DeePGA: A Privacy-Preserving Data Aggregation Game in Crowdsensing via Deep Reinforcement Learning

Yang Liu, *Member, IEEE,* Hongsheng Wang, Mugen Peng, *Fellow, IEEE,* Jianfeng Guan, *Member, IEEE,* Jia Xu, *Member, IEEE,* and Yu Wang, *Fellow, IEEE.*

***Abstract*—The Internet of Things (IoT) has such a profound impact that we have witnessed crowdsensing has emerged as the most popular sensing paradigm where participants sense and aggregate data to the platform by smart devices. However, the participants may not be willing to involve in data sensing and aggregation if they are not sufficiently compensated or their personalized private information are disclosed. In order to overcome the above issues, this paper proposes a payment-privacy protection level (PPL) game, where each participant submits his sensing data with a specified PPL while the platform chooses a corresponding payment to the participant. Additionally, we derive the Nash equilibrium (NE) point of the game. Considering that the payment-PPL model is unknown in practice, we employ a reinforcement learning technique, i.e., Q-learning to obtain the payment-PPL strategy in a dynamic payment-PPL game. We further use deep Q network (DQN), which combines a deep learning technique with Q-learning to accelerate learning speed. Through extensive simulations, we verify that our proposed algorithm using DQN achieves superior performance in terms of utilities of both platform and participants and data aggregation accuracy compared with the one using Q-learning.**

***Index Terms*—Data aggregation, crowdsensing, differential privacy, equilibrium, Q-learning, deep reinforcement learning.**

I. Introduction

THE development of IoT technology and the popularization of mobile smart terminals have become the cornerstone for the construction of smart cities. The emergence of various IoT based applications has fostered the development of wireless sensor networks, wireless body area networks, vehicular networks, and so on. Among them, a new paradigm, i.e., crowdsensing which finds ideas and solves large-scale computing or sensing tasks with the help of public wisdom has attracted great attention [1]. Compared with traditional paradigms, participants in crowdsensing not only act as the ultimate consumers of data, but also play more roles, including data transmission, analysis, and so on. At present, crowdsensing has entered a rapid development stage, and its application involves all aspects of people's work and life. More specifically, typical applications include environmental monitoring [2], intelligent transportation [3], social sensing [4], road condition detection [5], air quality monitoring [6], indoor positioning [7], etc.

When taking part in crowdsensing, participants need to finish sensing tasks through sensors such as camera, gyroscope, accelerometer, and so on, and they need to upload sensing data to the platform, incuring costs such as battery and bandwidth. In this case, a rational participant may provide sensing or computing services only if he is incentivized by reward. Therefore, it is necessary to design an incentive mechanism to encourage smartphone users to participate in crowsensing. Game theory is an important way to solve incentive problems, by adjusting the payments of participants to reflect user engagement, participation contributions, and so on.

An important feature of crowdsensing is that platform can collect sensitive personal information of participants such as locations and social relations [8], and thus deduce their occupation, preferences, etc., which are of great value, as shown in Fig. 1. Therefore, in crowdsensing, it is a challenging task to protect participants' individual information from being leaked, at the same time to extract accurate sensing data to guarantee that the sensing tasks can be completed. A traditional approach is, when there is a sensing request, the sensing data is uploaded to the platform first, then it is added noise uniformly. However, the privacy sensitivity of different participants for their sensing data is not considered in this approach.

Currently, differential privacy [9], among all the existing privacy protection mechanisms, has gained great attention due to the fact that it provides a strong theoretical guarantee for individual data in aggregated statistics. However, the utility of platform will be damaged if the participants add unified noise to the sensing data when uploading it to the platform. In this case, it is reasonable to consider different privacy levels. On the one hand, it can differentiate the privacy sensitivities for different participants. On the one hand, it can evaluate the privacy of participants more accurately.

In this paper, we propose a personalized privacy-preserving data aggregation game based on the interactions between platform payment and participants' PPLs, which is to meet

Yang Liu, Hongsheng Wang, Mugen Peng, and Jianfeng Guan are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China (email:{liu.yang, wanghongsheng, pmg, jfguan}@bupt.edu.cn) (Corresponding Authors: Yang Liu and Mugen Peng).

Yang Liu and Jia Xu are with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China (email: liu.yang@bupt.edu.cn, xujia@njupt.edu.cn).

Yu Wang is with the Department of Computer and Information Sciences, Temple University, Philadelphia, Pennsylvania 19122, USA, (email:wangyu@temple.edu).

Manuscript received August, 2019.

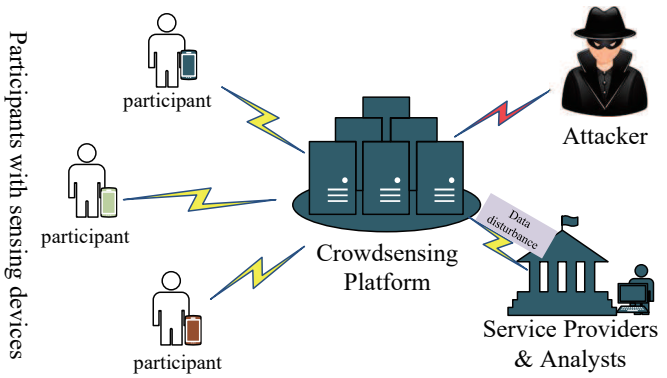


Fig. 1: An attacker can deduce participants' personal information from crowdsensing platform if the sensing data is not disturbed.

the requirement of different privacy sensitivities for different participants. For the privacy-preserving data aggregation game, only one sensing task is considered, and the platform verifies the correctness of participants' PPLs and pays the corresponding payment. Through such repeated interactions, the platform and participants constantly adjust their strategies, in which participants tend to obtain more payment, and the platform is keen to acquire more accurate sensing data. In this case, on the participant side, on the one hand, the impact of invalid data on the aggregated sensing data can be suppressed; on the other hand, participants' privacy can be personalized. On the platform side, on the one hand, it aims to encourage participants to participate in crowdsensing and reduce their privacy by enabling them to obtain the payment they deserve; on the other hand, it also aims to achieve a balance between platform utility and data aggregation accuracy.

