Resource Allocation for Federated Knowledge Distillation Learning in Internet of Drones

Jingjing Yao, *Member, IEEE*, Semih Cal, and Xiang Sun, *Member, IEEE*

***Abstract*—The Internet of Drones (IoD) integrates drone technology with the Internet of Things, enabling efficient data collection and communication applications. Federated learning (FL) in IoD networks facilitates collaborative model training while preserving data privacy but imposes significant computational and communication demands on resource-constrained drones. Federated knowledge distillation learning (FedKD) addresses this challenge by training both a large teacher model and a smaller student model locally but only updating the smaller student model, thereby reducing communication overhead. This article tackles the resource allocation problem in FedKD within IoD networks, focusing on optimizing CPU computing resource, wireless transmission power, and bandwidth allocation to minimize overall drone energy consumption. We formulate this as an optimization problem, considering constraints on latency, computing resource, bandwidth, and power. To effectively address this problem, we design a low-complexity algorithm. Extensive simulations validate our approach, showing it reduces energy consumption by an average of 85% compared to FedKD and 94% compared to FedAvg (a standard FL algorithm).**

***Index Terms*—Bandwidth allocation, CPU control, federated knowledge distillation learning (FedKD), federated learning (FL), Internet of drones (IoD), Internet of Things (IoT), power control, resource allocation.**

I. Introduction

With the development of emerging applications, such as smart cities, smart homes, smart industries, and healthcare, the number of Internet of Things (IoT) devices has increased explosively [1] and is expected to exceed 75 billion by 2025 [2]. The Internet of Drones (IoD), where drones act as IoT devices, has gained significant attention due to its high mobility and flexibility. IoD is widely applied in traffic monitoring, disaster management, precision agriculture, and more [3]. Modern IoD applications often require the integration of machine learning (ML) techniques to enable autonomous decision-making, pattern recognition, and real-time data analysis. Typically, centralized ML is applied in these scenarios, where large volumes of data collected by drones are transmitted to a remote cloud for processing and analysis. However, centralized ML poses privacy risks, as sensitive data from drones must be transmitted over potentially insecure wireless networks [4].

Compared to traditional centralized ML, federated learning (FL) is an emerging distributed approach that addresses privacy concerns by avoiding raw data transmission [5]. In IoD networks, FL allows multiple drones to collaboratively train a global model without sharing private data [6]. Instead, each drone keeps its data locally and transmits only the parameters of its trained model. In the standard FL process, known as FedAvg [7], the global model is sent to all drones at each iteration. Drones then update their local models with private data and upload the new parameters to a ground base station (BS). The BS aggregates these parameters to refine the global model and redistributes it to the drones. This iterative process of local training, parameter aggregation, and global updating continues until the desired model accuracy is achieved.

While FL offers a promising framework for privacy preservation, its implementation in IoD networks poses significant challenges, especially in resource-constrained environments [8]. One challenge is the size of modern deep learning models, which often contain billions or trillions of parameters [9], [10]. Transmitting these large parameters creates significant communication overhead, making it difficult for wireless networks with limited bandwidth to manage. Another challenge is the increased battery consumption caused by frequent parameter exchanges, which is problematic for drones with limited battery capacity, reducing the overall efficiency of FL in IoD networks.

Federated knowledge distillation (FedKD) [11] is a promising approach that integrates knowledge distillation (KD) [12] with FL to enhance the efficiency of distributed learning. In FedKD, both a large teacher model and a smaller student model are trained collaboratively on drones. The student model, designed to mimic the performance of the larger teacher model, captures essential knowledge without requiring the full complexity of the original model [13]. Crucially, only the lightweight student model is transmitted to the BS for aggregation, minimizing communication overhead. Therefore, FedKD mitigates the issue of transmitting large model parameters by limiting communication to the smaller student models, and also reduces the size of data exchanges, conserving drone battery life [14].

Investigating the resource allocation problem in FedKD is crucial due to the inherent constraints and challenges in

Received 14 November 2024; revised 22 January 2025; accepted 17 February 2025. Date of publication 21 February 2025; date of current version 26 March 2025. This work was supported in part by the National Science Foundation under Award under Grant OIA-2417062 and Grant CNS-2148178, where CNS-2148178 is supported in part by funds from federal agency and industry partners as specified in the Resilient & Intelligent NextG Systems (RINGS) program. *(Corresponding author: Jingjing Yao.)*

Jingjing Yao and Semih Cal are with the Department of Computer Science, Texas Tech University, Lubbock, TX 79409 USA (e-mail: jingjing.yao@ttu.edu; scal@ttu.edu).

Xiang Sun is with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131 USA (e-mail: sunxiang@unm.edu).

Digital Object Identifier 10.1109/JIOT.2025.3545006

IoD networks. One of the primary challenges is the limited computational capacity of drones, which necessitates efficient use of CPU computing resources for training both teacher and student models. It is important to carefully allocate CPU resources to ensure that both models can be trained effectively. Another challenge is the limited battery life of drones, which impacts their ability to continuously perform computational tasks and communicate with the BS. To address this, it is essential to design wireless power allocation strategies that optimize power usage across different drones. Additionally, the constrained communication bandwidth in IoD networks poses a challenge for transmitting model parameters. To mitigate this, it is crucial to allocate wireless bandwidth efficiently.

Based on the above analysis, we investigate the resource allocation problem for FedKD in IoD networks. Specifically, we jointly optimize CPU computing resource allocation, wireless power allocation, and wireless bandwidth allocation to minimize the energy consumption of all drones while adhering to constraints on CPU computing resource, training time, and bandwidth. The major contribution of this work can be summarized as follows.

1) We investigate the resource allocation problem in FedKD within resource-constrained IoD networks, optimizing the use of limited computing and communication resources.
2) We formulate the joint optimization of CPU computing resource allocation, wireless power allocation, and bandwidth allocation with the objective of minimizing the energy consumption of all drones while satisfying constraints on computing resource, training time, bandwidth, and power.
3) We design an alternative algorithm to address the problem by iteratively solving two subproblems: a) jointly optimizing CPU resource allocation and global iteration time and b) jointly optimizing wireless power and bandwidth. The process repeats until convergence.
4) Extensive simulations validate the effectiveness of our approach, showing improved performance on energy consumption and model accuracy. The results demonstrate that our algorithm reduces energy consumption while maintaining high accuracy, outperforming existing methods.

The remainder of this article is organized as follows. In Section II, we review the existing literature. Section III presents our system model and analyzes the drone energy model. In Section IV, we formulate our joint optimization problem to optimize CPU computing resource, wireless transmission power, and bandwidth allocation in FedKD within IoD networks. Section V elaborates on our designed algorithm to effectively address this problem. The performance of our algorithm is demonstrated and analyzed through simulation results in Section VI. Finally, we conclude this article in Section VII.

II. Related Works

Several researchers have conducted studies on IoD networks. Abualigah et al. [15] provided a comprehensive survey of IoD applications and the integration of IoD with technologies, such as neural networks, blockchain, and privacy protection. Shirabayashi and Ruiz [16] reviewed the literature on UAV path planning optimization, comparing various mathematical models and techniques, and highlighting the advantages, safety, and potential for integration with IoD networks. Grieco et al. [17] designed an open-source tool for IoD simulation called IoD-Sim, based on ns-3. It features a three-layer stack: 1) the underlying telecommunications platform; 2) the core for fundamental IoD scenario features; and 3) a simulation development platform for graphical design of use cases. Yao and Ansari [18] investigated the joint optimization of power control and energy harvesting in time-varying IoD networks by deep reinforcement learning.

Resource allocation in FL has been studied in various research. Yang et al. [19] tackled an optimization problem to minimize total energy consumption during the FL process by considering both local computation and transmission energy under a latency constraint. Mo and Xu [20] enhanced system energy efficiency in FL by jointly optimizing communication resource allocation for global ML-parameters aggregation and computation resource allocation for local ML-parameter updates. Chen et al. [21] explored communication-efficient FL over wireless IoT networks with limited resources, formulating a joint optimization problem for client scheduling and resource allocation while considering bandwidth and power constraints. Yao and Ansari [22] investigated the joint optimization of CPU frequency control and wireless transmission power control for all IoT devices in fog-aided IoT networks. Their study aimed to balance the tradeoff between device energy consumption and FL time.

