Secure Federated Learning by Power Control for Internet of Drones

Jingjing Yao, *Student Member, IEEE*, and Nirwan Ansari, *Fellow, IEEE*

***Abstract*—Fog-aided Internet of Drones (IoD), where massive training data are collected by drones and analyzed in the fog node, can leverage machine learning to provision various services. Aggregating all data in the fog node may incur huge network traffic and drone data privacy leakage. Federated learning (FL) is hence proposed to preserve drone data privacy by performing local training in drones and sharing training model parameters in the fog node without uploading drone raw data. However, drone privacy can still be divulged to ground eavesdroppers by wiretapping and analyzing uploaded parameters during the FL training process. In this paper, we investigate the power control of all drones to maximize the FL system security rate constrained by drone battery capacities and the quality of service (QoS) requirement (i.e., FL training time). We formulate this problem as a non-linear programming problem and design an algorithm to obtain the optimum solutions with a low computational complexity. Extensive simulations are conducted to demonstrate the performance of our proposed algorithm.**

***Index Terms*—Federated learning, Internet of Drones (IoD), security, fog computing, power control, energy consumption, quality of service (QoS).**

I. Introduction

INTERNET of Drones (IoD) utilizes drones as the Internet of Things (IoT) devices to provision services such as traffic surveillance, object tracking and disaster rescue [1], [2]. In IoD networks, multiple drones are deployed to collect information (e.g., images and videos) and send them to the IoD gateway for further processing. Fog-aided IoD networks provide fast service response by equipping the IoD gateway with a fog node where data can be analyzed and processed instead of being sent to the remote cloud [3], [4]. With the rapid development of machine learning technologies, fog-aided IoD becomes a promising architecture to provide novel services such as virtual reality, traffic prediction and object recognition [5].

To enable machine learning services in fog-aided IoD networks, the training data collected by drones are all sent to the fog node to train the machine learning models (e.g., traffic prediction and object recognition models) [6]. However, massive data transmission injects huge network traffic and degrades the quality of service (QoS) because of wireless bandwidth limitations. On the other hand, the data collected by drones may be sensitive and contain private information (e.g., military areas and human faces). Hence, aggregating all data in the fog node may pose the risk of privacy leakage [7].

Federated learning (FL) [8] is proposed to address the challenges of both the bandwidth limitation and privacy leakage in fog-aided IoD networks. Instead of sending the training data to the fog node in the conventional machine learning services, FL enables local training at each drone on its own data and then shares machine learning model parameters with the fog node. In this way, FL learns a shared global model in the fog node by aggregating the local model parameters derived from distributed drone data in a privacy preserved manner. Much wireless bandwidth can also be saved by avoiding the massive wireless data transmission [9].

The FL performance is usually determined by the FL training time which is composed of the local computation time for model training and the wireless transmission time for transmitting the local model parameters [10]. Hence, the FL performance depends on the drone computing resources and the wireless channels between drones and the fog node. There is a tradeoff between the FL training time and drone energy consumption [11]. To reduce the FL training time, more energy is required to reduce the computation time and wireless data transmission time. Therefore, the FL performance is also limited by the drone batteries, which are used for local training computation, wireless data transmission, and drone hovering in the air. Drone wireless transmission power determines the wireless transmission time and energy consumption for wireless data transmission, and hence is an important factor to be investigated in order to improve the FL performance [9].

As compared with conventional machine learning technologies, FL alleviates the privacy concern by local training. However, security concerns still exist because of data eavesdropping. The ground eavesdroppers may wiretap the local model parameters when drones upload them to the fog node [12]. If the uploaded model parameters are inferred by a malicious eavesdropper, they may leak the private information by model inversion attack [12]. It is hence critical to improve the security of FL communications. Security rate is a key metric to measure the security level of wireless communications, and it is defined as the rate of reliably transmitted information without eavesdropping (i.e., the difference of the data rate between a drone to the fog node and that between the drone and an eavesdropper) [13]. Wireless power control, which

Manuscript received November 25, 2020; revised March 10, 2021; accepted April 21, 2021. Date of publication April 28, 2021; date of current version December 9, 2021. This work was supported in part by the U.S. National Science Foundation under Grants No. CNS-1814748 and No. CNS-1647170. The associate editor coordinating the review of this article and approving it for publication was D. Niyato. *(Corresponding author: Jingjing Yao.)*

The authors are with the Advanced Networking Laboratory, Helen and John C. Hartmann Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: jy363@njit.edu; nirwan.ansari@njit.edu).

Digital Object Identifier 10.1109/TCCN.2021.3076167

determines the data rates to the fog node and an eavesdropper, is an option to improve the FL system security rate [14].

Utilizing power control to secure FL in IoD networks has not been readily addressed, and we hence investigate the power control problem for FL in fog-aided IoD networks to exploit the tradeoff among FL system security rate, FL training time, and drone energy consumption. Specifically, we optimize each drone's wireless transmission power to maximize the system security rate constrained by the FL training time requirement and each drone's battery capacity. We theoretically demonstrate the FL convergence bounds, based on which we formulate our power control problem and design an algorithm to solve it. The major contributions of our work are summarized below.

- We investigate the power control problem for FL in a fog-aided IoD network to counteract eavesdroppers.
- We theoretically demonstrate the convergence property of FL in the IoD network.
- We formulate the power control problem for secure FL in IoD as a non-linear programming model to maximize the system security rate constrained by the drone battery capacities and FL time requirement.
- We design the Power Control in Secure FL (PCSF) algorithm to solve the power control problem in our work.
- We demonstrate that our proposed algorithm performs better than the existing works by extensive simulations.

The rest of this paper is organized as follows. The related works are presented in Section II. In Section III, we describe the architecture of FL in IoD networks, the FL process, the FL convergence analysis, the FL training time, drone energy consumption, and the system security rate. We formulate the power control problem of FL in IoD networks in Section IV. We then design an algorithm to solve the problem in Section V. The performance of our proposed algorithm is evaluated in Section VI. This paper is finally concluded in Section VII.

II. Related Works

Fog-aided IoD networks have been investigated to provision services such as object tracking, traffic surveillance, and disaster rescue [15]. Gharibi *et al.* [16] proposed an IoD system to provide navigation services and described how to implement the IoD system. Motlagh *et al.* [17] surveyed various applications, the implementation, and challenges of IoD networks. Wazid *et al.* [18] proposed a user authentication scheme to access the data from drones in IoD networks. However, they do not consider utilizing FL in IoD networks.

Machine learning imparts intelligence into IoT networks by analyzing the data, which are collected by all IoT devices, in the fog node. Meidan *et al.* [19] collected the network traffic data from IoT devices to train a classification model to distinguish the traffic generated by IoT and non-IoT devices. Yao and Ansari [20] constructed a deep reinforcement learning model for the content placement problem in dynamic cache-enabled IoT networks. They also utilized a deep reinforcement learning model to optimize the wireless power control in energy harvesting aided time-varying IoD networks to minimize the average system energy cost [21].

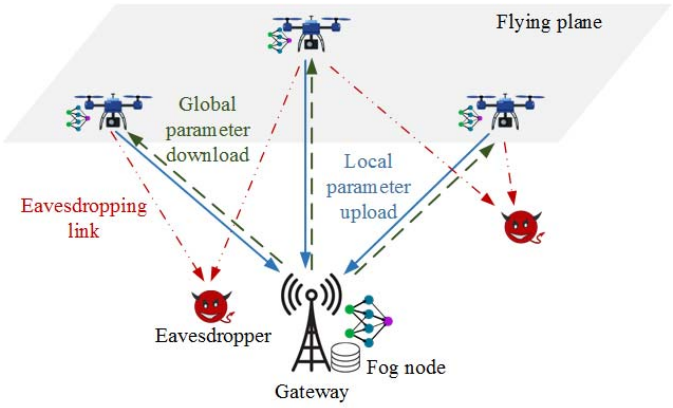


Fig. 1. Federated learning in a fog-aided IoD network with eavesdroppers.

FL has been investigated in wireless networks. Tran *et al.* [22] formulated FL over wireless networks as an optimization problem to balance the tradeoff of the FL learning time, accuracy level, and energy cost. Wang *et al.* [23] designed an intelligent framework to implement an FL system by utilizing the collaboration among devices and edge nodes and exchanging the learning model parameters. Yang *et al.* [24] formulated an optimization problem to minimize a weighted sum of the FL completion time, local computation energy, and transmission energy for FL in wireless communication networks. However, the above works do not consider the security issue nor the implementation of FL in IoD networks.

The security issue of FL has been studied in several works. Song *et al.* [7] explored the user-level privacy leakage in FL and proposed a multi-task generative adversarial network (GAN) framework to identify the anonymized updates of the clients. Xu *et al.* [25] proposed a secure federated training protocol to verify the correctness of results returned from the global aggregator while protecting user data privacy. Wei *et al.* [26] proposed a differential privacy based framework, which adds artificial noise to the uploaded model parameters, to prevent information leakage in FL.

