^5$ CPU cycles/sample, and the maximum CPU frequency $f_n^{\max}$ is randomly distributed within [1, 2] GHz. We consider $f_n^{\min} = f_n^{\max} - 1$ GHz. The local training accuracy and edge accuracy are, respectively, defined as $\theta = 0.9$ and $\epsilon = 0.9$. Similar to [30] and [31], we adopt the Rayleigh fading with a shadowing standard deviation of 4 dB, and the elevated non-line-of-sight path loss (in dB) $140.7 + 36.7 \log_{10} d$[km], where d denotes the distance between IIoT device and AP. Unless otherwise indicated, other default parameters are set as $N_e = 8$, $P_n^{\max} = 23$ dBm, $\psi_n = 100$ kb, $\alpha = 0.5$, $B = 1$ MHz, $\sigma_m^2 = -20$ dBm, $\varrho = 10^{-28}$, and $\lambda = 0.023$ dB. The setup of other parameters w.r.t. the FL training is described as follows.

1) Data Set: We use the well-known MNIST [42] data set, which contains a training set of 60 000 samples and a test set of 10 000 samples of handwritten single-digit classes from 0 to 9 with a size of 28 $\times$ 28 pixels. The data samples are first sorted by digit label, and then divided into 120 shards of size 500. We assign each of 30 clients with 4 shards. Thus, each IIoT device only has at most 4 digits.

2) Neural Network Model: The MNIST data set is trained on the convolutional neural network (CNN) model, which has three 5 $\times$ 5 convolution layers with 128, 256, and 512 filters, respectively, an output layer with the Softmax activation function, and a fully connected layer with the rectified linear unit (ReLu) activation function and 512 and 256 units, respectively.

3) FL Algorithms: We compare the proposed FL scheme under our overall alternating optimization algorithm with the state-of-the-art FL algorithm, federated average (FedAvg) [34]. In addition, we adopt the classic multiagent DRL-based FL algorithm, known as "Reinforcement on Federated [5]," where each ES serves as the edge agent that decentralizedly learns an optimal policy. The accuracy-optimal DRL algorithm considers the optimization of resource allocation and device selection jointly in hierarchical FL, with the aim of minimizing the FL evaluating loss, which can be regarded as the approximate optimal FL learning performance.

For the optimization performance evaluation, we compare our proposed scheme with the following different resource allocation and edge association schemes as the baselines.

1) *FL With Computation Capacity Allocation (FL-CCA):* In this scheme, each IIoT device only considers to optimize the computation capacity by solving (32), while applying the random policy to randomly select the ES and allocate the RB and power resources.
2) *FL With Transmit Power Allocation (FL-TPA):* This scheme is designed for only optimizing the transmit power of each IIoT device by solving (33). The random policy is also adopted to randomly select the ES and allocate the RB and computing resources.
3) *FL With Joint Optimization via Hypergraph Matching Algorithm (FL-JO-Hypergraph):* In this scheme, the RB assignment and edge association is jointly optimized by solving (31) via the LS-based algorithm for hypergraph matching, i.e., Algorithm 2, under the randomly allocated power and computing resources for each IIoT device.
4) *FL With Joint Optimization via Greedy Algorithm (FL-JO-Greedy):* This scheme is developed for each IIoT device to only use the greedy algorithm, i.e., Algorithm 1, to optimize the RB assignment and edge association jointly. Meanwhile, the random policy is employed to randomly allocate the power and computing resources.

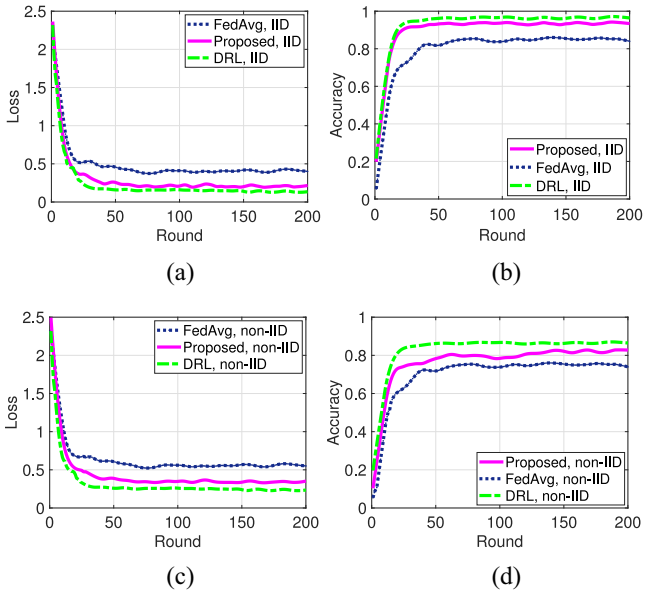


Fig. 3. Training loss and test accuracy with the MNIST data set under the IID and non-IID data distributions, respectively. (a) and (b) IID data on MNIST. (c) and (d) Non-IID data on MNIST.