Resource allocation in FL within IoD networks has been investigated in various research studies. Zhang et al. [23] proposed a semi-supervised FL framework to address data privacy and availability challenges in IoD networks. Shvetsov et al. [24] introduced a novel framework that integrates intelligent reflecting surfaces (IRS) and FL with drones to enhance 6G communication networks, improving coverage, reliability, and energy efficiency through smart signal reflection and collaborative learning. Shen et al. [25] minimized the overall training energy consumption of FL in IoD networks by jointly optimizing the local convergence threshold, local iterations, computation resource allocation, and bandwidth allocation while ensuring FL global accuracy and adhering to maximum training latency constraints. Yao and Ansari [26] focused on the power control of all drones to maximize the FL system security rate, taking into account drone battery capacities and FL latency requirements. Pham et al. [27] proposed an energy-efficient FL algorithm that optimizes UAV placement, power control, transmission time, model accuracy, bandwidth allocation, and computing resources to minimize total energy consumption for aerial servers and users. Wang et al. [28] developed a two-tier hierarchical FL scheme assisted by a UAV swarm, where UAVs offload their data to the BS during the local training phase and act as relays to assist the parameter server and local clients in forwarding ML models during the remaining time.

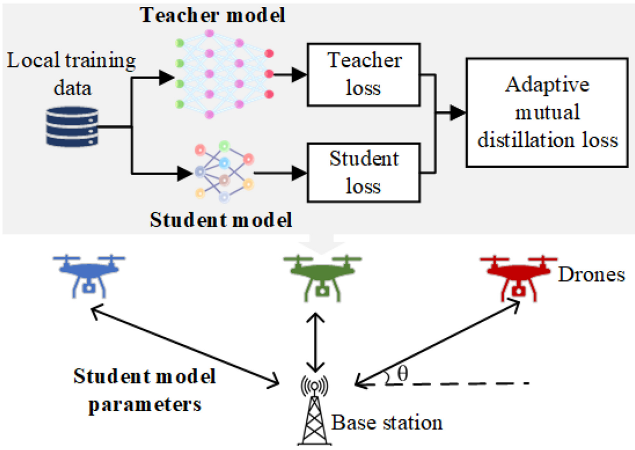

Fig. 1. FedKD in IoD networks.

KD was introduced by Hinton et al. [12], and it has gained attention as a method for accelerating learning by transferring knowledge from a teacher model to a student model. Li and Wang [29] were among the first to leverage KD in FL. Wu et al. [11] proposed the FedKD framework, which is based on adaptive mutual KD. This framework trains both a large model and a small model on the client side, transmitting only the smaller student model to the server, thereby reducing communication overhead while improving accuracy. Wen et al. [30] introduced Fed2KD, a communication-efficient FL framework that employs a novel KD scheme with an attention mechanism and metric learning. Chen et al. [31] designed a KD-based FL optimization problem that accounts for dynamic local resources. Gad et al. [32] explored path optimization and transmission organization algorithms to minimize flight time and extend the range of UAVs, using self-organizing maps for path planning and KD-based FL to reduce communication overhead.

Our recent work [33] investigated CPU computing resource optimization in FedKD within IoD networks. However, to the best of our knowledge, no research has addressed the joint optimization of CPU computing resource, wireless transmission power, and bandwidth allocation in the FedKD framework for IoD networks. To fill this gap, we investigate the resource allocation problem, aiming to minimize the energy consumption of all drones while considering constraints on latency, computing resource, power, and bandwidth.

III. System Model

In our system model, as shown in Fig. 1, there are K drones (indexed by set $\mathcal{K} = \{1, 2, \ldots, K\}$) deployed in the air. These drones collaborate with a ground BS to train a global ML model for applications, such as object recognition and traffic monitoring. The communication between the UAVs and the BS follows frequency division multiple access (FDMA) protocol, where the available bandwidth is divided into frequency bands, and each UAV is allocated a specific band for data transmission. In the FedKD framework, each drone contains a dual-model architecture including a larger, more complex teacher model, and a smaller, more efficient student model. The teacher model generates high-quality predictions and guides the training of the student model. To train this dual-model, an adaptive mutual distillation loss that combines both the loss from the teacher model and the loss from the student model is minimized. The parameters of the smaller student models from the drones are shared with the BS for aggregation to ensure system efficiency. The specific process of FedKD in IoD networks operates through a series of global iterations. In each global iteration, the BS broadcasts the current global ML model to all drones. Each drone then trains its dual-model locally and updates the parameters through multiple local iterations. After completing local training, only the student model parameters are sent back to the BS for aggregation. This process of global iterations is repeated until a desired level of accuracy is achieved or the maximum number of global iterations is reached.

A. FedKD Training Time

In each global iteration, each drone's FedKD training time consists of the local computation time for updating dual-model parameters and the wireless data transmission time for uploading the student model parameters. Note that the broadcasting time from the BS to drones is typically much smaller than the local training and parameter transmission time, so we neglect it in our work [34]. For drone k, the local computation time for the teacher model is $t_k^{ta} = (C_k^{ta} D_k / f_k^{ta})$, and for the student model, it is $t_k^{\text{stu}} = (C_k^{\text{stu}} D_k / f_k^{\text{stu}})$. Here, C_k^{ta} and C_k^{stu} are the numbers of CPU cycles required to train a data sample in one local iteration for the teacher and student models, respectively. f_k^{ta} and f_k^{stu} are the CPU frequencies of the teacher and student models, respectively, and D_k is the number of data samples in the dataset. Since the teacher and student models are trained simultaneously during the local training process, the total local computation time t_k^{cmp} is determined by the longer of t_k^{ta} and t_k^{stu}

$$t_k^{\text{cmp}} = I_k \max\left\{t_k^{ta}, t_k^{\text{stu}}\right\} = I_k \max\left\{\frac{C_k^{ta} D_k}{f_k^{ta}}, \frac{C_k^{\text{stu}} D_k}{f_k^{\text{stu}}}\right\} \tag{1}$$

where I_k is the number of local iterations.

The wireless data transmission time is influenced by the wireless channel between the BS and the drones. We adopt the widely accepted drone communication model [35], which assumes that the wireless channel between the BS and drones follows a probability model that can be either line-of-sight (LoS) or non-LoS (NLoS). The probabilities of being LoS and NLoS are given by $\Pr(\text{LoS}) = (1/[1 + \alpha \exp(-\beta[(180/\pi)\theta - \alpha])])$ and $\Pr(\text{NLo}S) = 1 - \Pr(\text{LoS})$, respectively, where α and β are environment-related constants (e.g., rural or urban areas), and θ is the elevation angle as shown in Fig. 1. The LoS and NLoS path loss models follow the free space propagation loss model, given by $\text{PL(LoS)} = 20 \log_{10}(4\pi f_c d/c) + \xi_{\text{LoS}}$ and $\text{PL(NLoS)} = 20 \log_{10}(4\pi f_c d/c) + \xi_{\text{NLoS}}$, where ξ_{LoS} and ξ_{NLoS} are environment-related constants, f_c is the carrier frequency, c is the speed of light, and d is the distance between a drone and the BS. The average path loss can be calculated as $\overline{PL} =$

$\Pr(\text{LoS}) \cdot \text{PL}_{\text{LoS}} + \Pr(\text{NLoS}) \cdot \text{PL}_{\text{NLoS}}$, and thus the wireless channel gain between a drone and the BS is $G = 10^{-(\overline{\text{PL}}/10)}$.

According to the Shannon equation, the data transmission rate from drone k to the BS is given by $r_k = B_k \log_2(1 + [p_k G_k / N_0 B_k])$, where B_k is the bandwidth allocated to drone k, p_k is its wireless transmission power, G_k is the wireless channel gain, and N_0 is the noise power spectral density. Consequently, the wireless parameter transmission time for drone k is

$$t_k^{\text{com}} = \frac{s_k}{r_k} \tag{2}$$

where s_k is the data size of the student model. In summary, the FedKD training time for drone k in one global iteration is

$$t_k = t_k^{\text{cmp}} + t_k^{\text{com}}. \tag{3}$$

B. FedKD Energy Consumption

The energy consumption of a drone arises from performing local model training and wireless data transmission. We assume that the drone's mobility and FedKD operations are powered separately by the drone's motor battery and the battery associated with the computing board, respectively. Therefore, we focus solely on the energy consumption related to FedKD operations in our work.

