From Eye to Brain: A Proactive and Distributed Crowdsensing Framework for Federated Learning

Hongjia Wu, Hui Zeng, Tongqing Zhou, Zhiping Cai, Zehui Xiong, *Member, IEEE*,
Dusit Niyato, *Fellow, IEEE*, and Zhu Han, *Fellow, IEEE*

***Abstract*—Massive amounts of high-quality data are the prerequisite and support for AI technologies. Due to the nature of privacy-preserving and low communication overheads, federated learning (FL) has garnered considerable attention in comparison with traditional data collection methods. However, the performance of FL is hampered by the lack of interested clients and limited local data due to selfishness and individual behavioral preferences. To this end, we propose *PractFL*, a proactive and distributed framework that incorporates the concept of mobile crowdsensing into the FL paradigm. Specifically, we design an incentive mechanism in the form of virtual red packets, which are a widely used way of monetary reward and gift-giving in social lives. In this article we extend this further by giving meaning to the locations, i.e., the red packets are only accessible at specific places. The virtual red packets' locations and monetary amounts can be dynamically updated by the cloud center to encourage clients to collect additional data that may benefit the FL process. Further, we propose a distributed behavioral decision engine based on multiarmed bandits (i.e., choose which red packet to go for) in response to the incentive mechanism enforced by the cloud. Considering the movement cost and conflicts with other clients, *K*-anonymity and probabilistic selection are introduced in the distributed behavioral decision to recommend the optimal red packet choice for clients without revealing their privacy. The experimental results demonstrate that *PractFL* outperforms the baselines in terms of classification accuracy. We also find that *PractFL* can effectively alleviate the overfitting problem caused by class imbalance during the training.**

***Index Terms*—Crowdsensing, federated learning (FL), multiarmed bandits (MABs), proactive sensing, virtual red packets.**

I. Introduction

With the advancement of technology, smart devices are entering our daily lives, allowing us to enjoy the convenience brought by the era of artificial intelligence [1], [2]. Artificial intelligence has brought convenience to people in the fields of transportation and travel [3], public safety, and healthcare [4]. However, these AI technologies [5] rely heavily on massive amounts of high-quality data to support their services. Nowadays, data collection methods with centralized and dedicated sampling devices are widely adopted. Yet, they suffer from privacy disclosure and excessive data transmission costs. Against this backdrop, federated learning (FL) [6] has attracted attention, which provides a way to leverage clients' local data to learn without revealing privacy. It addresses the privacy concerns of centralized collection and saves the high cost of dedicated sampling devices.

However, compared with traditional data collection methods, FL may face the problem of data insufficiency. It arises from two facts: 1) *client limitation*: volunteer participation is not realistic in FL, since model learning consumes enormous resources of the participating clients [7] and 2) *local data limitation*: clients, such as smartphones, mobile devices, and vehicles, have their daily repeating behavioral preferences and a limited range of activities, leading to a lack of data diversity. Specifically, we illustrate an application of FL in vehicle networks as an example. As shown in Fig. 1, few vehicles are inclined to participate in the FL due to the cost overheads involved in computation and communication. Moreover, the vehicles have their driving routes and lack flexibility, which prevents data from being collected in uncharted places, such as road conditions information on rural roads and inaccessible places for vehicles. As a result, the local data on the clients involved in FL is insufficient and undiverse.

To address such an issue, introducing an incentive mechanism [7], [8], [9], [10], [11], [12] can be an option to improve

Manuscript received 2 February 2022; revised 7 September 2022 and 1 November 2022; accepted 9 December 2022. Date of publication 19 December 2022; date of current version 25 April 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFC2003404; in part by the National Natural Science Foundation of China under Grant 62072465, Grant 62172155, Grant 62102425, Grant 62001483, and Grant 62171449; in part by the Science and Technology Innovation Program of Hunan Province under Grant 2022RC3061 and Grant 2021RC2071; in part by the China Scholarship Council; in part by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Future Communications Research and Development Programme; in part by SUTD under Grant SRG-ISTD-2021-165; in part by the SUTD-ZJU IDEA Grant under Grant SUTD-ZJU (VP) 202102; in part by the Ministry of Education, Singapore, under its SUTD Kickstarter Initiative under Grant SKI 20210204; in part by NSF under Grant CNS-2107216, Grant CNS-2128368, and Grant CMMI-2222810; in part by Toyota; and in part by Amazon. *(Corresponding authors: Tongqing Zhou; Zhiping Cai.)*

Hongjia Wu is with the College of Computer, National University of Defense Technology, Changsha 410082, Hunan, China, and also with the Information Systems Technology and Design, Singapore University of Technology and Design, Singapore (e-mail: wuhongjia19@nudt.edu.cn).

Hui Zeng, Tongqing Zhou, and Zhiping Cai are with the College of Computer, National University of Defense Technology, Changsha 410082, Hunan, China (e-mail: zenghui116@nudt.edu.cn; zhoutongqing@nudt.edu.cn; zpcai@nudt.edu.cn).

Zehui Xiong is with the Pillar of Information Systems Technology and Design, Singapore University of Technology and Design, Singapore (e-mail: zehui_xiong@sutd.edu.sg).

Dusit Niyato is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: dniyato@ntu.edu.sg).

Zhu Han is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 446-701, South Korea (e-mail: hanzhu22@gmail.com).

Digital Object Identifier 10.1109/JIOT.2022.3230050

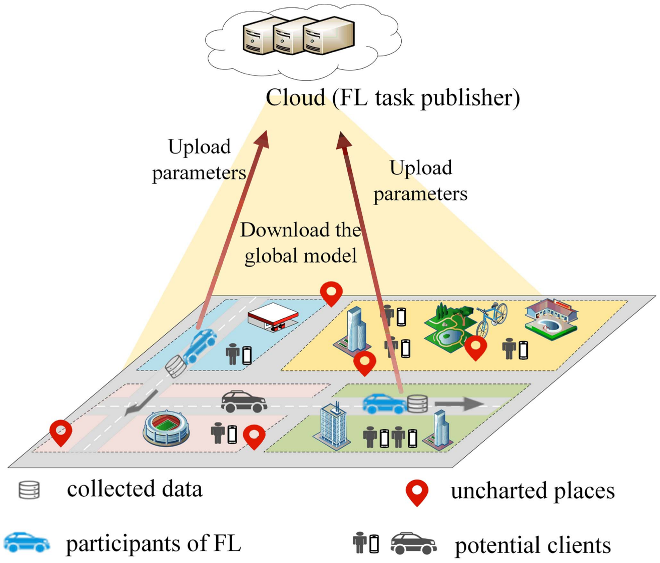


Fig. 1. Example of FL in vehicle networks.

FL performance by attracting high-quality clients to participate in learning. However, motivating more clients to participate does not fundamentally solve the problem of local data limitation. Another possible approach is to leverage federated transfer learning (FTL) [13], [14], [15], [16], i.e., applying the already mature knowledge of one domain to other scenarios, to cope with the lack of data through knowledge transfer. Nevertheless, transfer learning can only be applicable when the source data is very similar to the auxiliary data [17]. When the source data and the auxiliary data are relatively different, it is often difficult for instance-based transfer learning algorithms to find the common knowledge that can be transferred. Therefore, how to make more potentially valuable data available to clients and solve the problem of local data limitation is still a problem worthy of research attention.

In this article, we are inspired to introduce mobile crowdsensing (MCS) [18], [19], [20] into FL and propose a proactive and distributed crowdsensing framework (*PractFL*). As a high-level view, the *PractFL* is composed of two modules: 1) the incentive mechanism and 2) the distributed behavioral decision. The incentive mechanism attracts clients to actively participate in the crowdsensing process by strategically setting the locations and monetary amounts of virtual red packets. The distributed behavioral decision helps clients choose the optimal red packet to take based on circumstances, i.e., the optimal geographical location to collect data. After iterations of the above procedure, the clients collect additional data to enrich the local training data through rational guidelines of virtual red packets from the cloud, solving the problem of local data limitation and improving model performance. However, designing such a framework faces the following challenges.

1) How to design an incentive mechanism that can be adapted to dynamic environments to attract clients to proactively and effectively explore additional data that may benefit the learning process?
2) How can clients make independent decisions without exposing sensitive information while avoiding conflicts with other clients? There are two contradictions for clients: on the one hand, they are privacy-sensitive individuals who refuse to disclose information such as locations due to proactive sensing. On the other hand, they need to know information about other clients to avoid conflicts.

Given these challenges, we first introduce the concept of a heat map in the incentive mechanism. The cloud center updates the locations and monetary amounts of virtual red packets mostly based on the heat value. Such conscious and goal-oriented incentives would enable more clients to rationally collect additional data. For the second challenge, we develop a distributed behavioral decision algorithm based on multiarmed bandits (MABs). It works well to tackle conflicts without disclosing privacy and helps clients choose the optimal virtual red packets that maximize clients' rewards. The contributions of *PractFL* are summarized as follows.

