This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3014385, IEEE Transactions on Network Science and Engineering

Learning in the Air: Secure Federated Learning for UAV-Assisted Crowdsensing

Yuntao Wang[1], Zhou Su[1,2], Ning Zhang[3], and Abderrahim Benslimane[4]
[1]School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China
[2]School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China
[3]Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada
[4]Computer Science and Engineering, University of Avignon, Avignon, France
Corresponding author: Zhou Su (zhousu@ieee.org)

***Abstract*—Unmanned aerial vehicles (UAVs) combined with artificial intelligence (AI) have opened a revolutionized way for mobile crowdsensing (MCS). Conventional AI models, built on aggregation of UAVs' sensing data (typically contain private and sensitive user information), may arise severe privacy and data misuse concerns. Federated learning, as a promising distributed AI paradigm, has opened up possibilities for UAVs to collaboratively train a shared global model without revealing their local sensing data. However, there still exist potential security and privacy threats for UAV-assisted crowdsensing with federated learning due to vulnerability of central curator, unreliable contribution recording, and low-quality shared local models. In this paper, we propose SFAC, a secure federated learning framework for UAV-assisted MCS. Specifically, we first introduce a blockchain-based collaborative learning architecture for UAVs to securely exchange local model updates and verify contributions without the central curator. Then, by applying local differential privacy, we design a privacy-preserving algorithm to protect UAVs' privacy of updated local models with desirable learning accuracy. Furthermore, a two-tier reinforcement learning-based incentive mechanism is exploited to promote UAVs' high-quality model sharing when explicit knowledge of network parameters are not available in practice. Extensive simulations are conducted, and the results demonstrate that the proposed SFAC can effectively improve utilities for UAVs, promote high-quality model sharing, and ensure privacy protection in federated learning, compared with existing schemes.**

***Index Terms*—UAV, AI security, federated learning, blockchain, reinforcement learning, local differential privacy.**

I. Introduction

Mobile crowdsensing (MCS), which leverages the sensing capabilities of ubiquitous smart mobile devices with embedded sensors, has become an appealing paradigm for urban sensing [1]–[5]. However, traditional MCS has certain limitations in satisfying stringent quality of service (QoS) requirements for complex and extreme scenarios such as flooding, earthquakes, and city evacuation [6]. Unmanned aerial vehicles (UAVs) combined with artificial intelligence (AI) technology has opened up possibilities to assist the existing terrestrial MCS infrastructures to perform more challenging tasks [7]–[10]. Specifically, UAVs allow autonomous crowdsensing anytime and anywhere owing to their remarkable advantages of low cost, fast deployment, and flexible mobility [11], [12]. Moreover, UAVs equipped with rich sensors can be dispatched immediately for a variety of on-demand MCS applications such as traffic monitoring, search and rescue, and surveillance of public safety [13]. To enable diverse intelligent MCS services, in traditional AI models, a wealth of sensing data distributed across a large set of UAVs require to be migrated to a central curator (e.g., the mobile edge computing (MEC) node [14]–[17] located in the base station) for model training and analysis. Nevertheless, the sensory data may contain potentially private and sensitive information while the central curator may be honest-but-curious during data collection and training [18], [19]. As a result, such mechanisms can result in severe risk of privacy breach such as privacy leakage, data eavesdropping, and misuse of user's personal data [20]–[23].

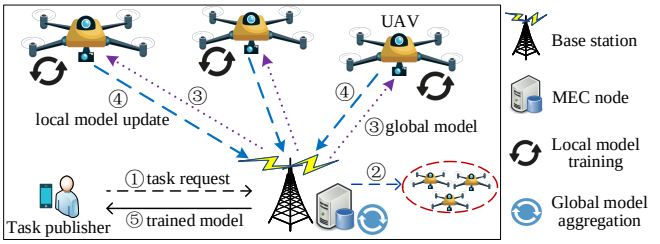


Fig. 1. An example scenario of federated learning framework for UAV-assisted MCS. (①: A task publisher sends its task request to an MEC node; ②: MEC node selects a set of worker UAVs; ③: MEC node delivers the global model to the selected UAVs; ④: UAVs iteratively train the global model with local sensing data and send back local model updates for aggregation; ⑤: Repeat ③∼④, then the trained model or learning result is delivered to task publisher.)

Federated learning, as a compelling distributed AI paradigm, can protect data privacy to a large extent by enabling a group of UAVs to collaboratively train a shared AI model while keeping the training data (i.e., sensed data) locally on their devices [24]–[26]. In federated learning as shown in Fig. 1, UAVs only need to send the local model updates (i.e., weights or gradients) trained on their local sensed data, instead of revealing the raw data to the central curator which synthesizes a global model. This training process is repeated until the accuracy of the global model attains a predefined desirable level. Accordingly, with federated learning, the process of data acquisition, storage, and training in a central server for AI applications can be decoupled.

To fully reap the benefits of federated learning in UAV-assisted MCS, a series of fundamental challenges need to be resolved. First, due to the selfishness and constrained resources, UAVs will not participate in the collaborative learning process. They may also share low-quality local model updates generated from low-quality sensed data and few training samples if without sufficient incentive or validation. Evidences have shown that the model accuracy depends on both the quantity and quality of training data [27], [28]. Accordingly, the accuracy of global model will be deteriorated. Second, the centralized curator for model aggregation can be vulnerable to diverse threats (e.g., single point of failure and DDoS attack), whose malfunction (e.g., distorting all local model updates) can lead to a failure of the whole learning process. Additionally, UAVs usually make different contributions in global model training. If the curator is compromised, the recorded contribution values, which are evidences for rewards of participants, may be tampered, removed or forged. Third, the federated learning can not eliminate all privacy concerns. As shown in [29], [30], smart adversaries can launch differential attacks and model inversion attacks to infer UAV's private training data and its participation in a certain task by analyzing the shared gradients, which may lead to unwanted sensitive data leakage and damp the enthusiasm of legitimate participants. Thus, how to design a secure and privacy-preserving model sharing scheme for UAVs while promoting their collaboration in the federated learning needs to be investigated.

Recently, a flurry of research works are reported towards efficient and secure federated learning. For example, in [31], an asynchronous federated learning mechanism is proposed for joint optimization of power and resource allocation to enable low-latency and ultra-reliable vehicular communications. Besides, a blockchain-based federated learning framework named BlockFL is presented in [32], where users' local model updates are publicly shared, exchanged, and verified over the blockchain. Meanwhile, in [33], a privacy-preserving federated learning scheme named DeepChain is devised for auditable deep learning, where the blockchain is leveraged to stimulate users to join in collaborative training. However, most of the current works cannot be directly applied for UAV-assisted MCS with the following reasons. First, few works consider the joint security defense and incentives in reliable and high-quality local model sharing for UAVs. The lack of security validations may prevent UAVs from participating in cooperation. Second, the parameters of cost model and network model in UAV-assisted MCS may be time-varying and not readily available, which are not fully investigated in existing works. For instance, UAV users may have different privacy cost for performing distinct sensing tasks and sharing relevant model updates. Third, UAVs are typically equipped with diverse kinds of sensors, whereas their diversified sensing capabilities are seldom considered in most of existing works. Hence, it is still an open and vital issue to secure the collaborative learning while motivating the high-quality model sharing of UAVs in the highly dynamic network.

To address the aforementioned issues, in this paper, we propose SFAC, a secure federated learning framework for UAV-assisted MCS. Specifically, a novel blockchain-based collaborative learning architecture is first proposed to facilitate efficient data transmission and model training for UAVs in MCS. Afterwards, by leveraging blockchain to replace the central curator, a decentralized federated learning mechanism is devised to securely exchange local model updates and record UAVs' contributions in collaborative training. Furthermore, based on local differential privacy (LDP), a privacy-preserving local model sharing algorithm is developed to achieve desirable aggregate accuracy and rigorous privacy preserving for UAVs via on-device perturbations. Finally, the interactions between UAVs (i.e., data owners) and task publishers are formulated as finite Markov decision processes (MDPs), and a two-tier reinforcement learning (RL)-based incentive mechanism is presented to motivate UAVs' high-quality model sharing in the fast-changing network. The main contributions of this work are summarized as below.

- By exploiting blockchain, LDP and RL technologies, we propose SFAC, a practical federated learning framework for secure and efficient AI model training in UAV-assisted MCS. We consider three attacks in the network and develop the corresponding measures to safeguard collaborative learning for UAVs.
- We investigate a consortium blockchain network for decentralized data training, model sharing, and contribution tracing for resource-limited UAVs in federated learning through immutable ledgers, mutual verification, and distributed consensus. Based on the blockchain, the security concerns arisen from the central curator can be mitigated and the contributions of UAVs in global model training can be validated and securely recorded. Furthermore, a differentially private algorithm is devised to protect the privacy of UAVs' shared local models with guaranteed global aggregation accuracy. In addition, based on RL approach, the optimal payment strategy of task publisher and the optimal local model quality strategies of UAVs in federated learning can be acquired without the awareness of accurate network parameters in the highly uncertain environment.
- The effectiveness of SFAC is evaluated via extensive simulations. We show that our SFAC can efficiently motivate high-quality local model sharing, attain optimal strategies and better utilities for participants, and ensure privacy preservation for UAVs in federated learning, compared with existing schemes.

The remainder of the paper is organized as follows. Section II reviews the related work. Section III introduces the system model. In Section IV, the proposed SFAC scheme is presented. Performance evaluation is given in Section V. Section VI concludes this paper and discusses the future work.

II. Related Works

In this section, we first review the UAV-assisted MCS, and then discuss the works of federated learning in wireless networks and blockchain for UAVs.

A. UAV-Assisted Mobile Crowdsensing

Recently, there have been an increasing number of studies leveraging UAVs for MCS. Zhou *et al*. [34] study the joint

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3014385, IEEE Transactions on Network Science and Engineering

energy-efficient trajectory planning and sensing task assignment problem for UAV-assisted MCS systems while the sensing latency and battery dynamics are taken into consideration. By applying deep RL (DRL) techniques, Zhang *et al*. [35] propose a UAV cruise route control algorithm for high-priority data sensing under dynamic environments, where driverless cars are employed to facilitate UAVs' battery recharging. Zhang *et al*. [36] investigate an energy-efficient mechanism for cellular UAV systems to optimize UAV sensing and data transmission under velocity and uplink rate constraints. Based on LSTM and proximal policy optimization, Piao and Liu [37] develop a novel sequential model for UAVs to optimize data collection ratio and reduce the energy consumption in executing sensing tasks. One can observe that safety and privacy challenges in UAV-assisted MCS are seldom studied in the existing literature.

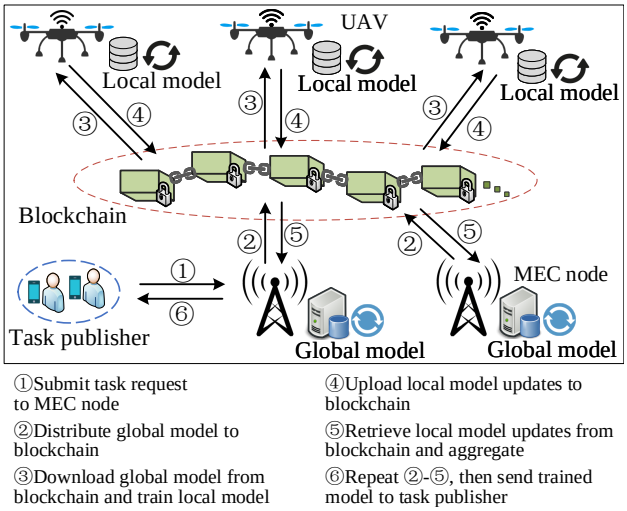


Fig. 2. An illustration of SFAC system.