The energy for local computation is consumed by training both the teacher and student models. The energy consumption for processing a CPU cycle of the teacher model is $\kappa (f_k^{ta})^2$, where κ represents the switch capacitance-related constant [36]. Thus, the energy consumption for training the teacher model in one local iteration is $E_k^{ta} = \kappa C_k^{ta} D_k (f_k^{ta})^2$, where C_k^{ta} is the required CPU cycles per data sample for the teacher model and D_k is the number of data samples. Similarly, the energy consumption for training the student model in one local iteration is $E_k^{\text{stu}} = \kappa C_k^{\text{stu}} D_k (f_k^{\text{stu}})^2$, where C_k^{stu} is the required CPU cycles per data sample for the student model. Therefore, the total energy consumed in local computation for drone k is

$$\begin{aligned} E_k^{\text{cmp}} &= I_k\big(E_k^{ta} + E_k^{\text{stu}}\big) \\ &= I_k\Big[\kappa C_k^{ta} D_k \big(f_k^{ta}\big)^2 + \kappa C_k^{\text{stu}} D_k \big(f_k^{\text{stu}}\big)^2\Big]. \end{aligned} \tag{4}$$

The energy consumption of wireless data transmission for uploading student model can be calculated as

$$E_k^{\text{com}} = p_k t_k^{\text{com}} = \frac{p_k s_k}{r_k}. \tag{5}$$

In summary, the energy consumption of drone k in a global iteration is

$$E_k = E_k^{\text{cmp}} + E_k^{\text{com}}. \tag{6}$$

IV. Problem Formulation

In this section, we formulate the resource allocation problem for optimizing CPU computing resources, wireless transmission power, and wireless bandwidth in FedKD within IoD networks. The objective is to minimize the total energy consumption of all drones while considering constraints on training time, CPU computing resource, wireless power, and wireless bandwidth. The problem can be formulated as follows:

$$\begin{aligned} \textbf{P0:} \ & \min_{f_k^{ta}, f_k^{\text{stu}}, b_k, p_k, \tau} \sum_{k=1}^{K} \Bigg\{ I_k\Big[\kappa C_k^{ta} D_k \big(f_k^{ta}\big)^2 + \kappa C_k^{\text{stu}} D_k \big(f_k^{\text{stu}}\big)^2\Big] \\ & \quad + \frac{s_k p_k}{b_k \log_2\Big(1 + \frac{p_k G_k}{N_0 b_k}\Big)} \Bigg\} \\ & \text{s.t.} \quad f_k^{\min} \le f_k^{ta} + f_k^{\text{stu}} \le f_k^{\max} \quad \forall k \in \mathcal{K} && (7) \\ & I_k \max\left\{ \frac{C_k^{ta} D_k}{f_k^{ta}}, \ \frac{C_k^{\text{stu}} D_k}{f_k^{\text{stu}}} \right\} \\ & \quad + \frac{s_k}{b_k \log_2\Big(1 + \frac{p_k G_k}{N_0 b_k}\Big)} \le \tau \quad \forall k \in \mathcal{K} && (8) \\ & \sum_{k=1}^{K} b_k \le B && (9) \\ & p_k^{\min} \le p_k \le p_k^{\max} \quad \forall k \in \mathcal{K}. && (10) \end{aligned}$$

The objective of problem **P0** is to minimize the total energy consumption of all drones, including energy used for both local computation and wireless data transmission, as defined in (6). Constraint (7) ensures that the total CPU computing resources allocated for training both the teacher model and the student model on each drone k remain within the physical and operational limits of the drone's computational capacity. Specifically, the sum of the CPU frequencies allocated to the teacher model f_k^{ta} and the student model f_k^{stu} should lie between a predefined minimum $f_k^{\min}$ and maximum $f_k^{\max}$. These bounds represent the operational range of the drone's CPU, where $f_k^{\min}$ corresponds to the minimum frequency required to ensure efficient operation, and $f_k^{\max}$ corresponds to the maximum frequency that the drone hardware can support without risking overheating or excessive power consumption. In each global iteration of FedKD, all drones transmit their parameters to the BS for aggregation. The BS should receive the parameters from all drones before performing the aggregation. Therefore, the total time t_k required for computation and communication of each drone in one global iteration should be within the global iteration time τ, as specified in (8). Constraint (9) imposes a bandwidth allocation limit, requiring the total bandwidth assigned to all drones to stay within the available bandwidth B. Constraint (10) ensures that each drone's wireless transmission power remains within the allowed range, bounded by $p_k^{\min}$ and $p_k^{\max}$. The lower bound $p_k^{\min}$ ensures the drone has sufficient power to maintain reliable communication with the BS, overcoming channel noise and path loss. The upper bound $p_k^{\max}$ limits the power to prevent hardware damage.

It is important to note that obtaining the solution to problem **P0** is challenging due to its nonconvex nature, which makes it difficult to solve using standard optimization techniques. To address this, we design a low-complexity algorithm that efficiently finds a solution by breaking down the problem into more manageable subproblems in next section.

V. Algorithm Design

To address the complex problem **P0**, we propose an alternating iteration method to optimize $\langle f_k^{ta}, f_k^{\text{stu}}, \tau \rangle$ and $\langle b_k, p_k \rangle$ in turn. In this method, we first solve the subproblem of optimizing f_k^{ta}, f_k^{stu}, and τ while keeping b_k and p_k fixed. Next, we address the subproblem of optimizing b_k and p_k while keeping f_k^{ta}, f_k^{stu}, and τ fixed. These two steps are repeated iteratively until the objective value of problem **P0** converges to a stable solution or the maximum number of iterations is reached.

A. Joint Optimization of CPU Frequency and Global Iteration Time

In this subproblem, we fix the wireless bandwidth and power $\langle b_k, p_k \rangle$ as $\langle b_k^*, p_k^* \rangle$ that satisfy (9) and (10). Then, problem **P0** becomes a joint optimization of CPU frequency and global iteration time

$$\begin{aligned}\textbf{P1:}\ &\min_{f_k^{ta}, f_k^{\text{stu}}, \tau} \sum_{k=1}^{K} I_k\left[\kappa C_k^{ta} D_k \left(f_k^{ta}\right)^2 + \kappa C_k^{\text{stu}} D_k \left(f_k^{\text{stu}}\right)^2\right]\\ &\text{s.t. } (9)\\ &I_k \max\left\{\frac{C_k^{ta} D_k}{f_k^{ta}}, \frac{C_k^{\text{stu}} D_k}{f_k^{\text{stu}}}\right\} + \frac{s_k}{b_k^* \log_2\left(1 + \frac{p_k^* G_k}{N_0 b_k^*}\right)} \le \tau\\ &\forall k \in \mathcal{K}. \end{aligned} \tag{11}$$

Note that the item $(s_k p_k^*/[b_k^* \log_2(1 + [p_k^* G_k / N_0 b_k^*])])$ in the objective function of problem **P0** is removed because it is a constant and does not affect the solution of the problem.

The difficulty in addressing problem **P1** lies in the global iteration time τ in (11), as it couples f_k^{ta} and f_k^{stu} for different drones. In other words, problem **P1** is straightforward to solve without (11) by decomposing it into K independent subproblems and addressing each subproblem separately. Since drones upload their parameters to the BS simultaneously, the global iteration time depends on the slowest drone, which has the longest FedKD training time. Motivated by this analysis, we enumerate τ as the FedKD training time for each drone, considering each drone as the potential slowest one. We assume drone j is the slowest, thus $\tau = \tau_j = I_j \max\{(C_j^{ta} D_j / f_j^{ta}), (C_j^{\text{stu}} D_j / f_j^{\text{stu}})\} + (s_j / [b_j^* \log_2(1 + [p_j^* G_j / N_0 b_j^*])])$. Then, problem **P1** can be transformed into two cases: one for $k = j$ and another for $k \neq j$.

For the case when $k = j$, problem **P1** becomes

$$\textbf{P1-1:}\ \min_{f_j^{ta}, f_j^{\text{stu}}, \tau_j} I_j\left[\kappa C_j^{ta} D_j \left(f_j^{ta}\right)^2 + \kappa C_j^{\text{stu}} D_j \left(f_j^{\text{stu}}\right)^2\right]$$

$$\text{s.t. } f_j^{\min} \le f_j^{ta} + f_j^{\text{stu}} \le f_j^{\max} \tag{12}$$

$$I_j \max\left\{\frac{C_j^{ta} D_j}{f_j^{ta}}, \frac{C_j^{\text{stu}} D_j}{f_j^{\text{stu}}}\right\} + \frac{s_j}{b_j^* \log_2\left(1 + \frac{p_j^* G_j}{N_0 b_j^*}\right)} = \tau_j \tag{13}$$

where we can get the specific value of τ_j so it can then be considered a constant.