B. Training Loss and Test Accuracy

We first examine the training loss and test accuracy of the proposed scheme compared with the algorithms of FedAvg and DRL and explore the effect of non-IID data distributions on the learning performance. Fig. 3 shows the results of the training loss and test accuracy with the MNIST data set over the training round under the IID and non-IID data distributions, respectively. We can see that all the algorithms can achieve faster convergence with the increasing of training rounds. From Fig. 3, two observations can be drawn as follows: 1) the proposed scheme achieves lower training loss and the fastest convergence compared to FedAvg and 2) the proposed scheme has much better model accuracy than FedAvg. As expected, both the training loss and test accuracy are quite stable for the proposed scheme. Moreover, our proposed scheme achieves similar performance compared with the accuracy-optimal DRL algorithm, which validates the effectiveness of the proposed scheme for improving the FL learning performance in both loss and accuracy. We can also observe from Fig. 3 that it has an adverse effect on the learning performance w.r.t. all the algorithms by performing the non-IID operation on the MNIST

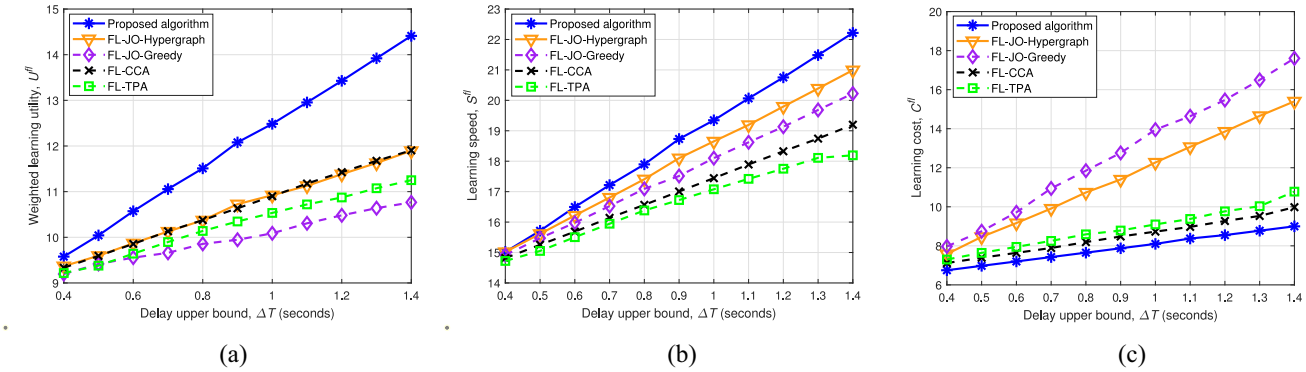


Fig. 4. Tradeoff between learning speed and learning cost under different delay upper bounds ΔT, when $|\mathcal{D}_n| = 200$ and $w = 0.75$. (a) Weighted learning utility versus ΔT. (b) Learning speed versus ΔT. (c) Learning cost versus ΔT.