1) We propose *PractFL*, a proactive and distributed crowdsensing framework for FL, including an incentive mechanism for the cloud and distributed behavioral decisions for clients. The *PractFL* essentially addresses the low-performance problem caused by insufficient local data and lack of clients.
2) We design an incentive mechanism in the form of virtual red packets that can well motivate clients to visit needed places (infrequently visited or important places) to collect data, thus benefiting the FL process. Such an incentive mechanism achieves the purpose of motivating clients to actively enrich their local data by dynamically updating the locations and monetary amounts of the red packets. Furthermore, a heat-based location selection algorithm is also proposed.
3) We explore a distributed behavioral decision algorithm based on MABs for clients, where each client makes decisions independently based on limited global information. Through the adoption of a K-anonymity and probabilistic selection design, we can mitigate the multiclient selection conflict problem caused by consistent global information while protecting client privacy.
4) We evaluate the framework by using three well-known data sets. Experimental results show that our proposal can achieve better classification accuracy than those of the baselines. Meanwhile, it can alleviate overfitting compared with the *Centralized learning*.

The remainder of this article is organized as follows. In Section II, we briefly illustrate related works. Then, we introduce our proposed framework *PractFL* and system model in Section III. We propose an incentive mechanism in the form of virtual red packets for the cloud center in Section IV. We design a distributed behavioral decision for clients in Section V. Extensive simulations and comparison experiments are conducted in Section VI. Finally, we conclude this article in Section VII.

II. Related Work

We summarize the existing studies that relate to our work and discuss the differences from our work. Related work can

TABLE I
CHARACTERISTICS OF THE RELATED WORK

Categories	References	Optimization methods	Main factors	Limitations
Incentive Mechanism in FL	[7]	Reverse auction	Learning quality estimation	Inadequate, undiversified local data; Insufficient clients; Lack of proactive sensing data; Ignoring the mobility of mobile clients.
	[8]	Contract theory	Reputation	
	[9]	Bayesian game	Truthful predictions	
	[10]	Auction theory	Multi-dimensional resources	
	[11]	Stackelberg game	Traning data size	
	[12]	Minority game	Volatility	
Federated Transfer Learning	[13]	Knowledge migration	Healthcare	Domain shift problem; Local data limitation.
	[14]	Transfer learning	Multiparty learning tasks	
	[15]	Secret sharing	Semi-honest model	
	[19]	Transfer learning	Smart manufacturing	
Integration of MCS and FL	[20]	Convolutional neural network	Unreliable user data	Focusing only on the benefits of FL used in MCS.
	[21]	Reinforcement learning	Privacy and data misuse	
	[22]	Privacy-preserving	Dilemma of privacy protection and data evaluation	
	[23]	Two stage stackelberg game	communication efficiency	

be primarily grouped into three categories: 1) incentive mechanism in FL; 2) FTL; and 3) the integration of MCS and FL. The characteristics of the related work are shown in Table I.

A. Incentive Mechanism in Federated Learning

Existing efforts in incentive design for FL are dedicated to client selection and performance optimization. For client selection, Kang et al. [8] introduced reputation as a fair metric to select reliable clients for FL to defend against unreliable model updates. An effective incentive mechanism was designed using contract theory to motivate highly reputable employees with high-quality data to participate in model training. Deng et al. [7] designed a new system called FAIR, in which the reverse auction was modeled to encourage the participation of high-quality learning clients. The FAIR has three main components: 1) learning quality estimation; 2) quality-aware incentives; and 3) model aggregation. For performance optimization, Weng et al. [9] customized an incentive mechanism based on Bayesian game theory and designed a federated prediction service framework empowered by blockchain. The prediction service was provided as accurately as possible in an open environment through real contributions from various source models. Zeng et al. [10] proposed an incentive mechanism FMore for multidimensional procurement auctions. FMore is not only lightweight and incentive compatible but also encourages more high-quality edge nodes to participate in learning at a lower cost, and ultimately improves the accuracy of FL. Zhan et al. [11] investigated an incentive mechanism based on the Stackelberg game for federated learning. Furthermore, a deep reinforcement learning (DRL)-based scheme was devised to determine the optimal pricing strategy for parameter servers and the optimal training strategy for edge nodes. Hu et al. [12] proposed a novel autonomous client participation scheme to incentivize clients in situations where they can make autonomous participation decisions. They modeled the autonomous client participation process in FL as a minority game by enabling each client to make decisions with incomplete information.

These efforts improve the performance of FL by selecting high-quality clients and optimizing learning parameters but still fail to address the limitations of local data. This is because clients have no concept of proactive sensing for data and ignore the mobility of mobile clients.

B. Federated Transfer Learning

The work on using transfer learning to address data insufficiency in FL has been conducted in [13], [14], [15], and [21]. For instance, Chen et al. [13] proposed FedHealth, the first federated migratory learning framework for wearable healthcare, which enabled personalized model learning through knowledge migration. Liu et al. [14] introduced a new technique and framework named FTL. The FTL allowed knowledge sharing without compromising client privacy and transferring complementary knowledge across domains in a data federation. Considering that most data is scattered across different organizations and therefore cannot be easily integrated, Sharma et al. [15] improved the efficiency of existing models by merging practical collaboration training under data federation. Wang et al. [21] proposed a new FTL framework to address the challenges of data scarcity and data privacy. The new applications can convert a base model to their target-domain models through transfer learning techniques for further application-specific accuracy improvements. However, the technique of transfer learning needs to cope with the domain shift problem, which still requires a certain amount of auxiliary data that is similar to the source data. Therefore, this technology fails to address the pain point of local data limitation.

C. Integration of MCS and FL

To address the risk of privacy leakage of existing MCS work, Jiang et al. [22] proposed a privacy-preserving mobile crowdsensing system (F-Sense) based on federated learning. F-Sense implemented privacy-preserving crowdsensing and developed incentives to improve task efficiency by encouraging local training on mobile devices. For unmanned aerial vehicles (UAVs) crowdsensing, Wang et al. [23] designed a secure federated learning framework for UAV-assisted MCS. Specifically, they introduced a blockchain-based collaborative learning architecture, while using a two-tier incentive mechanism based on reinforcement learning to facilitate high-quality model sharing for UAVs. Zhao et al. [24] integrated FL into

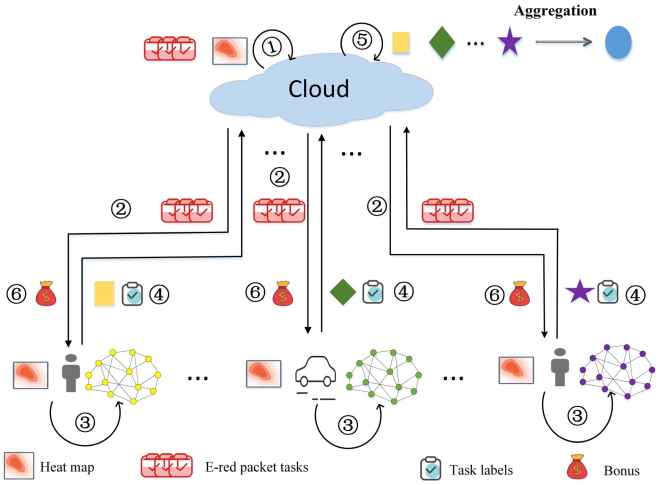


Fig. 2. System workflow of *PractFL*.

MCS and proposed a privacy-preserving MCS system, called CROWDFL. Pandey et al. [25] proposed a novel crowdsourcing framework to leverage FL to establish a high-quality and compact global model, improving communication efficiency during parameter exchange. These studies are not in line with our proposal to incorporate MCS in FL. They studied the privacy issues of the MCS, while we focused on how to leverage the features of the MCS to improve the performance of FL.

Different from the above work, we incorporate MCS in FL and introduce the idea of proactive sensing to fundamentally address the issue of local data limitations. More specifically, we consider how to attract mobile clients to participate and move consciously as needed to collect additional data in infrequent places. Moreover, efficient and privacy-protective behavioral guidelines are designed for clients to explore the world together.

III. Background and Overview of PractFL

PractFL is a proactive and distributed crowdsensing framework for federated learning designed with two major modules: 1) incentive mechanism and 2) distributed behavioral decision. The incentive mechanism is performed by the cloud and the distributed behavioral decision is designed for clients.

As shown in Fig. 2, we use a system workflow to describe this proactive sensing process in one FL round. In detail, the cloud generates a series of virtual red packets defined as red packet tasks, then initializes the heat map (see ①) and finally broadcasts the tasks and heat map to all clients (see ②). Next, the clients select the optimal red packet task for themselves based on the global task information and heat map, move to the corresponding red packet location to collect data, and participate in FL (see ③). After the FL model is trained locally, the client uploads its model parameter to the cloud along with the task labels[1] (see ④). The cloud aggregates the received parameters, updates the global model, and calculates the bonus assigned to the clients based on the received task labels (see ⑤). Finally, the cloud will faithfully distribute the bonus to clients[2] (see ⑥). Briefly, the main purpose of the incentive mechanism is to motivate clients to be actively involved in the MCS process for sampling (involving steps ①-②, ⑤-⑥). For distributed behavioral decisions, clients make their independent red packet task selection decisions based on global incentive information and their locations. Then, the collected additional data needs to be added to the local data for model training, and the model parameters are eventually uploaded to the cloud along with the selected task labels (involving steps ③-④).