For the case when $k \neq j$, problem **P1** can be separated into following subproblems:

$$\textbf{P1-2:}\ \min_{f_k^{ta}, f_k^{\text{stu}}} \sum_{k=1}^{K} I_k\left[\kappa C_k^{ta} D_k \left(f_k^{ta}\right)^2 + \kappa C_k^{\text{stu}} D_k \left(f_k^{\text{stu}}\right)^2\right]$$

$$\text{s.t. } f_k^{\min} \le f_k^{ta} + f_k^{\text{stu}} \le f_k^{\max} \quad \forall k \in \mathcal{K} \setminus j \tag{14}$$

$$I_k \max\left\{\frac{C_k^{ta} D_k}{f_k^{ta}}, \frac{C_k^{\text{stu}} D_k}{f_k^{\text{stu}}}\right\} + \frac{s_k}{b_k^* \log_2\left(1 + \frac{p_k^* G_k}{N_0 b_k^*}\right)} \le \tau_j \quad \forall k \in \mathcal{K} \setminus j. \tag{15}$$

It can be observed that the objective function of problem **P1-1** is an increasing function with regard to f_j^{ta} and f_j^{stu}. Hence, we can set $f_j^{ta} + f_j^{\text{stu}} = f_j^{\min}$ to minimize the objective function. We then substitute $f_j^{\text{stu}} = f_j^{\min} - f_j^{ta}$ into the objective function, which becomes $\mathcal{H}(f_j^{ta}) = I_j \kappa [C_j^{ta} (f_j^{ta})^2 + C_j^{\text{stu}} (f_j^{\min} - f_j^{ta})^2] D_j$. To minimize the objective function $\mathcal{H}(f_j^{ta})$, we take the first derivative and set it to zero. So, we have $\mathcal{H}'(f_j^{ta}) = I_j \kappa [2 C_j^{ta} f_j^{ta} - 2 C_j^{\text{stu}} (f_j^{\min} - f_j^{ta})] D_j = 0$. Therefore, the optimum solution of f_j^{ta} can be calculated as

$$f_j^{ta*} = \frac{c_j^{\text{stu}} f_j^{\min}}{c_j^{ta} + c_j^{\text{stu}}} \tag{16}$$

and the optimum solution of f_j^{stu} is

$$f_j^{\text{stu}*} = \frac{c_j^{ta} f_j^{\min}}{c_j^{ta} + c_j^{\text{stu}}}. \tag{17}$$

The global iteration time τ_j can then be calculated as

$$\tau_j = I_j \max\left\{\frac{C_j^{ta} D_j}{f_j^{ta*}}, \frac{C_j^{\text{stu}} D_j}{f_j^{\text{stu}*}}\right\} + \frac{s_j}{b_j^* \log_2\left(1 + \frac{p_j^* G_j}{N_0 b_j^*}\right)}. \tag{18}$$

Problem **P1-2** is a convex problem and so we address it by solving the Karush–Kuhn–Tucker (KKT) conditions. (15) is equivalent to $I_k (C_k^{ta} D_k / f_k^{ta}) + (s_k / [b_k^* \log_2(1 + [p_k^* G_k / N_0 b_k^*])]) \le \tau_j$ and $I_k (C_k^{\text{stu}} D_k / f_k^{\text{stu}}) + (s_k / [b_k^* \log_2(1 + [p_k^* G_k / N_0 b_k^*])]) \le \tau_j$. Hence, we have

$$f_k^{ta} \ge \frac{I_k C_k^{ta} D_k}{\tau_j - \frac{s_k}{b_k^* \log_2\left(1 + \frac{p_k^* G_k}{N_0 b_k^*}\right)}} \tag{19}$$

and

$$f_k^{\text{stu}} \ge \frac{I_k C_k^{\text{stu}} D_k}{\tau_j - \frac{s_k}{b_k^* \log_2\left(1 + \frac{p_k^* G_k}{N_0 b_k^*}\right)}}. \tag{20}$$

The Lagrangian function of the objective in problem **P1-2** is $\mathcal{L}(f_k^{ta}, f_k^{\text{stu}}, u, v) = I_k \kappa [C_k^{ta} (f_k^{ta})^2 + C_k^{\text{stu}} (f_k^{\text{stu}})^2] D_k + u(f_k^{\min} - f_k^{ta} - f_k^{\text{stu}}) + v(f_k^{ta} + f_k^{\text{stu}} - f_k^{\max})$. The KKT conditions are as follows:

$$\frac{\partial \mathcal{L}}{\partial f_k^{ta}} = 2 I_k \kappa C_k^{ta} f_k^{ta} D_k - u + v = 0 \tag{21}$$

$$\frac{\partial \mathcal{L}}{\partial f_k^{\text{stu}}} = 2 I_k \kappa C_k^{\text{stu}} f_k^{\text{stu}} D_k - u + v = 0 \tag{22}$$

$$u\left[f_k^{\min} - f_k^{ta} - f_k^{\mathrm{stu}}\right] = 0 \tag{23}$$

$$v\left[f_k^{ta} + f_k^{\mathrm{stu}} - f_k^{\max}\right] = 0 \tag{24}$$

$$u \geq 0, v \geq 0. \tag{25}$$

Based on (21) and (25), we can conclude $u > 0$. By substituting it to (23), we have $f_k^{\min} - f_k^{ta} - f_k^{\mathrm{stu}} = 0$, and so $f_k^{ta} + f_k^{\mathrm{stu}} - f_k^{\max} \neq 0$. Hence, we have $v = 0$ based on (24). Combining $v = 0$ with (21) and (22), we obtain $2I_k\kappa C_k^{ta} f_k^{ta} D_k = 2I_k\kappa C_k^{\mathrm{stu}} f_k^{\mathrm{stu}} D_k$. Therefore, the solution of the KKT conditions is $f_k^{ta} = (C_k^{\mathrm{stu}} f_k^{\min}/[C_k^{ta} + C_k^{\mathrm{stu}}])$ and $f_k^{\mathrm{stu}} = (C_k^{ta} f_k^{\min}/[C_k^{ta} + C_k^{\mathrm{stu}}])$. Combining this with (19) and (20), the solution of problem **P1-2** is

$$f_k^{ta} = \max\left\{\frac{C_k^{\mathrm{stu}} f_k^{\min}}{C_k^{ta} + C_k^{\mathrm{stu}}}, \frac{I_k C_k^{ta} D_k}{\tau_j - \frac{s_k}{b_k^* \log_2\left(1+\frac{p_k^* G_k}{N_0 b_k^*}\right)}}\right\} \tag{26}$$

and

$$f_k^{\mathrm{stu}} = \max\left\{\frac{C_k^{ta} f_k^{\min}}{C_k^{ta} + C_k^{\mathrm{stu}}}, \frac{I_k C_k^{\mathrm{stu}} D_k}{\tau_j - \frac{s_k}{b_k^* \log_2\left(1+\frac{p_k^* G_k}{N_0 b_k^*}\right)}}\right\}. \tag{27}$$

To obtain the optimal solution for problem **P1**, we choose the best result generated by enumerating all possible slowest drones $j \in \mathcal{K}$. Specifically, for each candidate slowest drone j, we first solve problem **P1-1** and then problem **P1-2**. We record the possible objective value and the corresponding f_k^{ta}, f_k^{stu}, and τ_j. After completing all enumerations, we select the best f_k^{ta}, f_k^{stu}, and τ_j that yield the minimum objective value as our final solution. The process for addressing problem **P1** is outlined in Algorithm 1. lines 1–3 initialize the candidate vector of possible objective values for each enumeration. Lines 4–12 enumerate each possible slowest drone. Line 5 calculates the CPU frequencies of the slowest drone, and line 6 calculates the global iteration time. Lines 7–9 determine the CPU frequencies for all other drones. Lines 10 and 11 store the possible objective value. Finally, lines 13 and 14 select the best solution among all possible objective values. Lines 1–3 execute in $\mathcal{O}(K)$ time. The loop in lines 7–9 has a time complexity of $\mathcal{O}(K)$, making the overall complexity of the loop from Lines 4–12 equal to $\mathcal{O}(K^2)$. lines 13 and 14 run in constant time $\mathcal{O}(1)$. Therefore, the total computational complexity of Algorithm 1 is $\mathcal{O}(K^2)$.