As shown in Fig. 2, the platform as the leader first broadcasts the payment to the participants, and then the participants as the followers upload their PPLs and sensing data to the platforms. We derive an NE for our personalized privacy-preserving data aggregation game. More specifically, more payment will enforce participants to choose lower PPLs, and less payment may lead to more invalid data which is beyond the data range of actual physical significance. In this way, the utilities of both platform and participants can be maximized by an appropriate payment-PPL matching strategy.

The platform's payment and the participants' PPLs can be seen as a finite Markov decision process. In this paper, we assume that the transition probabilities of payment levels under the current state and reward function are unknown to the participants, and the transition probabilities of PPLs under the current state are also unknown to the platform. The proposed game runs for several rounds, each round does not need to sample the information of the last round to update strategy. In this case, we propose to use Q-learning, a reinforcement learning algorithm to update strategy in real time. Under Q-learning algorithm, we introduce two Q-functions, which are the discounted long-term reward for a state-action pair. Platform use a Q-function which is related to data aggregation accuracy to obtain optimal payment, and participants use a Q-function which is related to their own payment to obtain optimal PPLs.

However, with the increase of the number of participants, PPLs, and payment levels, the size of Q-table increases exponentially and the utility convergence speed of platform will be greatly reduced. In order to overcome the above problem, we propose to employ DQN, which is a deep reinforcement learning technique. More specifically, on the participant side, we use Q-learning due to limited resources of smart phones. On the platform side, we use DQN to accelerate the acquisition of optimal payment policy, thereby increasing the utility of platform.

This paper's contributions are as follows:

- *Static Payment-PPL Game*: We formulate a payment-PPL game and derive its NE, which reveals the balance between platform payment and participants' PPLs.
- *Dynamic Payment-PPL Game Based Q-learning*: We propose to use Q-learning to learn the payment policy of platform and PPLs of participants respectively, so as to solve the dynamic game under an unknown payment-PPL model.
- *Dynamic Payment-PPL Game Based DQN*: We propose to use DQN to accelerate the speed of acquiring payment-PPL strategy. The proposed DQN based algorithm shows that compared with Q-learning both platform and participants can gain more utilities and the time to obtain the optimal strategies is reduced.

The remainder of paper is organized as follows. We review related work in Sec. II. Sec. III presents the system model for data aggregation. Sec. IV formulates the static privacy-preserving data aggregation game and derive the NE of the game. We propose a DQN-based payment-PPL strategy for the dynamic privacy-preserving data aggregation game in Sec. V. Sec. VI presents simulation results. Finally, Sec. VII concludes the paper.

II. Related Work

Privacy in crowdsensing has gained great attention with many pieces of work explored. In [10], the authors study the real-time release of data in an untrustworthy situation, and propose a distributed privacy protection framework based on differential privacy. The authors in [11] discuss a data privacy protection scheme based on one-way hashing, marking hybrid network and packet in the scenario of fog computing. The authors in [12] use k-anonymity to protect user privacy and use incentive payment mechanisms to reduce the loss of data information. In [13], the authors propose a personalized privacy protection and task allocation framework, in which privacy protection is based on differential privacy, and task assignment is a mechanism that utilizes probability to win. Several work has focused on data aggregation in privacy-preserving crowdsensing. In [14], the authors employ fog nodes in different regions to assist the crowdsensing server to achieve privacy-preserving task allocation and data aggregate. In [15], the authors propose an anonymized data collection method which is able to accurately estimate data distributions. In [16], the authors propose a k-anonymous privacy preserving scheme for multimedia sensing data by integrating data coding

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/JIOT.2019.2957400, IEEE Internet of Things Journal

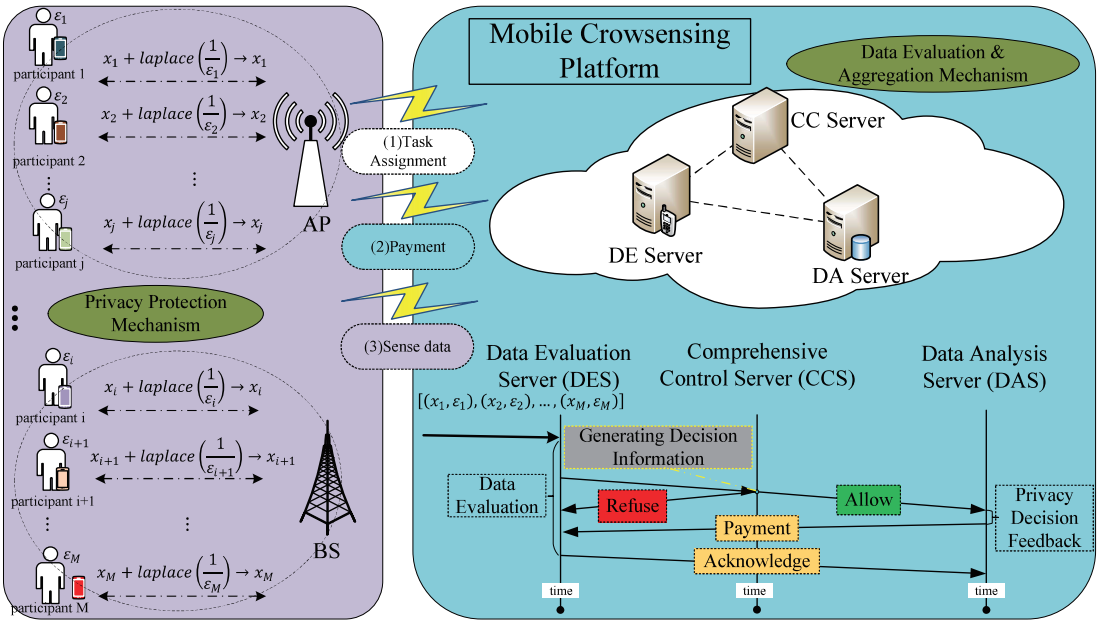


Fig. 2: Our proposed system consists of a platform which evaluates and aggregates sensing data and participants who execute a privacy protection mechanism.

and message transfer. The authors in [17], [18] achieve privacy protection by using blockchain. In [19], the authors infer the private information of users using embedded sensors in smart devices via deep learning. However, these work does not consider how to incentivize participants to participate in crowdsensing, where participants may be reluctant to contribute sensing services without sufficient incentives.