TABLE II
Frequency Used Notations

Notation	Description
$\mathcal{M}$	A set of clients
$\mathit{\Gamma}$	A set of red packet tasks
$\mathcal{I}$	A set of selected grids
$h(x_{\tau_i}, y_{\tau_i})$	Heat value where task τ_i is located
$L(x_{\tau_i}, y_{\tau_i})$	Location of task τ_i with coordinate (x_{τ_i}, y_{τ_i})
$Mon(\tau_i)$	Monetary amounts of red packet task τ_i
$r_m^t(\tau_i)$	Reward of task τ_i at round t for client m
$\hat{R}_m^t(\tau_i)$	Expected reward of task τ_i at round t for client m
$\mathit{\Gamma}_m^k$	A set of K anonymous tasks for client m

In this article, a network structure with geographic location identifiers is adopted and the whole sampling area is divided into $G \times G$ grids. Clients with different behavioral trajectories move on the grids and collect the corresponding data. We consider a federated learning instance consisting of a set $\mathcal{M} = \{1, 2, \ldots, M\}$ of clients with model training capacity and a centralized parameter server residing in the edge or cloud service. The mapping, navigation, and location-based services used in *PractFL* are provided by AutoNavi [27]. To ease the presentation, we summarize some important notations in Table II.

IV. Incentive Mechanism

Considering that clients usually have various destinations and trajectories in the real world, it is difficult for them to proactively visit elsewhere to collect additional data that could benefit the learning process. To this end, we design an incentive mechanism in the form of virtual red packets, which can be a good incentive for clients to sample at the designed locations. Actually, the whole incentive process aims to address the three key problems. The first one is how to define the virtual red packets. The second one is how to deploy virtual red packets properly. The last one is how to allocate the monetary amounts of the red packets to the clients. We define below the virtual red packet tasks, the location of where the red packets should be deployed and the monetary allocation of the virtual red packets.

[1] Uploading task labels, i.e., informing the cloud of the tasks that the clients have chosen to complete and providing the basis for the cloud to update the heat map and calculate bonus allocations for later use. The task labels contain both the real selected task and the other tasks introduced for privacy protection by K-Anonymous, which are described in more detail later.

[2] Here, we can ensure the truthfulness of the information based on third-party authority platforms [26], which allows the cloud center to get realistic heat map updates as well as the correct bonus feedback to the clients.

A. Definition of Virtual Red Packet Tasks

We formulate the cloud-issued virtual red packets as a series of red packet tasks $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_n\}$, which motivate clients to proactively collect data. Let $\tau_i = \langle L(x_{\tau_i}, y_{\tau_i}), \mathrm{Mon}(\tau_i), D\rangle$ be the ith task, where $L(x_{\tau_i}, y_{\tau_i})$ is the location of the ith task at the grid point with coordinate (x_{τ_i}, y_{τ_i}) and D indicates the duration of time that the tasks are available for selection by the clients. $\mathrm{Mon}(\tau_i)$ refers to the total monetary that red packet τ_i has, which is modeled as

$$\mathrm{Mon}(\tau_i) = \frac{\mu}{h\big(x_{\tau_i}, y_{\tau_i}\big)} + \varepsilon\big(\mathrm{ave}(\boldsymbol{h}) - h\big(x_{\tau_i}, y_{\tau_i}\big)\big) \quad (1)$$

where $h(x_{\tau_i}, y_{\tau_i})$ denotes the heat value of task τ_i with coordinate (x_{τ_i}, y_{τ_i}) in the grids, which is proportional to the total number of visits. $\mathrm{ave}(\boldsymbol{h})$ represents the average of the overall heat values, where $\boldsymbol{h} = \{h(x_{\tau_1}, y_{\tau_1}), \ldots, h(x_{\tau_i}, y_{\tau_i}), \ldots, h(x_{\tau_n}, y_{\tau_n})\}$. μ and ε are constant parameters [28], and their values are set according to the maximum monetary $\mathrm{Mon}(\tau_i)^{\max}$ that the cloud can afford. This monetary model consists of two items. The first term shows that the monetary amounts of the red packets are inversely proportional to the heat value of the red packet location, indicating that locations with lower historical visit frequency are more attractive to clients. The second term reflects the frequency of visits to the location compared to the average heat value. If it is higher than the average value, the monetary amounts are appropriately reduced, and vice versa.

B. Deployment of Virtual Red Packets

The deployment of virtual red packets[3] directly affects whether additional data can be collected. Considering the great dispersion of data distribution, we prefer to issue the virtual red packets uniformly on the sampling area and collect different categories of data as much as possible. This makes it possible for each location in the grids to be selected, avoiding the clustering of virtual red packets such as random placement that causes certain locations to be ignored. Further, we incline to attract clients to access infrequently visited and important places for collecting data that may be beneficial to model training. These two aspects above need to be considered together, i.e., it is ideal to select locations with low historical visit frequency as virtual red packet placement points under the premise of a uniform distribution.

We first propose to model the selection of virtual red packet locations as the distribution entropy (DE) of virtual red packets in the grids. We divide the entire $G \times G$ grids into $Area$ regions I_{area}, where each I_{area} contains $|(G \times G)/\text{Area}|$ grids. Given the grids, we define the DE with

$$DE(I) = -\sum_{\text{area}=1}^{\text{Area}} \frac{|I_{\text{area}}|}{|I|} \cdot \log_2\left(\frac{|I_{\text{area}}|}{|I|}\right), |I_{\text{area}}| \neq 0 \quad (2)$$

where $|I_{\text{area}}|$ is the number of virtual red packets in I taken in the I_{area}. I ($I \subset G \times G$, $|I| \leq |\Gamma|$) denotes the selected grid set.

Algorithm 1: Heat-Based Location Selection

```
Input: $Area$, Heat map $\boldsymbol{h}$ and Task set $\Gamma$.
Output: The location $L(x_{\tau_i}, y_{\tau_i})$, $\tau_i \in \Gamma$
Initialization: $I = I_{area} = \phi$;
for each $\tau_i$ in $\Gamma$ do
    for each area in Area do
        $\Delta_{DE}^{area} \leftarrow DE(I_{area} \cup \{\tau_i\}) - DE(I)$;
    $area = argmax_{1 \leq area \leq Area}\{\Delta_{DE}^{area}\}$;
    $|I_{area}| = |I_{area}| + 1$;
    $|I| = |I| + 1$;
repeat
    for each $I_{area}$ in $I$ do
        if $|I_{area}| > 0$ then
            $L(x_{\tau_i}, y_{\tau_i}) \leftarrow$ Select the location in $I_{area}$ with
              the smallest heat value $h(x_{\tau_i}, y_{\tau_i})$;
            $|I_{area}| = |I_{area}| - 1$;
            $|I| = |I| - 1$;
until $|I| \leq 0$;
return $L(x_{\tau_i}, y_{\tau_i})$
```

Inspired by the properties of entropy [29] (if the locations of virtual red packets are close to a uniform distribution, and then the entropy is high), we favor to find the virtual red packet distribution with the maximum value of DE.

Then, we prioritize the $|I_{\text{area}}|$ grids with low heat values within each I_{area} as the locations of virtual red packets. Based on the above two steps, a set of locations with low heat values in a uniform distribution is selected. However, it is challenging to find an optimal solution since it belongs to NP-hard [20], [29]. To relieve this pitfall, we design a heat-based location selection algorithm to find an approximate solution. Our heat-based location selection algorithm is listed in Algorithm 1.

There are two primary processes in Algorithm 1. To begin, we divide the entire grids into multiple areas of the equal size and distribute the red packets as evenly as possible to collect data thoroughly. For each red packet, we place it in an area that can increase the total entropy. After all red packs have been assigned to areas, specific red pack locations within each area will be chosen. Low heat values in each area will be prioritized for the deployment of red packets. Algorithm 1 is used to update the locations of the red packets every D min.