B. Federated Learning in Wireless Networks

The emerging federated learning in wireless networks has attracted much attention from both academia and industry. Mills *et al*. [38] devise an adapting FedAvg algorithm via Adam optimization and model compression to mitigate communication overhead for Internet of things (IoT) devices during federated learning. Sattler *et al*. [39] develop a novel sparse ternary compression protocol to promote robustness for non-IID and unbalanced data in federated learning by compressing both the upstream and downstream communications. Zhan *et al*. [28] design a game theoretical model to stimulate the participation of IoT devices in federated learning, where a learning-based approach is also exploited for contribution evaluation. Chen *et al*. [40] propose a novel federated deep learning framework with temporally weighted aggregation and asynchronous parameter update to improve learning efficiency and relieve communication costs. Aimed to minimize the learning loss under resource budget constraints, Wang *et al*. [41] present a control algorithm in edge computing architecture to optimally balance the local model training and global model aggregation by dynamically adapting global aggregation frequency. However, to realize existing works on federated learning in reality, the joint security, privacy, and incentive concerns for UAVs in MCS should be further studied and resolved.

C. Blockchain for UAVs

In recent times, many efforts have been made on the security of UAV networks by leveraging the promising blockchain technology. Based on consortium blockchain, Qiu *et al*. [42] present a secure spectrum sharing scheme for UAV-aided cellular networks, where a Stackelberg game model is proposed to derive optimal spectrum trading strategies for network operator and UAVs. Asheralieva and Niyato [43] design a blockchain-as-a-service (BaaS) platform integrated with MEC for IoT devices, where UAVs serve as arial base stations for computation-intensive blockchain task offloading. Islam and Shin [44] investigate a secure data collection framework for MEC-enabled IoT networks, where flying UAVs operate as relay nodes for identity authentication before forwarding information to MEC servers. Zhu *et al*. [45] propose a blockchain-based decentralized architecture for data sharing in air-to-ground IoT network, where a Cournot model is formulated to attain maximum benefits for air and ground sensors. By exploiting private blockchain and mutual-healing protocol, Li *et al*. [46] develop a decentralized group key distribution mechanism for secure and efficient key recovery in UAV ad-hoc networks. While there are a number of blockchain approaches proposed for UAVs, few of them focus on the implementation of blockchain network under federated learning framework for MCS applications.

In this paper, different from existing works, the proposed scheme studies the secure and privacy-preserving federated learning for UAV-assisted MCS. A blockchain-based decentralized federated learning architecture with user privacy protection functions is presented to safeguard data training and contribution verification among UAVs. In addition, a reinforcement learning (RL)-based algorithm with two-tier is exploited to obtain the optimal strategies of both task publisher and worker UAVs in the dynamic environment, without the whole knowledge of accurate network parameters.

III. System Model

In this section, we introduce the system model including the network model, federated learning model, task model, mobility model, communication model, and threat model. Table I summarizes the notations used.

A. Network Model

Fig. 2 shows the scenario of blockchain-enabled federated learning for UAV-assisted MCS, which includes multiple UAVs, task publishers, MEC nodes, base stations, and a consortium blockchain.

Task publishers. The task publishers are users who submit sensing tasks to nearby MEC nodes with specific purposes such as air pollution monitoring and disaster rescue. A set of sensing tasks are hosted by MEC nodes, denoted as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3014385, IEEE Transactions on Network Science and Engineering

TABLE I
THE KEY NOTATIONS

Notations	Description
$\mathbb{J}$	The set of sensing tasks.
$\mathbb{I}_j$	The set of worker UAVs in task j.
$\mathbb{M}$	The set of MEC nodes in a given area of interest.
$\mathbb{T}$	The set of time slots.
$\mathbb{G}$	The set of sensors equipped on UAVs.
$sc_{i,g}, s_{i,j}$	Sensing capacity of UAV i on type g sensor/performing task j.
$\mathbf{l}_i(t)$	Instant 3D location of UAV i at time slot t.
$z_i, \overline{V}_i$	Hovering altitude/maximum flight velocity of UAV i in performing sensing mission.
$\mathbb{D}_{i,j}$	The dataset of local sensory data shared by UAV i in model training for task j.
$D_{i,j}$	The size of dataset $\mathbb{D}_{i,j}$.
Ψ_j^k	Global model parameters in k-th global training round of task j.
$\Psi_{i,j}^k$	Local model updates of UAV i in k-th global training round of task j.
$f(\Psi_j, \mathbf{x}_n, y_n)$	The loss function for training each data sample $d_n = \{\mathbf{x}_n, y_n\}$ of task j.
$\ell_i(\Psi_j)$	The loss function of UAV i in local training for task j.
$\ell(\Psi_j)$	The global loss function in data training for task j.
$\epsilon_{i,j}^k$	Allocated privacy budget of UAV i at k-th global training round for task j.
$\epsilon_{i,j}$	Total privacy budget of UAV i for task j.
$q_{i,j}$	Quality of local model update (QoLM) of worker UAV i in model training for task j.
$p_{i,j}$	Payment to UAV i for task j.
$\Lambda_{i,m}, b_{i,m}^{up}, \gamma_{i,m}$	Channel gain/uplink bandwidth/uplink transmission rate between UAV i and MEC node m.
$\mathbf{s}^t$	QoLM state vector of worker UAVs in task j at time slot t.
$\mathcal{Q}\left(s_i^t, p_{i,j}^t\right)$	Q-function at state s_i^t with action $p_{i,j}^t$.
$\tilde{s}_i^t$	Payment state of task publisher of task j at time slot t.
$\tilde{\mathcal{Q}}\left(\tilde{s}_i^t, q_{i,j}^t\right)$	Q-function at state $\tilde{s}_i^t$ with action $q_{i,j}^t$.

$\mathbb{J} = \{1, \cdots, j, \cdots, J\}$, where each task j is published to workers (i.e., UAVs) equipped with a set of required sensors $\mathbb{G}_j$ in the specified location loc_j. Distinguished from the centralized AI approaches, by applying federated learning, a set of distributed sensing data owned by UAVs can be utilized to train a globally shared model while the training data are kept on their local devices. As such, the data privacy of UAVs can be better preserved. For instance, instead of returning the raw images captured by UAVs in a disaster area, only the results of survivor detection or damage evaluation, which are learned from the trained model, are provided towards the task queries.

UAVs. UAVs equipped with diverse types of sensors (e.g., cameras, GPS, and humidity sensor) perform sensing tasks by collecting the sensing data from surrounding environments. Let $\mathbb{I} = \{1, \cdots, i, \cdots, I\}$ denote the set of UAVs in the network. Due to limited energy capacity and sensing range of UAVs, multiple UAVs involved in a common sensing task can operate collaboratively to cover the entire sensing area and ensure the data collection ratio [37]. Suppose that there are I_j UAVs involved in task j, and the set of which is denoted as $\mathbb{I}_j = \{1, \cdots, i, \cdots, I_j\}$. UAVs in $\mathbb{I}_j$ cooperatively train a global model Ψ_j by using their local sensing data and only upload the local model updates (i.e., weights or gradients) to the MEC node for aggregation. Assume that there are G types of sensors mounted on UAVs, the set of which is denoted as $\mathbb{G} = \{1, \cdots, g, \cdots, G\}$. Typically, UAVs have distinct sensing capacities on the same type of sensors. Let $sc_{i,g} \in [0, 1]$ be the sensing capacity of UAV i on type g sensor, where $sc_{i,g} = 1$ and $sc_{i,g} = 0$ mean the highest and lowest sensing capacity on type g sensor of UAV i, respectively. If UAV i is not equipped with type g sensor, we define $sc_{i,g} = 0, \forall i \in \mathbb{I}, \forall g \in \mathbb{G}$.

MEC nodes. The set of MEC nodes in a given area of interest is denoted as $\mathbb{M} = \{1, \cdots, m, \cdots, M\}$. Each MEC node is deployed at a base station to provide edge computing and wireless communication capacities for task publishers and UAVs in its coverage to facilitate federated learning services. The communication range of each MEC node m is considered as a circle with radius ra_m. MEC nodes can work in a collaborative way such that each published sensing task j can be assigned to an appropriate MEC node whose communication coverage matches the specified locations loc_j. In addition, MEC nodes are responsible for global model building via aggregation of UAVs' local model updates (e.g., taking weighted average in [41]).

Consortium blockchain. In consortium blockchain, only authorized nodes registered at the certification authority (CA) can participate in the decentralized network [43], [47]. The CA is a trusted agent (e.g., a government department) which is determined and recorded in the genesis block and serves for entity registration and key management in the network [48], [49]. Here, we consider two kinds of authorized nodes, namely, full node and lightweight node [7], [50].

- *Full nodes* store all the block data and serve as consensus nodes in the blockchain for new transaction verification, ledge management, and blockchain update.
- *Lightweight nodes* only store the header of blocks instead of the entire blockchain, and can download the latest blockchain data from full nodes. Lightweight nodes can generate, forward, and exchange transactions in the blockchain but not participate in the consensus process (e.g., mine blocks).

In our work, MEC nodes, which have powerful communication, computing and storage capabilities, act as full nodes; whereas UAVs perform as lightweight nodes because of their

resource constraints [42].

B. Federated Learning Model

For each task j, let $\mathbb{D}_{i,j}$ be the dataset of local training samples (i.e., sensing data) with size $D_{i,j} = |\mathbb{D}_{i,j}|$ owned by UAV i in training global model Ψ_j. Then, the set of overall training samples of UAVs involved in task j is denoted as $\mathbb{D}_j = \cup_{i=1}^{I_j} \mathbb{D}_{i,j}$, which can be described by a collection of input-output pairs $\{\mathbf{x}_n, y_n\}_{n=1}^{D_j}$. Here, $\mathbf{x}_n \in \Re^d$ is a column input vector with d-dimension (e.g., a captured image of disaster area), $y_n \in \Re$ is a scalar output value of $\mathbf{x}_n$ (e.g., the labeled output of a sensing image), and $D_j = \sum_{i=1}^{I_j} D_{i,j}$ is the total number of training samples for task j. Moreover, $\mathbb{D}_{i,j} \cap \mathbb{D}_{i',j} = \emptyset, \forall i, i' \in \mathbb{I}_j, i \neq i'$.

In a typical machine learning problem, given a collection of training data samples of task j, the loss function captures the error of model on training samples, and the target is to learn the model parameters Ψ_j which minimizes the loss function. For each data sample $d_n = \{\mathbf{x}_n, y_n\}$ of task j, the loss function is defined as $f(\Psi_j, \mathbf{x}_n, y_n)$, which we write as $f_n(\Psi_j)$ in short. The detailed loss function is usually defined according to the optimization models, such as linear regression, support vector machines, and convolutional neural networks (CNN) [41]. Then, for each UAV $i \in \mathbb{I}_j$ with dataset $\mathbb{D}_{i,j}$, its loss function is defined as:

$$\ell_i(\Psi_j) = \frac{1}{D_{i,j}} \sum\nolimits_{n=1}^{D_{i,j}} f_n(\Psi_j). \tag{1}$$

The aim of our federated learning is to train a global model $\Psi_j \in \Re^d$ for task $j \in \mathbb{J}$ which optimizes a global loss function $\ell(\Psi_j)$ on all the distributed datasets [40], i.e.,

$$\min_{\Psi_j} \ell(\Psi_j) = \sum\nolimits_{i=1}^{I_j} \frac{D_{i,j}}{D_j} \ell_i(\Psi_j). \tag{2}$$

Note that it is usually impossible to acquire a closed-form solution of Eq. (2). Hence, it is often solved through distributed gradient descent methods such as stochastic gradient descent (SGD) algorithm [39]. Since the training samples are distributed across a group of UAVs, to solve Eq. (2) by federated learning in blockchain context, there are the following three phases at k-th global training round ($k = 0, 1, 2, \cdots$).