In crowdsensing, an efficient incentive mechanism can greatly promote the enthusiasm of participants. In the crowdsensing environment of the Internet of Vehicles, the authors in [20] rely on reinforcement learning and game theory to solve the trade-off between sensing accuracy and the overall payment of crowdsensing server. In [21], the authors use game theory to solve the problem of profit distribution among providers and to associate service quality with privacy levels. CENTURION [22] offers a double auction mechanism to stimulate data requesters to publish sensing requests and workers to participate in sensing tasks in order to achieve data aggregation. However, auction mechanisms focus on multiple tasks in crowdsensing. For one sensing task, game theory is usually employed to design incentive mechanisms. In [23], the authors present a crowdsourcer-centric model where an incentive mechanism is designed by using a Stackelberg game. In [24], the authors design an incentive mechanism by applying a two-stage Stackelberg game to determine the incentive of service provider and the participation levels of mobile users. In [25], the authors motivate mobile users to participate in crowdsensing by using a multi-stage stochastic programming based game. In [26], the authors motivate data carrier and mobile relay users to contribute data collection in mobile opportunistic crowdsensing by using a two-user cooperative game. However, these work all assumes the interactions between the platform and participants have a specific model. In [27], the authors design an incentive mechanism by formulating a model-free multi-leader and multi-follower Stackelberg game, and use deep reinforcement learning to obtain the optimal pricing strategies of task initiators. A pioneer study on networked data integration with machine learning methods is conducted in [28], where a directionality learning model is proposed with edge-based network representation. However, all the above work do not consider privacy issues in crowdsensing.

Some work have been done to encourage participants to take part in privacy-preserving crowdsensing. In [29], the authors design a differential privacy auction mechanism to minimize the platform's payment by taking each worker's bid privacy into consideration. The authors in [30] propose a truthful incentive mechanism with location privacy-preserving for mobile crowdsourcing systems. INCEPTION [31] introduces a crowdsensing framework which taking incentive, data aggregation and perturbation into consideration. In [32], the authors propose to put the participants into multiple groups and perform auctions within the groups to protect bid information. However, these work considers that participants perform multiple sensing tasks, which is not consistent with our work where only one data aggregation task is needed by participants. REAP [33] offers different contracts to participants with different privacy preferences to reconcile fusion center's aggregation accuracy and individual participant's data privacy. However, it does not consider how to maximize the utilities of both platform and participants.

III. System Model and Design

In this section, we first give an overview of our personalized privacy-preserving system and describe our task model, private protection model, data evaluation and aggregation model.

TABLE I: The main notations through the paper.

M	The number of participants
$\boldsymbol{\varepsilon}$	The action set of participants
$\boldsymbol{p}$	The action set of platform
$\varepsilon_i^{(t)}$	PPL of participant i in time slot t
$p^{(t)}$	Payment policy strategy
J	The number of PPLs
N	The number of payment levels
c_i	The cost of participant i excluding privacy cost
$\boldsymbol{s}^{(t)}$	The state of platform in time slot t
$\boldsymbol{s}_i^{(t)}$	The state of participant i in time slot t
α	The learning rate of Q-learning/DQN
$\varphi^{(t)}$	State sequence in time slot t
$\theta^{(t)}$	The weight of CNN in time slot t
B	The minibatch size of CNN
W	The experience size of CNN input sequence

A. System Overview

The goal of our crowdsensing platform is to recruit M smartphone users located in the areas of interests to collect sensing data and create a crowdsensing application, as shown in Fig. 2. The crowdsensing platform first selects its payment policy and broadcasts a recruitment message that lists the payment-PPL pairs. Each selfish and rational participant chooses his PPL, according to his energy consumption and data perturbation cost.

The platform first broadcasts location-based sensing tasks to the participants. Then, the participants use their sensors (such as smartphones, portable computers and environmental monitoring sensors) to collect sensing data and send them with PPLs to the platform through base stations (BSs) and access points (APs).

The platform first evaluates each participant's data through a data evaluation server (DES). We assume that each participant is rational such that the submitted PPL is accurate. As a result, a participant who offers a higher PPL requires more payment from the platform. The payment strategy of platform is given by a comprehensive control server (CCS). The data analysis server (DAS) of platform is set up for data aggregation visualization, e.g., a web server. The crowsensing system provides a trade off between platform payment and participants' PPLs. Also if the platform has a higher budget, it can pay more to motivate participants to accept sensing tasks. For the ease of our reference, our commonly used notations are summarized in Table I.

B. Task Model

Considering the time sensitivity of sensing data, we define time-invariant tasks τ_1 and time-varying tasks τ_2, so the sensing tasks are indicated by $\boldsymbol{\tau} = \tau_1 \cup \tau_2$. For time-invariant tasks such as sensing a profile of an object, in order to avoid the fact that the sensing data is far from the actual data, we consider collecting sensing data multiple times. If collecting V times, the sensing results for participant i is given as follows:

$$\boldsymbol{x}_i = \{x_{1,i}, x_{2,i}, \ldots, x_{V,i}\}, \tag{1}$$

and we take average of the sensing results as the final result, i.e.,

$$x_i = \frac{1}{V}\sum_{j=1}^{V} x_{j,i}. \tag{2}$$

In a time-varying task such as monitoring the air condition of some areas of interest within a time period, we consider sensing data for multiple locations. If sensing U locations, the sensing results for participant i in time slot t is given as follows:

$$\boldsymbol{x}_i^{(t)} = \left\{x_{1,i}^{(t)}, x_{2,i}^{(t)}, \ldots, x_{U,i}^{(t)}\right\}, \tag{3}$$

then we use the average of all locations as the final result in time slot t, i.e.,

$$\bar{x}_i^{(t)} = \frac{1}{U}\sum_{j=1}^{U} x_{j,i}^{(t)}. \tag{4}$$

If the sensing time range is T, the sensing results for participant i for time period T is given as follows:

$$\boldsymbol{x}_i^{(T)} = \left\{x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(T)}\right\}, \tag{5}$$

then we use the average of all time slots as the final result for time period T, i.e.,

$$\bar{x}_i^{(T)} = \frac{1}{T}\sum_{t=1}^{T} x_i^{(t)}. \tag{6}$$

C. Privacy Protection Model

We use differential privacy to provide PPLs for the data collected by each participant. Each participant can independently conduct obfuscation of the data collected, and then upload the obfuscated sensing data and PPL to the platform for evaluation. Here, we assume that the platform can accurately detect each participant's PPL.