Utilizing power control to alleviate the FL's privacy leakage, which is caused by the ground eavesdroppers during the learning parameter uploading in IoD networks, has not been investigated yet. To fill this gap, we optimize the drone wireless transmission powers to maximize the FL system security rate with the consideration of the QoS requirement (i.e., FL training time) and drone battery capacities.

III. System Model

In a fog-aided IoD network (Fig. 1), N drones are hovering in the air in the flying plane to collect local data samples and provide the FL service in concert with the fog node to IoD users. We denote $\mathcal{N} = \{1, \ldots, N\}$ as a set of indexes for indexing drones. The aim of the FL service is to obtain a global machine learning model (e.g., traffic prediction and object recognition). In FL, each drone iteratively downloads the global FL model parameter, updates the parameter with its own local data by local training, and sends it back to the

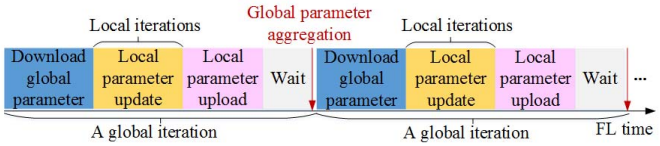


Fig. 2. Federated learning process.

fog node, while the fog node iteratively gathers all updated local parameters and aggregates them to a new global model. The local training is based on the local data samples $\mathcal{D}_n = \{(x_k, y_k)\}$ where x_k is sample k's input (e.g., image pixels) and y_k is the output (e.g., label of the image). A loss function $f_k(w)$ is defined to measure the error of the local model based on data sample k, where w is the parameter of the local model. Then, drone n's local training process is to minimize the local loss function [11]

$$F_n(w) = \frac{1}{|\mathcal{D}_n|} \sum_{k \in \mathcal{D}_n} f_k(w), \ \forall n \in \mathcal{N}, \tag{1}$$

where $|\mathcal{D}_n|$ is the number of data samples. For simplicity, we define $D_n = |\mathcal{D}_n|$ thereafter. Common examples of loss function $f_k(w)$ include $f_k(w) = \frac{1}{2}\|x_k^T w - y_k\|^2$ for linear regression and $f_k(w) = \{0, 1 - y_k x_k^T w\}, y_k \in \{-1, 1\}$ for support vector machine [27].

Note that M eavesdroppers on the ground aim to steal information from the drones. We denote $\mathcal{M} = \{1, \ldots, M\}$ as the set of indexes for indexing eavesdroppers. Locations of all eavesdroppers are assumed known, and they can be detected by the leaked power of their radio frequency (RF) front ends [28]. All drones adjust their wireless transmission powers to reduce the possibility of information leakage of the local model parameters.

A. Federated Learning Process

There are global FL iterations and local FL iterations in the FL process (Fig. 2). In a global iteration, each drone downloads the global parameter from the fog node, trains the model with its local data, and sends the updated local parameter to the fog node. The fog node finally aggregates all updated local parameters into a new global model parameter. The local model parameters are updated by the gradient descent algorithm [29]. In each local iteration, the local parameter is updated according to the gradient of the loss function and learning rate. The relationship between the global iteration and local iteration is shown in Fig. 2.

The objective of FL is to minimize the global loss function $F(w)$. In the t-th global iteration, all the drones first download the global parameter w^t from the fog node, and calculate the gradients of their local loss function $\nabla F_n(w^t)$. Then, the fog node collects all local gradients and calculates the gradient of the global loss function

$$\nabla F(w^t) = \frac{1}{N} \sum_{n \in \mathcal{N}} \nabla F_n(w^t), \tag{2}$$

which is broadcast to all the drones for local trainings.

Each drone n solves the local training problem

$$\min_w \ G_n^t(w) = F_n(w) - \left[\nabla F_n(w^t) - \eta \nabla F(w^t)\right]^\top w, \tag{3}$$

where $G_n^t(w)$ is the modified loss function of drone n in the t-th global iteration, and η is a positive constant to control the FL convergence rate [29]. The local training problem is solved by the gradient descent algorithm. We define $w_n^{t,i}$ as drone n's local model parameter at global iteration t and local iteration i, and $w_n^{t,*}$ as the local model parameter after convergence, i.e.,

$$w_n^{t,*} = \operatorname*{argmin}_w G_n^t(w). \tag{4}$$

In the i-th local iteration, the local model parameter $w_n^{t,i}$ is updated according to the gradient of $G_n^t(w)$ and the learning rate δ, i.e.,

$$w_n^{t,i+1} = w_n^{t,i} - \delta \nabla G_n^t\left(w_n^{t,i}\right), \tag{5}$$

where $w_n^{t,0} = w^t$ because it is downloaded from the fog node. According to Eq. (3), $\nabla G_n^t(w_n^{t,i})$ can be calculated as

$$\nabla G_n^t\left(w_n^{t,i}\right) = \nabla F_n\left(w_n^{t,i}\right) - \nabla F_n(w^t) + \eta \nabla F(w^t). \tag{6}$$

Since $w_n^{t,*}$ is the converged local model parameter, we have

$$\nabla G_n^t(w_n^{t,*}) = \nabla F_n(w_n^{t,*}) - \nabla F_n(w^t) + \eta \nabla F(w^t) = 0, \tag{7}$$

The local iteration continues until the local model accuracy ϵ_l is reached, which is defined as

$$G_n^t\left(w_n^{t,i}\right) - G_n^t(w_n^{t,*}) \le \epsilon_l \left[G_n^t(w^t) - G_n^t(w_n^{t,*})\right]. \tag{8}$$

After all local model parameters $w_n^{t,*}$ are collected, the fog node aggregates all of them into a new global model parameter w^{t+1}, i.e.,

$$w^{t+1} = \frac{1}{N} \sum_{n \in \mathcal{N}} w_n^{t,*}, \tag{9}$$

The global iteration continues until the global model accuracy ϵ_g is reached, which is defined as

$$F(w^t) - F(w^*) \le \epsilon_g \left[F\left(w^0\right) - F(w^*)\right]. \tag{10}$$

B. Federated Learning Convergence Analysis

It is generally impossible to know the exact number of FL iterations, and hence we utilize the convergence bounds to approximate both the local FL iterations and global FL iterations [11]. To analyze the convergence rate of FL, the local loss function $F_n(w)$ of each drone n follows the following assumptions [29]:

1) $F_n(w)$ is α-strongly convex,
2) $F_n(w)$ is β-smooth.

Assumption 1 implies that [30]

$$\alpha \|w - w'\| \le \|\nabla F_n(w) - \nabla F_n(w')\|, \ \forall w, w', \tag{11}$$

and

$$F_n(w) - F_n(w') - \nabla F_n(w)^\top (w - w') \le -\frac{\alpha}{2}\|w - w'\|^2 \tag{12}$$

where $\|x\|$ denotes the 2-norm of matrix x and $x^\top$ is the transpose of matrix x.

Assumption 2 implies that [30]

$$\left\|\nabla F_n(w) - \nabla F_n\left(w'\right)\right\| \leq \beta \left\|w - w'\right\|, \ \forall w, w', \quad (13)$$

and

$$F_n(w) - F_n\left(w'\right) - \nabla F_n\left(w'\right)^\top \left(w - w'\right) \leq \frac{\beta}{2}\left\|w - w'\right\|^2. \quad (14)$$

Lemma 1: $F(w)$ is α-strongly convex and β-smooth.

Proof: According to Eq. (12), we can derive $\frac{1}{N}\sum_{n\in\mathcal{N}} F_n(w) - \frac{1}{N}\sum_{n\in\mathcal{N}} F_n(w') - \frac{1}{N}\sum_{n\in\mathcal{N}} \nabla F_n(w)^\top (w - w') \leq \frac{1}{N}\sum_{n\in\mathcal{N}}\{-\frac{\alpha}{2}\|w - w'\|^2\}$. Combined with Eq. (2), we have $F(w) - F(w') - \nabla F(w)^\top (w - w') \leq -\frac{\alpha}{2}\|w - w'\|^2$, which proves that $F(w)$ is α-strongly convex. Similarly, by combining Eq. (14) and Eq. (2), we can prove that $F(w)$ is β-smooth. ■

Lemma 2: If both $F(w)$ and $F_n(w)$ are α-strongly convex, the following inequations hold:

$$\left\|\nabla F\left(w^t\right)\right\|^2 \geq \alpha\left[F\left(w^t\right) - F(w^*)\right], \ \forall t, \quad (15)$$

and

$$\left\|\nabla G_n^t\left(w_n^{t,i}\right)\right\|^2 \geq \alpha\left[G_n^t\left(w_n^{t,i}\right) - G_n^t\left(w_n^{t,*}\right)\right], \ \forall i. \quad (16)$$

Proof: See Appendix A. ■

Lemma 3: Local FL problem (4) of drone n with the local accuracy ϵ_l can be solved by the gradient descend method after $I = \frac{2}{(2-\delta\beta)\delta\alpha}\ln(\frac{1}{\epsilon_l})$ iterations, if the local learning rate $\delta < \frac{2}{\beta}$.