C. Allocation of Bonus

Every D min, the cloud center divides monetary amounts of virtual red packets based on the feedback collected from all clients. To achieve a reasonable bonus, clients need to inform the cloud of their task selection. However, this will lead to the disclosure of sensitive information for clients (e.g., visited places). To solve the problem, we formally adopt K-anonymous means [30] to hide the clients' real virtual red packet selection. Specifically, each client finds the closest $(k-1)$ virtual red packets to the selected one and takes the location area composed of the k virtual red packets as the reported information. Thus, given the feedback $\Gamma_m^k = \{\tau_1, \tau_2, \ldots, \tau_k\}$,

[3] There is no material cost overhead here for the deployment of red packets, which is done electronically at the logical layer. The cloud deploys virtual red packets by broadcasting the locations of the red packets and then updates the next deployment with the task labels uploaded by the clients.

the cloud can calculate the bonus of client m by the following formula:

$$\mathrm{bonus}(m) = \frac{\sum_{\tau_i \in \Gamma_m^k} \mathrm{Mon}(\tau_i) \cdot \frac{1}{\mathrm{col}(\tau_i)}}{k^2} \tag{3}$$

where $\mathrm{col}(\tau_i)$ indicates the conflict value of task τ_i selected by multiple clients, i.e., the number of clients who select task τ_i. Here, the specific value of k is determined by each client.[4] If the client determines that the location information is not sensitive, k can be set smaller. By reporting accurate information in this way, the client can get more bonuses. If the client is more concerned about privacy, k can be set larger, and privacy protection can be obtained by sacrificing certain bonuses.

The incentive mechanism completes the generation, deployment, and allocation of bonuses for the virtual red packets. Based on the above incentive mechanism, we next describe the distributed behavioral decision designed for clients to help them choose the red packets that reward them the most based on their respective movement costs.

V. Distributed Behavioral Decision

Our goal is to design a distributed behavioral decision that helps clients choose the optimal red packets to get the most rewards given their movement costs. An intuitive idea is that the cloud uses the collected client information to make global optimal decisions. However, this will lead to the exposure of client trajectories. Such threats will degrade the motivation of clients to join sensing tasks. To avoid the leakage of client-side privacy, we address this problem in a distributed way. In this case, each client makes a choice based on the limited global information without exposing trajectory information.

A. Problem Formulation

The clients dynamically learn and select the optimal red packet tasks, which can be modeled as a multiagent MAB (MAMAB) problem. The clients and tasks are regarded as agents and arms, respectively, where the $|\mathcal{M}|$ agents facing $|\Gamma|$ arms need to decide which arm to play. The essence is to make a tradeoff between exploitation (i.e., selecting the optimal task based on the rewards) and exploration (i.e., trying some suboptimal tasks to find the potentially optimal task).

At each round $t \leq T$ within D,[5] client m selects a red packet task $\tau_t^* \in \{\tau_1, \ldots, \tau_n\}$ with the largest expected reward $\hat{R}_m^t(\tau_t^*) = \overline{R}_m^{t-1}(\tau_t^*) + c_m^t(\tau_t^*)$. Thus, a behavioral decision based on the idea of the upper confidence bound (UCB) [31] for client m can be formalized as follows:

$$\tau_t^* = \arg\max_{\tau_i \in \Gamma} \left(\overline{R}_m^{t-1}(\tau_i) + c_m^t(\tau_i) \right) \tag{4}$$

where $\overline{R}_m^{t-1}(\tau_i)$ is the empirical average of the observed rewards, which is defined as

$$\overline{R}_m^{t-1}(\tau_i) = \frac{1}{N_m^{t-1}(\tau_i)} \sum_{s=1}^{t-1} r_m^t(\tau_i) \mathbb{I}_{(\tau_t^* = \tau_i)} \tag{5}$$

where

$$N_m^{t-1}(\tau_i) = \sum_{s=1}^{t-1} \mathbb{I}_{(\tau_t^* = \tau_i)} \tag{5a}$$

$$\mathbb{I}_{(\tau_t^* = \tau_i)} = \begin{cases} 1, & \text{if } \tau_t^* = \tau_i \\ 0, & \text{otherwise.} \end{cases} \tag{5b}$$

Here, $r_m^t(\tau_i)$ denotes the reward of task τ_i at round t for client m. $N_m^{t-1}(\tau_i)$ indicates the number of times that task τ_i is chosen until t. The padding function $c_m^t(\tau_i)$ [32] can help achieve a good tradeoff between exploration and exploitation, which is given by

$$c_m^t(\tau_i) = \sqrt{\frac{\delta \log n_m^t}{N_m^t(\tau_i)}} \tag{6}$$

where

$$n_m^t = \sum_{i=1}^{n} N_m^t(\tau_i) \tag{7}$$

where $\delta > 0$[6] is a parameter that controls the probability of exploration and determines the confidence level [34]. In brief, $\overline{R}_m^{t-1}(\tau_i)$ and $c_m^t(\tau_i)$ stand for exploitation and exploration, respectively. When the current behavior has been selected less often, $N_m^t(\tau_i)$ is constant, while n_m^t is increasing and $c_m^t(\tau_i)$ becomes larger. Thus, the uncertainty increases, making it more likely to be selected, and vice versa.

Besides, if task τ_i is chosen at round t, the average reward $\overline{R}_m^t(\tau_i)$ of client m with task τ_i is updated as

$$\overline{R}_m^t(\tau_i) = \frac{\left[N_m^t(\tau_i) - 1\right]\overline{R}_m^{t-1}(\tau_i) + r_m^t(\tau_i)}{N_m^t(\tau_i)}. \tag{8}$$

B. Definition of Reward Function

The client's reward consists of two metrics, namely, the bonus allocated from the red packet and the movement cost required to access the red packet. The clients prefer to go for the red packet with a high bonus and proximity, thus we define the reward function for client m with task τ_i as

$$r_m^t(\tau_i) = \mathrm{bonus}(\tau_i) - \alpha \cdot e^{d_{m\tau_i}} \tag{9}$$

where $d_{m\tau_i} = \sqrt{(x_m - x_{\tau_i})^2 + (y_m - y_{\tau_i})^2}$ denotes the linear-distance between the client m and the task τ_i. $\alpha(m/cent)$ is the positive moving cost coefficient.

The above two metrics determine the client's rewards and relate to the client's selection of red packets. On the one hand, the red packet locations with low historical visit frequency are more attractive to clients. On the other hand, the cost function as a penalty term increases exponentially with distance, making clients tend to prefer nearby red packets. Clients need to consider these two metrics together in a game-like manner to obtain the best reward.

[4] Here, the value of K is set at random to reflect the privacy sensitivity of individual clients and the determination of the optimal K value is not the focus of this article.

[5] The client can communicate with the cloud for (T-1) rounds to gain experience for the last decision to find the optimal red packet within duration D. After the duration D finishes, the client moves to get the red packet decided in the Tth round. During time D, the client can either be idle or participate in FL training. The additional data currently obtained will be used in the next FL round. In addition, this T-communication cost is only for the task selection upload and bonus feedback, which is negligible compared to the training of FL.

[6] Here, δ is set to 2. It has been proved that the corresponding regret is sublinear as long as $\delta > 0.5$ [33].

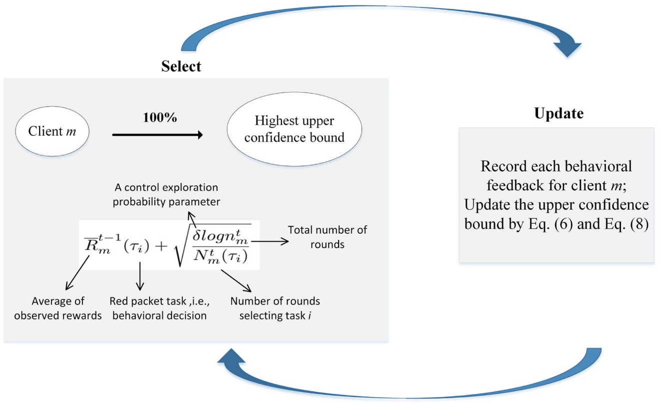

Select
Client m
100%
Highest upper confidence bound
A control exploration probability parameter
$\overline{R}_m^{t-1}(\tau_i) + \sqrt{\frac{\delta log n_m^t}{N_m^t(\tau_i)}}$
Total number of rounds
Average of observed rewards
Red packet task ,i.e., behavioral decision
Number of rounds selecting task i
Update
Record each behavioral feedback for client m; Update the upper confidence bound by Eq. (6) and Eq. (8)

Fig. 3. Basic principles of UCB.

C. Distributed Behavioral Decision

Different from the traditional MAB problem [35] solved by the UCB bandit algorithm, we tailor the UCB algorithm to better solve our problem. Considering the potential conflicts that may arise among multiple clients due to consistent global information (e.g., red packet locations and monetary amounts), we incorporate a probabilistic selection design. In detail, instead of selecting only one optimal task, we generate an optimal set S_{τ_t} by selecting the top $n' \leq |\Gamma|$ red packet tasks in terms of expected reward $\hat{R}_m^t(\tau_t)$ at round t. Next, the probability of red packet task $\tau_t \in S_{\tau_t}$ being selected can be calculated by

$$\text{pro} = \frac{\hat{R}_m^t(\tau_t)}{\sum_{\tau_t \in S_{\tau_t}} \hat{R}_m^t(\tau_t)}. \tag{10}$$

Such a probability factor design using reward ratios allows for the appropriate selection of "suboptimal" but potentially better rewarding behavior.[7] This can alleviate the problem of multiple clients selecting the same red packets due to consistent global incentive information and the fact that clients tend to concentrate their activities in popular urban areas. The details of the distributed behavioral decision (i.e., red packet task selection) algorithm for each client are shown in Algorithm 2.