Different types of sensing data may have different tolerable error ranges, i.e., sensitivity. For example, a human body temperature's tolerable error range is 1, while the tolerable error range of a vehicle's velocity is 10, we normalize sensing data to the range $[0, 1]$. If the data submitted to the platform is out of range, the system will prohibit this participant from participating in this task. We define sensitivity as follows:

Definition 1 (**l-Sensitivity**). *$D_{(d)}$ and $D'_{(d)}$ are adjacent data which satisfies l-sensitivity, if $\left\|D_{(d)} - D'_{(d)}\right\|_1 \leq l$, where l is the data range of sensing data, d is data dimension, $||_1$ is the 1-th order norm distance.*

Next, we define differential privacy as follows:

Definition 2 (**Differencial Privacy [9]**). *Suppose ε is a positive real number, f represents a random algorithm. For two adjacent data sets D and D', if any output result x of algorithm f on data sets D and D' satisfies the following inequality,*

$$\Pr\left[f\left(D\right) = x\right] \leq \exp(\varepsilon)\Pr\left[f\left(D'\right) = x\right], \tag{7}$$

f satisfies ε-differential privacy.

For different types of sensing data, privacy protection methods are different. There are two common noise adding

mechanisms, namely Laplace mechanism and exponential mechanism. The former is for numerical results and the latter is for non-numeric results. In this paper, we only consider numerical data, so participant i uploads data x_i with Laplace mechanism, i.e., $x_i = x_i + Laplace(0, \frac{l}{\varepsilon_i})$. As shown in Fig. 3, if the obfuscated data of participant i is x_i, its actual data may be $\tilde{x}_i$ or $\tilde{x}_i^{'}$ with probability p_i and $p_i^{'}$, so an attacker is difficult to determine the actual sensing data. Additionally, q_i and $q_i^{'}$ represent the probabilities that the actual data d_i is selected under different PPLs. We can see from the figure that the higher the PPL, the higher probability that the actual sensing data is selected, the easier the data is leaked.

For time-invariant tasks, we have the following threom:

Theorem 1. *For a time-invariant task, a participant performs sensing V times. For the i-th sensing, the sensing data achieves $\varepsilon_{j,i}$-differential privacy. By adding noise which follows Laplace distribution with probability density function (pdf) $\frac{1}{2b}e^{-\frac{|x_{j,i}|}{b}}$, where $b = \frac{Vl}{\sum_{i=1}^{V}\varepsilon_{j,i}}$, the time-invariant task satisfies $\sum_{i=1}^{V}\varepsilon_{j,i}$-differential privacy.*

Proof. See Appendix A. □

Similarly, the following theorem is for time-varying tasks:

Theorem 2. *For a time-varying task at time slot t, a participant performs sensing U locations. For the i-th location, the sensing data achieves $\varepsilon_i^{(t)}$-differential privacy. By adding noise which follows Laplace distribution with pdf $\frac{1}{2b}e^{-\frac{|x_{j,i}^{(t)}|}{b}}$, where $b = \frac{l}{\max \varepsilon_{j,i}^{(t)}}$, the time-varying task satisfies $\max \varepsilon_{j,i}^{(t)}$-differential privacy.*

Proof. See Appendix B. □

Theorem 3. *For a time-varying task for time period T, by adding noise which follows Laplace distribution with pdf $\frac{1}{2b}e^{-\frac{|x_i^{(t)}|}{b}}$, where $b = \frac{l}{\max\limits_{1\le t\le T}\max\limits_{1\le k\le U}\varepsilon_{k,i}^{(t)}}$, the time-varying task satisfies $\max\limits_{1\le t\le T}\max\limits_{1\le k\le U}\varepsilon_{k,i}^{(t)}$-differential privacy.*

Proof. The proof is similar to that of Theorem 2. □

D. Data Evaluation and Aggregation Model

Before data aggregation, we first evaluate the effectiveness of sensing data, we give the follow definition:

Definition 3 (β-**Effectiveness**). *For the sensing data with noise, i.e., x uploaded by a participant at one time, if it satisfies:*

$$D(x) < \beta, \tag{8}$$

x is valid, otherwise x is considered invalid, where $D(x)$ is defined as follows:

$$D(x) = \sqrt{(x-\mu)^T H^{-1}(x-\mu)}, \tag{9}$$

where μ is the mean value of sensing data x, the calculation method is shown in Eqs. (2) and (4), and H^{-1} is covariance of sensing data x.

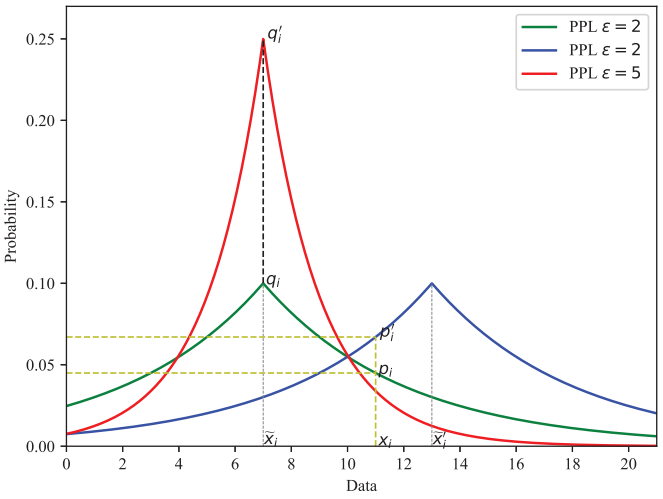


Fig. 3: Illustration of Private Protection Model.

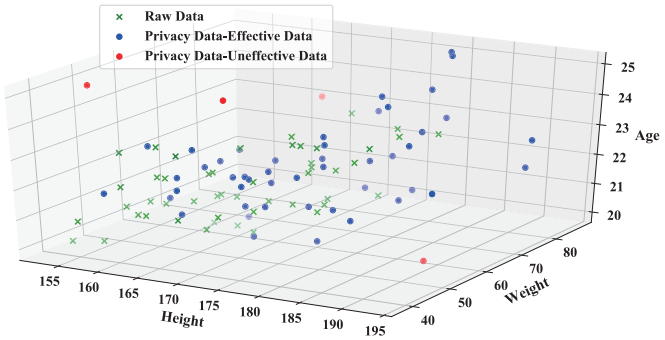


Fig. 4: Data evaluation for the student information of a college, where β is 6.