Proof: See Appendix B. ■

Lemma 4: The global FL algorithm converges after $T = \frac{2\beta^2}{(2\alpha-\beta\eta)\alpha\eta}\ln(\frac{1}{\epsilon_g})$ iterations, if $\eta \in (0, \frac{\alpha}{\beta})$.

Proof: See Appendix C. ■

C. Drone Data Transmission Rate

A drone's data transmission rate depends on the air-to-ground channel between the drone and the fog node. We adopt the widely used probability model where the signal between the drone and the fog node is either Line-of-Sight (LoS) with probability Pr^{LoS} or Non-Line-of-Sight (NLoS) with probability Pr^{NLoS} [31]. The probabilities of LoS and NLoS signals are $Pr^{LoS} = \frac{1}{1+a\exp(-b[\frac{180}{\pi}\arcsin(\frac{H}{d})-a])}$ and $Pr^{NLoS} = 1 - Pr^{LoS}$, where a and b are environment-related (e.g., rural and urban) constants, H is the flying height, and d is the distance between the drone and the fog node. The path losses for LoS and NLoS signals are $PL^{LoS} = 20\log_{10}(\frac{4\pi f_c d}{c}) + \psi^{LoS}$ and $PL^{NLoS} = 20\log_{10}(\frac{4\pi f_c d}{c}) + \psi^{NLoS}$, where ψ^{LoS} and ψ^{NLoS} are environment-related constants, f_c is the carrier frequency, and c is the speed of light. Therefore, the path loss between the drone and the fog node can be calculated as $PL = Pr^{LoS}PL^{LoS} + Pr^{NLoS}PL^{NLoS}$. The wireless channel gain between drone n and the fog node is $G_n^D = 10^{-\frac{PL}{10}}$. Hence, drone n's wireless data transmission rate to the fog node can be calculated as

$$r_n = B_n \log_2\left(1 + \frac{p_n G_n^D}{N_0 B_n}\right), \quad (17)$$

where B_n is the allocated bandwidth to drone n, p_n is drone n's wireless transmission power, and N_0 is the noise power spectrum density. Similarly, we can calculate the wireless data transmission rate from drone n to eavesdropper m (i.e., eavesdropping rate):

$$\pi_{n,m} = B_n \log_2\left(1 + \frac{p_n G_{n,m}^E}{N_0 B_n}\right), \quad (18)$$

where $G_{n,m}^E$ is the wireless channel gain between drone n and eavesdropper m.

D. Security Rate

We utilize the security rate to measure the system security level, which is defined as the difference between the drone data transmission rate and the maximum eavesdropping rate [32]. Hence, drone n's security rate is

$$R_n^{SEC} = \left[r_n - \max_{\forall m\in\mathcal{M}} \pi_{n,m}\right]^+, \quad (19)$$

where $[x]^+ \triangleq \max\{x, 0\}$, r_n is drone n's data transmission rate, and $\pi_{n,m}$ is the eavesdropping rate from drone n to eavesdropper m. Note that we intend to maximize the security rates of all drones, and we hence define the system security rate R^{SEC} as the summation of all drones' security rates, i.e.,

$$R^{SEC} = \sum_{n\in\mathcal{N}} R_n^{SEC} = \sum_{n\in\mathcal{N}}\left[r_n - \max_{\forall m\in\mathcal{M}} \pi_{n,m}\right]^+. \quad (20)$$

E. Federated Learning Training Time

The FL time in each global iteration consists of both the local computation time for local training and the wireless transmission time to transmit the updated local parameters. Note that we neglect the global parameter download time because it is usually very small. We denote the number of CPU cycles to process one data sample of drone n as C_n, which can be measured offline [33]. The number of drone n's data samples is denoted as D_n. Hence, the number of CPU cycles for a local iteration is $C_n D_n$. Drone n's local computation time for one local iteration can then be calculated as $\frac{C_n D_n}{f_n}$, where f_n is the CPU computation capacity in CPU cycles per second [33]. Hence, drone n's local computation time is

$$\tau_n^c = I\frac{C_n D_n}{f_n} = \frac{2\ln(1/\epsilon_l)}{(2-\delta\beta)\delta\alpha}\frac{C_n D_n}{f_n}. \quad (21)$$

Each drone uploads the updated local parameter to the fog node, and the wireless data transmission time for uploading parameters of drone n can be calculated as $\tau_n^w = \frac{s_n}{r_n}$. Note that the global model parameters can only be aggregated until all local model parameters are received in a global iteration. The duration of a global iteration is hence determined by the

longest local FL time among all drones. Hence, the FL time of a global iteration can be calculated as

$$\tau^l = \max_{n\in\mathcal{N}}\{\tau_n^c + \tau_n^w\} = \max_{n\in\mathcal{N}}\left\{\tau_n^c + \frac{s_n}{B_n\log_2\left(1+\frac{p_n G_n^D}{N_0 B_n}\right)}\right\}. \quad (22)$$

In summary, the total FL time of all global iterations is

$$\tau = T\tau^l = T\max_{n\in\mathcal{N}}\left\{\tau_n^c + \frac{s_n}{B_n\log_2\left(1+\frac{p_n G_n^D}{N_0 B_n}\right)}\right\}. \quad (23)$$

F. Drone Energy Consumption

The drone's energy is consumed for local model training, wireless data transmission, and hovering in the air.

1) Local Computation: We utilize the widely used energy consumption model which assumes that drone *n*'s energy consumption for processing a single CPU cycle is γf_n^2, where γ is a constant related to the switched capacitance [34]. Then, drone *n*'s energy consumption for local computation in each global iteration is

$$E_n^c = IC_nD_n\gamma f_n^2 = \frac{2\ln(1/\epsilon_l)}{(2-\delta\beta)\delta\alpha}C_nD_n\gamma f_n^2, \quad (24)$$

where C_nD_n is the total number of CPU cycles for one local iteration, and *I* is the number of local iterations.

2) Wireless Data Transmission: Drone *n*'s energy consumption for uploading the updated local model parameter can be calculated as

$$E_n^w = p_n\tau_n^w = \frac{p_n s_n}{r_n} = \frac{p_n s_n}{B_n\log_2\left(1+\frac{p_n G_n^D}{N_0 B_n}\right)}. \quad (25)$$

3) Drone Hovering Energy: The energy consumed for hovering is used for the drone to remain stationary in the air. The drone's generated hovering power is expressed as [35]

$$P^{hov} = \sqrt{\frac{(mg)^3}{2\pi r_p^2 n_p \rho}}, \quad (26)$$

where *m* is the drone's weight, *g* is the earth gravitational acceleration, r_p is the radius of the propellers, n_p is the number of propellers, and ρ is the air density. We assume that these parameters of all drones are the same. Drone *n*'s hovering time τ^l in each global iteration depends on the longest local FL time among all drones. Hence, drone *n*'s hovering energy can be calculated as

$$E_n^{hov} = P^{hov}\tau^l = P^{hov}\max_{n\in\mathcal{N}}\{\tau_n^c + \tau_n^w\}. \quad (27)$$

In summary, the total energy consumption of all drones is

$$E_n = T(E_n^w + E_n^c + E_n^{hov}) = T\frac{p_n s_n}{r_n} + TE_n^c + P^{hov}\tau. \quad (28)$$

IV. Problem Formulation

We formulate the power control problem for secure FL in a fog-aided IoD network that maximizes the system security rate constrained by the QoS requirement and battery constraint, as problem **P0**.

$$\mathbf{P0}: \max_{\boldsymbol{p}} \sum_{n\in\mathcal{N}}\left[r_n - \max_{\forall m\in\mathcal{M}}\pi_{n,m}\right]^+ \quad (29)$$

$$s.t.\ \ 0 \le p_n \le P_n^m,\ \forall n\in\mathcal{N}, \quad (30)$$

$$\tau \le Q^{th}, \quad (31)$$

$$E_n \le B_n^{max},\ \forall n\in\mathcal{N}, \quad (32)$$

where τ and E_n are defined in Eq. (23) and Eq. (28), respectively. The objective in Eq. (29) is to maximize the system security rate. Equation (30) imposes the wireless transmission power to be positive and less than the maximum value P_n^m. Equation (31) is the QoS requirement for the FL service which imposes the FL time not to surpass the requirement Q^{th}. Equation (32) implies that drone *n*'s energy consumption should be less than its battery capacity B_n^{max}. It is challenging to solve problem **P0** because of its non-convexity.