B. Joint Optimization of Wireless Power and Bandwidth Allocation

In this subproblem, we fix $\langle f_k^{ta}, f_k^{\mathrm{stu}}, \tau\rangle$ as $\langle f_k^{ta*}, f_k^{\mathrm{stu}*}, \tau^*\rangle$, and the problem **P0** becomes

$$\begin{aligned}
\textbf{P2:}\ &\min_{b_k, p_k} \sum_{k=1}^{K} \frac{s_k p_k}{b_k \log_2\left(1 + \frac{p_k G_k}{N_0 b_k}\right)}\\
&\text{s.t.}\quad (9),\ (10)\\
&I_k \max\left\{\frac{C_k^{ta} D_k}{f_k^{ta*}}, \frac{C_k^{\mathrm{stu}} D_k}{f_k^{\mathrm{stu}*}}\right\}\\
&+ \frac{s_k}{b_k \log_2\left(1 + \frac{p_k G_k}{N_0 b_k}\right)} \leq \tau^* \quad \forall k \in \mathcal{K}. \qquad (28)
\end{aligned}$$

Algorithm 1: Algorithm for Joint Optimization of CPU Frequency and Global Iteration Time

1 **for** *each* $j \in \mathcal{K}$ **do**
2   Initialize candidate vector $V[j] = 0$;
3 **end**
4 **for** *each* $j \in \mathcal{K}$ **do**
5   Calculate drone j's CPU frequencies f_j^{ta} and f_j^{stu} according to (16) and (17) ;
6   Calculate the global iteration time τ according to (18) ;
7   **for** *each* $k \in \mathcal{K} \setminus j$ **do**
8   Calculate drone k's CPU frequencies f_k^{ta} and f_k^{stu} according to (26) and (27) ;
9   **end**
10   Calculate the objective value R of problem **P1** ;
11   Assign $V[j] = R$ and record corresponding f_k^{ta} and f_k^{stu} ;
12 **end**
13 Choose j that achieves the minimum $V[j]$;
14 Select the corresponding f_k^{ta} and f_k^{stu} as the optimum solution of problem **P1** ;

Note that the item $I_k[\kappa C_k^{ta} D_k (f_k^{ta*})^2 + \kappa C_k^{\mathrm{stu}} D_k (f_k^{\mathrm{stu}*})^2]$ is removed in the objective for simplicity because it is a constant and does not affect the optimal solution. It is challenging to address problem **P2** because of its nonconvexity, so we introduce another variable $t_k = (s_k/[b_k \log_2(1 + [p_k G_k/N_0 b_k])])$ to simplify this problem. Then, problem **P2** becomes

$$\begin{aligned}
\textbf{P3:}\ &\min_{b_k, p_k, t_k} \sum_{k=1}^{K} p_k t_k\\
&\text{s.t.}\quad (9),\ (10)\\
&t_k \leq \tau^* - I_k \max\left\{\frac{C_k^{ta} D_k}{f_k^{ta*}}, \frac{C_k^{\mathrm{stu}} D_k}{f_k^{stu*}}\right\} \quad \forall k \in \mathcal{K} \qquad (29)\\
&t_k b_k \log_2\left(1 + \frac{p_k G_k}{N_0 b_k}\right) = s_k \quad \forall k \in \mathcal{K}. \qquad (30)
\end{aligned}$$

It is still challenging to solve problem **P3**. Therefore, we design an iterative method to address it. In this method, we first fix t_k and then use the obtained values of b_k and p_k to update t_k. We repeat this process until the objective value of problem **P3** converges to a stable value.

We first fix t_k as t_k^* that satisfies (29). Then, problem **P3** can be transformed into

$$\begin{aligned}
\textbf{P3-1:}\ &\min_{b_k, p_k} \sum_{k=1}^{K} p_k t_k^*\\
&\text{s.t.}\quad (9),\ (10)\\
&t_k^* b_k \log_2\left(1 + \frac{p_k G_k}{N_0 b_k}\right) = s_k \quad \forall k \in \mathcal{K}. \qquad (31)
\end{aligned}$$

Equation (31) is equivalent to $p_k = (N_0 b_k/G_k)(2^{(s_k/t_k^* b_k)} - 1)$. By substituting it into problem **P3-1**, we have

$$\textbf{P3-2:} \min_{b_k} \sum_{k=1}^{K} \frac{N_0 b_k}{G_k}\left(2^{\frac{s_k}{t_k^* b_k}} - 1\right) t_k^*$$
$$\text{s.t.} \quad (9)$$
$$p_k^{\min} \le \frac{N_0 b_k}{G_k}\left(2^{\frac{s_k}{t_k^* b_k}} - 1\right) \le p_k^{\max} \quad \forall k \in \mathcal{K}. \quad (32)$$

Lemma 1: Problem **P3-2** is a convex optimization problem.

Proof: To demonstrate the convexity of problem **P3-2**, we obtain the first derivative of the objective function $g(b_k) = \sum_{k=1}^{K} p_k t_k^* = \sum_{k=1}^{K} (N_0 b_k/G_k)(2^{(s_k/t_k^* b_k)} - 1)t_k^*$ is

$$g'(b_k) = t_k^* \frac{\partial p_k}{\partial b_k} = \frac{N_0 t_k^*}{G_k}\left(2^{\frac{s_k}{t_k^* b_k}} - 1 - \frac{s_k \ln 2}{t_k^* b_k} 2^{\frac{s_k}{t_k^* b_k}}\right)$$
$$= \left(\frac{N_0 t_k^*}{G_k} - \frac{N_0 s_k \ln 2}{G_k b_k}\right) 2^{\frac{s_k}{t_k^* b_k}} - \frac{N_0 t_k^*}{G_k}. \quad (33)$$

The second derivative is

$$g''(b_k) = t_k^* \frac{\partial p_k}{\partial^2 b_k} = \frac{N_0 s_k^2 (\ln 2)^2}{G_k t_k^* b_k^3} 2^{\frac{s_k}{t_k^* b_k}} > 0. \quad (34)$$

Therefore, the objective function is a convex function and p_k is a convex function with regard to b_k. (32) is hence a convex set. Additionally, (9) is a convex set. In summary, problem **P3-2** is a convex optimization problem. ■

Because we have the second derivative $g''(b_k) = (\partial p_k/\partial^2 b_k) > 0$, $g'(b_k) = (\partial p_k/\partial b_k)$ is an increasing function. Note that $(\partial p_k/\partial^2 b_k) \to 0$ when $b_k \to \infty$. Hence, we have $(\partial p_k/\partial b_k) < 0$, and so p_k is a decreasing function with regard to b_k. Let $b_k^{\min}$ and $b_k^{\max}$ be the minimum and maximum values of b_k, respectively. Then, we have

$$p_k^{\min} = \frac{N_0 b_k^{\max}}{G_k}\left(2^{\frac{s_k}{t_k^* b_k^{\max}}} - 1\right) \quad (35)$$

and

$$p_k^{\max} = \frac{N_0 b_k^{\min}}{G_k}\left(2^{\frac{s_k}{t_k^* b_k^{\min}}} - 1\right). \quad (36)$$

Therefore, (32) is equivalent to

$$b_k^{\min} \le b_k \le b_k^{\max} \quad \forall k \in \mathcal{K}. \quad (37)$$

Since problem **P3-2** is a convex problem, we can obtain its solution by solving its KKT conditions. The Lagrangian function of the objective function is $\mathcal{L}(b_k, \lambda) = \sum_{k=1}^{K} (N_0 b_k/G_k)(2^{(s_k/t_k^* b_k)} - 1)t_k^* + \lambda(\sum_{k=1}^{K} b_k - B)$. The KKT conditions include

$$\frac{\partial \mathcal{L}(b_k, \lambda)}{\partial b_k} = \left(\frac{N_0 t_k^*}{G_k} - \frac{N_0 s_k \ln 2}{G_k b_k}\right) 2^{\frac{s_k}{t_k^* b_k}} - \frac{N_0 t_k^*}{G_k} + \lambda = 0 \quad (38)$$

$$\lambda\left(\sum_{k=1}^{K} b_k - B\right) = 0 \quad (39)$$

$$\lambda \ge 0 \quad (40)$$

$$b_k^{\min} \le b_k \le b_k^{\max} \quad \forall k \in \mathcal{K}. \quad (41)$$

Based on (38), we have $g'(b_k) + \lambda = 0$. Since $g'(b_k) < 0$, we have $\lambda > 0$. Substituting it into (39), we then have

$$\sum_{k=1}^{K} b_k - B = 0. \quad (42)$$

Assume $\tilde{b}_k$ and $\tilde{\lambda}$ is the solution of the set of equations from (38) and (42). Combining these with (37), the solution of problem **P3-2** can be calculated as

$$b_k^* = \begin{cases} b_k^{\min}, & \text{if } \tilde{b}_k \le b_k^{\min} \\ b_k^{\max}, & \text{if } \tilde{b}_k \ge b_k^{\max} \\ \tilde{b}_k, & \text{otherwise.} \end{cases} \quad (43)$$

Based on (31), we have $p_k^* = (N_0 b_k^*/G_k)(2^{(s_k/t_k^* b_k^*)} - 1)$. After that, we can update the value of t_k^* according to

$$t_k^* = \frac{s_k}{b_k^* \log_2\left(1 + \frac{p_k^* G_k}{N_0 b_k^*}\right)}. \quad (44)$$

Then, we repeat the process of addressing problem **P3-2** and updating t_k^* until the objective value of problem **P3** converges.