Fig. 4 presents an example of data evaluation for the student information of a college.

Definition 4 ((λ,η)-**Accuracy**). *The aggregated result $\hat{x}$ of sensing data can achieve (λ,η)-accuracy, if*

$$\Pr[|\hat{x} - x| \ge \lambda] \le 1 - \eta \tag{10}$$

where x is the sensing data with noise.

This definition means that probability of aggregation error λ is limited by $1-\eta$. From the perspective of estimation theory, λ represents the confidence interval, η represents the confidence level. Then, we propose to deduce the relationship between participants' PPLs and platform aggregation error.

Theorem 4. *For a given $\eta \le 1$, the aggregation error λ of sensing data under our privacy protection mechanism is given by the following formula:*

$$\lambda = l\frac{(1-\sigma)\sum_{i=1}^{M}\frac{1}{\varepsilon_i} + \sqrt{(1-\sigma)^2(\sum_{i=1}^{M}\frac{1}{\varepsilon_i})^2 + 8(1-\eta)\sigma\sum_{i=1}^{M}\frac{1}{\varepsilon_i^2}}}{2M(1-\eta)}, \tag{11}$$

where $\sigma \in (0,1)$ is a control parameter. We can see that the platform hopes to obtain higher PPLs to reduce aggregation error, while participants hope to adopt lower PPLs to protect their privacy better.

Proof. We introduce a generalized form of Chebyshev inequality combined with Markov inequality:

$$P(|x - E(x)| \geq \lambda) = P(|x| \geq \lambda) \leq \sigma \frac{Var(x)}{\lambda^2} + (1-\sigma)\frac{E(|x|)}{\lambda}, \sigma \in (0,1) \tag{12}$$

thus we have:

$$(1-\sigma)\lambda^2 - [(1-\sigma)\frac{l}{M}\sum_{i=1}^{M}\frac{1}{\varepsilon_i}]\lambda - \frac{2l^2}{M^2}\sum_{i=1}^{M}\frac{1}{\varepsilon_i^2} = 0. \tag{13}$$

Then we have

$$\lambda = l\frac{(1-\sigma)\sum_{i=1}^{M}\frac{1}{\varepsilon_i} + \sqrt{(1-\sigma)^2(\sum_{i=1}^{M}\frac{1}{\varepsilon_i})^2 + 8(1-\eta)\sigma\sum_{i=1}^{M}\frac{1}{\varepsilon_i^2}}}{2M(1-\eta)} \tag{14}$$

For the ease of the following discussion, we let $\sigma = 1$, then we have

$$\lambda = \frac{\sqrt{2}l}{M\sqrt{1-\eta}}\sqrt{\sum_{i=1}^{M}\frac{1}{\varepsilon_i^2}}. \tag{15}$$

□

After the participants upload sensing data to the platform, PES first evaluates sensing data by using Eq. (9). We denote $\hat{N}$ as the number of valid data, which is given as follows:

$$\hat{N} = \sum_{i=1}^{K} I(\beta(x_i)) \tag{16}$$

where $I(\cdot)$ is the indication function which judges whether x_i satisfies $\beta(x_i)$-Effectiveness and K is the number of uploaded sensing data .

For time-invariant tasks, the valid sensing result is:

$$\bar{x}_i = \frac{1}{\hat{N}}\sum_{j=1}^{\hat{N}} x_{j,i}, \tag{17}$$

where $x_{\hat{N},i}$ represents effective data, and $\hat{N} \leq V$.

According to Theorem 1 we can obtain that the aggregated sensing data of time-invariant tasks satisfies $(\lambda, \frac{2}{\lambda^2\hat{N}^2}\sum_{i=1}^{\hat{N}}(\frac{l}{\sum_{j=1}^{V}\varepsilon_{j,i}})^2)$-accuracy.

For time-varying tasks, the valid sensing result at time slot t is:

$$\bar{x}_i^{(t)} = \frac{1}{\hat{N}}\sum_{j=1}^{\hat{N}} x_{j,i}^{(t)}, \tag{18}$$

where $x_{j,i}^{(t)}$ represents effective data, and $\hat{N} \leq U$.

Similarly, according to Theorem 2 we can obtain that the aggregated sensing data of time-varying tasks satisfies $(\lambda, \frac{2}{\lambda^2\hat{N}^2}\sum_{i=1}^{\hat{N}}(\frac{l}{\max\varepsilon_{j,i}^{(t)}})^2)$-accuracy.

If the sensing data is invalid, CCS refuses the data and feeds back the corresponding information to the participant, otherwise passes the data to DAS for further processing. Finally, the participant will be notified the corresponding information such as upload success or upload fail.

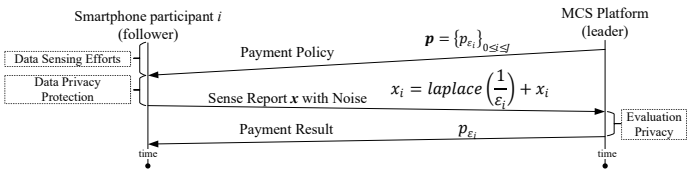


Fig. 5: Game overview. The stackelberg game is between the platform and the participants. Firstly, the platform acts as a leader to broadcast a payment list, then the participants act as followers, choosing their PPLs and uploading the sensing data. Finally, the participants get payment from the platform.