To simplify constraint Eq. (31), we combine it with Eq. (22) and (23), and we have $T\max_{n\in\mathcal{N}}\{\tau_n^c + \frac{s_n}{B_n\log_2(1+\frac{p_nG_n^D}{N_0B_n})}\} \le Q^{th}$, which can be transformed into $T(\tau_n^c + \frac{s_n}{B_n\log_2(1+\frac{p_nG_n^D}{N_0B_n})}) \le Q^{th},\ \forall n\in\mathcal{N}$. Hence, the lower bound of drone *n*'s wireless transmission power p_n can be calculated as $p_n \ge \frac{N_0B_n}{G_n^D}[2^{\frac{s_n}{B_n(\frac{Q^{th}}{T}-\tau_n^c)}} - 1]$. We denote $\tilde{p}_n = \frac{N_0B_n}{G_n^D}[2^{\frac{s_n}{B_n(\frac{Q^{th}}{T}-\tau_n^c)}} - 1]$ for simplicity. Then, p_n satisfies

$$p_n \ge \tilde{p}_n. \quad (33)$$

To simplify the objective function Eq. (29), we have

$$\sum_{n\in\mathcal{N}}\left[r_n - \max_{\forall m\in\mathcal{M}}\pi_{n,m}\right]^+ = \sum_{n\in\mathcal{N}} B_n\log_2\left(\frac{1+\frac{p_nG_n^D}{N_0B_n}}{1+\frac{p_n\max_{\forall m\in\mathcal{M}}G_{n,m}^E}{N_0B_n}}\right), \quad \text{if } G_n^D \ge \max_{\forall m\in\mathcal{M}} G_{n,m}^E. \quad (34)$$

We denote $\gamma_n = \frac{G_n^D}{N_0B_n}$, $\gamma_n' = \frac{\max_{\forall m\in\mathcal{M}}G_{n,m}^E}{N_0B_n}$, and $\mathcal{N}' = \{n | n\in\mathcal{N}, G_n^D \ge \max_{\forall m\in\mathcal{M}} G_{n,m}^E\}$. Then, the objective becomes

$$\sum_{n\in\mathcal{N}}\left[r_n - \max_{\forall m\in\mathcal{M}}\pi_{n,m}\right]^+ = \sum_{n\in\mathcal{N}'} B_n\log_2\left(\frac{1+\gamma_n p_n}{1+\gamma_n' p_n}\right) \quad (35)$$

Problem **P0** can then be transformed into problem **P1**:

$$\mathbf{P1}: \max_{p_n} \sum_{n\in\mathcal{N}'} B_n\log_2\left(\frac{1+\gamma_n p_n}{1+\gamma_n' p_n}\right) \quad (36)$$

$$s.t.\ \ \tilde{p}_n \le p_n \le P_n^m,\ \forall n\in\mathcal{N}, \quad (37)$$

$$T\frac{p_n s_n}{B_n \log_2(1+\gamma_n p_n)} + TE_n^c + P^{hov}\tau \le B_n^{max},\ \forall n \in \mathcal{N}, \quad (38)$$

$$\tau = T \max_{n\in\mathcal{N}}\left\{\tau_n^c + \frac{s_n}{B_n \log_2(1+\gamma_n p_n)}\right\}, \quad (39)$$

where T is the number of FL global iterations defined in Lemma 4, E_n^c, defined in Eq. (24), is drone n's energy consumption for computation in each global iteration, and τ_n^c, defined in Eq. (21), is drone n's computation time in each FL global iteration. It is still difficult to solve problem **P1** because of its non-convexity. Approaches such as exhaustive search and branch-and-bound are computationally expensive. We hence design an algorithm to tackle this problem with a much lower computational complexity in the next section.

V. Problem Solution

We propose the Power Control in Secure FL (PCSF) algorithm in this section to solve problem **P0**. The basic idea of PCSF is to enumerate each possible FL time, optimize all drones' wireless transmission powers, and then choose the best FL time and its corresponding power control policy which achieves the largest system security rate.

A. Subproblem Transformation

Note that the difficulty of problem **P1** lies in the total FL time τ which couples all p_n together. In order to solve this challenge, we propose to enumerate each $\tau = T\tau_j^c + \frac{Ts_j}{B_j \log_2(1+\gamma_j p_j)}, \forall j \in \mathcal{N}$ and then compare all derived objective values by different τ. Since τ is determined, all drones' p_n are independent. It can be observed that $B_n \log_2(\frac{1+\gamma_n p_n}{1+\gamma_n' p_n})$ is an increasing function with regard to p_n when $G_n^D \ge \max_{\forall m\in\mathcal{M}} G_{n,m}^E$. Then, maximizing the summation of $B_n \log_2(\frac{1+\gamma_n p_n}{1+\gamma_n' p_n})$ is equivalent to maximizing each p_n when the condition $G_n^D \ge \max_{\forall m\in\mathcal{M}} G_{n,m}^E$ is satisfied. Otherwise, if the drones do not satisfy the condition, their security rates are always zero and do not contribute to the system security rate. To minimize the FL training time, we can choose the maximum wireless transmission power. In summary, all drones try to maximize their wireless transmission power p_n to maximize the system security rate. Therefore, problem **P1** can be transformed into solving N drones' subproblems **P2**:

$$\textbf{P2:}\quad \max_{p_n}\ p_n \quad (40)$$

$$s.t.\ \tilde{p}_n \le p_n \le P_n^m, \quad (41)$$

$$T\frac{p_n s_n}{B_n \log_2(1+\gamma_n p_n)} + TE_n^c + P^{hov}\tau_j \le B_n^{max}, \quad (42)$$

$$\tau_j = T\tau_j^c + \frac{Ts_j}{B_j \log_2(1+\gamma_j p_j)}, \quad (43)$$

where Eq. (43) means drone j incurs the largest FL time.

B. FL Time Calculation

Since τ_j is related to variable p_j, we first solve the subproblem of drone j. Then, τ_j can be calculated according to p_j and help determine the solutions of other drones' subproblems. By combining constraints Eq. (42) and Eq. (43), we have

$$\left(\frac{B_j^{max}}{T} - E_j^c - P^{hov}\tau_j^c\right) B_j \log_2(1+\gamma_j p_j) - s_j p_j - P^{hov} s_j \ge 0. \quad (44)$$

We define function $g(p_j) = (\frac{B_j^{max}}{T} - E_j^c - P^{hov}\tau_j^c)B_j \log_2(1+\gamma_j p_j) - s_j p_j - P^{hov} s_j \ge 0$. Therefore, the subproblem **P2** of drone j is to find the maximum p_j which satisfies $g(p_j) \ge 0, \tilde{p}_j \le p_j \le P_j^m$. We can calculate the derivative of $g(p_j)$ as

$$g'(p_j) = \left(\frac{B_j^{max}}{T} - E_j^c - P^{hov}\tau_j^c\right) B_j \frac{\gamma_j \log_2 e}{1+\gamma_j p_j} - s_j. \quad (45)$$

Then, we can observe that $g(p_j)$ monotonically increases (i.e., $g'(p_j) < 0$) when $p_j < (\frac{B_j^{max}}{T} - E_j^c - P^{hov}\tau_j^c)B_j \frac{\log_2 e}{s_j} - \frac{1}{\gamma_j}$, and $g(p_j)$ monotonically decreases (i.e., $g'(p_j) > 0$) when $p_j > (\frac{B_j^{max}}{T} - E_j^c - P^{hov}\tau_j^c)B_j \frac{\log_2 e}{s_j} - \frac{1}{\gamma_j}$. Hence, to satisfy $g(p_j) \ge 0$, we have $p_j \in [\lambda, u]$, where $g(\lambda) = 0, g(u) = 0$. Meanwhile, the constraint $\tilde{p}_j \le p_j \le P_j^m$ should also be satisfied. We then have $p_j \in [\max\{\lambda, \tilde{p}_j\}, \min\{u, P_j^m\}]$. Therefore, the solution of p_j can be expressed as $p_j = \min\{u, P_j^m\}$. Since $g(p_j)$ decreases when $p_j > (\frac{B_j^{max}}{T} - E_j^c - P^{hov}\tau_j^c)B_j \frac{\log_2 e}{s_j} - \frac{1}{\gamma_j}$, we utilize the binary search method [36] to calculate u which makes $g(u) = 0$.

The basic idea of the binary search method is to repeatedly dividing the search interval in half. Initially, we choose the search interval $[\lambda_1, \lambda_2]$, where $g(\lambda_1) > 0$ and $g(\lambda_2) < 0$. If the value in the middle of the search interval $g(\frac{\lambda_1+\lambda_2}{2}) = 0$, then we find $u = \frac{\lambda_1+\lambda_2}{2}$ and stop the search. Otherwise, if $g(\frac{\lambda_1+\lambda_2}{2}) > 0$, we narrow the search interval to $[\frac{\lambda_1+\lambda_2}{2}, \lambda_2]$ and continue the search. If $g(\frac{\lambda_1+\lambda_2}{2}) < 0$, we narrow the search interval to $[\lambda_1, \frac{\lambda_1+\lambda_2}{2}]$ and continue the search. By the binary search method, we can obtain the value u and $p_j = \min\{u, P_j^m\}$. Based on p_j, the FL time $\tau_j = T\tau_j^c + \frac{Ts_j}{B_j \log_2(1+\gamma_j p_j)}$ can be calculated.