Algorithm 2 consists of two main components, namely, the basic UCB algorithm and a probabilistic selection factor introduced to avoid conflicts among multiple client behaviors. The UCB logic is illustrated in Fig. 3. As the number of tests increases, the estimate of the reward that can be obtained for each action becomes more accurate. This is mathematically equivalent to increasing the confidence level of the empirical error $c_m^t(\tau_i)$. Then, the suboptimal n' behaviors τ_t' are obtained sequentially and the final behavioral decision is made according to the probability derived from (10).

D. Regret Analysis

As the choices of other clients are unknown, there is no guarantee that the action that results in the greatest reward will always be made. Therefore, in each round, when a red packet

Algorithm 2: Distributed Behavioral Decision

```
Input: Heat map $\boldsymbol{h}$, Task set $\Gamma$.
Output: The decision making $\tau_t^*$ for client $m$.
for $t = 1$ to $T$ do
    choose $\tau_t^* = \arg\max_{\tau_i \in \Gamma} \hat{\mathbf{R}}$;
    update $N_m^t(\tau_t^*)$ (refer to (5a)) and $\overline{R}_m^t(\tau_t^*)$ (refer to (8));
$\Gamma \leftarrow \Gamma \setminus \{\tau_t^*\}$;
$S_{\tau_t} = S_{\tau_t} \cup \{\tau_t^*\}$;
repeat
    choose $\tau_t' = \arg\max_{\tau_i \in \Gamma} \hat{\mathbf{R}}$;
    $\Gamma \leftarrow \Gamma \setminus \{\tau_t'\}$;
    $S_{\tau_t} = S_{\tau_t} \cup \{\tau_t'\}$;
until $(|S_{\tau_t}|) > n'$;
Calculate the probability of each $\tau_t \in S_{\tau_t}$ using pro (refer to (10));
$\tau_t^* \leftarrow \tau_t \in S_{\tau_t}$ is chosen with probability pro;
$r_m^t(\tau_t^*) \leftarrow$ Get the reward of $\tau_t^*$;
return $\tau_t^*$
```

τ_i is selected, there is a nonnegative gap $\Delta_m^t(\tau_i)$ between its reward $r_m^t(\tau_i)$ and the optimal reward $r_m^t(\tau_i^*)$. We define *regret* $\mathbb{R}_m^t$ of client m as the sum of the gaps over the rounds below

$$\mathbb{R}_m^t = \sum_{i=1}^{n} \sum_{t=1}^{T} \Delta_m^t(\tau_i) \mathbb{I}_{(\tau_t^* = \tau_i)}. \tag{11}$$

By accumulating upper bounds for each gap, we establish an upper bound on the regret of the proposed behavioral decision algorithm with the following theorem.

Theorem 1: For all rounds $t \leq T$, the proposed behavioral decision algorithm achieves an upper regret bound of

$$E\left[\mathbb{R}_m^t\right] = \mathcal{O}\left(\sqrt{|\Gamma| t \log T}\right) \tag{12}$$

where $|\Gamma|$ is the total number of red packets.

Proof: Here, we consider the last round $t \leq T$ when the deactivation rule was invoked and red packet task τ_i remained active, and let τ^* be an optimal red packet task. As per in [36, Ch. 1] and Hoeffding inequality, the confidence intervals of τ_i and τ^* must overlap at round t. Then, we can have

$$\Delta_m^t(\tau_i) = r_m^t\left(\tau^*\right) - r_m^t(\tau_i) \leq 2\left(c_m^t\left(\tau^*\right) + c_m^t(\tau_i)\right). \tag{13}$$

Since the algorithm has been alternating active tasks and both τ_i and τ^* have been active before round t, $c_m^t(\tau_i) = c_m^t(\tau^*)$ [i.e., $N_m^t(\tau_i) = N_m^t(\tau^*)$], we have $\Delta_m^t(\tau_i) = 4c_m^t(\tau_i)$.

By the selection of round t, red packet τ_i can be selected at most once afterward: $N_m^T(\tau_i) \leq 1 + N_m^t(\tau_i)$. Hence, we have the crucial property as follows:

$$\Delta_m^t(\tau_i) \leq \mathcal{O}\left(c_m^T(\tau_i)\right) = \mathcal{O}\left(\sqrt{\frac{\text{Log} n_m^T}{N_m^T(\tau_i)}}\right) \tag{14}$$

where $r_m^t(\tau_i) < r_m^t(\tau^*)$.

At round t, the contribution of red packet task τ_i to regret $\mathbb{R}_m^t(\tau_i)$ can be expressed by $\Delta_m^t(\tau_i)$ for each round when this

[7] Fig. 6 shows the benefits of the probability factor introduced in Section VI.

TABLE III
STATISTICS OF DATA SETS

Dataset	# train images	# test images	# classes
CIFAR-10	50000	10000	10
GTSRB	39209	12630	43
CIFAR-100	50000	10000	100

Fig. 4. Some cases in the three data sets.

red packet is selected. According to (14), this quantity can be bounded by

$$\begin{aligned}\mathbb{R}_m^t(\tau_i) &= N_m^t(\tau_i)\Delta_m^t(\tau_i)\\ &\le N_m^t(\tau_i)\mathcal{O}\left(\sqrt{\frac{\log n_m^T}{N_m^t(\tau_i)}}\right)\\ &= \mathcal{O}\left(\sqrt{N_m^t(\tau_i)\log n_m^T}\right). \qquad (15)\end{aligned}$$

Then, by summing up over all red packet tasks for client m, we obtain

$$\mathbb{R}_m^t = \sum_{\tau_i\in\Gamma}\mathbb{R}_m^t(\tau_i) \le \mathcal{O}\left(\sqrt{\log n_m^T}\right)\sum_{\tau_i\in\Gamma}\sqrt{N_m^t(\tau_i)}. \qquad (16)$$

Given with the fact that $f(x) = \sqrt{(x)}$ is a real concave function and $\sum_{\tau_i\in\Gamma} N_m^t(\tau_i) = t$, and Jensen's inequality, we have

$$\frac{1}{|\Gamma|}\sum_{\tau_i\in\Gamma}\sqrt{N_m^t(\tau_i)} \le \sqrt{\frac{1}{|\Gamma|}\sum_{\tau_i\in\Gamma}\sqrt{N_m^t(\tau_i)}} = \sqrt{\frac{t}{|\Gamma|}}. \qquad (17)$$

By substituting the above into (16), we can obtain $\mathbb{R}_m^t \le \mathcal{O}(\sqrt{|\Gamma| t \log T})$, and thus the upper bound $E[\mathbb{R}_m^t] = \mathcal{O}(\sqrt{|\Gamma| t \log T})$ is proved. ■

VI. EVALUATION

In this section, we first implement our proposed *PractFL* by Pytorch. Then, we examine the effectiveness of the proposed *PractFL* and evaluate its performance by comparing it with three baselines.

A. Experimental Setup

1) Data Sets: We conduct our experiments on three well-known data sets with pretrained ResNet-18 [37]. Table III shows the statistics of these data sets. The CIFAR-10 consists of ten classes of 32 × 32 images with three RGB channels. CIFAR-100 has the same format as CIFAR-10 but contains 100 classes, each with 600 color images. GTSRB [38] is a German Traffic Sign Recognition data set, which was recorded while driving on different road types in Germany during the daytime. Some cases in the three data sets are shown in Fig. 4.

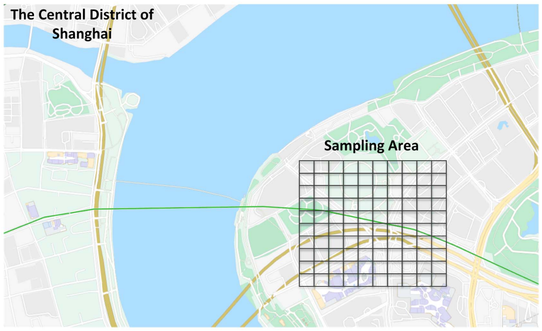


Fig. 5. Map of the experimental area.

2) Grids Division: As shown in Fig. 5, we consider a certain place with an area of 4 km^2 in a city (e.g., the central district of Shanghai) as the experimental area. Taking into account the sampling capacity of the clients, each grid is set to 0.04 km^2, and finally 100 grids are divided to achieve good coverage. Besides, the number of grids can be adjusted adaptively upon the demands on the total sampling area.