IV. Static Privacy-Preserving Data Aggregation Game

In this section, we propose to make use of static Stackelberg game to solve the conflict between leaders and followers [34], where the platform as a leader first broadcasts payment strategy to each participant who participates in sensing tasks, and then the participant as a follower spontaneously chooses a PPL to obtain the corresponding payment. For simplicity, the PPL of participant i, i.e., ε_i is quantified as $J+2$ levels, $\varepsilon_i \in \boldsymbol{\varepsilon} = \{a_{-1}, a_0, a_1, \cdots, a_J\}$. For example, if $\varepsilon_i = a_{-1}$, participant i adds too much noise to the sensing data, if $\varepsilon_i = a_0$, participant i does not provide sensing service because he is not willing to reveal his privacy, if $\varepsilon_i = J$, it fully performs sensing tasks without considering his privacy, and other conditions indicate that the participant personalizes his privacy to participate in sensing tasks. Based on the evaluation algorithm, it is assumed that the platform knows ε_i by evaluating the sensing data. Participants who provide higher PPLs require higher payment from the platform. The payment for participant i with PPL ε_i is represented by $p_{\varepsilon_i} \in \boldsymbol{p}$. Since each PPL matches one payment, the payment of platform is quantified as $N+2$ levels, and is represented by $\boldsymbol{p} = [p_{-1}, p_0, p_1, p_2, \cdots, p_N]$, where p_{-1} is the payment for PPL a_{-1} and p_0 is the payment for PPL a_0. We define $\boldsymbol{y}_n = \{y_{nj}\}_{1\leq n\leq N, 1\leq j\leq J}$, where y_{nj} is the payment for level n and PPL j and monotonically increasing. And p_{ε_i} is chosen from $\boldsymbol{y}_n$, where $\varepsilon_i \in \{a_1, \cdots, a_J\}$ and $1 \leq n \leq N$. The static privacy-preserving data aggregation game is shown in Fig. 5.

When receiving a sensing task from the platform, participant i determines whether to participate in the sensing task and his PPL. If participant i sends a sensing data with PPL ε_i, the utility u_i can be expressed as:

$$u_i(\varepsilon_i, p_{\varepsilon_i}) = p_{\varepsilon_i} - \varepsilon_i c_{\varepsilon_i} - c_i, \tag{19}$$

where c_{ε_i} is the unit cost for ε_i, c_i is the cost of participant i excluding privacy such as energy consumption.

Based on Eq. (15), the platform benefit is:

$$Benefit(\boldsymbol{\varepsilon}, \boldsymbol{p}) = \frac{R}{\frac{\sqrt{2}l}{M\sqrt{1-\eta}}\sqrt{\sum_{i=1}^{M}\frac{1}{\varepsilon_i^2}}}, \tag{20}$$

where R is a constant.

We can observe the benefit of platform from Fig. 6(a) that, as the range of PPLs increases, the overall benefit of platform increases. We can see from Fig. 6(b) that when PPLs are

within a given range, the finer the granularity of the range, the greater the overall benefit of platform. For different ranges of sensing data, the benefit of platform decreases as the data range increases, as shown in Fig. 6(c).

Thus the utility of platform is given as follows:

$$u_s(\boldsymbol{\varepsilon}, \boldsymbol{p}) = Benefit(\boldsymbol{\varepsilon}, \boldsymbol{p}) - \sum_{i=1}^{M} p_{\varepsilon_i}. \tag{21}$$

The NE of the game is denoted by $[\boldsymbol{\varepsilon}^*, \boldsymbol{p}^*]$, where $\boldsymbol{\varepsilon}^* = [\varepsilon_i^*]_{0\le i\le M}$ and $\boldsymbol{p}^* = [p_j^*]_{-1\le j\le J}$, where ε^* and p^* are given by

$$\varepsilon_i^* = \arg\max_{\varepsilon_i \in \boldsymbol{\varepsilon}} u_i\left(\varepsilon_i, \boldsymbol{p}^*\right), \quad 1 \le i \le M \tag{22}$$

$$p_j^* = \arg\max_{p_j \in \boldsymbol{p}} u\left(\boldsymbol{\varepsilon}^*, p_j\right). \quad -1 \le j \le J \tag{23}$$

We consider a special case with 2 PPLs for data aggregation, i.e., $J = 1$. In this case, participant i either sends sensing data with PPL $\varepsilon_i = a_1$ or does not attend the task with $\varepsilon_i = a_0$. When $\varepsilon_i = a_0$, from Eqs. (21) and (23), we have $p_0^* = 0$. Thus the NE of the payment strategy of platform is given by $\boldsymbol{p}^* = [0, p_1^*]$.

Theorem 5. *If* $\left(\frac{R\sqrt{1-\eta}}{\sqrt{2M}l} - c_{a_1}\right) a_1 > \max_{1\le i\le M} c_i$, *then the unique optimal NE of the static payment-PPL game G with* $J = 1$ *is as follows*

$$\begin{aligned} &\varepsilon_i^* = a_1, 1 \le i \le M \\ &\boldsymbol{p}^* = \left[0, \max_{1\le i\le M} c_{\varepsilon_i} + c_i\right]. \end{aligned} \tag{24}$$

Proof. From Eq. (19), if $p_1^* = \max_{1\le i\le M} \varepsilon_i c_{\varepsilon_i} + c_i$, we have $u_i(a_1, p^*) = p_1^* - a_1 c_{a_1} - c_i \ge 0 = p_0^* = u_i(a_0, p^*)$. Thus, if $p_1^* \ge \max_{1\le i\le M} a_1 c_{a_1} + c_i$, Eq. (22) holds for $\varepsilon_i^* = a_1$, $\forall 1 \le i \le M$. From Eq. (21), u monotonically decreases with p_1, yielding $u_s([a_1, \cdots, a_1], [0, p_1]) = \frac{R\sqrt{M(1-\eta)}a_1}{\sqrt{2}l} - Mp_1 < \frac{R\sqrt{M(1-\eta)}a_1}{\sqrt{2}l} - Mp_1^* = u_s\left([a_1, \cdots, a_1], \left[0, p_1^*\right]\right)$, $\forall p_1 > p_1^*$, and from Eq. (23), we have $p_1^* = \max_{1\le i\le M} a_1 c_{a_1} + c_i$. If $\left(\frac{R\sqrt{1-\eta}}{\sqrt{2M}l} - c_{a_1}\right) a_1 > \max_{1\le i\le M} c_i$, we have $u_s\left([a_1, \cdots, a_1], \left[0, p_1^*\right]\right) > 0$. Thus, Eq. (23) holds for Eq. (24), which is an NE of the game.