C. Subproblem Solution

We then calculate the subproblems of drone n ($n \in \mathcal{N} \setminus j$) based on τ_j. Combining Eqs. (42) and (43) yields

$$\left(B_n^{max} - TE_n^c - P^{hov}\tau_j\right) B_n \log_2(1+\gamma_n p_n) - Ts_n p_n \ge 0. \quad (46)$$

We define function $\xi(p_n) = (B_n^{max} - TE_n^c - P^{hov}\tau_j)B_n \log_2(1+\gamma_n p_n) - Ts_n p_n \ge 0$ and hence $\xi(p_n) \ge 0$. The derivative of $\xi(p_n)$ is

$$\xi'(p_n) = \left(B_n^{max} - TE_n^c - P^{hov}\tau_j\right) B_n \frac{\gamma_n \log_2 e}{1+\gamma_n p_n} - Ts_n. \quad (47)$$

Algorithm 1: PCSF

```
Initialize the candidate vector V = ∅;
for each j ∈ 𝒩 do
    Calculate p_j according to the binary search method in Section V-B;
    Calculate FL time τ_j = Tτ_j^c + Ts_j / (B_j log_2(1+γ_j p_j));
    for each n ∈ 𝒩 \ j do
        Calculate p_n according to the binary search method in Section V-C;
    end
    if Candidate condition Eq. (48) is satisfied then
        Calculate the system security rate R^SEC;
        Assign V[j] = R^SEC;
    else
        Assign V[j] = 0;
    end
end
Choose j that achieves the largest V[j];
Choose the FL time τ_j and its corresponding p_n as the optimum solution.
```

It can be observed that $\xi(p_n)$ monotonically increases (i.e., $\xi'(p_n) > 0$) when $p_n < (B_n^{max} - TE_n^c - P^{hov}\tau_j)B_n\frac{\log_2 e}{Ts_n} - \frac{1}{r_n}$ and monotonically decreases (i.e., $\xi'(p_n) < 0$) when $p_n > (B_n^{max} - TE_n^c - P^{hov}\tau_j)B_n\frac{\log_2 e}{Ts_n} - \frac{1}{r_n}$. Hence, when $\xi(p_n) \geq 0$, p_n falls within the interval $[\tilde{\lambda}, \tilde{u}]$, where $\xi(\tilde{\lambda}) = 0$ and $\xi(\tilde{u}) = 0$. Note that Eq. (41) should also be satisfied, and then $p_n \in [\max\{\tilde{\lambda}, \tilde{p}_n\}, \min\{\tilde{u}, P_n^m\}]$.

To calculate $\tilde{u}$, we utilize the binary search method similar to that in Section V-B. Specifically, we first initialize the search interval $[\tilde{\lambda}_1, \tilde{\lambda}_2]$, where $\xi(\tilde{\lambda}_1) > 0$ and $\xi(\tilde{\lambda}_2) < 0$. If $\xi(\frac{\tilde{\lambda}_1+\tilde{\lambda}_2}{2}) = 0$, we stop the search and assign $\tilde{u} = \frac{\tilde{\lambda}_1+\tilde{\lambda}_2}{2}$. If $\xi(\frac{\tilde{\lambda}_1+\tilde{\lambda}_2}{2}) > 0$, we change the search interval to $[\frac{\tilde{\lambda}_1+\tilde{\lambda}_2}{2}, \tilde{\lambda}_2]$ and continue the search. If $\xi(\frac{\tilde{\lambda}_1+\tilde{\lambda}_2}{2}) < 0$, we change the search interval to $[\tilde{\lambda}_1, \frac{\tilde{\lambda}_1+\tilde{\lambda}_2}{2}]$ and continue the search. Since we try to maximize p_n, we have $p_n = \min\{\tilde{u}, P_n^m\}$.

Note that we assume that $\tau_j = T\max_{n\in\mathcal{N}}\{\tau_n^c + \frac{s_n}{B_n\log_2(1+\gamma_n p_n)}\} = T\tau_j^c + \frac{Ts_j}{B_j\log_2(1+\gamma_j p_j)}$. Hence, we have $\tau_n = T\tau_n^c + \frac{Ts_n}{B_n\log_2(1+\gamma_n p_n)} \leq \tau_j$, which indicates that $p_n \geq \frac{1}{\gamma_n}[2^{\frac{s_n}{B_n(\tau_j/T-\tau_n^c)}} - 1]$. By combining with the QoS requirement Eq. (33), p_n should satisfy

$$p_n \geq \max\left\{\frac{1}{\gamma_n}\left[2^{\frac{s_n}{B_n\left(\tau_j/T-\tau_n^c\right)}} - 1\right], \tilde{p}_n\right\}, \ \forall n \in \mathcal{N}\setminus j, \tag{48}$$

to denote the candidate condition on checking whether the assumption that drone *j* has the longest FL training time leads to a feasible solution of problem **P1**.

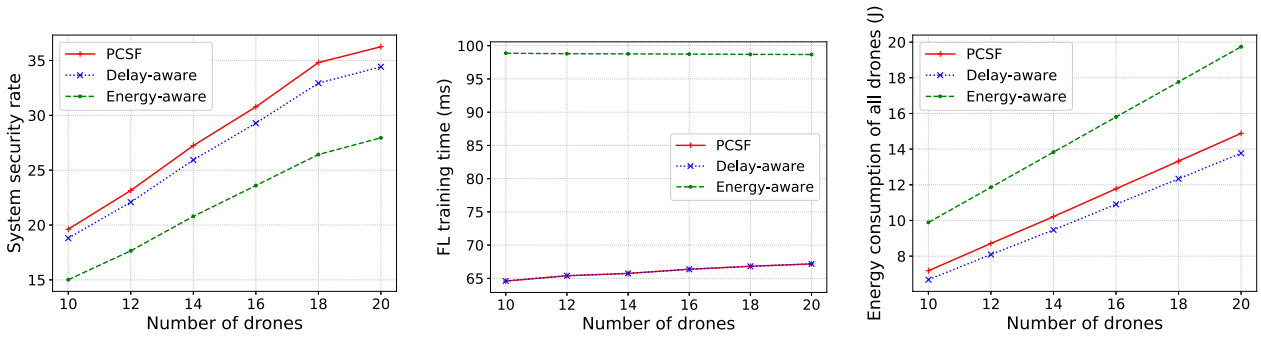
D. Proposed Algorithm

The detailed process of our proposed algorithm is delineated in Algorithm 1. Lines 2-14 enumerate each possible FL time. Lines 3-4 calculate the FL time τ_j. Lines 5-7 calculate p_n of all other drones. Lines 8-13 check whether the derived solutions by the current FL time satisfy the candidate condition. Lines 15-16 choose the best solution by comparing all the FL time possibilities. Note that the running time of PCSF is dominated by the binary search in line 6 in the nested loop. The computational complexity of the binary search is $\mathcal{O}(\log_2(\lambda^- - \lambda^+))$, where $[\lambda^+, \lambda^-]$ is the initial interval of the binary search and satisfies $\xi(\lambda^+) > 0$ and $\xi(\lambda^-) < 0$. Therefore, PCSF yields a computational complexity of $\mathcal{O}(N^2\log_2(\lambda^- - \lambda^+))$.

VI. Performance Evaluation

We set up simulations to evaluate the performance of our proposed algorithm PCSF in this section. We compare PCSF with the existing algorithm (denoted as "Delay-aware") inspired by [10] which minimizes the FL training time. We also use the existing algorithm (denoted as "Energy-aware") as the comparison algorithm which is inspired by [9] where the energy consumption for wireless data transmission is minimized.

In our simulations, there are $N = 16$ drones hovering in the flying plane within a 1000 *m* × 1000 *m* area to provide the FL service. The drones' locations are randomly distributed in this area and the height of the flying plane is $H = 100$ *m*. The fog node is located in the center of this area to communicate with all drones. There are $M = 3$ eavesdroppers randomly distributed in this area. To calculate wireless channels between drones and the fog node, the environment-related constants *a* and *b* are respectively 9.6 and 0.28, the speed of light $c = 3 \times 10^8$ *m/s*, the carrier frequency $f_c = 2$ *GHz*, and the environment-related constants $\psi^{LoS} = 1$ *dB* and $\psi^{NLoS} = 20$ *dB*. The allocated wireless bandwidth $B = 2$ *MHz* and the noise power density $N_0 = -174$ *dBm/Hz*. The above parameters related to drone wireless communications are consistent with [31]. The maximum wireless transmission power $Pm = 3$ *W*. To calculate drones' hovering power, each drone's mass $m = 500$ *g* and the earth gravitational acceleration $g = 9.8$ *m/s*^{2}, constants r_p, n_p and ρ in Eq. (26) are 20 *cm*, 4, and 1.225 *kg/m*^{3}, respectively. The above parameters related to drone hovering power are consistent with [37]. Each drone updates the local model by its local training data, and the number of data samples D_n is randomly chosen from 300 to 500. Each data sample requires C_n, randomly chosen from 30 to 50, CPU cycles for computation. The computation capacity of each drone $f = 2 \times 10^9$ CPU cycles per second. The constant γ which contributes to the CPU energy consumption of drones is 10^{-28} [34]. The battery capacity of each drone $B_n^{max} = 1$ *J*. Drone *n*'s uploaded local model parameter $s_n = 5$ *Kb* and the QoS requirement of the FL service $Q^{th} = 200$ *ms*. To analyze FL convergence, the loss function is $\alpha = 2$ strongly convex and $\beta = 4$ smooth, constant $\eta = \frac{1}{3}$ in Eq. (3), and the learning step size of the gradient descent algorithm $\delta = \frac{1}{4}$. The above parameters for FL convergence analysis are consistent with [24]. Note that the above parameters are default values and may change as needed.