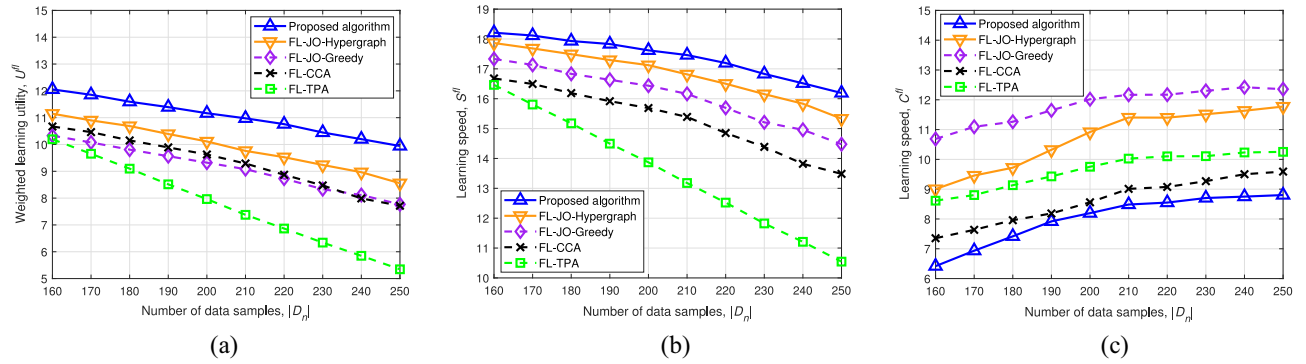


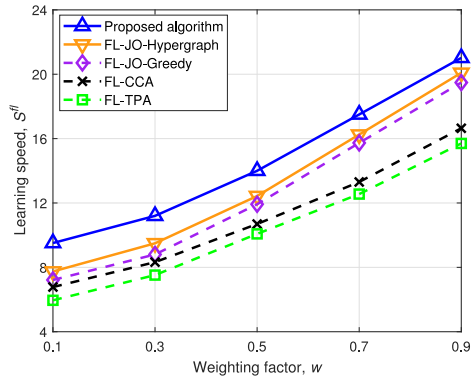
Fig. 5. Tradeoff between learning speed and learning cost under different number of data samples $|\mathcal{D}_n|$, when $\Delta T = 0.8$ s and $w = 0.75$. (a) Weighted learning utility versus $|\mathcal{D}_n|$. (b) Learning speed versus $|\mathcal{D}_n|$. (c) Learning cost versus $|\mathcal{D}_n|$.

data set. For the proposed scheme, it has been shown that compared with the IID setting, the non-IID data distributions increase the FL loss by up to 43% and also harm the model accuracy by around 14.7%.

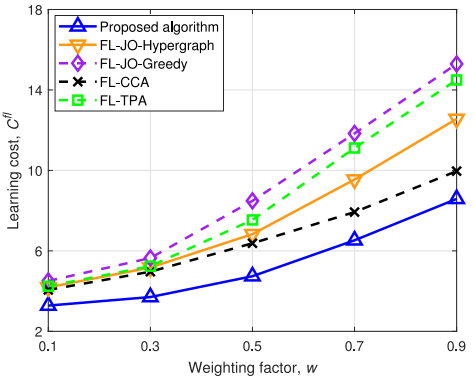
C. Learning Speed and Cost Tradeoff

In Fig. 4, we evaluate the tradeoff between learning speed and learning cost for the proposed scheme and the baselines by varying the delay upper bound ΔT. It can be seen from Fig. 4(a) that as ΔT grows, the system-wide learning utility shows an obvious increasing trend for all schemes. This is because, with the evolution of ΔT, more IIoT devices can satisfy the delay upper bound constraint of (26), which results in the increasing number of IIoT devices engaging in the FL. The larger the number of scheduled IIoT devices in the FL, the higher the learning speed is obtained, which improves the learning utility. This observation confirms the results in Fig. 4(b). From Fig. 4(a), we see that the proposed scheme outperforms the baselines, and the gap between them becomes larger as ΔT increases. To explain, compared with the baselines of FL-CCA and FL-TPA, the proposed scheme improves the system throughput and reduces the edge-level delay by jointly optimizing the edge association and RB assignment. As for the baselines of FL-JO-Hypergraph and FL-JO-Greedy, the proposed scheme considers to optimize both computation capacity and transmit power of IIoT device, which incurs the decreasing of energy consumption. Noteworthy, the learning cost is accordingly reduced for all scheduled IIoT devices, which is in line with the results in Fig. 4(c). From Fig. 4(b) and (c), we can also observe that, comparing to the baselines, the proposed scheme obtains the highest learning speed while keeping the lowest learning cost. This observation further shows that the best tradeoff between learning speed and learning cost can be achieved for the proposed scheme.

Fig. 5 illustrates the effect of the number of data samples $|\mathcal{D}_n|$ on the tradeoff between learning speed and cost for the proposed scheme and the baselines. As can be seen from Fig. 5(a), the learning utility of all schemes presents a moderately decreasing trend as $|\mathcal{D}_n|$ grows. The reason is that, with the increase of $|\mathcal{D}_n|$, each participating IIoT device could incur more energy consumption for local model training in (14), thus increasing the learning cost. This observation confirms the results in Fig. 5(c). The larger the learning cost incurred by energy consumption, the lower the learning utility is obtained, as shown in Fig. 5(a). As expected, the proposed scheme always outperforms the baselines in terms of the learning utility for different $|\mathcal{D}_n|$, and the gap between the proposed scheme and the FL-TPA becomes larger as $|\mathcal{D}_n|$ grows, which shows the superiority of the proposed scheme. This is because the FL-TPA could neither select the IIoT devices with lower learning cost to engage in FL, nor optimize the allocation of computation capacities of IIoT devices to reduce the energy consumption of local training. Thereby, the FL-TPA could only schedule less IIoT devices to reduce the learning cost, which incurs the reduced learning speed, as