3) Data Distribution on Different Grids: We distribute these training samples into the grids that we divide above. For simulating the non-IID data distribution in the real world, we refer to the method in [39], each grid only contains partial labels of samples. Moreover, we consider that some labels might only exist in some specific grids. For example, the details about the Oriental Pearl TV Tower can only be seen in its vicinity. We distribute (1/2) of these labels in only (1/2) grids, the rest is distributed uniformly in each grid.

4) Training and Control Parameters in FL: In all our experiments, the configurations of local training are local batch size $B = 10$, local epoch $E = 5$, local learning rate $\eta = 0.01$, and the sample ratio $q = 0.3$, following the settings in [39].

B. Impact of the Red Packet Client Ratio

We first use a case study ($G = 4$, $|\mathcal{M}| = 3$, $|\Gamma| = 2$) to illustrate the process of red packet location selection and distributed behavioral decision, and we compare the proposed distributed behavioral decision with the *Greedy* strategy (where clients always choose the closest red packet).

As it can be seen in Fig. 6, the cloud center divides the grids into two major regions on the left and right and selects the location with the lowest heat value in both regions as the red packet placement point after each duration time D. Red packets are evenly distributed while taking into account the historical visit frequency, which better attracts clients to sample in infrequently visited places. Different from the *Greedy*, we can well avoid the situation where all clients select the same red packet due to the closest distance [see Fig. 6(a)]. Our strategy makes clients more interested in red packets with lower heat values and higher bonuses. This allows clients to reduce conflicts and get more rewards, and also promotes more collection of additional data.

After clients complete data collection at the red packet locations, the heat value of the sampled location increases according to the number of visitors [see Fig. 6(a) and (b)]. Furthermore, it can be seen from [Fig. 6(b)–(f)] that clients

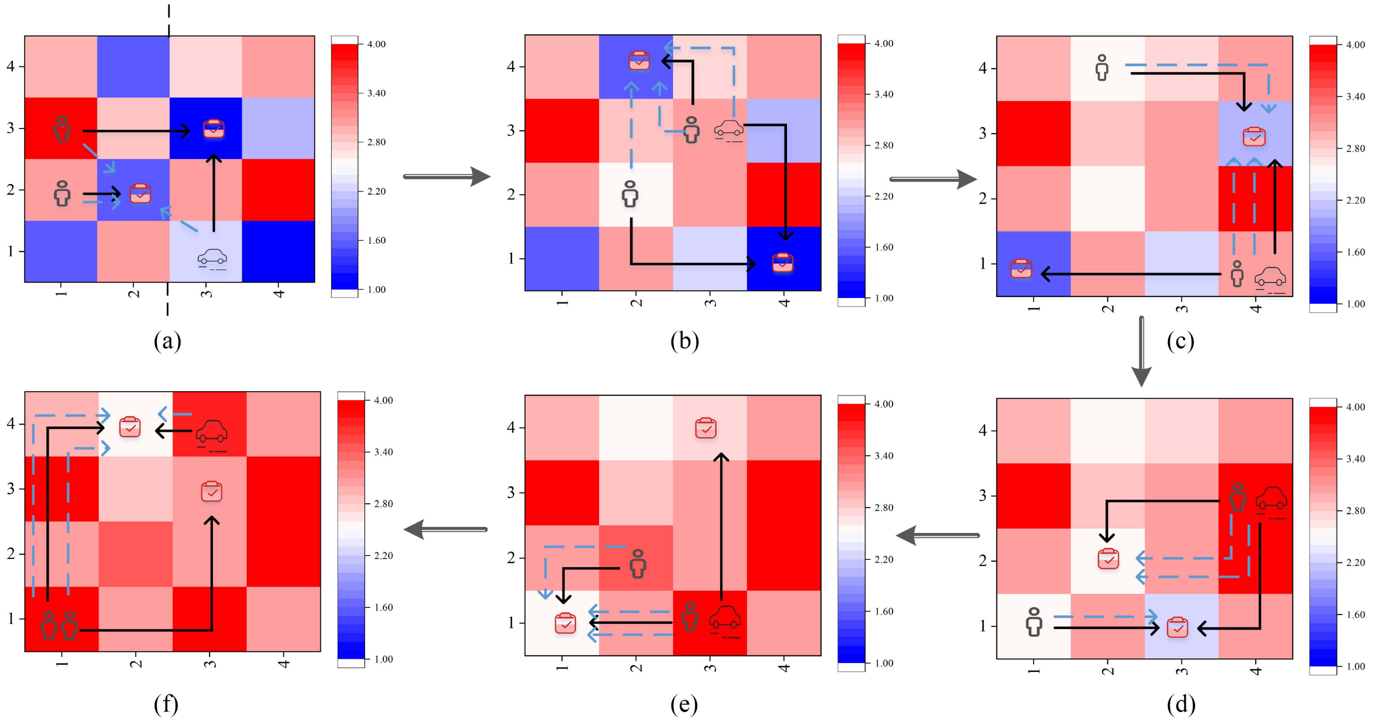


Fig. 6. Illustration for the process of red packet location selection and client behavioral decision. Here, the solid black line is the behavior of the proposed distributed behavioral decision algorithm. The blue dashed line represents the behavior of Greedy. The solid and dashed lines clearly show the different processes of the two algorithms for choosing red packets. Taking (a) and (b) as examples, the proposed distributed behavioral decision can avoid collisions well, favoring locations with low heat values, while the Greedy focuses only on physical distance, which not only leads to increased competition but also wastes a red packet. Finally, the heat map gradually changes from blue to red under the guidance of distributed behavioral decision, achieving an adequate collection of data on the grids in fewer rounds.

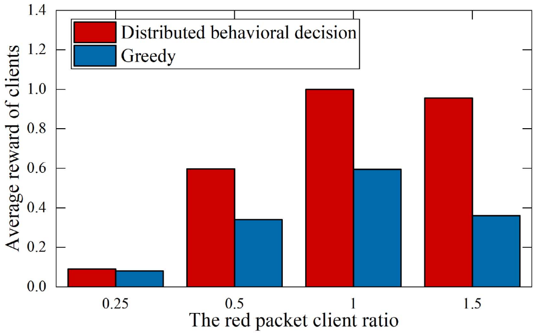


Fig. 7. Impact of red packet client ratio on the average reward of clients.

will select suboptimal red packets with a certain probability compared to the fixed choices of *Greedy*. This is because the probabilistic selection introduced in our approach plays an important role in effectively mitigating the selection conflicts due to multiple clients in the same location. Finally, we can see that the heat map gradually changes from blue to red after certain iterations of red packet issuance. With the attraction of red packets and combined with the distributed behavioral decision, the data in the grids can be fully collected in fewer rounds.

Next, we study the impact of the red packet client ratio (the ratio of the number of red packets to the number of clients) on the average reward of clients and the classification accuracy. We fix $|\mathcal{M}| = 20$, vary $|\Gamma|$ according to [5, 10, 20, 30] and set the total iteration rounds of FL to 100.

TABLE IV
AVERAGE ACCURACY UNDER DIFFERENT RED PACKET CLIENT RATIOS ON THREE DATA SETS. HERE, WE LIST THE AVERAGE ACCURACY PERFORMANCE AFTER THE LEARNING PROCESS GETS CONVERGENCE

	The red packet client ratio			
	0.25	0.5	1.0	1.5
CIFAR-10	0.7149	**0.8384**	0.7952	0.8249
GTSRB	0.9581	**0.9686**	0.9672	0.9668
CIFAR-100	0.6078	**0.6255**	0.6245	0.6165

From Fig. 7, when the red packet client ratio changes from 0.25 to 1.5, the average reward of clients increases significantly and is higher than that of the *Greedy*. This is because the increase in the number of red packets makes clients more selective and reduces the possibility of conflict. However, when the red packet client ratio exceeds 1.0, the average reward is not affected much. The reason is that we allow each client to select only one red packet task per selection in the *PractFL*. Therefore, after exceeding the number of clients, the increase in the number of red packets does not result in a greater reward for the clients. Since *Greedy* focuses only on the cost of movement, it chooses the closest red packet to itself, resulting in certain randomness. This randomness is related to the distance of the clients from the red packets.

Table IV shows that the performance of all three data sets is the lowest when the red packet client ratio is 0.25. This is because the number of red packets is too small, making it slow to converge with not much additional data collected in each FL round. We also find that the classification accuracy achieves the best at the ratio of 0.5 under our experimental setting. The accuracy does not improve as the red packet client ratio keeps increasing. Because when the ratio is 0.5, the data of most locations can be collected in a few FL rounds, and each location is visited by two clients on average. For the ratios of 1.0 and 1.5, the number of red packets is greater than or equal to the number of clients. This results in some red packets not being completed since the clients favor the most beneficial selection, and each location is visited by 0 or 1 client on average. The above facts leave some locations at a disadvantage in terms of data collection and the data is not as diverse as 0.5. However, it can cover most areas quickly due to the high red packet client ratio, so the performance does not degrade tremendously.