(a) System security rate (b) FL training time (c) Drones' energy consumption

Fig. 3. Key performance metrics vs number of drones.

We first evaluate PSCF's performances in Fig. 3(a) with different numbers of drones ranging from 10 to 20. Fig. 3(a), Fig. 3(b), and Fig. 3(c) depict the performances of the system security rate, FL training time, and all drones' energy consumption, respectively. In Fig. 3(a), more drones lead to a larger system security rate for all three algorithms because the system security rate is the summation of all drones' security rates. PCSF provides a larger system security rate than those of Delay-aware and Energy-aware. In Fig. 3(b), the FL training time does not change much when the number of drones increases because all drones' computations are operated in parallel. PCSF achieves similar FL training time as that of Delay-aware and performs better than Energy-aware. From the objective function of problem **P1**, we can observe that a larger wireless transmission power leads to a larger system security rate. Hence, PCSF prefers larger wireless power to maximize the system security rate. Delay-aware maximizes the transmission power to minimize the FL training time. Therefore, Delay-aware performs close to PCSF as shown in Fig. 3(a) and Fig. 3(b). Delay-aware performs better than Energy-aware because Delay-aware prefers higher transmission powers to minimize the FL training time and thus to help improve the system security rate, while Energy-aware prefers lower transmission powers to minimize the energy consumption. In Fig. 3(c), the energy consumption increases when the number of drones increases because more drones consume more energy. Counterintuitively, Energy-aware incurs the most energy consumption because Energy-aware minimizes the energy consumption for wireless data transmission, while a drone's energy consumption is mostly composed of the hovering energy consumption which is determined by the FL training time. Since Energy-aware incurs the largest FL training time, it incurs the most drone energy consumption. Similarly, Delay-aware achieves the smallest FL training time and hence the least energy consumption. Note that the performance of drones' energy consumption is determined by the FL training time, and we hence only show the performance of FL training time and ignore that of the energy consumption thereafter. In summary, PCSF achieves the largest system security rate and also a small FL training time.

Fig. 4 illustrates the performances of three algorithms with different numbers of eavesdroppers ranging from 2 to 7. In Fig. 4, the system security rates of all three algorithms decrease when the number of eavesdroppers increase because more data can be wiretapped. PCSF provides the highest system security rate and Delay-aware the second. Energy-aware achieves the smallest system security rate because of the similar reason as that in Fig. 3(a).

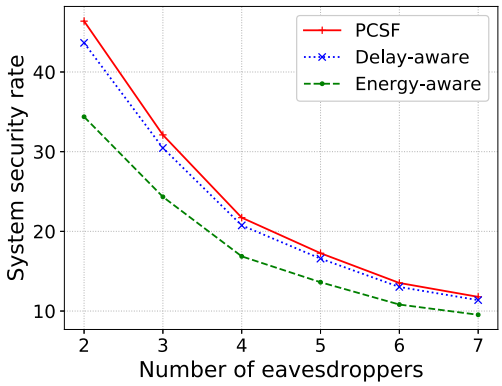
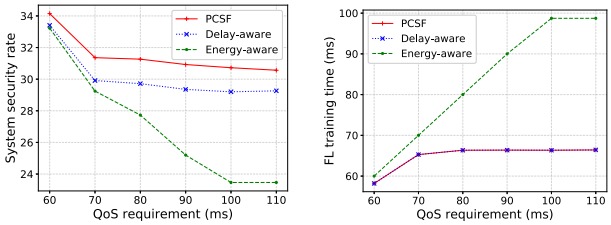


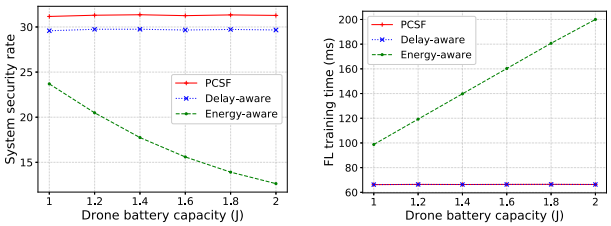
Fig. 4. System security rate vs number of eavesdroppers.



(a) System security rate (b) FL training time

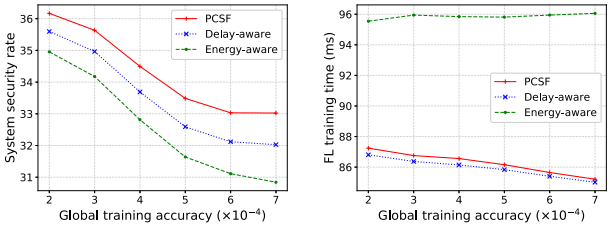
Fig. 5. Key performance metrics vs QoS requirement.

We then investigate the impact of the QoS requirement (i.e., FL training time requirement), ranging from 60 to 110 *ms*, on our proposed algorithm in Fig. 5. The performances of system security rate and FL training time are shown in Fig. 5(a) and Fig. 5(b), respectively. In Fig. 5(a), the system security rates of all three algorithms decrease with the increase of the QoS requirement. When the QoS requirement is small (i.e., strict), higher transmission powers are required to satisfy the QoS requirement and hence the system security rate is higher. Delay-aware tries to minimize the FL training time and does not increase much when the QoS requirement is larger than 70 *ms* in Fig. 5(b). Similar to Fig. 3(a) and Fig. 4, PCSF achieves the largest system security rate as shown in Fig. 5(a) and a small FL training time 5(b).



(a) System security rate (b) FL training time

Fig. 6. Key performance metrics vs drone battery capacity.



(a) System security rate (b) FL training time

Fig. 7. Key performance metrics vs global training accuracy.

Fig. 6 evaluates the performances of PCSF with different drone battery capacities ranging from 1 to 2 J. The performances of system security rate and FL training time are shown in Fig. 6(a) and Fig. 6(b), respectively. The system security rate in Fig. 6(a) and the FL training time in Fig. 6(b) of PCSF and Delay-aware do not change with the increase of the drone battery capacity because the drone battery capacity restricts the minimum wireless transmission power while PCSF and Delay-aware tend to choose the largest transmission power. Energy-aware's system security rate decreases in Fig. 6(a) and its FL training time increases in Fig. 6(b) when the drone battery capacity increases, because Energy-aware prefers lower transmission powers which are affected by the increasing drone battery capacity. Moreover, PCSF performs the best among the three algorithms in Fig. 6(a) and achieves a small FL training time in Fig. 6(b).

We explore the performances of PCSF with different global training accuracy ranging from 2×10^{-4} to 7×10^{-4} in Fig. 7. Fig. 7(a) and Fig. 7(b) illustrate the performance of the system security rate and FL training time, respectively. In Fig. 7(a), the system security rate of all three algorithms decreases when the global training accuracy becomes large. This is because a larger global training accuracy means less global iterations is required and more time and energy consumption are allowed to finish one global iteration, hence reducing the wireless transmission power to meet the QoS and battery capacity constraints. PCSF provides the largest system security rate among the three algorithms. In Fig. 7(b), the FL training time of PCSF and Delay-aware decreases with the increase of the global training accuracy while Energy-aware does not change much. Since the transmission power becomes smaller, all three algorithms have larger FL time in one global iteration. Meanwhile, the number of global iterations decreases. Therefore, the FL time in one global iteration increases more than those of PCSF and Delay-aware, and hence the FL training time of Energy-aware remains almost the same while those of PCSF and Delay-aware decrease. We can also observe that PCSF performs close to Delay-aware.

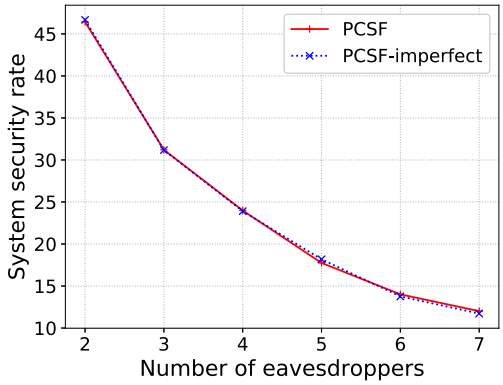


Fig. 8. System security rate vs number of eavesdroppers.