1) At $k = 0$, the local model parameters $\Psi_{i,j}^k$'s of all UAVs are initialized to the same value.
2) *Local update.* For $k \geq 1$, each UAV $i \in \mathbb{I}_j$ computes its local model update $\Psi_{i,j}^k$ by training the previous global model Ψ_j^{k-1} on its local dataset via SGD, and then it uploads $\Psi_{i,j}^k$ to the blockchain for mutual verification.
3) *Global aggregation.* MEC node m updates the current global model Ψ_j^k by aggregating these local model updates retrieved from the blockchain.

Until the predefined accuracy is attained, the learning process stops and the trained model or learning result is delivered to the task publisher. Here, a *global training round* is defined in federated learning which consists of the distributed local update step on UAVs followed by a global aggregation step on MEC node.

C. Task Model

In the system, each specific task $j \in \mathbb{J}$ hosted on an MEC node m can be formulated as:

$$task_j = \left\langle ID_j || desc_j || loc_j || z_j || T_j^{\max} || \mathbb{G}_j || time_j || Sig_j \right\rangle, \tag{3}$$

where ID_j is the unique identity of task j, and $desc_j$ is the description of task j created by task publisher. loc_j and z_j are the specified horizontal location and altitude required for performing mission j, respectively. $T_j^{\max}$ is the time-to-live (TTL) of task j, $\mathbb{G}_j$ is the set of required sensor types for executing mission j, $time_j$ is publish time of task j, and Sig_j is the signature of task publisher.

Note that different UAVs have distinct contributions in global model training. According to [27], [30], a higher quality of sensing information and a larger training sample size of UAV i in executing task j can lead to higher quality of UAVs' local model updates, and a faster convergence of the local model in Eq. (1) and the global model in Eq. (2) to the target value. Therefore, the contributions of participating UAVs in federated learning tasks are related to their quality and quantity of contributed sensing data in local model training. To measure the contribution of UAVs within each task j, we define a *quality of local model update (QoLM)* metric in federated learning. Intuitively, the higher sensing capacity of sensors equipped in a UAV, the better precision of sensing data of its training samples. Therefore, it is reasonable to assume that the quality of sensing data of UAV i in performing mission j is positively related to its average sensing capacity of $\mathbb{G}_j$ sensors (i.e., $s_{i,j}$). Here, $s_{i,j} = \frac{1}{|\mathbb{G}_j|} \sum_{g \in \mathbb{G}_j} sc_{i,g}$. As such, the QoLM of UAV i within task j can be formulated as:

$$q_{i,j} = F(s_{i,j}, D_{i,j}). \tag{4}$$

The detailed design of function $F(.)$ is elaborated in the next section.

D. Mobility Model

Let $\mathbb{T} = \{1, \cdots, t, \cdots, T\}$ be the finite time horizon, which can be further divided into T time slot with equal length τ. As τ can be sufficiently small, the location of UAV i can be approximately fixed within each time slot. By applying the 3D Cartesian coordinate system, the instant location of UAV i at time slot t is described as:

$$\mathbf{l}_i(t) = [x_i(t), y_i(t), z_i], \forall t \in \mathbb{T}, \forall i \in \mathbb{I}, \tag{5}$$

where z_i is the fixed hovering altitude of UAV i for continuous flying without frequent descending and ascending when performing sensing missions [34]. In addition, in the working area, UAV i's start and end locations are preset and denoted as $\mathbf{l}_i(1)$ and $\mathbf{l}_i(T)$, respectively. Hence, within time horizon $\mathbb{T}$, the flying trajectory of UAV i can be expressed as:

$$\begin{cases} \mathbf{l}_i = \{\mathbf{l}_i(1), \cdots, \mathbf{l}_i(t), \cdots, \mathbf{l}_i(T)\}, & (6) \\ \text{s.t.} \left\| \mathbf{l}_i(t+1) - \mathbf{l}_i(t) \right\| \leq \tau \overline{V}_i, 1 \leq t < T, & (7) \end{cases}$$

where formula (7) means the mobility constraint of UAV i, and $\overline{V}_i$ denotes UAV i's maximum flight velocity.

E. Communication Model

In the task area, worker UAVs can cooperatively transmit their local model updates to MEC nodes through multi-hop air-to-air (A2A) communications and air-to-ground (A2G) communications. Meanwhile, UAVs can receive the latest parameters of global model recorded in the blockchain from MEC node via ground-to-air (G2A) communications [7]. The A2G and G2A path loss between UAVs and base stations is dominated by line-of-sight (LoS) transmission, which can be described by the quasi-static block fading channel model [45]. In LoS link, the wireless channel keeps constant during each fading block and the power attenuation is distance-dependent. As such, the channel gain between UAV i and MEC node m within time slot t can be formulated as:

$$\Lambda_{i,m}(t) = \psi_0 \left[d_{i,m}(t)\right]^{-\varrho} = \frac{\psi_0}{\left[\left(x_i(t) - x_m\right)^2 + \left(y_i(t) - y_m\right)^2 + \left(z_i - z_m\right)^2\right]^{\varrho/2}}, \quad (8)$$

where ψ_0 is the channel power gain at a reference distance, $d_{i,m}(t)$ is the Euclidean distance between UAV i and MEC node m, $[x_m, y_m, z_m]$ is the 3D location of base station m, and $\varrho > 1$ is the path loss exponent [45]. Let $b_{i,m}^{up}$ be the allocated uplink bandwidth between UAV i and MEC node m, and P_i be the transmit power of UAV i. Then, the available uplink data transmission rate between UAV i and MEC node m is

$$\gamma_{i,m} = b_{i,m}^{up} \log_2\left(1 + \frac{P_i \Lambda_{i,m}(t)}{b_{i,m}^{up} \varphi_0}\right), \quad (9)$$

where φ_0 means the power spectral density of the additive white Gaussian noise [51], [52].

F. Threat Model

Compared with centralized AI methods, federated learning mitigates the data privacy concerns by splitting the learning phase into local training and global aggregation. Nevertheless, it brings some new security issues as well. We define the following threats on UAVs' privacy and security in MCS during federated learning process.

Privacy Leakage Attack. In distributed learning process, UAVs' updated local model parameters may still leak information about the certain data that was used during training. In addition, attackers can infer whether a UAV participated in the certain task from their local model updates through differential attacks. Since each mission has specified sensing locations, the location privacy of involved UAVs may be disclosed.

Low-quality Local Model Update Attack. We assume that participating UAVs are selfish and rational, whose targets are to maximize their own benefits. If there is not enough compensations for their costs in local training, UAVs may contribute low-quality local model updates to reduce cost. Consequently, the accuracy of the global model will be deteriorated, which may cause a failure of the whole learning process and prevent participation of other honest UAVs.

Contribution Records Tampering Attack. Since the records of UAVs' contributions in the learning phase are traced and stored in the central curator (i.e., MEC node), adversaries may launch a variety of attacks (e.g., single point failure attack and DDoS attack) to delete, tamper, forge, and replace these contribution records for monetary purposes. For instance, an attacker may illegally increase his/her portion in mission reward or impersonate another node with high contribution by tampering with the contribution records.

IV. Proposed SFAC Scheme

In this section, we propose SFAC to address secure federated learning in UAV-assisted MCS. We first introduce the blockchain-based decentralized federated learning mechanism for secure local model sharing and contribution recording for UAVs. Then, the privacy-preserving local model update algorithm is devised to protect UAVs' privacy of participation and shared local models during federated learning. After that, with the formulation of utility functions for UAVs and task publishers, we present an RL-based incentive mechanism to motivate UAVs' high-quality local model contribution in collaborative model learning under highly dynamic environment.

A. Blockchain Based Decentralized Federated Learning

Due to the intermittent and unreliable wireless channels in UAV networking and security vulnerabilities of the central curator, a decentralized federated learning mechanism is developed based on blockchain, aimed to secure the local model update sharing and the global aggregation process. In the blockchain, each block is divided into its header and body parts. The header of block mainly consists of a hash pointer to the previous block (i.e., $preHash$), a solution of the proof-of-work (PoW) puzzle (i.e., $nonce$), block generation rate λ_b, and block producer. The block body stores a series of valid transactions in the network. The following three types of transactions are considered in the blockchain.

- *Task request transaction (trTx).* A trTx records a specific task publishing event with all involved entities, task information, and initial model parameters.
- *Local model update transaction (lmTx).* An lmTx records the updated local model parameters of a UAV in a global training round.
- *Aggregated global model transaction (gmTx).* A gmTx records the aggregated global model parameters of the MEC node in a global training round.

In addition, we consider two types of blocks, i.e., ordinary block (orBlock) and local model update block (lmBlock).

- An *ordinary block (orBlock)* stores the valid trTx and gmTx transactions during a consensus epoch ΔT_c.
- The special *local model update block (lmBlock)* records the lmTx transactions of UAVs involved in a specific task.

The detailed implementation of decentralized federated learning to solve Eq. (2) is presented with the following phases.

1) Entity Registration and Role Selection. After registration at CA, each UAV and MEC node become authorized in the network by binding their true identities, e.g., the registration certificate of UAV and the business license of MEC node issued by the government. Each authorized node u obtains its public/private key pair (PK_u, SK_u), wallet address W_u,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3014385, IEEE Transactions on Network Science and Engineering

and certificate Cer_u. CA maintains all registered nodes and their corresponding identities. In the blockchain network, each authorized MEC node selects its role as the full node, while each authorized UAV serves as the lightweight node.

2) Task Publishing and Learning Initialization. The task publisher can submit its sensing task $task_j$ with its requirements to a nearby MEC node. The MEC node m whose communication coverage matches the specified sensing locations loc_j hosts the corresponding task j. For each task j, a set of $\mathbb{I}_j$ worker UAVs are selected. For learning initialization, the MEC node chooses an appropriate type of global AI model $\hbar_j$ for task j, and initializes the model parameters Ψ_j^0. Then, MEC node m generates a trTx transaction and broadcasts it to the network. The form of trTx is:

$$trTx = \left\langle task_j || time || \{PK_i\}_{i=1}^{I_j} || \hbar_j || \Psi_j^0 || H(trTx) || Sig_m || mSig_{\mathbb{I}_j} \right\rangle, \tag{10}$$

where $task_j$ is the task j defined in Eq. (3), $time$ is the creation time of trTx, $H(.)$ is the secure hash function, Sig_m is the signature of MEC node m, and $mSig_{\mathbb{I}_j}$ is the multi-signature of UAVs in $\mathbb{I}_j$. Other MEC nodes who receive the trTx transaction will verify its correctness independently. Invalid transactions are discarded, while valid ones are packaged and stored in its candidate orBlock. After reaching consensus, all valid trTx transactions included in the orBLock are immutably recorded in the blockchain.