(a)



(b)

Fig. 6. Effect of weighting factor w on the learning speed and learning cost, respectively, when $\Delta T = 0.8$ s and $|\mathcal{D}_n| = 200$. (a) Learning speed versus w. (b) Learning cost versus w.

shown in Fig. 5(b). Compared with the other baselines, the FL-TPA has the gradual decreasing learning utility as $|\mathcal{D}_n|$ grows. Furthermore, we can observe from Fig. 5(b) that, with the increase of $|\mathcal{D}_n|$, all schemes will decrease the learning speed to balance the tradeoff. Noteworthy, compared to the baselines, the proposed scheme is less affected over the varying values of $|\mathcal{D}_n|$. Particularly, the proposed scheme can always obtain the highest learning speed, as presented in Fig. 5(b), while keeping the lowest learning cost, as depicted in Fig. 5(c), thus obtaining the best tradeoff of interest.

D. Effect of Weighting Factor

In Fig. 6, we show the learning speed and cost performance of the proposed scheme against the baselines by adjusting the weighting factor w. It can be observed from Fig. 6(a) that the learning speed always increases with w for all schemes. The reason is that when w becomes larger, all schemes are more likely to increase the learning speed by scheduling more IIoT devices to engage in FL for obtaining more learning utilities, instead of reducing the learning cost. As can be found from Fig. 6, the proposed scheme outperforms the baselines by adjusting w. Besides, under different values of w, the proposed scheme always obtains the highest learning speed [as in Fig. 6(a)] with the lowest learning cost [as in Fig. 6(b)] when compared with the baselines. These results are attributed to the fact that the proposed scheme can jointly optimize the edge association as well as the allocations of RBs, computation capacities, and transmit powers of IIoT devices for enhancing the learning speed performance while guaranteeing the lower learning cost. Such observation bolsters the importance of choosing an appropriate weighting factor for the FL training process to obtain the best tradeoff between the learning speed and learning cost.

VI. Conclusion

The focus of this article lies on the joint optimization of edge association and resource allocation for achieving an optimized tradeoff between learning speed and cost in the three-layer FL-enabled IIoT systems. In particular, the weighted learning utility function was invoked for identifying such a tradeoff, and thus the weighted learning utility maximization problem was formulated to jointly optimize the edge association, RBs, computation capacities, and transmit powers of all IIoT devices. To track such a mixed-integer nonconvex optimization problem, we decomposed it into three solvable subproblems, and then proposed a low-complexity alternating optimization algorithm to solve the subproblems iteratively. Finally, numerical experiments are performed to validate our theoretical analysis and algorithm design. The results demonstrate that 1) the proposed scheme has a faster convergence rate and obtains the enhanced learning performance in both the training loss and model accuracy and 2) the proposed scheme can achieve the optimized tradeoff between learning speed and learning cost by properly choosing the system parameters, e.g., the delay upper bound, number of data samples, and weighting factor.

References

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[2] L. Zhang et al., "Digital twin-assisted edge computation offloading in Industrial Internet of Things with NOMA," *IEEE Trans. Veh. Technol.*, vol. 72, no. 9, pp. 11935–11950, Sep. 2023.

[3] D. C. Nguyen et al., "Federated learning for Industrial Internet of Things in future industries," *IEEE Wireless Commun.*, vol. 28, no. 6, pp. 192–199, Dec. 2021.

[4] Q. Guo, F. Tang, and N. Kato, "Federated reinforcement learning-based resource allocation for D2D-aided digital twin edge networks in 6G Industrial IoT," *IEEE Trans Ind. Informat.*, vol. 19, no. 5, pp. 7228–7236, May 2023.

[5] W. Zhang et al., "Optimizing federated learning in distributed Industrial IoT: A multi-agent approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3688–3703, Dec. 2021.

[6] W. Yang, W. Xiang, Y. Yang, and P. Cheng, "Optimizing federated learning with deep reinforcement learning for digital twin empowered Industrial IoT," *IEEE Trans Ind. Informat.*, vol. 19, no. 2, pp. 1884–1893, Feb. 2023.