Note that PCSF assumes that the locations of eavesdroppers are perfectly known by detecting their leaked power [28]. In practice, eavesdroppers may hide their positions, and hence the channel power gain from a drone to an eavesdropper is considered as a random parameter. The average channel gain is usually adopted to calculate the system security rate [38]. Fig. 8 compares the performances of our proposed algorithm with perfectly known and imperfectly known channel models (denoted as PCSF and PCSF-imperfect, respectively). In PCSF-imperfect, the locations of eavesdroppers are randomly chosen within the region 100×100 m^2 of those in PCSF, and we average the system security rate of PCSF-imperfect over 1000 simulations. Similar to Fig. 4, the system security rate decreases when the number of eavesdroppers increases in Fig. 8. We can also observe that PCSF-imperfect performs close to PCSF because PCSF-imperfect averages all system security rates with different eavesdropper locations; averages of these locations can be considered as the eavesdropper locations of PCSF.

VII. Conclusion

In this paper, we have proposed the secure FL in fog-aided IoD networks to counteract eavesdroppers. The FL convergence has been analyzed and demonstrated. We have investigated the wireless transmission power control problem to maximize the system security rate constrained by the FL training time requirement and drone battery capacities. This problem has been formulated as a non-linear programming problem to optimize each drone's wireless transmission power. We have designed an algorithm PCSF to obtain the solution of this problem. Simulation results have demonstrated that PCSF performs better than two existing algorithms and achieves both a high system security rate and a small FL training time.

Appendix A
Proof of Lemma 2

Since w^* and $w_n^{t,*}$ are the optimal solution of $F(w)$ and $G_n^t(w)$ respectively, we can derive that $\nabla F(w^*) = 0$ and

$\nabla G_n^t(w_n^{t,*}) = 0$. We then have

$$\begin{aligned}\left\|\nabla F(w^t)\right\|^2 &= \left\|\nabla F(w^t) - \nabla F(w^*)\right\|^2 \\ &\overset{(11)}{\geq} \alpha\left\|\nabla F(w^t) - \nabla F(w^*)\right\|\left\|w^t - w^*\right\|^2 \\ &= \alpha\nabla F(w^t)^\top (w^t - w^*) \\ &\overset{(12)}{\geq} \alpha\Big[F(w^t) - F(w^*) + \frac{\alpha}{2}\left\|w^t - w^*\right\|^2\Big] \\ &\geq \alpha\big[F(w^t) - F(w^*)\big], \end{aligned} \tag{49}$$

which proves Eq. (15). Meanwhile, we have

$$\begin{aligned}&\left\|\nabla G_n^t\Big(w_n^{t,i}\Big)\right\|^2 = \left\|\nabla G_n^t\Big(w_n^{t,i}\Big) - \nabla G_n^t(w_n^{t,*})\right\|^2 \\ &\overset{(7)}{=} \left\|\nabla G_n^t\Big(w_n^{t,i}\Big)\right\|\left\|\nabla F_n\Big(w_n^{t,i}\Big) - \nabla F_n(w_n^{t,*})\right\| \\ &\overset{(11)}{\geq} \alpha\left\|\nabla G_n^t\Big(w_n^{t,i}\Big)\right\|\left\|w_n^{t,i} - w_n^{t,*}\right\| \\ &\overset{(6)}{=} \alpha\Big[\nabla F_n\Big(w_n^{t,i}\Big) - \nabla F_n(w^t) + \eta\nabla F(w^t)\Big]^\top \\ &\quad\times \Big(w_n^{t,i} - w_n^{t,*}\Big) \\ &= \alpha\Big\{\nabla F_n\Big(w_n^{t,i}\Big)^\top \Big(w_n^{t,i} - w_n^{t,*}\Big) \\ &\qquad -\big[\nabla F_n(w^t) - \eta\nabla F(w^t)\big]^\top \Big(w_n^{t,i} - w_n^{t,*}\Big)\Big\} \\ &\overset{(12)}{\geq} \alpha\Big\{F_n\Big(w_n^{t,i}\Big) - F_n(w_n^{t,*}) \\ &\qquad - \big[\nabla F_n(w^t) - \eta\nabla F(w^t)\big]^\top \Big(w_n^{t,i} - w_n^{t,*}\Big)\Big\} \\ &\overset{(6)}{=} \alpha\Big[G_n^t\Big(w_n^{t,i}\Big) - G_n^t(w_n^{t,*})\Big], \end{aligned} \tag{50}$$

which proves Eq. (16).

APPENDIX B
PROOF OF LEMMA 3

We demonstrate the relationship between $G_n^t(w_n^{t,i}) - G_n^t(w_n^{t,*})$ and $G_n^t(w^t) - G_n^t(w_n^{t,*})$. The demonstration process is inspired by [24].

$$\begin{aligned}&G_n^t\Big(w_n^{t,i+1}\Big) \overset{(3)}{=} F_n\Big(w_n^{t,i+1}\Big) \\ &\quad - \big[\nabla F_n(w^t) - \eta\nabla F(w^t)\big]^\top w_n^{t,i+1} \\ &\overset{(5),(14)}{\geq} F_n\Big(w_n^{t,i}\Big) - \delta\nabla F_n\Big(w_n^{t,i}\Big)^\top \nabla G_n^t\Big(w_n^{t,i}\Big) \\ &\quad + \frac{\delta^2\beta}{2}\left\|\nabla G_n^t\Big(w_n^{t,i}\Big)\right\|^2 \\ &\quad + \big[\nabla F_n(w^t) - \eta\nabla F(w^t)\big]^\top w_n^{t,i+1} \\ &\overset{(3),(5)}{=} G_n^t\Big(w_n^{t,i}\Big) - \delta G_n^t\Big(w_n^{t,i}\Big)^\top G_n^t\Big(w_n^{t,i}\Big) \\ &\quad + \frac{\delta^2\beta}{2}\left\|\nabla G_n^t\Big(w_n^{t,i}\Big)\right\|^2 \\ &= G_n^t\Big(w_n^{t,i}\Big) - \frac{(2-\delta\beta)\delta}{2}\left\|\nabla G_n^t\Big(w_n^{t,i}\Big)\right\|^2 \\ &\overset{(16)}{\leq} G_n^t\Big(w_n^{t,i}\Big) - \frac{(2-\delta\beta)\delta\alpha}{2}\Big[G_n^t\Big(w_n^{t,i}\Big) - G_n^t(w_n^{t,*})\Big]. \end{aligned}$$

Based on the above analysis, we have

$$\begin{aligned}&G_n^t\Big(w_n^{t,i}\Big) - G_n^t(w_n^{t,*}) \\ &\leq \left[1 - \frac{(2-\delta\beta)\delta\alpha}{2}\right]\Big[G_n^t\Big(w_n^{t,i-1}\Big) - G_n^t(w_n^{t,*})\Big] \leq \cdots \leq \\ &\leq \left[1 - \frac{(2-\delta\beta)\delta\alpha}{2}\right]^i \big[G_n^t(w^t) - G_n^t(w_n^{t,*})\big] \\ &\leq e^{-i\frac{(2-\delta\beta)\delta\alpha}{2}}\big[G_n^t(w^t) - G_n^t(w_n^{t,*})\big], \end{aligned}$$

where the last inequality holds because $1 - x \leq e^{-x}, x \geq 0$. If we assign the local accuracy $e_l = e^{-i\frac{(2-\delta\beta)\delta\alpha}{2}}$, i.e., $I = \frac{2}{(2-\delta\beta)\delta\alpha}\ln(\frac{1}{\epsilon_l})$, the local convergence definition (Eq. (8)) holds. Therefore, Lemma 3 is proved.