From Table IV and Fig. 7, we can conclude that:

1) the average reward for clients is influenced by the red packet client ratio. If the number of red packets is smaller than that of clients, the more red packets there are, the larger average reward will be received by the clients;
2) the classification accuracy is not greatly influenced by the red packet client ratio. However, the small amount of red packets causes its convergence to be too slow. Thus, it should be ensured that the number of red packets issued is not less than half of the number of clients. In the rest of the experiments, we set 0.5 as the default value of the red packet client ratio. Note that the cloud can easily adjust the number of red packets issued to meet the changing number of clients.

C. Performance Evaluation

Next, we evaluate the performance of our proposed *PractFL* on federated learning for classification by comparing it with the following approaches.

1) *Centralized Learning:* The cloud collects data sampled by all clients in each FL round for model training.
2) *Traditional FL:* We design the baseline based on [39], [40]. Clients are randomly selected in each FL round, and selected clients train their models based on local data and upload model parameters to the cloud for aggregation.
3) *Mobile FL:* We design the baseline inspired by [41] and [42]. The clients have their location preferences and routes. For example, some clients will go to urban hot spots and certain clients will go to suburban areas. They can collect additional data from this daily movement and increase the amount of local data. These trajectories are set in advance as prior knowledge for the experiment.

For each data set, we examine the performance of these four schemes. Figs. 8–10 summarize the evaluation results.

We can see that the accuracy performance of *PractFL* is greater than that of *Traditional FL* and *Mobile FL*. This is because *Traditional FL* lacks the concept of proactive sensing and ignores the mobility of clients. Even though the accuracy can be increased by selecting high-quality clients [40], it is still limited to local data on the client side, leading to unsatisfactory performance. For *Mobile FL*, additional data can be obtained by movement, which greatly improves performance compared to *Traditional FL*. However, its perceptual scope is constrained by client preferences, making it difficult to collect more data. In contrast, the incentive mechanism in *PractFL* can be a promising way to attract clients to collect data in infrequently visited places.

Moreover, we note the accuracy of *PractFL* is initially lower than *Centralized learning*, but presents slightly better than *Centralized learning* after some FL rounds. This is because in our framework, clients are continuously motivated to collect additional data. However, common data can be collected in different places, while rare data are only found in some areas, which leads to more data but class imbalance. This imbalance is even more pronounced for *Centralized learning* that is trained with the entire data, resulting in a high tendency of overfitting. The rare data (i.e., minor classes) collected has

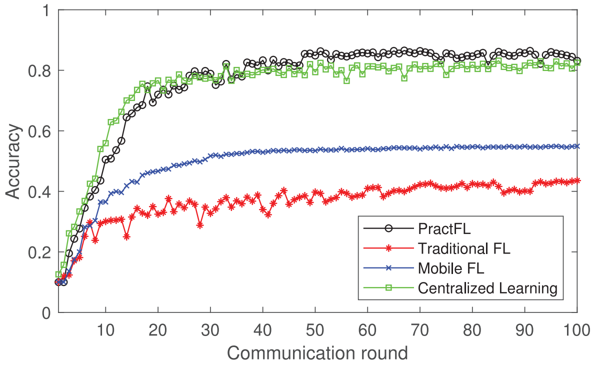


Fig. 8. Evaluation results for CIFAR-10.

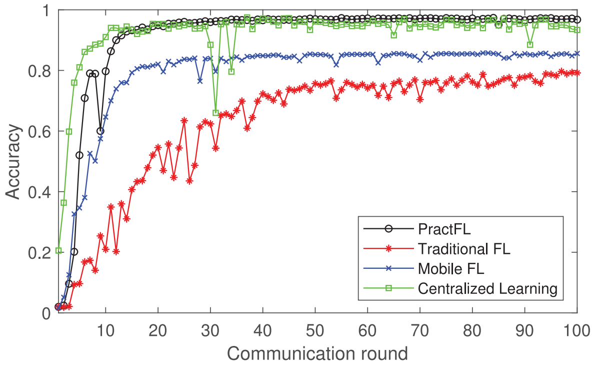


Fig. 9. Evaluation results for GTSRB.

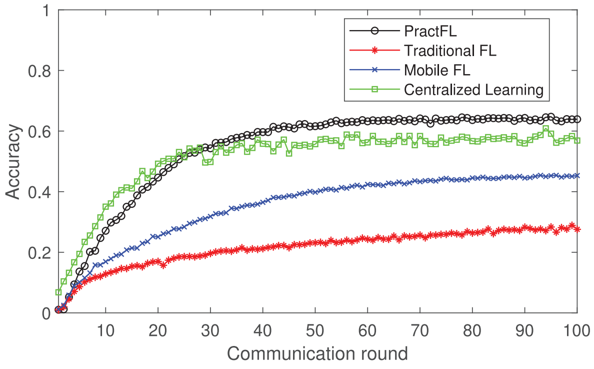


Fig. 10. Evaluation results for CIFAR-100.

a little gain on the model due to its small volume. However, *PractFL* can mitigate the class imbalance caused by data concentration. As the minor classes for *Centralized learning* are the major classes in the local data of the clients, the client local model can learn information about the rare data well, which in turn improves the global model performance. Meanwhile, the global model of *PractFL* makes a compromise among the models of clients, effectively avoiding the overfitting problem caused by class imbalance.

For CIFAR-10, we find that the performance of *Traditional FL* can only reach about 40%. The reason is that the data distribution in our experiments is non-IID, and the model performance of FL in this setting is greatly affected [40], [43]. Similarly, the performance of *Traditional FL* is only about 20% in CIFAR-100 (shown in Fig. 10) due to the impact of the non-IID setting. This is because CIFAR-100 has more categories than CIFAR-10. This leads to a more complex non-IID, which further reduces its performance. Based on the above analysis, we learn that *PractFL* can alleviate the low-performance problem caused by non-IID. In addition, we note that the performance of *Traditional FL* is significantly higher in GTSRB than in CIFAR-100 and CIFAR-10. This is because the data in GTSRB is relatively simple and has fewer categories, and is less affected by the data distribution.

Finally, we can find that proactive sensing is essential for federated learning in the real world through comparison with *Traditional FL*. Moreover, the importance of red packets guidance in the proposed incentive mechanism is emphasized compared to *Mobile FL*.

VII. Conclusion

Through the design, implementation, and evaluation of *PractFL*, this article tackles the inherent problem of data insufficiency in federated learning. Compared to state-of-the-art solutions, *PractFL* introduces the crowdsensing idea in the federated learning paradigm, empowering clients with initiatives of proactive sensing. More importantly, *PractFL* addresses two main challenges by proposing an incentive mechanism in the form of virtual red packets and a distributed behavioral decision engine based on MABs. We conduct extensive experiments on three real data sets. Experimental results demonstrate the effectiveness of our proposal and show the performance advantages over *Traditional FL*, *Mobile FL*, and *Centralized learning*.

References

[1] A. Nikitas, K. Michalakopoulou, E. T. Njoya, and D. Karampatzakis, "Artificial intelligence, transport and the smart city: Definitions and dimensions of a new mobility era," *Sustainability*, vol. 12, no. 7, p. 2789, 2020.

[2] G. Fortino, C. Savaglio, G. Spezzano, and M. Zhou, "Internet of Things as system of systems: A review of methodologies, frameworks, platforms, and tools," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 1, pp. 223–236, Jan. 2021.

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

[4] R. Chatterjee, T. Maitra, S. H. Islam, M. M. Hassan, A. Alamri, and G. Fortino, "A novel machine learning based feature selection for motor imagery EEG signal classification in Internet of Medical Things Environment," *Future Gener. Comput. Syst.*, vol. 98, pp. 419–434, Sep. 2019.

[5] M. M. Hassan, M. G. R. Alam, M. Z. Uddin, S. Huda, A. Almogren, and G. Fortino, "Human emotion recognition using deep belief network architecture," *Inf. Fusion*, vol. 51, pp. 10–18, Nov. 2019.

[6] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, 3rd Quart., 2021.

[7] Y. Deng et al., "FAIR: Quality-aware federated learning with precise user incentive and model aggregation," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May, 2021, pp. 1–10.

[8] 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 Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.

[9] J. Weng, J. Weng, H. Huang, C. Cai, and C. Wang, "FedServing: A federated prediction serving framework based on incentive mechanism," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May, 2021, pp. 1–10.

[10] R. Zeng, S. Zhang, J. Wang, and X. Chu, "FMore: An incentive scheme of multi-dimensional auction for federated learning in MEC," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Singapore, Jul. 2020, pp. 278–288.

[11] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.

[12] M. Hu, D. Wu, Y. Zhou, X. Chen, and M. Chen, "Incentive-aware autonomous client participation in federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 10, pp. 2612–2627, Oct. 2022.

[13] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "FedHealth: A federated transfer learning framework for wearable healthcare," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 83–93, Jul./Aug. 2020.

[14] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 70–82, Jul./Aug. 2020.