Now we prove that this NE is unique. We assume another NE is (ε', p'), where $(\boldsymbol{\varepsilon}^*, \boldsymbol{p}^*) \ne (\boldsymbol{\varepsilon}', \boldsymbol{p}')$. We assume $\varepsilon_i' = a_0$ is participant i's PPL. As shown in Eq. (19), $u_i\left(\varepsilon_i', \boldsymbol{p}'\right) = 0 < u_i\left(\varepsilon_i^*, \boldsymbol{p}^*\right)$. Thus, $(\boldsymbol{\varepsilon}^*, \boldsymbol{p}^*)$ is unique. □

Remark 1. *In this scenario, all M participants perform sensing tasks including time-invariant tasks and time-varying tasks.*

We now consider the scenario with 3 PPLs, in which participant i submits a high-PPL sensing data, i.e., $\varepsilon_i = a_1$, a low-PPL sensing data, i.e., $\varepsilon_i = a_2$, or does not attend the task, i.e., $\varepsilon_i = a_0$. As $p_0^* = 0$, the NE of payment strategy is given by $\boldsymbol{p}^* = [0, p_1^*, p_2^*]$.

Proposition 1. *Participant i submits a low-PPL sensing data with* $\varepsilon_i^* = a_2$, *if*

$$p_2^* = \max\left(a_2 c_{a_2} + c_i, p_1^* + a_2 c_{a_2} - a_1 c_{a_1}\right) \tag{25}$$

Proof. If $p_2^* > a_2 c_{a_2} + c_i$, from Eq. (22) we have

$$u_i\left(a_2, \boldsymbol{p}^*\right) = p_2^* - a_2 c_{a_2} - c_i \ge 0 = p_0^* = u_i\left(a_0, \boldsymbol{p}^*\right) \tag{26}$$

If $p_2^* > p_1^* + a_2 c_{a_2} - a_1 c_{a_1}$, we have

$$u_i\left(a_2, \boldsymbol{p}^*\right) = p_2^* - a_2 c_{a_2} - c_i \ge p_1^* - a_1 c_{a_1} - c_i = u_i\left(a_1, \boldsymbol{p}^*\right) \tag{27}$$

Combining Eqs. (26) and (27), we have $\varepsilon_i^* = a_2$, if $p_2^* \ge \max\left(a_2 c_{a_2} + c_i, p_1^* + a_2 c_{a_2} - a_1 c_{a_1}\right)$, As u_s decreases with payment, if $0 \le p_1^* \le a_1 c_{a_1} + c_i$, we have $p_2^* = a_2 c_{a_2} + c_i$; otherwise, if $p_1^* > a_1 c_{a_1} + c_i$, we have $p_2^* = p_1^* + a_2 c_{a_2} - a_1 c_{a_1}$. □

Proposition 2. *Participant i submits high-PPL sensing data with* $\varepsilon_i^* = a_1$, *if*

$$p_1^* = \max\left(a_1 c_{a_1} + c_i, p_2^* - a_2 c_{a_2} + a_1 c_{a_1}\right) \tag{28}$$

Proof. The proof is similar to that of Proposition 1. □

Proposition 3. *Participant i does not submit sensing data, i.e.,* $\varepsilon_i^* = a_0$, *if*

$$p_1^* < a_1 c_{a_1} + c_i \quad and \quad p_2^* < a_2 c_{a_2} + c_i \tag{29}$$

Proof. The proof is similar to that of Proposition 1. □

Theorem 6. *If* $\frac{R\sqrt{1-\eta}a_2}{\sqrt{2M}l} > \max_{1\le i\le M} a_2 c_{a_2} + c_i$, *we have* $u([a_2, \ldots, a_2], p^*) > 0$, *then NE of the static payment-PPL game G for the sensing tasks with* $J = 2$ *is as follows*

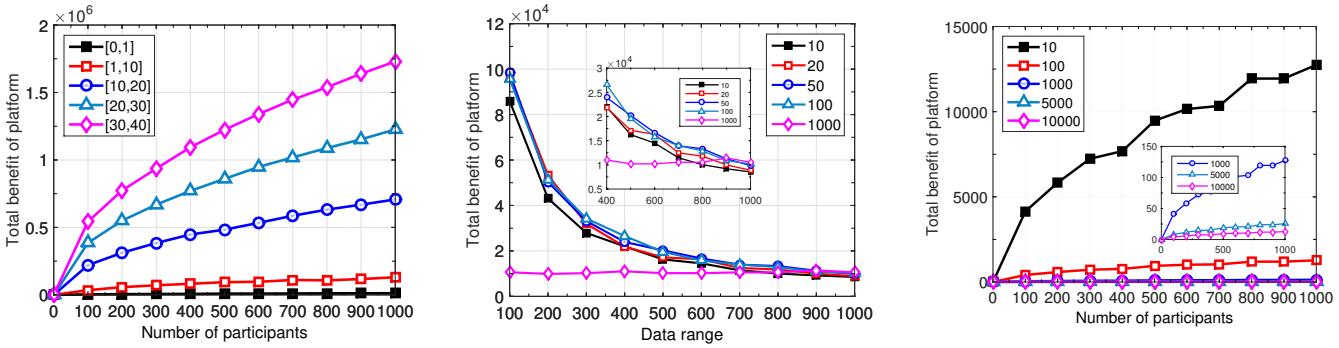
$$\begin{aligned} &\varepsilon_i^* = a_2, 1 \le i \le M \\ &\boldsymbol{p}^* = \left[0, 0, \max_{1\le i\le M} a_2 c_{a_2} + c_i\right]. \end{aligned} \tag{30}$$

Proof. If $p_2^* \ge \max\left(a_2 c_{a_2} + c_i, p_1^* + a_2 c_{a_2} - a_1 c_{a_1}\right)$, we have $\varepsilon_i^* = a_2$, $\forall 1 \le i \le M$. By Eq. (21), u_s monotonically decreases with p_2, yielding $u_s([a_2, \cdots, a_2], [0, p_1, p_2]) = \frac{R\sqrt{M(1-\eta)}a_2}{\sqrt{2}l} - Mp_2 < \frac{R\sqrt{M(1-\eta)}a_2}{\sqrt{2}l} - Mp_2^* = u_s\left([a_2, \cdots, a_2], \left[0, p_1, p_2^*\right]\right), \forall p_2 > p_2^*$. Therefore, from Eq. (23), we have $p_2^* = \max_{1\le i\le M} a_2 c_{a_2} + c_i$ and $p_1^* = 0$ in this case. If $\frac{R\sqrt{1-\eta}a_2}{\sqrt{2M}l} > \max_{1\le i\le M} a_2 c_{a_2} + c_i$, we have $u_s([a_2, \ldots, a_2], p^*) > 0$. Thus, Eq. (23) holds for Eq. (24), which is an NE of the game. □