APPENDIX C
PROOF OF LEMMA 4

We demonstrate the relationship between $F(w^t) - F(w^*)$ and $F(w^0) - F(w^*)$.

$$\begin{aligned}&F\Big(w^{t+1}\Big) \overset{(9)}{=} F\left(w^t + \frac{1}{N}\sum_{n\in\mathcal{N}}\big(w_n^{t,*} - w^t\big)\right) \\ &\overset{\text{Lemma 1}}{\leq} F(w^t) + \frac{1}{N}\sum_{n\in\mathcal{N}}\nabla F(w^t)^\top\big(w_n^{t,*} - w^t\big) \\ &\quad + \frac{\beta}{2}\left\|\frac{1}{N}\sum_{n\in\mathcal{N}}\big(w_n^{t,*} - w^t\big)\right\|^2 \\ &\overset{(3)}{=} F(w^t) + \frac{1}{N\eta}\sum_{n\in\mathcal{N}} \\ &\quad\times\Big[G_n^t(w_n^{t,*}) - F_n(w_n^{t,*}) + \nabla F_n(w^t)^\top w_n^{t,*}\Big] \\ &\quad - \frac{1}{N}\sum_{n\in\mathcal{N}}\nabla F(w^t)^\top w^t + \frac{\beta}{2}\left\|\frac{1}{N}\sum_{n\in\mathcal{N}}\big(w_n^{t,*} - w^t\big)\right\|^2, \\ &\overset{(12)}{\leq} F(w^t) + \frac{1}{N\eta}\sum_{n\in\mathcal{N}}\Big\{G_n^t(w_n^{t,*}) - F_n(w^t) \\ &\qquad + \big[\nabla F_n(w^t) - \eta\nabla F(w^t)\big]^\top w^t\Big\} \\ &\quad + \frac{\beta}{2}\left\|\frac{1}{N}\sum_{n\in\mathcal{N}}\big(w_n^{t,*} - w^t\big)\right\|^2 \\ &\leq F(w^t) + \frac{1}{N\eta}\sum_{n\in\mathcal{N}}\Big\{G_n^t(w_n^{t,*}) - F_n(w^t) \\ &\qquad + \big[\nabla F_n(w^t) - \eta\nabla F(w^t)\big]^\top w^t\Big\} \\ &\quad + \frac{\beta}{2N}\sum_{n\in\mathcal{N}}\left\|w_n^{t,*} - w^t\right\|^2 \\ &\overset{(3)}{=} F(w^t) + \frac{1}{N\eta}\sum_{n\in\mathcal{N}}\Big\{G_n^t(w_n^{t,*}) - G_n^t(w^t) \\ &\qquad - \frac{\alpha-\beta\eta}{2}\left\|w_n^{t,*} - w^t\right\|^2\Big\} \\ &\overset{(3)}{=} F(w^t) - \frac{1}{N\eta}\sum_{n\in\mathcal{N}}\Big\{\frac{\alpha-\beta\eta}{2}\left\|w_n^{t,*} - w^t\right\|^2 \\ &\qquad + F_n(w^t) - F_n(w_n^{t,*}) \\ &\qquad + \nabla F_n(w_n^{t,*})^\top\big(w_n^{t,*} - w^t\big)\Big\} \end{aligned}$$

Then,

$$\begin{aligned} F\left(w^{t+1}\right) &\overset{(12)}{\leq} F\left(w^t\right) - \frac{1}{N\eta}\sum_{n\in\mathcal{N}}\frac{2\alpha-\beta\eta}{2}\left\|w_n^{t,*}-w^t\right\|^2 \\ &\overset{(11)}{\leq} F\left(w^t\right) - \frac{2\alpha-\beta\eta}{2N\eta\beta^2} \\ &\quad\times\sum_{n\in\mathcal{N}}\left\|\nabla F_n\left(w_n^{t,*}\right)-\nabla F_n\left(w^t\right)\right\|^2 \\ &\overset{(7)}{=} F\left(w^t\right) - \frac{(2\alpha-\beta\eta)\eta}{2\beta^2}\left\|\nabla F\left(w^t\right)\right\|^2 \\ &\overset{(15)}{\leq} F\left(w^t\right) - \frac{(2\alpha-\beta\eta)\alpha\eta}{2\beta^2}[F\left(w^t\right)-F(w^*)]. \end{aligned}$$

Based on the above analysis, we have

$$\begin{aligned} F\left(w^t\right)-F(w*) &\leq \left[1-\frac{(2\alpha-\beta\eta)\alpha\eta}{2\beta^2}\right] \\ &\quad\times\left[F\left(w^{t-1}\right)-F(w^*)\right] \\ &\leq\cdots\leq\left[1-\frac{(2\alpha-\beta\eta)\alpha\eta}{2\beta^2}\right]^t \\ &\quad\times\left[F\left(w^0\right)-F(w^*)\right] \\ &\leq e^{-t\frac{(2\alpha-\beta\eta)\alpha\eta}{2\beta^2}}\left[F\left(w^0\right)-F(w^*)\right]. \end{aligned}$$

We assign the global accuracy $\epsilon_g = e^{-t\frac{(2\alpha-\beta\eta)\alpha\eta}{2\beta^2}}$, i.e., $T = \frac{2\beta^2}{(2\alpha-\beta\eta)\alpha\eta}\ln(\frac{1}{\epsilon_g})$, the global FL problem is converged. Therefore, Lemma 4 is proved.

REFERENCES

[1] J. Yao and N. Ansari, "Online task allocation and flying control in fog-aided Internet of Drones," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5562–5569, May 2020.

[2] A. M. Almasoud and A. E. Kamal, "Data dissemination in IoT using a cognitive UAV," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 849–862, Dec. 2019.

[3] J. Yao, T. Han, and N. Ansari, "On mobile edge caching," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2525–2553, 3rd Quart. 2019.

[4] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the Internet of Things," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 22–29, Dec. 2016.

[5] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu, and J. Qin, "A survey on application of machine learning for Internet of Things," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 8, pp. 1399–1417, 2018.

[6] N. Ansari, Q. Fan, X. Sun, and L. Zhang, "SoarNet," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 37–43, Dec. 2019.

[7] M. Song *et al.*, "Analyzing user-level privacy attack against federated learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2430–2444, Oct. 2020.

[8] 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*, Apr. 2017, pp. 1273–1282.

[9] 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, doi: 10.1109/JIOT.2020.3022590.

[10] Z. Yang *et al.*, "Delay minimization for federated learning over wireless communication networks," 2020. [Online]. Available: arXiv:2007.03462.

[11] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[12] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2134–2143, Mar. 2020.

[13] P. K. Gopala, L. Lai, and H. E. Gamal, "On the secrecy capacity of fading channels," *IEEE Trans. Inf. Theory*, vol. 54, no. 10, pp. 4687–4698, Oct. 2008.

[14] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: Attack and defense strategies," *IEEE Netw.*, vol. 20, no. 3, pp. 41–47, May–Jun. 2006.

[15] J. Yao and N. Ansari, "QoS-aware power control in Internet of Drones for data collection service," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6649–6656, Jul. 2019.

[16] M. Gharibi, R. Boutaba, and S. L. Waslander, "Internet of Drones," *IEEE Access*, vol. 4, pp. 1148–1162, 2016.

[17] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based Internet of Things services: Comprehensive survey and future perspectives," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 899–922, Dec. 2016.

[18] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of Drones deployment," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3572–3584, Apr. 2019.

[19] Y. Meidan *et al.*, "ProfilIoT: A machine learning approach for IoT device identification based on network traffic analysis," in *Proc. ACM SAC*, 2017, pp. 506–509. [Online]. Available: https://doi.org/10.1145/3019612.3019878

[20] J. Yao and N. Ansari, "Caching in dynamic IoT networks by deep reinforcement learning," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3268–3275, Mar. 2021, doi: 10.1109/JIOT.2020.3004394.

[21] J. Yao and N. Ansari, "Power control in Internet of Drones by deep reinforcement learning," in *Proc. IEEE ICC*, 2020, pp. 1–6.

[22] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE INFOCOM*, Apr. 2019, pp. 1387–1395.

[23] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-Edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Netw.*, vol. 33, no. 5, pp. 156–165, Sep./Oct. 2019.

[24] 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, doi: 10.1109/TWC.2020.3037554.

[25] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: Secure and verifiable federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 911–926, 2020.

[26] K. Wei *et al.*, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[27] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

[28] A. Mukherjee and A. L. Swindlehurst, "Detecting passive eavesdroppers in the MIMO wiretap channel," in *Proc. IEEE ICASSP*, Mar. 2012, pp. 2809–2812.

[29] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016. [Online]. Available: arXiv:1610.02527.

[30] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Mach. Learn.*, vol. 8, nos. 3–4, pp. 231–357, Nov. 2015.

[31] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *Proc. IEEE GLOBECOM*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

[32] Y. Zhou *et al.*, "Secure communications for UAV-enabled mobile edge computing systems," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 376–388, Jan. 2020.

[33] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. USENIX HotCloud*, Berkeley, CA, USA, 2010, p. 4.

[34] 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.

[35] J. Gundlach, *Designing Unmanned Aircraft Systems: A Comprehensive Approach*, Amer. Inst. Aeronaut. Astronaut., Reston, VA, USA, 2012.

[36] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, vol. 23. Englewood Cliffs, NJ, USA: Prentice Hall, 1989,

[37] H. Ghazzai, M. B. Ghorbel, A. Kadri, M. J. Hossain, and H. Menouar, "Energy-efficient management of unmanned aerial vehicles for underlay cognitive radio systems," *IEEE Trans. Green Commun. Netw.*, vol. 1, no. 4, pp. 434–443, Dec. 2017.

[38] Y. Wu *et al.*, "Secrecy-driven resource management for vehicular computation offloading networks," *IEEE Netw.*, vol. 32, no. 3, pp. 84–91, May/Jun. 2018.