3) Local Model Training and Perturbation on UAVs. At a global training round k ($1 \leq k \leq K_j$), after downloading the previous global model Ψ_j^{k-1} from blockchain, each UAV $i \in \mathbb{I}_j$ adopts the SGD algorithm to train a local model $\Psi_{i,j}^k$ which minimizes $\ell_i(\Psi_j)$ by taking several steps of mini-batch gradient descent. K_j is the maximum number of global training rounds for task j. Here, UAV i's local model (i.e., the average gradient on its local training data) can be updated by the following equation:

$$\Psi_{i,j}^k = \Psi_j^{k-1} - \varphi \nabla \ell_i \left(\Psi_j^{k-1}\right), \tag{11}$$

where φ is the learning rate of SGD. It is notable that the recorded data in the blockchain are transparent to all authorized nodes. To defend against the privacy leakage attack defined in Sect. III-F, each UAV $i \in \mathbb{I}_j$ applies the LDP mechanism [30] to protect its privacy of local model updates. Specifically, instead of uploading the true local model updates $\Psi_{i,j}^k$, each UAV $i \in \mathbb{I}_j$ generates a sanitized version $\tilde{\Psi}_{i,j}^k$ in the lmTx transaction. LDP mechanisms ensure that attackers with arbitrary background knowledge cannot distinguish the true model updates of any UAV from others with strong probability, whereas the MEC node can still attain accurate inference from the aggregate of sanitized data [30]. The detailed local model perturbation mechanism is elaborated in the next subsection. Here, the form of lmTx generated by UAV i is:

$$lmTx = \left\langle \tilde{\Psi}_{i,j}^k || time || H(trTx) || H(lmTx) || Sig_i \right\rangle, \tag{12}$$

where $time$ is the creation time of lmTx, and Sig_i is the signature of UAV i. Each MEC node verifies the received local model updates and records the verified ones related with a task j into a special candidate lmBlock via the filling procedure [32]. The filling procedure ends until this candidate lmBlock reaches the block size I_j (i.e., the number of lmTx transactions included in lmBlock) or a maximum waiting time ΔT_w. All valid lmTx transactions associated with task j at k-th global training round are included in an lmBLock $lmBlock_j$, which are recorded in the blockchain after reaching consensus.

4) Global Model Aggregation on MEC Node. The MEC node m fetches the local model updates from $lmBlock_j$ and performs global aggregation via the weighted aggregation of all local model parameters. Note that a higher QoLM of participating UAVs can lead to a faster convergence rate of federated learning process and an improved learning efficiency in terms of less training time and less energy consumption. Consequently, the updated local model with a higher QoLM should be assigned with a larger weight in the aggregated result. Then, the global model can be aggregated by the following equation:

$$\Psi_j^k = \sum\nolimits_{i=1}^{I_j} \frac{q_{i,j}}{\sum_{i=1}^{I_j} q_{i,j}} \tilde{\Psi}_{i,j}^k. \tag{13}$$

Then, MEC node m generates a gmTx transaction as:

$$gmTx = \left\langle \Psi_j^k || time || H(trTx) || H(gmTx) || H(lmBlock_j) || Sig_m \right\rangle, \tag{14}$$

where Ψ_j^k is the current global model, $time$ is the creation time of gmTx, and Sig_m is the signature of MEC node m. All valid gmTx transactions can be recorded into the blockchain when the consensus is reached. The above local training and global aggregation process is repeated until the global model obtains a desirable accuracy ϑ, i.e., $\left\| \ell\left(\Psi_j^k\right) \right\| \leq \vartheta \left\| \ell\left(\Psi_j^{k-1}\right) \right\|$.

Furthermore, to efficiently reach consensus in blockchain network, the following procedures need to undertake.

1) Block Generation. After building its candidate block $\mathcal{B}$ (i.e., orBlock or lmBlock) within a consensus epoch, each MEC node competes against each other to solve the PoW puzzle, i.e.,

$$H\left(preHash || H\left(bHeader\right) || nonce\right) \leq \sigma, \tag{15}$$

where $bHeader$ is the header of current block, $nonce$ is a random number indicating the mining results, and σ is the PoW target hash value. The PoW difficulty can be controlled by the block generation rate λ_b, i.e., the smaller λ_b, the lower σ and the higher PoW difficulty. The winner of mining competition (i.e., the first one to solve the PoW puzzle) becomes the block producer and earns the relevant mining rewards. The block producer (i.e., MEC node $\bar{m} \in \mathbb{M}$) broadcasts its block proposal $\mathcal{B}_{\bar{m}}$ which includes its miming result $nonce_{\bar{m}}$ and signature $Sig_{\bar{m}}$ to other MEC nodes.

2) Block Propagation. During block propagation phase, the time needed for a block proposal $\mathcal{B}_m$ to reach consensus in the blockchain network is jointly determined by the block dissemination delay T_m^d and the block verification delay T_m^v. According to [53], the average time for a block $\mathcal{B}_m$ to reach consensus can be modeled as a function of the block size π_m (i.e., the number of transactions packaged in $\mathcal{B}_m$), and is defined as:

$$T_{bp}(\pi_m) = T_m^d + T_m^v = \frac{\pi_m}{\mu \kappa_1} + \kappa_2 \pi_m, \tag{16}$$

where μ is the average effective channel capacity among MEC nodes, κ_1 is the network scale coefficient, and κ_2 is the coefficient decided by both the scale of network and the average verification speed of each MEC node. Once receiving $\mathcal{B}_m$, each MEC node audits the correctness of the PoW solution and all transactions in the received block proposal. If verification passes, each MEC node adds the newly created block $\mathcal{B}_m$ to its local blockchain linearly and chronologically. Furthermore, due to the latency of the broadcast of block $\mathcal{B}_m$ over blockchain, another MEC node $m' \in \mathbb{M}$, $m' \neq m$, may succeed in solving the PoW puzzle within $T_{bp}(\pi_m)$ and generates its block proposal $\mathcal{B}_{m'}$. As such, part of MEC nodes may mistakenly append the secondly generated block $\mathcal{B}_{m'}$ to their local blockchains, and a forking of blockchain occurs. By modeling the occurrence of solving the PoW puzzle as a Poisson process [54], the probability of experiencing a blockchain forking for an MEC node m can be formulated as:

$$\Pr_m^{fork} = 1 - e^{-\lambda_b \cdot T_{bp}(\pi_m)}, \tag{17}$$

where λ_b represents the expected block arrival rate of the Poisson process (e.g., $\lambda_b = \frac{1}{600\,\text{sec}}$ in Bitcoin). Eq. (17) shows that the forking frequency increases with the block generation rate λ_b and the block propagation delay $T_{bp}(\pi_m)$.

3) Contribution Recording and Rewarding. Let $c_{i,j}$ denote the contribution of UAV i in training Ψ_j, where $\sum_{i\in\mathbb{I}_j} c_{i,j} = 1$. The rewards in the blockchain contains three types, i.e., local model training reward, global model aggregation reward, and mining reward. The local model training reward $p_{i,j}$ in task j is provided to each involved UAV $i \in \mathbb{I}_j$ for collecting sensing data and performing local model training. The global model aggregation reward fee_m is provided to the MEC node for synthesizing the global model. The mining reward p_{mine} is earned by the block producer which generates a valid block in the longest blockchain.

As analyzed in Eq. (4), the QoLM (i.e., $q_{i,j}$) of UAV $i \in \mathbb{I}_j$ during training global model Ψ_j is positively related to UAV i's sensing capacity $s_{i,j}$ and training sample size $D_{i,j}$. Experiments in [28] show that, the QoLM function F of the trained model can be modeled as a concave function with respect to the amount of training data $D_{i,j}$, i.e., $\frac{dF(D_{i,j})}{dD_{i,j}} \geq 0$, and $\frac{d^2F(D_{i,j})}{dD_{i,j}{}^2} < 0$. Based on [28], the widely used natural logarithmic function is employed to formulate $F(D_{i,j})$. Therefore, the QoLM of UAV i in task j can be defined as:

$$q_{i,j} = F\left(s_{i,j}, D_{i,j}\right) = \xi_j\, s_{i,j} \ln\left(1 + D_{i,j}\right), \tag{18}$$

where $\xi_j = \frac{1}{\ln\left(1+D_j^{\max}\right)}$ is a positive normalization factor, and $D_j^{\max}$ is the predefined maximum training sample size of a UAV in executing task j. Let $D_{i,j}^{\max}$ be the maximum number of training samples of UAV i for performing task j. Then, the maximum QoLM of UAV i in task j is $q_{i,j}^{\max} = \xi_j s_{i,j} \ln\left(1 + D_{i,j}^{\max}\right)$. Accordingly, we have $0 \leq q_{i,j} \leq q_{i,j}^{\max} \leq 1$. Then the contribution of UAV i in task j can be derived by $c_{i,j} = \frac{q_{i,j}}{\sum_{i\in\mathbb{I}_j} q_{i,j}}$. When the global model attains a predefined accuracy level, the collaborative federated learning process ends and each UAV uploads its QoLM records into the blockchain. The criteria for evaluating the QoLM of UAVs are given by the system and are known for all the nodes. Each MEC node will independently verify these QoLM records and calculate the contribution distributions for all the tasks. After the consensus is reached, the contributions of UAVs can be immutably recorded in the blockchain.

Security Analysis. By dynamically changing public keys in different transactions to hide the true identity, the anonymity and unlinkability of UAVs and task publishers can be attained. Due to the special hash-chained data structure of the blockchain which is maintained by all MEC nodes, if adversaries attempt to tamper with the data (e.g., UAVs' contribution records) in one block, they need to recalculate the PoW puzzles of this block and all subsequent ones. Hence, the *contribution records tampering attack* can be addressed well.

B. Privacy-Preserving Local Model Update

To improve the accuracy in global model aggregation while ensuring LDP, a privacy-preserving local model update sharing mechanism is developed for participating UAVs. First, to protect UAVs' privacy of their shared local model updates and their participation in the task, each participating UAV utilizes a perturbation function $\mathcal{F}$ and uploads obfuscated local model parameters to the blockchain.

Definition 1. By applying a randomized perturbation function $\mathcal{F}$, participating UAV $i \in \mathbb{I}_j$ can achieve $\epsilon_{i,j}^k$-LDP when submitting its local model update at k-th global training round for task j, if the following inequality holds:

$$\Pr\left[\mathcal{F}\left(\Psi_{i,j}^k\right) = \tilde{\Psi}_{i,j}^k\right] \leq e^{\epsilon_{i,j}^k} \Pr\left[\mathcal{F}\left(\Psi_{i,j}^{k\ \prime}\right) = \tilde{\Psi}_{i,j}^k\right], \tag{19}$$

where $\epsilon_{i,j}^k$ is the allocated privacy budget of UAV i at k-th round for task j, which measures UAV i's privacy loss in sharing local trained model $\Psi_{i,j}^k$. $\Psi_{i,j}^k$ is the actual local model update of UAV i, $\tilde{\Psi}_{i,j}^k$ is the obfuscated local model parameters of UAV i, and $\Psi_{i,j}^{k\ \prime}$ is the data that has γ_j-adjacency with $\Psi_{i,j}^k$ (i.e., $\left\|\Psi_{i,j}^{k\ \prime} - \Psi_{i,j}^k\right\| \leq \gamma_j$). Here, γ_j is the adjacency parameter denoting whether two local model updates are γ_j-adjacent or not, and $\|\cdot\|$ stands for the 1-norm.