The process of addressing problem **P3** is outlined in Algorithm 2. Lines 1–3 initialize the auxiliary variable, the iteration number, and the maximum iteration number I_1. Lines 4–14 iteratively calculate the wireless power and bandwidth and update the auxiliary variable. Specifically, lines 7–9 compute each drone's bandwidth, line 10 determines each drone's wireless transmission power, and line 11 updates the auxiliary variable. The while loop in lines 4–14 runs I_1 times in the worst case scenario. Within each iteration, lines 6–12 execute K times. Specifically, line 8 involves solving a system of equations by solvers, often with Newton's method, which has a time complexity of $\mathcal{O}(K^2 \cdot I)$, where I is the number of iterations required for convergence [37]. Thus, the total computational complexity of Algorithm 2 is $\mathcal{O}(I_1 \cdot K^3 \cdot I)$.

In summary, our proposed algorithm for addressing the resource allocation problem **P0** involves iteratively optimizing two components: 1) the joint optimization of CPU frequency and 2) global iteration time using Algorithm 1, and the joint optimization of wireless power and bandwidth allocation using Algorithm 2. This iterative process continues until convergence or the maximum number of iterations is reached. The complete process of our proposed algorithm, Federated KD with optimized CPU frequency, wireless transmission power, and bandwidth (FedKD-FPB), is outlined in Algorithm 3. Lines 1–4 initialize the wireless power, bandwidth, iteration number, and maximum number of iterations I_2. Lines 6–9 alternatively calculate the two subproblems. The while loop in lines 5–10 runs I_2 times in the worst case scenario. Line 7 executes in $\mathcal{O}(K^2)$ time from Algorithm 1, and line 8 has a complexity of $\mathcal{O}(I_1 \cdot K^3 \cdot I)$ from Algorithm 2. Thus, the overall computational complexity of the proposed FedKD-FPB algorithm is $\mathcal{O}(I_2 \cdot (K^2 + I_1 \cdot K^3 \cdot I)) = \mathcal{O}(I_2 \cdot I_1 \cdot K^3 \cdot I)$, which is in polynomial time.

VI. Performance Evaluation

In this section, we set up simulations to evaluate our proposed algorithm *FedKD-FPB*. The simulations are conducted on a high-performance Dell tower station equipped

Algorithm 2: Algorithm for Joint Optimization of Wireless Power and Bandwidth Allocation

Initialize $t_k = t_k^*$;
Initialize iteration number $i = 0$;
Initialize maximum iteration number I_1;
while *Objective of* ***P3*** *does not converge and* $i \le I_1$ **do**
  $i = i + 1$;
  for *each* $k \in \mathcal{K}$ **do**
    Calculate drone k's minimum and maximum bandwidth $b_k^{\min}$ and $b_k^{\max}$ according to (35) and (36);
    Calculate drone k's bandwidth $\tilde{b}_k$ by solving (38) and (42);
    Calculate drone k's bandwidth b_k^* according to (43);
    Calculate drone k's wireless transmission power p_k^* according to (31);
    Update t_k^* according to (44);
  end
  Calculate the objective value of **P3**;
end

Algorithm 3: FedKD-FPB Algorithm

Initialize wireless power $p_k = p_k^*$;
Initialize wireless bandwidth $b_k = b_k^*$;
Initialize iteration number $i = 0$;
Initialize maximum iteration number I_2 ;
while *Objective of* ***P0*** *does not converge and* $i \le I_2$ **do**
  $i = i + 1$;
  Calculate CPU frequency f_k^* of all drones and global iteration time τ^* according to Algorithm 1;
  Calculate wireless power p_k^* and bandwidth b_k^* of all drones according to Algorithm 2 ;
  Calculate the objective value of **P0**;
end

with an Intel Xeon W-2245 CPU running at 3.90 GHz across 16 cores, an NVIDIA Quadro RTX 6000/8000 GPU, and 128 GB of RAM. For comparison, we utilize five existing algorithms. The *FedKD* algorithm [11] is a federated KD framework without any resource optimization. The *FedKD-F* algorithm [33] optimizes the CPU frequencies of teacher and student models within the FedKD framework. The *FedAvg* algorithm [7] represents the original FedAvg approach, using a single model for local training. The *FedAvg-F* algorithm optimizes CPU frequency within the FedAvg framework. Lastly, the *FedAvg-PB* algorithm [21] includes optimizations for wireless transmission power and bandwidth allocation within the FedAvg framework.

In our simulation, we have $K = 30$ drones hovering at a height of $H = 100$ m above a square region measuring 1000×1000 m. The BS is located at the center of this region, with drones uniformly distributed across the area. For the drone wireless channel, the environmental parameters α and β are set to 9.6 and 0.28, respectively. The speed

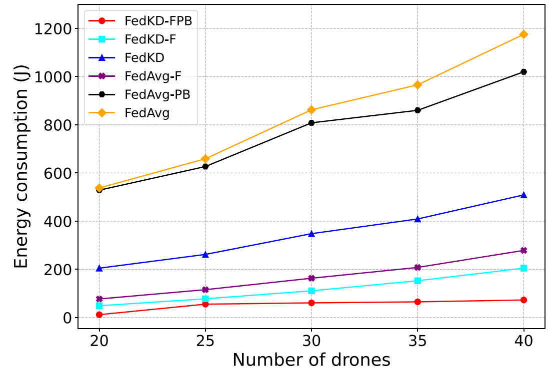

Fig. 2. Energy consumption versus number of drones.

of light c is 3×10^8 m/s, and the carrier frequency f_c is 2 GHz. The environment-related parameters ξ_{LoS} and ξ_{NLoS} are 1 and 20 dB, respectively. The noise power density N_0 is set to -114 dBm/Hz. The CPU switched capacitance κ is 10^{-28}. The minimum CPU frequency of each drone $f_k^{\min}$ is randomly selected from $\mathcal{U}(0.2, 1) \times 10$ GHz, while the maximum CPU frequency $f_k^{\max}$ is randomly selected from $\mathcal{U}(2.2, 3) \times 10$ GHz. The total bandwidth B is set to 20 MHz. The minimum and maximum wireless transmission powers $p_k^{\min}$ and $p_k^{\max}$ are 0.1 and 1 W, respectively. For implementing the FedKD framework, the teacher model consists of three convolutional layers and one linear layer, while the student model comprises two convolutional layers and one linear layer. These models are trained using the MNIST dataset [38]. Each drone has $D_k = 1100$ data samples. The required CPU cycles for the teacher model C_k^{ta} are randomly chosen from 2×10^5 to 4×10^5, and the student model's C_k^{stu} is one-third of C_k^{ta}. The student model size s_k is 28.1 Mb. The number of local iterations I_k is set to 1. The above parameters are consistent with [34], [35], [39]. Note that these values are default settings and may be adjusted to investigate their impacts on performance.

Fig. 2 compares the energy consumption performance across varying numbers of drones, ranging from 20 to 40. We observe that as the number of drones increases, the energy consumption of all algorithms also rises. This increase in energy consumption is due to the additional energy required for local drone training and wireless data transmission. Our proposed algorithm *FedKD-FPB* outperforms all other algorithms due to its efficient resource utilization. *FedKD-F* performs worse than *FedKD-FPB* because it does not optimize wireless transmission power and bandwidth. Similarly, *FedKD* consumes more energy than *FedKD-FPB* due to its lack of resource allocation optimization. *FedAvg-F*, *FedAvg-PB*, and *FedAvg* have higher energy consumption compared to *FedKD-FPB* because their larger model sizes of FedAvg lead to increased computation and wireless transmission energy consumption, and they lack resource optimization.

Fig. 3 illustrates the energy consumption of various algorithms with data sample sizes ranging from 1100 to 2700. As the number of data samples increases, more computation is required, resulting in higher energy consumption for all algorithms. Our proposed algorithm, *FedKD-FPB*, demonstrates the best performance due to the advantages of the

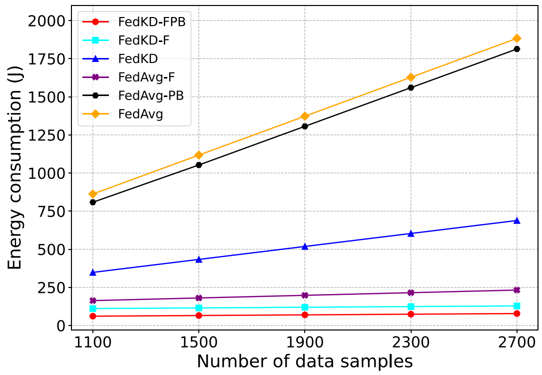

Fig. 3. Energy consumption versus number of data samples.