[7] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

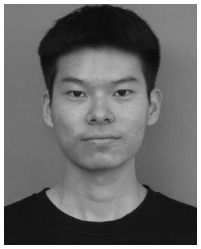
[8] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, Jun. 2020.

[9] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[10] X. Deng et al., "Low-latency federated learning with DNN partition in distributed Industrial IoT networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 3, pp. 755–775, Mar. 2023.
[11] J. Yao and N. Ansari, "Enhancing federated learning in fog-aided IoT by CPU frequency and wireless power control," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3438–3445, Mar. 2021.
[12] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.
[13] T. Zhang and S. Mao, "Energy-efficient federated learning with intelligent reflecting surface," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 2, pp. 845–858, Jun. 2022.
[14] J. Feng, W. Zhang, Q. Pei, J. Wu, and X. Lin, "Heterogeneous computation and resource allocation for wireless powered federated edge learning systems," *IEEE Trans. Commun.*, vol. 70, no. 5, pp. 3220–3233, May 2022.
[15] 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. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, Jan. 2021.
[16] H. Chen, S. Huang, D. Zhang, M. Xiao, M. Skoglund, and H. V. Poor, "Federated learning over wireless IoT networks with optimized communication and resources," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16592–16605, Sep. 2022.
[17] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, Feb. 2021.
[18] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, Apr. 2021.
[19] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *Proc. IEEE ICC Workshops*, Dublin, Ireland, Jun. 2020, pp. 1–6.
[20] S. Liu, G. Yu, X. Chen, and M. Bennis, "Joint user association and resource allocation for wireless hierarchical federated learning with IID and non-IID data," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 7852–7866, Oct. 2022.
[21] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu, "HFEL: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6535–6548, Oct. 2020.
[22] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 11, pp. 2687–2700, Nov. 2022.
[23] T. Zhao, F. Li, and L. He, "DRL-based joint resource allocation and device orchestration for hierarchical federated learning in NOMA-enabled Industrial IoT," *IEEE Trans Ind. Informat.*, vol. 19, no. 6, pp. 7468–7479, Jun. 2023.
[24] V.-D. Nguyen, S. Chatzinotas, B. Ottersten, and T. Q. Duong, "FedFog: Network-aware optimization of federated learning over wireless fog-cloud systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8581–8599, Oct. 2022.
[25] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690–7703, Nov. 2020.
[26] J. Konečný, Z. Qu, and P. Richtárik, "Semi-stochastic coordinate descent," *Optim. Methods Softw.*, vol. 32, no. 5, pp. 993–1005, Mar. 2017.
[27] C. Ma, J. Konečný, M. Jaggi, V. Smith, M. I. Jordan, P. Richtárik, and M. Takáč, "Distributed optimization with arbitrary local solvers," *Optim. Methods Softw.*, vol. 32, no. 4, pp. 813–848, 2017.
[28] L. U. Khan, M. Alsenwi, I. Yaqoob, M. Imran, Z. Han, and C. S. Hong, "Resource optimized federated learning-enabled cognitive Internet of Things for smart industries," *IEEE Access*, vol. 8, pp. 168854–168864, 2020.
[29] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.
[30] V. D. Tuong, W. Noh, and S. Cho, "Delay minimization for NOMA-enabled mobile edge computing in Industrial Internet of Things," *IEEE Trans Ind. Informat.*, vol. 18, no. 10, pp. 7321–7331, Oct. 2022.
[31] M. Li, C. Chen, H. Wu, X. Guan, and X. Shen, "Edge-assisted spectrum sharing for freshness-aware industrial wireless networks: A learning-based approach," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 7737–7752, Sep. 2022.
[32] C. T. Dinh et al., "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–409, Feb. 2021.
[33] 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.
[34] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, Fort Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282.
[35] K. Cumanan, R. Krishna, L. Musavian, and S. Lambotharan, "Joint beamforming and user maximization techniques for cognitive radio networks based on branch and bound method," *IEEE Trans. Wireless Commun.*, vol. 9, no. 10, pp. 3082–3092, Oct. 2010.
[36] L. Zhang, H. Zhang, C. Guo, H. Xu, L. Song, and Z. Han, "Satellite-aerial integrated computing in disasters: User association and offloading decision," in *Proc. IEEE ICC*, Dublin, Ireland, Jun. 2020, pp. 554–559.
[37] L. Wang, H. Wu, Y. Ding, W. Chen, and H. V. Poor, "Hypergraph-based wireless distributed storage optimization for cellular D2D underlays," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2650–2666, Oct. 2016.
[38] M. Razaviyayn, "Successive convex approximation: Analysis and applications," Ph.D. dissertation, Dept. Elect. Eng., Univ. Minnesota. Minneapolis, MN, USA, May 2014.
[39] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Feb. 2017.
[40] Z. Zhao, J. Xia, L. Fan, X. Lei, G. K. Karagiannidis, and A. Nallanathan, "System optimization of federated learning networks with a constrained latency," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 1095–1100, Jan. 2022.
[41] M. M. Wadu, S. Samarakoon, and M. Bennis, "Joint client scheduling and resource allocation under channel uncertainty in federated learning," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5962–5974, Sep. 2021.
[42] Y. LeCun, C. Cortes, and C. J. C. Burges. "The mnist database of handwritten digits." Accessed: Jan. 2023. [Online]. Available: http://yann.lecun.com/exdb/mnist/