According to LDP mechanism, the Laplace distribution is employed to design the perturbation function $\mathcal{F}$ by injecting a random noise to the actual local model update, which can be expressed as:

$$\tilde{\Psi}_{i,j}^k = \mathcal{F}\left(\Psi_{i,j}^k\right) = \eta_{i,j}^k + \Psi_{i,j}^k, \tag{20}$$

where $\eta_{i,j}^k$ is the random noise injected to $\Psi_{i,j}^k$ and it follows the Laplace distribution with scale parameter $\frac{\gamma_j}{\epsilon_{i,j}^k}$, i.e., $\eta_{i,j}^k \sim Lap\left(0, \frac{\gamma_j}{\epsilon_{i,j}^k}\right)$.

Privacy Analysis. Note that in Eq. (20), a smaller privacy budget implies a larger injected noise on the updated local model and a smaller risk of privacy violation. In this manner, each UAV i can fully control its privacy by independently sanitizing its local model parameters to an extent that matches its privacy preference $\epsilon_{i,j}$ for task j. The added random noises can protect user privacy from being disclosed, however, attackers can still exactly deduce a UAV's private and sensitive information by collecting all local model updates at all global training rounds. For a total of K_j global training rounds, each

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3014385, IEEE Transactions on Network Science and Engineering

UAV i involved in task j should restrict its total privacy budget $\epsilon_{i,j}$ such that $\epsilon_{i,j} = \sum_{k \in K_j} \epsilon_{i,j}^k$. For simplicity, given a fixed K_j, the privacy budget is evenly split across each round, i.e., $\epsilon_{i,j}^k = \epsilon_{i,j}/K_j$, for any $1 \le k \le K_j$. As such, both MEC nodes and eavesdroppers cannot deduce the true information and data source of any UAV from the sanitized version with strong probability. Hence, the *privacy leakage attack* can be settled well.

Apart from the privacy protection for participating UAVs, the data utility (i.e., aggregation accuracy) during federated learning process should be ensured. Nevertheless, the random noise injected in local model updates inevitably impairs the accuracy of aggregated result (i.e., the synthetic global model). The (α, β)-accuracy is defined to quantize the data utility of aggregated results, which is given in Definition 2.

Definition 2. The aggregated global model at k-th round for task j achieves $\left(\alpha_j^k, \beta_j^k\right)$-accuracy, if the following inequality holds:

$$\Pr\left[\left\|\tilde{\Psi}_j^k - \Psi_j^k\right\| \le \alpha_j^k\right] \ge \beta_j^k, \tag{21}$$

where $\tilde{\Psi}_j^k = \sum_{i=1}^{I_j} c_{i,j} \tilde{\Psi}_{i,j}^k$ is the aggregated global model from UAVs' perturbed local model updates, and $\Psi_j^k = \sum_{i=1}^{I_j} c_{i,j} \Psi_{i,j}^k$ is the aggregation result of actual local model updates of UAVs. α_j^k stands for the confidence interval and β_j^k is the confidence level of task j at k-th round. Eq. (21) indicates that the aggregation error is less than or equal to α_j^k with probability at least β_j^k. Apparently, for a given β_j^k, a smaller α_j^k indicates a better aggregation accuracy, namely, a smaller difference between the actual aggregated result and the obfuscated one. Thus, it is reasonable to employ the confidence interval α_j^k to measure the *aggregation error* in k-th global model aggregation, i.e., $\Im_j^k = \left\|\tilde{\Psi}_j^k - \Psi_j^k\right\|$.

In Theorem 1, we give the quantitative relationship between UAV's privacy preservation in local model training and MEC node's aggregation accuracy in global model learning.

Theorem 1. At k-th global model aggregation for task j, given the confidence level $\beta_j^k \le 1$, the confidence interval α_j^k is given by:

$$\alpha_j^k = \sqrt{2}\gamma_j \sqrt{\frac{\sum_{i=1}^{I_j} \left(\frac{q_{i,j}}{\sum_{i=1}^{I_j} q_{i,j}}\right)^2 \left(\epsilon_{i,j}^k\right)^{-2}}{1 - \beta_j^k}}. \tag{22}$$

Proof: The aggregation error of k-th global model aggregation for task j can be denoted as

$$\begin{aligned}\tilde{\Psi}_j^k - \Psi_j^k &= \sum\nolimits_{i=1}^{I_j} c_{i,j}\left(\Psi_{i,j}^k + \eta_{i,j}^k\right) - \sum\nolimits_{i=1}^{I_j} c_{i,j}\Psi_{i,j}^k \\ &= \sum\nolimits_{i=1}^{I_j} c_{i,j}\eta_{i,j}^k. \end{aligned} \tag{23}$$

Note that the variance of Laplace noise $\eta_{i,j}^k \sim Lap\left(0, \frac{\gamma_j}{\epsilon_{i,j}^k}\right)$ equals to $2(\frac{\gamma_j}{\epsilon_{i,j}^k})^2$. Since the random noises are generated by the Laplace distribution, the mean value of $\tilde{\Psi}_j^k - \Psi_j^k$ is zero and its variance is

$$Var\left(\tilde{\Psi}_j^k - \Psi_j^k\right) = \sum\nolimits_{i=1}^{I_j} \left(c_{i,j}\right)^2 \cdot 2\left(\gamma_j \frac{1}{\epsilon_{i,j}^k}\right)^2. \tag{24}$$

According to Chebyshev's inequality [55], we have

$$\Pr\left[\left\|\tilde{\Psi}_j^k - \Psi_j^k\right\| \le \alpha_j^k\right] \ge 1 - \frac{2}{(\alpha_j)^2} \sum\nolimits_{i=1}^{I_j} \left(\frac{c_{i,j}\gamma_j}{\epsilon_{i,j}^k}\right)^2. \tag{25}$$

Therefore, according to Definition 2, the confidence level β_j^k can be denoted as $\beta_j^k = 1 - \frac{2}{(\alpha_j)^2} \sum_{i=1}^{I_j} (\frac{c_{i,j}\gamma_j}{\epsilon_{i,j}^k})^2$. As such, we can attain the detailed expression of α_j^k, as given in Eq. (22). Theorem 1 is proved. ■

C. Utility Analysis

Utility function of task publisher. The utility function of task publisher in task j can be defined as the difference between its satisfaction and the payment for task j, i.e.,

$$\mathbf{U}_j\left(\mathbf{p}_j\right) = \omega \mathcal{S}\left(\mathbf{q}_j\right) - \left(1 - \omega\right)\omega_p \left(\sum\nolimits_{i=1}^{I_j} p_{i,j} q_{i,j} + fee_m\right), \tag{26}$$

where $\mathbf{p}_j = \left(p_{1,j}, \cdots, p_{i,j}, \cdots, p_{I_j,j}\right)$ is the payment vector of task publisher for mission j, and $\mathbf{q}_j = \left(q_{1,j}, \cdots, q_{i,j}, \cdots, q_{I_j,j}\right)$ is the QoLM vector of UAVs in set $\mathbb{I}_j$ during the learning process for task j, $\mathcal{S}(.)$ means the satisfaction function of task publisher by collecting UAVs' local model updates with QoLM $q_{i,j}$'s, ω is the weight factor of its satisfaction, and ω_p is the adjustment factor to balance the satisfaction and payment. Here, the satisfaction function $\mathcal{S}(.)$ contains two parts, i.e., the accumulated QoLM of participating UAVs and the accuracy of aggregated global model. Then, we have

$$\mathcal{S}\left(\mathbf{q}_j\right) = \mu \sum\nolimits_{i=1}^{I_j} q_{i,j} - \nu\omega_l \sum\nolimits_{k=1}^{K_j} \alpha_j^k, \tag{27}$$

where μ and ν are two weight factors for accumulated QoLM of UAVs and the aggregation error, respectively. ω_l is the adjustment factor to balance the accumulated QoLM and aggregation error. Thus, the explicit utility function of the task publisher is shown as:

$$\begin{aligned}\mathbf{U}_j\left(\mathbf{p}_j\right) = \omega \Bigg\{ & \mu \sum\nolimits_{i=1}^{I_j} q_{i,j} - \nu\,\omega_l \times \\ & \sum\nolimits_{k=1}^{K_j} \sqrt{2}\gamma_j \sqrt{\frac{\sum_{i=1}^{I_j} \left(\frac{q_{i,j}}{\sum_{i=1}^{I_j} q_{i,j}}\right)^2 \left(\epsilon_{i,j}^k\right)^{-2}}{1 - \beta_j^k}} \Bigg\} \\ & - \left(1 - \omega\right)\omega_p \left(\sum\nolimits_{i=1}^{I_j} p_{i,j} q_{i,j} + fee_m\right), \end{aligned} \tag{28}$$

$$\text{s.t.} \begin{cases} \Pr\left[\Im_j^k \le \alpha_j^k\right] \ge \beta_j^k, & (29) \\ t_j \le T_j^{\max}, & (30) \\ p_{i,j} \le p_j^{\max}, & (31) \end{cases}$$

where formula (29) is the aggregation accuracy constraint, formula (30) is the execution time constraint, t_j is the overall time for executing task j, formula (31) is the price constraint, and $p_j^{\max}$ is the maximum payment to a worker UAV in task j.

Utility function of UAVs. The utility function of UAV $i \in \mathbb{I}_j$ can be denoted as its revenue $p_{i,j}q_{i,j}$ minus its cost $\mathcal{C}_{i,j}$, i.e.,

$$\mathbf{U}_{i,j}\left(q_{i,j}\right) = \varpi p_{i,j} q_{i,j} - \left(1 - \varpi\right)\lambda_c \mathcal{C}_{i,j}, \tag{32}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3014385, IEEE Transactions on Network Science and Engineering

where ϖ is the weighted factor of the revenue, and λ_c is the adjustment factor to balance the revenue and cost. During federated learning process, the cost of UAV i in performing task j (i.e., $\mathcal{C}_{i,j}$) consists of the sensing cost $C^s_{i,j}$, the privacy cost $C^p_{i,j}$, and the energy consumption cost of local model training $E^{tr}_{i,j}$ and local model uploading $E^{up}_{i,j}$.