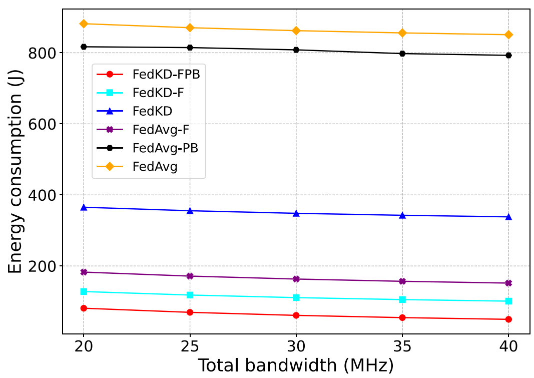

Fig. 4. Energy consumption versus total bandwidth.

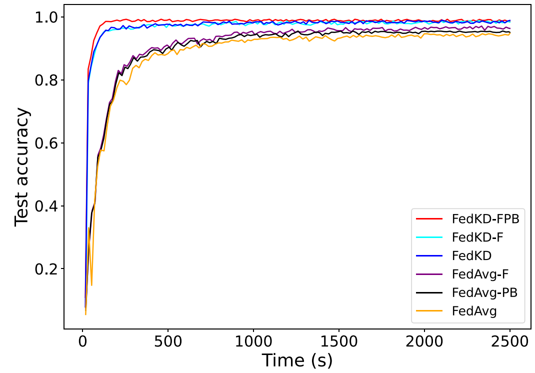

Fig. 5. Test accuracy versus training time for different algorithms.

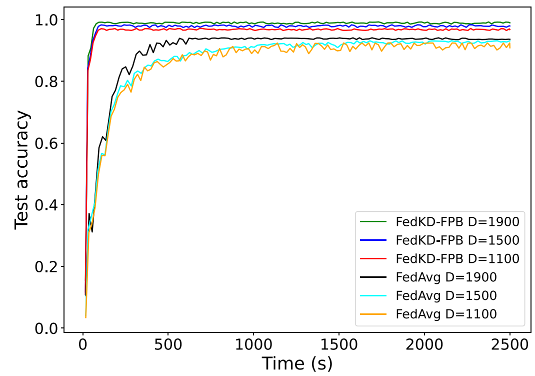

Fig. 6. Test accuracy versus training time with different data sample sizes.

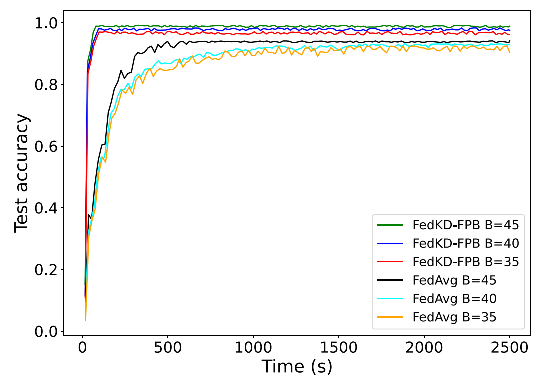

Fig. 7. Test accuracy versus training time with different total bandwidth.

FedKD framework and efficient resource optimization. *FedAvg* performs the worst, as it involves a larger local model and lacks optimized resource allocation. *FedKD-F* and *FedKD* perform worse than *FedKD-FPB* due to their inefficient resource management. *FedAvg-F* and *FedAvg-PB* have higher energy consumption because they utilize the FedAvg framework, which includes a larger local model.

Fig. 4 evaluates the performance of our proposed algorithm with varying total bandwidths ranging from 20 to 40. We observe that the energy consumption of all algorithms decreases as the available bandwidth increases. This is because a higher bandwidth allows for a greater data rate, reducing data transmission time and, consequently, the energy required for data transmission. Our proposed algorithm, *FedKD-FPB*, consistently consumes the least energy and performs the best among all the algorithms.

Fig. 5 illustrates the test accuracies of various algorithms over time. It can be observed that *FedKD-FPB*, *FedKD-F*, and *FedKD* consistently achieve higher accuracies compared to *FedAvg-F*, *FedAvg-PB*, and *FedAvg*. This performance improvement can be attributed to the advantages of the FedKD framework. Additionally, our proposed *FedKD-FPB* shows a slight edge over *FedKD-F* and *FedKD* in terms of both accuracy and convergence speed, due to more comprehensive resource optimization.

Fig. 6 depicts the accuracy performance of our proposed algorithm *FedKD-FPB* compared to *FedAvg* over time with different data sample sizes. As the data sample size increases, the learning performance of both *FedKD-FPB* and *FedAvg* improves, resulting in higher accuracy. We can also observe that *FedKD-FPB* outperforms *FedAvg* in both accuracy and convergence speed due to its resource optimization and smaller local model.

Fig. 7 illustrates the accuracy performance of *FedKD-FPB* and *FedAvg* over time with varying total bandwidths. Increasing the system bandwidth results in a higher wireless transmission rate, which leads to shorter learning times and improved training performance. Consequently, both *FedKD-FPB* and *FedAvg* achieve better accuracy with larger total bandwidth. Additionally, *FedKD-FPB* outperforms *FedAvg* due to the advantages provided by the FedKD framework.

Fig. 8 illustrates the accuracy performance of *FedKD-FPB* and *FedAvg* over time with both independent and identically

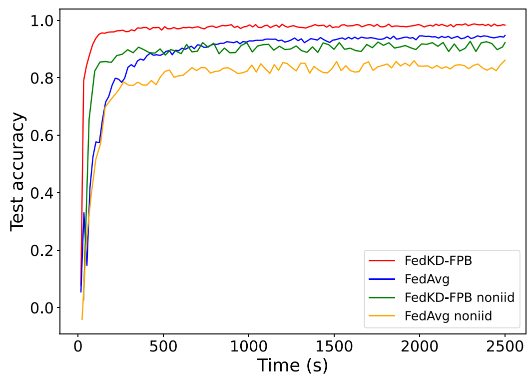

Fig. 8. Test accuracy versus training time with iid and non-iid data.

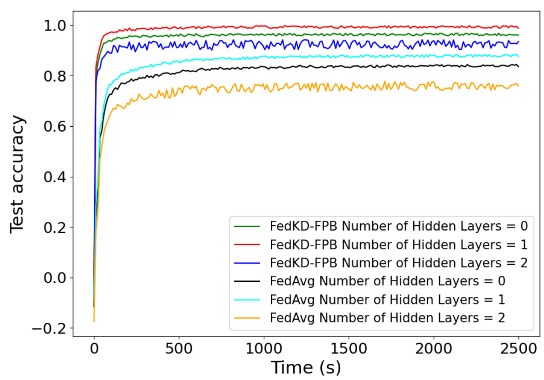

Fig. 9. Test accuracy versus training time with different number of hidden layers.

distributed (IID) and non-IID data. To simulate a non-IID data distribution, we utilize the Dirichlet distribution [40]. The distribution is controlled by a concentration parameter, which we set to 0.1, dictating the level of heterogeneity in the data distribution across all drones. Training with non-IID datasets leads to performance degradation compared to IID datasets due to the imbalance of local data samples. Consequently, both *FedKD-FPB* and *FedAvg* achieve better accuracy with IID datasets. Additionally, *FedKD-FPB* outperforms *FedAvg* due to the advantages provided by the FedKD framework.

Fig. 9 illustrates the test accuracy performance of our proposed algorithm with different numbers of hidden layers in the training model. The model is evaluated with 0, 1, and 2 hidden layers. It can be observed that the configuration with 1 hidden layer achieves the best accuracy for both *FedKD-FPB* and *FedAvg* because it strikes an optimal balance between complexity and generalization, avoiding underfitting of simpler models and overfitting of deeper ones. Moreover, *FedKD-FPB* performs better than *FedAvg* with all different numbers of hidden layers.