Long Zhang (Member, IEEE) received the B.S. degree from China University of Geosciences, Wuhan, China, in 2006, and the Ph.D. degree from the University of Science and Technology Beijing, Beijing, China, in 2012.

In 2017, he was a Visiting Scholar with Tokyo Institute of Technology, Tokyo, Japan. From 2018 to 2020, he was a Postdoctoral Fellow with the University of Houston, Houston, TX, USA. He is currently a Professor with the School of Information and Electrical Engineering, Hebei University of Engineering, Handan, China. His research interests include wireless communications, resource optimization, and mobile computing.



Suiyuan Wu received the B.S. degree from Hebei University of Engineering, Handan, China, in 2020, where he is currently pursuing the M.S. degree with the School of Information and Electrical Engineering.

His current research interests include federated learning, resource optimization, and wireless communications.

Haitao Xu (Member, IEEE) received the B.S. degree in communication engineering from Sun Yat-sen University, Guangzhou, China, in 2007, the M.S. degree in communication system and signal processing from the University of Bristol, Bristol, U.K., in 2009, and the Ph.D. degree in communication and information system from the University of Science and Technology Beijing (USTB), Beijing, China, in 2014.

He is currently a Professor with the Department of Communication Engineering, USTB. His research interests include resource management, wireless communications, game theory, and big data.

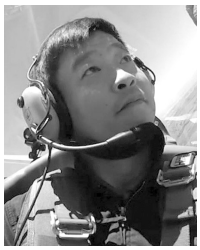
Qilie Liu received the Ph.D. degree in instruments science and technology from Chongqing University, Chongqing, China, in 2012.

He is currently a Professor with Chongqing University of Posts and Telecommunications, Chongqing. His research interests include broadband wireless access technology, wireless network virtualization, and sensor networks.

Choong Seon Hong (Senior Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Kyung Hee University (Seoul Campus), Seoul, South Korea, in 1983 and 1985, respectively, and the Ph.D. degree from Keio University, Tokyo, Japan, in 1997.

In 1988, he joined KT, Seongnam, South Korea, where he was involved in broadband networks as a Technical Staff Member. Since 1993, he has been with Keio University. He was with the Telecommunications Network Laboratory, KT, as a Senior Member of Technical Staff and the Director of the Networking Research Team until 1999. Since 1999, he has been a Professor with the Department of Computer Science and Engineering, Kyung Hee University, Yongin, South Korea. His research interests include future Internet, ad hoc networks, network management, and network security.

Prof. Hong was an Associate Editor of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT and the IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS. He currently serves as an Associate Editor for the *International Journal of Network Management* and an Associate Technical Editor for the *IEEE Communications Magazine*.



Zhu Han (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland at College Park, College Park, MD, USA, in 1999 and 2003, respectively.

From 2000 to 2002, he was a Research and Development Engineer with JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate with the University of Maryland at College Park. From 2006 to 2008, he was an Assistant Professor with Boise State University, Boise, ID, USA. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department and the Computer Science Department, University of Houston, Houston, TX, USA. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid.

Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the *Journal on Advances in Signal Processing* in 2015, the IEEE Leonard G. Abraham Prize in the field of Communications Systems (Best Paper Award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. He has been a 1% Highly Cited Researcher since 2017 according to Web of Science. He is also the winner of the 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: "for contributions to game theory and distributed management of autonomous communication networks." He was an IEEE Communications Society Distinguished Lecturer from 2015 to 2018 and has been an AAAS Fellow since 2019 and an ACM Distinguished Member since 2019.