Intuitively, the higher sensing capacity and larger amount of local sensing data in task execution, the higher sensing cost of a UAV. Thus, the sensing cost of UAV i in performing task j can be considered to be directly proportional to its QoLM, i.e.,

$$C^s_{i,j} = c^s_{i,j} \cdot q_{i,j}, \tag{33}$$

where $c^s_{i,j}$ is the sensing cost of UAV i to execute task j with the highest QoLM (i.e., $q_{i,j} = 1$). Commonly, different UAV users may have distinct privacy preferences, and even a user may suffer distinct privacy losses in performing different sensing tasks. Here, we define $c^p_{i,j}$ as the unit cost of UAV i for privacy leakage, which indicates how much UAVs care about their privacy losses. The privacy cost of UAV i is in direct proportion to its privacy budget $\epsilon_{i,j}$, i.e.,

$$C^p_{i,j} = c^p_{i,j} \cdot \epsilon_{i,j}. \tag{34}$$

Thirdly, according to [27], the energy consumption cost of UAV i in local model training can be expressed as

$$E^{tr}_{i,j} = \sum\nolimits_{k=1}^{K_j} \varsigma\, L^k_{i,j} \chi_i D_{i,j} (f_i)^2, \tag{35}$$

where ς is an effective capacitance factor, $L^k_{i,j}$ is the number of local training iterations of UAV i in k-th round of task j, χ_i is the number of CPU cycles to perform one data sample in local training, and f_i is UAV i's CPU cycle frequency implying the utilized computing resources in local training. Lastly, the energy consumption of UAV i in local model uploading can be derived as

$$E^{up}_{i,j} = \sum\nolimits_{k=1}^{K_j} \frac{\Psi^k_{i,j}}{\gamma_{i,m}} \cdot P_i. \tag{36}$$

Then the utility function of UAV $i \in \mathbb{I}_j$ can be rewritten as:

$$\mathbf{U}_{i,j}\left(q_{i,j}\right) = \varpi p_{i,j} q_{i,j} - (1-\varpi)\,\lambda_c \left\{ \phi_1 c^s_{i,j}\, q_{i,j} + \phi_2 c^p_{i,j}\, \epsilon_{i,j} + \phi_3 \left(\sum\nolimits_{k=1}^{K_j} \varsigma\, L^k_{i,j} \chi_i D_{i,j} f_i^2 + \sum\nolimits_{k=1}^{K_j} \frac{\Psi^k_{i,j}}{\gamma_{i,m}} P_i \right) \right\}, \tag{37}$$

$$\text{s.t. } 0 \le q_{i,j} \le q^{\max}_{i,j}, \tag{38}$$

where $\phi_k\,(k = 1, 2, 3)$ are positive adjustment factors, and formula (38) is the QoLM constraint of UAV i.

D. RL Based Incentive Mechanism

In practice, the accurate parameters of cost model and network model during federated learning process might not be available for participating UAVs and task publishers. For UAVs and the task publisher, the RL approaches such as Q-learning are exploited to derive their optimal QoLM and pricing strategies via trials when accurate system parameters not readily available.

Q-learning based pricing strategy making. In general, a high payment for model training decreases the task publisher's immediate utility while it motivates more UAVs to provide high QoLM in the future. Apparently, the task publisher's current payment strategy affects the future learning accuracy and its future benefit. Since the pricing strategy making process can be modeled as a finite MDP, the Q-learning, which is a model-free RL method, can be employed to enable the task publisher to attain its optimal pricing strategy without the explicit private parameters of participating UAVs. At each time slot t, the task publisher observes the system state vector $\mathbf{s}^t = \left(s^t_1, \cdots, s^t_i, \cdots, s^t_{I_j}\right)$ which consists of the previous QoLM sequences of participating UAVs, i.e., $\mathbf{s}^t = \mathbf{q}^{t-1}_j$. For simplicity, the task publisher uniformly quantizes the payment into $H+1$ levels and selects its payment level $p \in \{\frac{h}{H} \cdot p^{\max}_j\}_{0 \le h \le H}$. Let $\mathcal{Q}\left(\mathbf{s}^t, \mathbf{p}^t_j\right)$ denote the Q-function at state $\mathbf{s}^t$ with action vector $\mathbf{p}^t_j = \left(p^t_{1,j}, \cdots, p^t_{i,j}, \cdots, p^t_{I_j,j}\right)$. To attain a tradeoff between exploration and exploitation in the learning process, the task publisher applies the δ-greedy policy by behaving greedily (i.e., select the payment strategy which maximizes the Q-function) with a high probability $1-\delta$, while randomly picks other actions (which may lead to better future return) with a small probability δ. Here, δ lies in $(0, 1]$. The payment policy $p^t_{i,j}$ at time slot t is given by

$$\Pr\left[p^t_{i,j} = p^*_{i,j}\right] = \begin{cases} 1-\delta, & p^*_{i,j} = \arg\max_{p_{i,j}} \mathcal{Q}\left(s^t_i, p_{i,j}\right), \\ \delta, & \text{otherwise.} \end{cases} \tag{39}$$

At each time slot t with state $\mathbf{s}^t$, after the task publisher chooses the payment action $\mathbf{p}^t_j$ and observes a reward (i.e., its utility $\mathbf{U}_j\left(\mathbf{s}^t, \mathbf{p}^t_j\right)$), a new state vector $\mathbf{s}^{i+1}$ is generated in the system. The value function is denoted as $\mathcal{V}\left(\mathbf{s}^t\right)$, which implies the highest Q-function at state $\mathbf{s}^t$. Based on the iterative Bellman equation, the task publisher updates its Q-function over time slots as below:

$$\mathcal{Q}\left(s^t_i, p^t_{i,j}\right) \leftarrow \mathcal{Q}\left(s^t_i, p^t_{i,j}\right) + \zeta_1 \left\{ \mathbf{U}_j\left(s^t_i, p^t_{i,j}\right) + \rho_1 \mathcal{V}\left(s^{t+1}_i\right) - \mathcal{Q}\left(s^t_i, p^t_{i,j}\right) \right\}, \tag{40}$$

$$\mathcal{V}\left(s^{t+1}_i\right) \leftarrow \max_{p_{i,j}} \mathcal{Q}\left(s^{t+1}_i, p^{t+1}_{i,j}\right), \tag{41}$$

where $\zeta_1 \in (0, 1]$ is the learning rate of pricing strategy, and $\rho_1 \in [0, 1]$ is the discount factor implying the importance of the future rewards over the immediate one. The Q-learning based pricing strategy making algorithm for task publisher is summarized in Algorithm 1.

Q-learning based QoLM strategy making. The QoLM strategy making process can be formulated as an MDP with finite states. Since each UAV has little explicit knowledge of the task publisher's private parameters, its optimal QoLM strategy can not be immediately derived. The UAV $i \in \mathbb{I}_j$ applies Q-learning to find its optimal QoLM strategy via trial and error. The state $\tilde{s}^t_i$ for UAV i in the Q-learning is composed of the previous payment of the task publisher, i.e., $\tilde{s}^t_i = p^{t-1}_{i,j}$. For simplicity, each UAV uniformly quantizes the QoLM into $N+1$ levels and selects the QoLM level $q_{i,j} \in \{\frac{n}{N} \cdot q^{\max}_{i,j}\}_{0 \le n \le N}$. Let $\tilde{\mathcal{Q}}\left(\tilde{s}^t_i, q^t_{i,j}\right)$ be the Q-function

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3014385, IEEE Transactions on Network Science and Engineering

Algorithm 1 Q-learning based Pricing Algorithm

Initialize: ζ_1, ρ_1, $\mathbf{s}^1 = 0$, $\mathcal{Q}(\mathbf{s}, \mathbf{p}_j) = 0$, $\mathcal{V}(\mathbf{s}) = 0$, $\forall \mathbf{s}, \mathbf{p}_j$.
for $t = 1, 2, 3, \cdots$ **do**
  Observe $\mathbf{s}^t = \mathbf{q}_j^{t-1}$.
  Select $p_{i,j}^t$ through δ-greedy algorithm via Eq. (39).
  Calculate $\mathbf{U}_j\left(s_i^t, p_{i,j}^t\right)$ via Eq. (28).
  Update $\mathcal{Q}\left(s_i^t, p_{i,j}^t\right)$ via Eq. (40).
  Update $\mathcal{V}\left(s_i^t\right)$ via Eq. (41).
end for

Algorithm 2 Q-learning based QoLM Algorithm

Initialize: ζ_2, ρ_2, $\tilde{s}_i^1 = 0$, $\tilde{\mathcal{Q}}(\tilde{s}_i, q_{i,j}) = 0$, $\tilde{\mathcal{V}}(\tilde{s}_i) = 0$, $\forall \tilde{s}_i, q_{i,j}$.
for $t = 1, 2, 3, \cdots$ **do**
  Observe $\tilde{s}_i^t = p_{i,j}^{t-1}$.
  Select $q_{i,j}^t$ through δ-greedy algorithm via Eq. (42).
  Calculate $\mathbf{U}_{i,j}\left(\tilde{s}_i^t, q_{i,j}^t\right)$ via Eq. (37).
  Update $\tilde{\mathcal{Q}}\left(\tilde{s}_i^t, q_{i,j}^t\right)$ via Eq. (43).
  Update $\tilde{\mathcal{V}}\left(\tilde{s}_i^t\right)$ via Eq. (44).
end for

at state $\tilde{s}_i^t$ with action $q_{i,j}^t$. Based on δ-greedy policy, UAV i selects the optimal QoLM strategy which maximizes its Q-function with a high probability $1-\delta$, and picks the other QoLM strategies randomly with a small probability δ. The QoLM policy $q_{i,j}^t$ at time slot t is given by

$$\Pr\left[q_{i,j}^t = q_{i,j}^*\right] = \begin{cases} 1-\delta, & q_{i,j}^* = \arg\max_{q_{i,j}} \tilde{\mathcal{Q}}\left(\tilde{s}_i^t, q_{i,j}\right), \\ \delta, & \text{otherwise.} \end{cases} \tag{42}$$

For UAV i, the reward in Q-learning is defined as its utility $\mathbf{U}_{i,j}\left(\tilde{s}_i^t, q_{i,j}^t\right)$. After UAV i selects a QoLM action $q_{i,j}^t$ at time slot t with state $\tilde{s}_i^t$, it evaluates the reward and observes the new generated state $\tilde{s}_i^{t+1}$. Let $\tilde{\mathcal{V}}\left(\tilde{s}_i^t\right)$ be the value function, i.e., the highest Q-value at state $\tilde{s}_i^t$. Each UAV $i \in \mathbb{I}_j$ updates its Q-function based on the iterative Bellman equation as follows:

$$\tilde{\mathcal{Q}}\left(\tilde{s}_i^t, q_{i,j}^t\right) \leftarrow \tilde{\mathcal{Q}}\left(\tilde{s}_i^t, q_{i,j}^t\right) + \zeta_2\left\{\mathbf{U}_{i,j}\left(\tilde{s}_i^t, q_{i,j}^t\right) + \rho_2 \tilde{\mathcal{V}}\left(\tilde{s}_i^{t+1}\right) - \tilde{\mathcal{Q}}\left(\tilde{s}_i^t, q_{i,j}^t\right)\right\}, \tag{43}$$

$$\tilde{\mathcal{V}}\left(\tilde{s}_i^{t+1}\right) \leftarrow \max_{q_{i,j}} \tilde{\mathcal{Q}}\left(\tilde{s}_i^{t+1}, q_{i,j}^{t+1}\right), \tag{44}$$

where $\zeta_2 \in (0, 1]$ means the learning rate of QoLM strategy, and $\rho_2 \in [0, 1]$ denotes the discount factor of UAV. The Q-learning based QoLM strategy making algorithm for worker UAV i is summarized in Algorithm 2. The overall computational complexity of the proposed two-tier Q-learning based approach in Algorithms 1 and 2 is $\mathcal{O}\left(t \times I_j\right)$, where t is the number of iterations (i.e., time slots). We can observe that computational cost in the proposed approach for each task j increases linearly with the number of participating UAVs (i.e., I_j).

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of SFAC by using Python. The simulation setup is first introduced, followed by the numerical results and discussions.

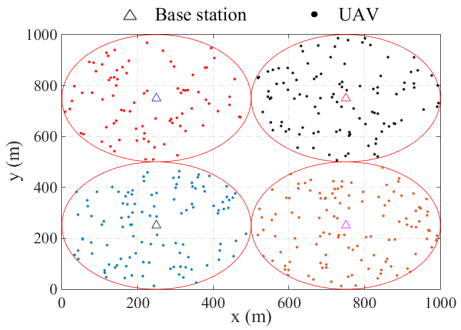


Fig. 3. The simulation area.