Fig. 10 shows the test accuracy performance of our proposed algorithm over time with varying numbers of neurons in the hidden layers of our training model. Specifically, the model is evaluated with 256, 512, and 1024 neurons. It can be observed that both *FedKD-FPB* and *FedAvg* with 512 neurons achieve the highest test accuracy and exhibit faster convergence compared to those with 256 or 1024 neurons. This is because 512 neurons provide an optimal balance between model capacity and generalization, avoiding underfitting with

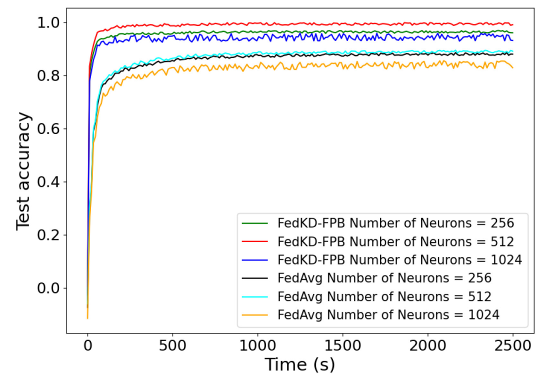

Fig. 10. Test accuracy versus training time with different number of neurons.

256 neurons and overfitting or increased training complexity with 1024 neurons. In addition, *FedKD-FPB* consistently achieves higher accuracy and faster convergence than *FedAvg* across all neuron configurations.

VII. Conclusion

In this article, we have investigated the resource allocation problem in FedKD within IoD networks. We have formulated the joint optimization problem of CPU computing resources, wireless transmission power, and bandwidth allocation with the objective of minimizing overall drone energy consumption while considering constraints on latency, computing resources, bandwidth, and power. We have designed a low-complexity algorithm to effectively address this optimization problem by iteratively solving subproblems. We have validated our approach through extensive simulations, demonstrating that our method improves energy efficiency and model accuracy compared to existing methods. In future work, we aim to incorporate drone mobility to enhance system adaptability in dynamic environments for real-world applications.

References

[1] J. Yao and N. Ansari, "Task allocation in fog-aided mobile IoT by Lyapunov online reinforcement learning," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 2, pp. 556–565, Jun. 2020.
[2] S. Al-Sarawi, M. Anbar, R. Abdullah, and A. B. Al Hawari, "Internet of Things market analysis forecasts, 2020–2030," in *Proc. 4th WorldS4*, 2020, pp. 449–453.
[3] J. Yao and N. Ansari, "QoS-aware machine learning task offloading and power control in Internet of Drones," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 6100–6110, Apr. 2023.
[4] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1622–1658, 3rd Quart., 2021.
[5] D. Manu, J. Yao, W. Liu, and X. Sun, "GraphGANFed: A federated generative framework for graph-structured molecules towards efficient drug discovery," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 21, no. 2, pp. 240–253, Mar./Apr. 2024.
[6] J. Yao and X. Sun, "Energy-efficient federated learning in Internet of Drones networks," in *Proc. IEEE 24th Int. Conf. HPSR*, 2023, pp. 185–190.
[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Stat.*, 2017, pp. 1273–1282.
[8] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, 2021.

[9] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Sci. China Technol. Sci.*, vol. 63, no. 10, pp. 1872–1897, 2020.
[10] X. Lian et al., "Persia: An open, hybrid system scaling deep learning-based recommenders up to 100 trillion parameters," in *Proc. 28th ACM SIGKDD KDD*, 2022, pp. 3288–3298.
[11] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nat. Commun.*, vol. 13, no. 1, p. 2032, 2022.
[12] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
[13] H. Seo, J. Park, S. Oh, M. Bennis, and S. L. Kim, "Federated knowledge distillation," in *Machine Learning and Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2022, pp. 457–485.
[14] Z. Zhao et al., "Towards efficient communications in federated learning: A contemporary survey," *J. Franklin Inst.*, vol. 360, no. 12, pp. 8669–8703, 2023.
[15] L. Abualigah, A. Diabat, P. Sumari, and A. H. Gandomi, "Applications, deployments, and integration of Internet of Drones (IoD): A review," *IEEE Sensors J.*, vol. 21, no. 22, pp. 25532–25546, Nov. 2021.
[16] J. V. Shirabayashi and L. B. Ruiz, "Toward UAV path planning problem optimization considering the Internet of Drones," *IEEE Access*, vol. 11, pp. 136825–136854, 2023.
[17] G. Grieco, G. Iacovelli, P. Boccadoro, and L. A. Grieco, "Internet of drones simulator: Design, implementation, and performance evaluation," *IEEE Internet Things J.*, vol. 10, no. 2, pp. 1476–1498, Jan. 2023.
[18] J. Yao and N. Ansari, "Wireless power and energy harvesting control in IoD by deep reinforcement learning," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 2, pp. 980–989, Jun. 2021.
[19] 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.
[20] X. Mo and J. Xu, "Energy-efficient federated edge learning with joint communication and computation design," *J. Commun. Netw.*, vol. 6, no. 2, pp. 110–124, Jun. 2021.
[21] 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.
[22] 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.
[23] Z. Zhang et al., "Robust semisupervised federated learning for images automatic recognition in Internet of Drones," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 5733–5746, Apr. 2023.
[24] A. V. Shvetsov et al., "Federated learning meets intelligence reflection surface in drones for enabling 6G networks: Challenges and opportunities," *IEEE Access*, vol. 11, pp. 130860–130887, 2023.
[25] Y. Shen, Y. Qu, C. Dong, F. Zhou, and Q. Wu, "Joint training and resource allocation optimization for federated learning in UAV swarm," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2272–2284, Feb. 2023.
[26] J. Yao and N. Ansari, "Secure federated learning by power control for Internet of Drones," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 4, pp. 1021–1031, Dec. 2021.
[27] Q.-V. Pham, M. Le, T. Huynh-The, Z. Han, and W.-J. Hwang, "Energy-efficient federated learning over UAV-enabled wireless powered communications," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4977–4990, May 2022.
[28] T. Wang, X. Huang, Y. Wu, L. Qian, B. Lin, and Z. Su, "UAV swarm-assisted two-tier hierarchical federated learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 1, pp. 943–956, Jan./Feb. 2024.
[29] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," 2019, *arXiv:1910.03581*.
[30] H. Wen, Y. Wu, J. Hu, Z. Wang, H. Duan, and G. Min, "Communication-efficient federated learning on non-IID data using two-step knowledge distillation," *IEEE Internet Things J.*, vol. 10, no. 19, pp. 17307–17322, Oct. 2023.
[31] Z. Chen, P. Tian, W. Liao, X. Chen, G. Xu, and W. Yu, "Resource-aware knowledge distillation for federated learning," *IEEE Trans. Emerg. Topics Comput.*, vol. 11, no. 3, pp. 706–719, Jul.–Sep. 2023.
[32] G. Gad, A. Farrag, A. Aboulfotouh, K. Bedda, Z. M. Fadlullah, and M. M. Fouda, "Joint self-organizing maps and knowledge-distillation-based communication-efficient federated learning for resource-constrained UAV-IoT systems," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 15504–15522, May 2024.
[33] S. Cal, X. Sun, and J. Yao, "Energy-efficient federated knowledge distillation learning in Internet of Drones," in *Proc. IEEE ICC Workshops*, 2024, pp. 1256–1261.
[34] L. Yu, R. Albelaihi, X. Sun, N. Ansari, and M. Devetsikiotis, "Jointly optimizing client selection and resource management in wireless federated learning for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4385–4395, Mar. 2022.
[35] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *Proc. IEEE GLOBECOM*, 2015, pp. 1–6.
[36] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
[37] N. Eagan, G. Hauser, and T. Flaherty, *Newton's Method on a System of Nonlinear Equations*, Carnegie Mellon Univ., Pittsburgh, PA, USA, 2014.
[38] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[39] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.
[40] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *Proc. 36th ICML*, 2019, pp. 7252–7261.

Jingjing Yao (Member, IEEE) received the Ph.D. degree in computer engineering from New Jersey Institute of Technology, Newark, NJ, USA, in 2021.

She is an Assistant Professor with the Department of Computer Science, Texas Tech University, Lubbock, TX, USA. Her research interests include IoT, applied AI in wireless communication and networking, drone-assisted networks, and resource optimization.

Semih Cal is currently pursuing the Ph.D. degree with the Department of Computer Science, Texas Tech University, Lubbock, TX, USA.

His research interests include wireless communication networks, energy-efficient algorithms, and federated learning for IoT and drone networks.

Xiang Sun (Member, IEEE) received the Ph.D. degree in electrical engineering from New Jersey Institute of Technology, Newark, NJ, USA, in 2018.

He is an Assistant Professor with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA. His research interests span a variety of topics, including free space optics, wireless networks, IoT, edge computing, UAV swarm control, and drug discovery. Additionally, he focuses on developing innovative methodologies in deep reinforcement learning, federated learning, generative adversarial networks, and nonconvex optimization to address diverse research challenges.

Dr. Sun is an Associate Editor of the IEEE OPEN JOURNAL OF THE COMPUTER SOCIETY and *Digital Communications and Networks*.