[15] S. Sharma, C. Xing, Y. Liu, and Y. Kang, "Secure and efficient federated transfer learning," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Los Angeles, CA, USA, Dec. 2019, pp. 2569–2576.

[16] S. Saha and T. Ahmad, "Federated transfer learning: Concept and applications," *Intelligenza Artificiale*, vol. 15, no. 1, pp. 35–44, 2021.

[17] F. Zhuang et al., "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.

[18] X. Zhang et al., "Incentives for mobile crowd sensing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 54–67, 1st Quart., 2016.

[19] T. Zhou, Z. Cai, and F. Liu, "The crowd wisdom for location privacy of crowdsensing photos," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 5, pp. 1–23, Sep. 2021.

[20] T. Zhou, B. Xiao, Z. Cai, and M. Xu, "A utility model for photo selection in mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 20, no. 1, pp. 48–62, Jan. 2021.

[21] K. I.-K. Wang, X. Zhou, W. Liang, Z. Yan, and J. She, "Federated transfer learning based cross-domain prediction for smart manufacturing," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4088–4096, Jun. 2022.

[22] Y. Jiang, R. Cong, C. Shu, A. Yang, Z. Zhao, and G. Min, "Federated learning based mobile crowd sensing with unreliable user data," in *Proc. IEEE Int. Conf. High Perform. Comput. Commun. (HPCC)*, Dec. 2020, pp. 320–327.

[23] Y. Wang, Z. Su, N. Zhang, and A. Benslimane, "Learning in the air: Secure federated learning for UAV-assisted Crowdsensing," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1055–1069, Apr.-Jun. 2021.

[24] B. Zhao, X. Liu, and W.-N. Chen, "When crowdsensing meets federated learning: Privacy-preserving mobile crowdsensing system," 2021, *arXiv:2102.10109*.

[25] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, "A crowdsourcing framework for on-device federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3241–3256, May 2020.

[26] S. M. Usman, F. A. S. Bukhari, M. Usman, D. Badulescu, and M. S. Sial, "Does the role of media and founder's past success mitigate the problem of information asymmetry? evidence from a U.K. crowdfunding platform," *Sustainability*, vol. 11, no. 3, p. 692, 2019.

[27] X. Zhu, "Case VI: Changing with the times: AutoNavi's autonomous development," in *China's Technology Innovators*. Singapore: Springer, 2018, pp. 111–134.

[28] D. Wu et al., "When sharing economy meets IoT: Towards fine-grained urban air quality monitoring through mobile crowdsensing on bike-share system," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 4, no. 2, pp. 1–26, 2020.

[29] T. Zhou, B. Xiao, Z. Cai, M. Xu, and X. Liu, "From uncertain photos to certain coverage: A novel photo selection approach to mobile crowdsensing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Honolulu, HI, USA, Apr. 2018, pp. 1979–1987.

[30] H. Li, L. Gong, B. Wang, F. Guo, J. Wang, and T. Zhang, "*k*-anonymity based location data query privacy protection method in mobile social networks," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Ürümqi, China, 2020, pp. 326-334.

[31] H. Wang, K. Wu, J. Wang, and G. Tang, "Rldish: Edge-assisted QoE optimization of HTTP live streaming with reinforcement learning," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 706–715.

[32] A. Garivier and E. Moulines, "On upper-confidence bound policies for switching bandit problems," in *Algorithmic Learn. Theory*, J. Kivinen, C. Szepesvári, E. Ukkonen, and T. Zeugmann, Eds. Berlin, Germany: Springer, 2011, pp. 174–188.

[33] B. Wu, T. Chen, K. Yang, and X. Wang, "Edge-centric bandit learning for task-offloading allocations in multi-RAT heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3702–3714, Apr. 2021.

[34] X. Nie, Y. Zhao, D. Pei, G. Chen, K. Sui, and J. Zhang, "Reducing Web latency through dynamically setting tcp initial window with reinforcement learning," in *Proc. IEEE/ACM Int. Symp. Qual. Service (IWQoS)*, Banff, AB, Canada, Jun. 2018, pp. 1–10.

[35] F. Li, D. Yu, H. Yang, J. Yu, H. Karl, and X. Cheng, "Multi-armed-bandit-based spectrum scheduling algorithms in wireless networks: A survey," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 24–30, Feb. 2020.

[36] A. Slivkins, "Introduction to multi-armed bandits," *Found. Trends®Mach. Learn.*, vol. 12, nos. 1–2, pp. 1–286, 2019. [Online]. Available: http://dx.doi.org/10.1561/2200000068

[37] Q. A. Al-Haija, M. A. Smadi, and S. Zein-Sabatto, "Multi-class weather classification using ResNet-18 CNN for autonomous IoT and CPS applications," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Las Vegas, USA, Dec. 2020, pp. 1586–1591.

[38] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, pp. 323–332, Aug. 2012.

[39] H. Zeng, T. Zhou, Y. Guo, Z. Cai, and F. Liu, "FedCav: Contribution-aware model aggregation on distributed heterogeneous data in federated learning," in *Proc. Int. Conf. Parallel Process. (ICPP)*, Chicago, IL, USA, Aug. 2021, p. 75.

[40] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-IID data with reinforcement learning," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 1698–1707.

[41] J. Wang et al., "HyTasker: Hybrid task allocation in mobile crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 3, pp. 598–611, Mar. 2020.

[42] B. Guo, C. Chen, D. Zhang, Z. Yu, and A. Chin, "Mobile crowd sensing and computing: When participatory sensing meets participatory social media," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 131–137, Feb. 2016.

[43] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

Hongjia Wu received the B.S. degree from Liaoning Technical University, Fuxin, China, in 2016, and the M.S. degree from Guangxi University, Nanning, China, in 2018. She is currently pursuing the Ph.D. degree with the National University of Defense Technology, Changsha, China.

She is currently a visiting student with the Singapore University of Technology and Design, Singapore. Her research interests include computation offloading, game theory, resource allocation, and dispersed computing.

Hui Zeng received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2020, and the M.S. degree from the National University of Defense Technology, Changsha, China, in 2022, where he is currently pursuing the Ph.D. degree.

His research interests include data privacy, federated learning, and distributed systems.

Tongqing Zhou received the bachelor's, master's, and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), Changsha, China, in 2012, 2014, and 2018, respectively.

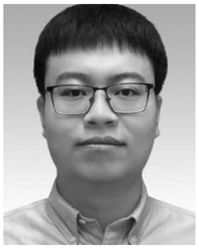
He was a Research Assistant with Hong Kong Polytechnic University, Hong Kong. He is currently a Postdoctoral Fellow with the College of Computer, NUDT. His main research interests include ubiquitous computing, mobile sensing, and data privacy.

Dr. Zhou is the recipient of the Outstanding Ph.D. Dissertation Award and the Outstanding Postdoctoral Fellow, both of Hunan Province, China.

Zhiping Cai received the B.Eng., M.A.Sc., and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), Changsha, China, in 1996, 2002, and 2005, respectively.

He is a Full Professor with the College of Computer, NUDT. His current research interests include artificial intelligence, network security, and big data.

Prof. Cai is a Distinguished Member of CCF.



Zehui Xiong (Member, IEEE) received the Ph.D. degree from Nanyang Technological University, Singapore, in June 2020.

He is currently an Assistant Professor with the Pillar of Information Systems Technology and Design, Singapore University of Technology and Design, Singapore. Prior to that, he was a Researcher with Alibaba–NTU Joint Research Institute, Singapore. He was a Visiting Scholar with Princeton University, Princeton, NJ, USA, and the University of Waterloo, Waterloo, ON, Canada. He has published more than 120 research papers in leading journals and flagship conferences and seven of them are ESI Highly Cited Papers. His research interests include wireless communications, network games and economics, blockchain, and edge intelligence.

Dr. Xiong has won numerous best paper awards in international conferences and technical committee. He is the recipient of the Chinese Government Award for Outstanding Students Abroad in 2019 and the NTU SCSE Best Ph.D. Thesis Runner-Up Award in 2020. He is the Founding Vice Chair of Special Interest Group on Wireless Blockchain Networks in IEEE Cognitive Networks Technical Committee. He is currently serving as an Editor or a Guest Editor for many leading journals, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, and IEEE SYSTEMS JOURNAL.

Dusit Niyato (Fellow, IEEE) received the B.Eng. degree from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Winnipeg, MB, Canada, in 2008.

He is a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are in the areas of sustainability, edge intelligence, decentralized machine learning, and incentive mechanism design.

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

From 2000 to 2002, he was a Research and Development Engineer with JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate with the University of Maryland at College Park. From 2006 to 2008, he was an Assistant Professor with Boise State University, Boise, ID, USA. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department and the Computer Science Department, University of Houston, Houston, TX, USA. His main research targets on the novel game theory-related concepts critical to enabling efficient and distributive use of wireless networks with limited resources. His other research interests include wireless resource allocation and management, wireless communications and networking, quantum computing, data science, smart grid, and security and privacy.

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