TABLE II
SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
ϖ	0.8	ω	0.5
ϕ_1, ϕ_2, ϕ_3	$1, 1, 0.01$	λ_c	1
$p_j^{\max}$	5	$sc_{i,g}$	uniform in $[0, 1]$
fee_m	5	φ	0.01 [41]
μ	11	ν	5
ω_l	5	ω_p	2
P_i	23 dBm [43]	P_m	43 dBm [43]
V_i	10 m/s [43]	ϱ	2
δ	0.95	χ_i	5
μ	100	ΔT_w	50 ms [32]
κ_1	5	κ_2	1×10^{-3}

A. Simulation Setup

As shown in Fig 3, we consider a simulation scenario with 4 base stations in a 1000×1000 m^2 terrain area, where each base station has the same radius $ra_m = 250$m and serves a group of UAVs uniformly distributed in $[80, 120]$. The flying altitudes of UAVs are random in $[100, 300]$m. Each base station performs one federated learning task on digit classification by using the MNIST dataset, which includes 60 000 training samples and 10 000 test samples of 28×28 gray-scale images of handwritten digits in 10 classes. Within each base station, the dataset is randomly split into I_j shards, and each shard is assigned to a participating UAV. The size of each training image of UAV i is compressed in proportional to its average sensing capacity. The CNN model is adopted for model training. The TTL of each task is generated from $[10, 15]$ minutes at random. The privacy budget $\epsilon_{i,j}^k$ of each UAV is randomly picked within $[0.1, 0.5]$. The adjacency parameter γ_j is randomly selected in $[0.3, 0.7]$ and the confidence level β_j^k of each task is chosen in $[0.8, 0.9]$ at random. The unit sensing cost is set as 0.3 and unit cost of privacy leakage is set as 0.05. The optimal block generation rate λ_b is calculated based on [32]. In the Q-learning, the learning rates for both QoLM and payment strategies are 0.8, and the discount factors for both UAVs and task publisher are 0.9 [56]. The simulation time is $T = 9000$. Other parameters in the simulation are listed in Table II.

The performance of our proposed scheme is evaluated by comparing with the following conventional schemes:

- *One-tier RL-based scheme.* In this scheme, each UAV

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3014385, IEEE Transactions on Network Science and Engineering

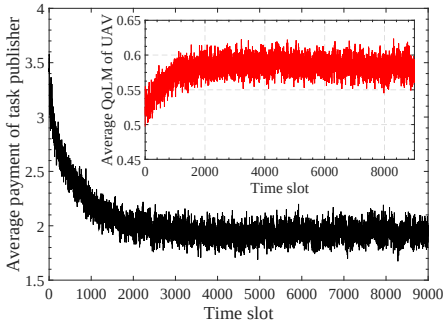


Fig. 4. Evolution of strategies of task publisher and UAVs over time.

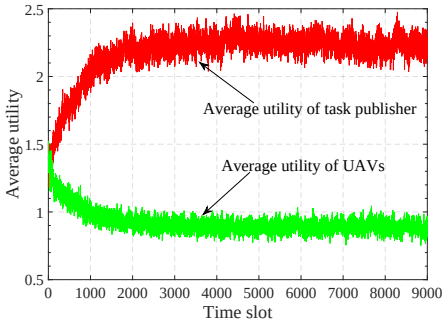


Fig. 5. Evolution of average utilities of task publisher and UAVs over time.

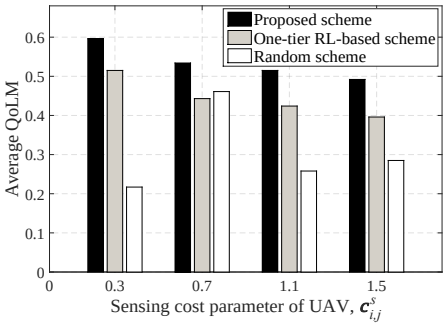


Fig. 6. Average QoLM vs. sensing cost parameter of UAV in three schemes.

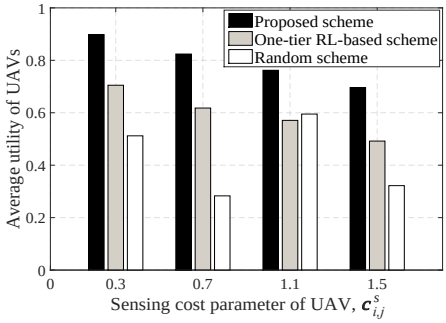


Fig. 7. Average utility of UAVs vs. sensing cost parameter of UAV in three schemes.

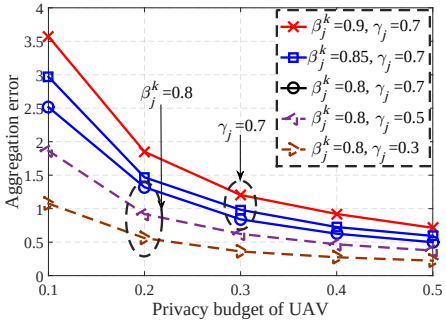


Fig. 8. Aggregation error vs. privacy budget of UAV with different values of β_j^k and γ_j.

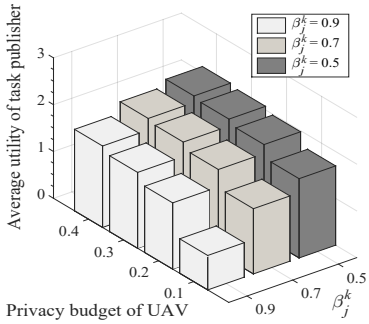


Fig. 9. Average utility of task publisher vs. privacy budget of UAV with different values of β_j^k.

applies Q-learning to derive its optimal QoLM strategy, while the pricing strategy of task publisher is fixed.

- *Random scheme.* In this scheme, the task publisher and all participating UAVs randomly decide their pricing and QoLM strategies during federated learning, respectively.

B. Numerical Results

Fig. 4 depicts the evolution of Q-learning based average payment of task publisher and average QoLM of UAVs over time. As seen in Fig. 4, our proposed scheme can effectively motivate the high-quality local model updates of participating UAVs with low payment. Moreover, both the QoLM strategy of UAVs and the payment strategy of task publisher can converge to stable values. In specific, the average payment of task publisher first decreases and then converges to be stable, while the average QoLM of UAVs first increases and then reaches a stable value. It can be explained as follows: on one hand, motivated by the initial high payment, UAVs are willing to offer local models with high quality and then task publisher tends to reduce its payment to gain more profits. On the other hand, after observing the previous payment sequences, UAVs continuously pursues their optimal QoLM strategies to maximize their utilities.

Fig. 5 shows the evolution of average utilities of task publisher and UAVs with Q-learning over time. It can be seen that the average utilities of both sides are convergent. Besides, the average utility of task publisher increases, while that of UAVs declines as time grows. This is because that the initial high payment promotes UAV to provide high-quality model sharing for improved utility, while task publisher prefers reducing its payment gradually to pursue maximum utility after observing the high QoLM of UAVs.

TABLE III
COMPARISON OF CONVERGENCE TIME IN THE PROPOSED TWO-TIER RL-BASED SCHEME, ONE-TIER RL-BASED SCHEME, AND RANDOM SCHEME

	Two-tier RL	One-tier RL	Random scheme
Convergence time (time slot)	2180	920	Infinity

Table III shows the comparison of convergence time in three schemes. In the one-tier RL-based scheme, since the task publisher's pricing strategy is fixed, the worker UAVs can observe the corresponding payment information of the task after just one iteration. As such, UAVs can quickly attain their optimal QoLM strategies with Q-learning in the relatively stable environment. In the random scheme, as the strategies of all participants are picked at random, their strategies will never convergent to stable values. In the proposed two-tier RL-based scheme, both the strategies of UAVs and task publisher are dynamically determined based on their historical interactions (i.e., observed states, evaluated utilities, and updated Q-tables), resulting a relatively longer convergence time.

Fig. 6 and Fig. 7 demonstrate the average QoLM and average utility of UAVs in different schemes, where the sensing cost parameter of UAV varies from 0.3 to 1.5. We can observe that the proposed scheme outperforms other two schemes by attaining higher QoLM and better utilities of UAVs. This is because that in the one-tier RL-based scheme, task publisher's pricing strategy is fixed and relatively low and is not alterable

with the change of UAVs' QoLM strategies, resulting in that the QoLM and utility of UAVs are not globally optimal. In the random scheme, the quality and payment strategy in model training are chosen at random, causing that the quality of delivered local models and UAVs' utilities are not the maximum. In our proposal, by applying Q-learning at both sides in the dynamic network, the optimal QoLM strategy and the corresponding pricing strategy of UAV and task publisher can be derived in pursuit of their maximum utilities.

Fig. 8 illustrates the aggregation error in global model aggregation, where the privacy budget $\epsilon_{i,j}^k$ increases from 0.1 to 0.5. From Fig. 8, we can see that the aggregation error declines with the increase of the privacy budget, as a larger privacy budget offers a more accurate aggregated result. Furthermore, given the privacy budget, with the increase of confidence level β_j^k and adjacency parameter γ_j, the aggregation error also increases. The reason is that a larger β_j^k implies that the task publisher has a higher accuracy requirement on aggregated global model. Meanwhile, a large γ_j means that a large noise is injected on actual local model updates. As a result, the smaller β_j^k and γ_j and larger $\epsilon_{i,j}^k$ can lead to a more accurate aggregation result and smaller aggregation error, which in accordance with Theorem 1.

Fig. 9 shows the average utility of task publisher when the privacy budget $\epsilon_{i,j}^k$ and confidence level β_j^k are varied. For this simulation, we set $\gamma_j = 0.7$. Other settings are unchanged. As seen in Fig. 9, the average utility of task publisher increases with the privacy budget. This is because that a larger privacy budget ensures a more accurate global model aggregation, so as to increase the task publisher's utility. Moreover, when the privacy budget is fixed, the average utility of task publisher keeps decreasing with the increase of β_j^k. The reason is that the large β_j^k deteriorates the accuracy of aggregated global model, so that the task publisher can not acquire high utility.

VI. Conclusion

In this paper, we have proposed SFAC, a secured federated learning scheme in UAV-assisted MCS for collaborative AI model training without revealing the local sensing data. Firstly, three attacks have been introduced and the corresponding defenses have been investigated towards secured cooperative learning for UAVs. Secondly, by implementing the blockchain network, a decentralized federated learning mechanism has been formulated to record and trace UAVs' contributions in an immutable manner while securing local model exchange among UAVs. Thirdly, based on LDP mechanism, a privacy-preserving local model sharing algorithm has been devised to offer rigorous privacy protection for UAVs with guaranteed aggregate accuracy. In addition, a two-tier RL-based incentive model has been developed to derive the optimal strategies of task publisher and UAVs without the awareness of the private parameters of both sides. Finally, simulation results have shown that the proposed scheme can achieve improved utilities for UAVs, converged strategies, and enhanced QoLM in federated learning process, compared with conventional schemes. In the future work, the reliable worker UAV selection model in federated learning and multi-agent DRL approach for accelerating strategy making process will be investigated.

Acknowledgement

This work is supported in part by NSFC (nos. U1808207, 91746114), and the Project of Shanghai Municipal Science and Technology Commission, 18510761000.

References

[1] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2419–2465, thirdquarter 2019.

[2] Y. Hui, Z. Su, and S. Guo, "Utility based data computing scheme to provide sensing service in internet of things," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 2, pp. 337–348, 2019.

[3] Z. Zhou, H. Liao, B. Gu, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, "Robust mobile crowd sensing: When deep learning meets edge computing," *IEEE Network*, vol. 32, no. 4, pp. 54–60, 2018.

[4] Z. Su, Q. Xu, and Q. Qi, "Big data in mobile social networks: a qoe-oriented framework," *IEEE Network*, vol. 30, no. 1, pp. 52–57, 2016.

[5] W. Li, Z. Su, K. Zhang, and X. Qi, "Abnormal crowd traffic detection with crowdsourcing-based rss fingerprint position in heterogeneous communications networks," *IEEE Transactions on Network Science and Engineering*.

[6] J. A. Ansere, G. Han, L. Liu, Y. Peng, and M. Kamal, "Optimal resource allocation in energy efficient internet of things networks with imperfect csi," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[7] Z. Su, Y. Wang, Q. Xu, and N. Zhang, "Lvbs: Lightweight vehicular blockchain for secure data sharing in disaster rescue," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2020.

[8] J. Gu, T. Su, Q. Wang, X. Du, and M. Guizani, "Multiple moving targets surveillance based on a cooperative network for multi-uav," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 82–89, 2018.

[9] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, and M. Guizani, "When mobile crowd sensing meets uav: Energy-efficient task assignment and route planning," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5526–5538, 2018.

[10] D. Wang, D. Chen, B. Song, N. Guizani, X. Yu, and X. Du, "From iot to 5g i-iot: The next generation iot-based intelligent algorithms and 5g technologies," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 114–120, 2018.

[11] M. Liu, J. Yang, and G. Gui, "Dsf-noma: Uav-assisted emergency communication technology in a heterogeneous internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5508–5519, 2019.

[12] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. Shen, "Software defined space-air-ground integrated vehicular networks: Challenges and solutions," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 101–109, 2017.

[13] M. Liu, G. Gui, N. Zhao, J. Sun, H. Gacanin, and H. Sari, "Uav-aided air-to-ground cooperative nonorthogonal multiple access," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2704–2715, 2020.

[14] C. Lin, G. Han, X. Qi, M. Guizani, and L. Shu, "A distributed mobile fog computing scheme for mobile delay-sensitive applications in sdn-enabled vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5481–5493, 2020.

[15] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, "6g: Opening new horizons for integration of comfort, security and intelligence," *IEEE Wireless Communications*, pp. 1–7, 2020.

[16] M. Huang, A. Liu, N. N. Xiong, T. Wang, and A. V. Vasilakos, "An effective service-oriented networking management architecture for 5g-enabled internet of things," *Computer Networks*, vol. 173, p. 107208, 2020.

[17] W. Li, Z. Su, K. Zhang, A. Benslimane, and D. Fang, "Defending malicious check-in using big data analysis of indoor positioning system: An access point selection approach," *IEEE Transactions on Network Science and Engineering*.

[18] J. Xiong, H. Guo, and J. Liu, "Task offloading in uav-aided edge computing: Bit allocation and trajectory optimization," *IEEE Communications Letters*, vol. 23, no. 3, pp. 538–541, 2019.

[19] T. Xu, G. Han, X. Qi, J. Du, C. Lin, and L. Shu, "A hybrid machine learning model for demand prediction of edge-computing based bike sharing system using internet of things," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[20] U. Challita, A. Ferdowsi, M. Chen, and W. Saad, "Machine learning for wireless connectivity and security of cellular-connected uavs," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 28–35, February 2019.

[21] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, vol. abs/1602.05629, 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[22] Q. Xu, Z. Su, and R. Lu, "Game theory and reinforcement learning based secure edge caching in mobile social networks," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2020.

[23] Y. Liu, M. Dong, K. Ota, and A. Liu, "Activetrust: Secure and trustable routing in wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2013–2027, 2016.

[24] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "Verifynet: Secure and verifiable federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2020.

[25] 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 *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, April 2019, pp. 1387–1395.

[26] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.

[27] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, Dec 2019.

[28] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[29] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *CoRR*, vol. abs/1712.07557, 2017. [Online]. Available: http://arxiv.org/abs/1712.07557

[30] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2134–2143, March 2020.

[31] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146–1159, Feb 2020.

[32] H. Kim, J. Park, M. Bennis, and S. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, pp. 1–1, 2019.

[33] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2019.

[34] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, and M. Guizani, "When mobile crowd sensing meets uav: Energy-efficient task assignment and route planning," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5526–5538, Nov 2018.

[35] B. Zhang, C. H. Liu, J. Tang, Z. Xu, J. Ma, and W. Wang, "Learning-based energy-efficient data collection by unmanned vehicles in smart cities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1666–1676, April 2018.

[36] S. Zhang, H. Zhang, B. Di, and L. Song, "Joint trajectory and power optimization for uav sensing over cellular networks," *IEEE Communications Letters*, vol. 22, no. 11, pp. 2382–2385, Nov 2018.

[37] C. Piao and C. H. Liu, "Energy-efficient mobile crowdsensing by unmanned vehicles: A sequential deep reinforcement learning approach," *IEEE Internet of Things Journal*, pp. 1–1, 2019.

[38] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in iot," *IEEE Internet of Things Journal*, pp. 1–1, 2019.

[39] F. Sattler, S. Wiedemann, K. Mller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2019.

[40] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2019.

[41] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, June 2019.

[42] J. Qiu, D. Grace, G. Ding, J. Yao, and Q. Wu, "Blockchain-based secure spectrum trading for unmanned-aerial-vehicle-assisted cellular networks: An operators perspective," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 451–466, Jan 2020.

[43] A. Asheralieva and D. Niyato, "Distributed dynamic resource management and pricing in the iot systems with blockchain-as-a-service and uav-enabled mobile edge computing," *IEEE Internet of Things Journal*, pp. 1–1, 2019.

[44] A. Islam and S. Y. Shin, "Buav: A blockchain based secure uav-assisted data acquisition scheme in internet of things," *Journal of Communications and Networks*, vol. 21, no. 5, pp. 491–502, Oct 2019.

[45] Y. Zhu, G. Zheng, and K. Wong, "Blockchain-empowered decentralized storage in air-to-ground industrial networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3593–3601, June 2019.

[46] X. Li, Y. Wang, P. Vijayakumar, D. He, N. Kumar, and J. Ma, "Blockchain-based mutual-healing group key distribution scheme in unmanned aerial vehicles ad-hoc network," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11 309–11 322, Nov 2019.

[47] Y. Wang, Z. Su, and N. Zhang, "Bsis: Blockchain-based secure incentive scheme for energy delivery in vehicular energy network," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3620–3631, June 2019.

[48] Q. Xu, Z. Su, and Q. Yang, "Blockchain-based trustworthy edge caching scheme for mobile cyber-physical system," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1098–1110, 2020.

[49] J. Gu, B. Sun, X. Du, J. Wang, Y. Zhuang, and Z. Wang, "Consortium blockchain-based malware detection in mobile devices," *IEEE Access*, vol. 6, pp. 12 118–12 128, 2018.

[50] K. Zhang, Y. Zhu, S. Maharjan, and Y. Zhang, "Edge intelligence and blockchain empowered 5g beyond for the industrial internet of things," *IEEE Network*, vol. 33, no. 5, pp. 12–19, 2019.

[51] Y. Sun, F. Tong, Z. Zhang, and S. He, "Throughput modeling and analysis of random access in narrowband internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1485–1493, 2018.

[52] M. Zhang, S. He, C. Yang, J. Chen, and J. Zhang, "Vanet-assisted interference mitigation for millimeter-wave automotive radar sensors," *IEEE Network*, vol. 34, no. 2, pp. 238–245, 2020.

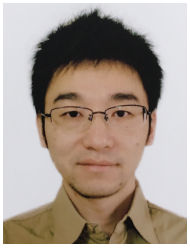
[53] X. Liu, W. Wang, D. Niyato, N. Zhao, and P. Wang, "Evolutionary game for mining pool selection in blockchain networks," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 760–763, Oct 2018.

[54] Z. Xiong, J. Kang, D. Niyato, P. Wang, and V. Poor, "Cloud/edge computing service management in blockchain networks: Multi-leader multi-follower game-based admm for pricing," *IEEE Transactions on Services Computing*, pp. 1–1, 2019.

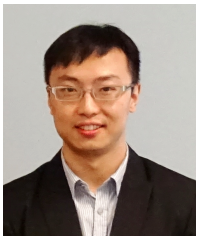
[55] W. J. J. Rey, "On the upper bound of the probability of error, based on chebyshev's inequality, in two-class linear discrimination," *Proceedings of the IEEE*, vol. 64, no. 3, pp. 361–362, March 1976.

[56] L. Xiao, T. Chen, C. Xie, H. Dai, and H. V. Poor, "Mobile crowdsensing games in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1535–1545, Feb 2018.

Yuntao Wang is working on his Ph.D degree with the school of Cyber Science and Engineering of Xi'an Jiaotong University, Xi'an, P. R. China. His research interests include security and privacy in wireless network architecture and vehicular networks.



Zhou Su received the Ph.D degree from Waseda University, Tokyo, Japan, in 2003. He is an Associate Editor of IET Communications, and Associate Editor of IEICE Trans on Communications. He is the Chair of the Multimedia Services and Applications over Emerging Networks Interest Group (MENIG) of the IEEE Comsoc Society, the Multimedia Communications Technical Committee. He received the best paper award of IEEE CyberSciTech2017, WiCon2016, CHINACOM2008, and Funai Information Technology Award for Young Researchers in 2009.



Ning Zhang (M'15-SM'18) received the Ph.D degree from University of Waterloo, Canada, in 2015. After that, he was a postdoc research fellow at University of Waterloo and University of Toronto, Canada, respectively. He is an Associate Professor at University of Windsor, Canada. He serves as an Associate Editor of IEEE Internet of Things Journal, IEEE Transactions on Cognitive Communications and Networking, IEEE Access, and IET Communications, and Vehicular Communications (Elsevier); and a Guest Editor of several international journals, such as IEEE Wireless Communications, IEEE Transactions on Industrial Informatics, and IEEE Transactions on Cognitive Communications and Networking. He also serves/served as a track chair for several international conferences and a co-chair for several international workshops. He received the Best Paper Awards from IEEE Globecom in 2014, IEEE WCSP in 2015, and Journal of Communications and Information Networks in 2018, IEEE ICC in 2019, IEEE Technical Committee on Transmission Access and Optical Systems in 2019, and IEEE ICCC in 2019, respectively. He has been a senior member of IEEE since 2018.

Abderrahim Benslimane is Full Professor of Computer-Science at the Avignon University/France since 2001. He is Chair of the ComSoc Technical Committee of Communication and Information Security. He is EiC of Inderscience Int. J. of Multimedia Intelligence and Security (IJMIS), Area Editor of Security in IEEE IoT journal, Area Editor of Wiley Security and Privacy journal and editorial member of IEEE Wireless Communication Magazine, Elsevier Ad Hoc, IEEE Systems and Wireless Networks Journals. He is founder and serves as General-Chair of the IEEE WiMob since 2005 and of iCOST and MoWNet international conference since 2011. He was Board committee member, Vice-chair of Student activities of IEEE France section/Region 8; he was Publication Vice-chair and Conference Vice-Chair of the ComSoc TC of Communication and Information Security.