Theorem 7. *If* $\frac{R\sqrt{1-\eta}a_1}{\sqrt{2M}l} > \max_{1\le i\le M} a_2 c_{a_2} + c_i$, *then the NE of the static payment-PPL game G with* $\boldsymbol{\varepsilon} = [a_{-1}, a_0, a_1]$ *is as follows*

$$\begin{aligned} &\varepsilon_i^* = a_1, 1 \le i \le M \\ &p^* = \left[0, 0, \max_{1\le i\le M} a_2 c_{a_2} + c_i\right]. \end{aligned} \tag{31}$$

Proof. If $p_1^* > a_2 c_{a_2} + c_i$, we have $u_i(a_1, p^*) = p_1^* - a_1 c_{a_1} - c_i \ge 0 = p_0^* = u_i(a_0, p^*)$ and $u_i(a_1, p^*) = p_1^* - a_1 c_{a_1} - c_i \ge p_{-1}^* - a_{-1} c_{a_{-1}} - c_i = u_i(a_{-1}, p^*)$. Thus, if $p_1^* > \max_{1\le i\le M} a_2 c_{a_2}$, Definition 3 holds for $\varepsilon_i^* = a_1, \forall 1 \le i \le M$. From Eq. (21), u_s monotonically decreases with p_1,



(a) PPL ranges are 0-1, 1-10, 10-20, 20-30, 30-40, the number of intervals is 10, and confidence level $\eta = 0.95$.

(b) The intervals of PPL range are 10, 20, 50, 100, 1000, the number of participants is M=400, confidence level is $\eta = 0.95$.

(c) Data ranges are 10, 100, 1000, 5000, 10000, the number of participants is M=100, and confidence level is $\eta = 0.95$.

Fig. 6: Performance of the static payment-privacy game.

yielding $u_s\left([a_1,\cdots,a_1],[0,0,p_1]\right) = \frac{R\sqrt{M(1-\eta)a_1}}{\sqrt{2}l} - Mp_1 < \frac{R\sqrt{M(1-\eta)a_1}}{\sqrt{2}l} - Mp_1^* = u_s\left([a_1,\cdots,a_1],\left[0,0,p_1^*\right]\right), \forall p_1 > p_1^*$. Therefore, from Eq. (23), we have $p_1^* = \max_{1\le i\le M} a_2 c_{a_2} + c_i$. If $\frac{R\sqrt{1-\eta}a_1}{\sqrt{2M}l} > \max_{1\le i\le M} a_2 c_{a_2} + c_i$, we have $u_s\left([a_1,\cdots,a_1]\right) > 0$. Thus, Eq. (22) holds for Eq. (31), which is an NE of the game. □

In summary, we have discussed the scenarios with $J = 1$ and $J = 2$, i.e., participant i selects to send data with low PPL with $\varepsilon_i = a_2$, high PPL with $\varepsilon_i = a_1$, does not join in the task with $\varepsilon_i = a_0$, or submit over-noise data with $\varepsilon_i = a_{-1}$. The NEs of the static payment-PPL game in these scenarios show the impacts of privacy cost and other cost.

V. Dynamic Learning In Privacy-Preserving Data Aggregation Game

In this section, the interactions between platform and M participants can be formulated as a dynamic game. On the platform side, on the one hand, a higher payment for accurate sensing data will reduce the utility of platform, but will stimulate more participants to participate in sensing tasks in the future. On the other hand, over-payment may cause some illegal participants to join, thus reducing the long-term utility of platform. On the participant side, participants usually choose PPLs and upload sensing data according to the payment history of platform. Long-term low payment will inhibit participants' participation. In view of the inability to timely and accurately evaluate system parameters between the two sides of the system, we apply reinforcement learning which is a trial error method and does not need to know the specific parameters of the overall system model, such as Q-learning, (DQN) and so on, to obtain the optimal strategies of both sides.

A. Payment Based on Q-Learning

A finite Markov Decision Process (MDP) can formulate the payment decision process of platform. Therefore, the platform can dynamically adjust the payment strategy. In each time slot, the state of platform is composed of each participant's PPL. According to the current state, the platform selects the corresponding payment strategy by using ξ-greedy strategy. We assume that the privacy evaluation algorithm is valid for all the sensing data, as shown in Section III. In time slot t, the platform state $\boldsymbol{s}^{(t)}$ is composed of the number of different PPLs of M participants. The number of sensing data with PPLs received by the platform in time slot t is:

$$\hat{N}_j^{(t)} = \sum_{i=1}^{M} I\left(\beta(x)\right), 0 \le j \le J \tag{32}$$

Taking into account the different natures of two tasks, for the type of time-invariant tasks τ_1, participant i's PPL $\varepsilon_i = \sum_{j=1}^{\hat{N}} \varepsilon_{j,i}$ where $\varepsilon_{j,i}$ means the PPL of participant i's j-th sensing data, and for the type of time-varying tasks τ_2, participant i's PPL $\varepsilon_i = \max \varepsilon_{j,i}^{(t)}$ where $\varepsilon_{j,i}$ means participant i's PPL at location j. Under these conditions, the benefit of platform can be written as:

$$Benefit(\boldsymbol{\varepsilon},\boldsymbol{p}) = \begin{cases} \frac{R}{\frac{\sqrt{2}l}{M\sqrt{1-\delta}}\sqrt{\sum_{i=1}^{M}\frac{1}{\left(\sum \varepsilon_{j,i}\right)^2}}}, \tau_1 \in \boldsymbol{\tau}, 1 \le i \le M \\ \frac{R}{\frac{\sqrt{2}l}{M\sqrt{1-\delta}}\sqrt{\sum_{i=1}^{M}\frac{1}{\left(\max \varepsilon_{j,i}^{(t)}\right)^2}}}, \tau_2 \in \boldsymbol{\tau}, 1 \le i \le M \end{cases} \tag{33}$$

Note that for the tasks of the above two types, when participant i submits sensing data in each time slot, the platform provides payment to the participant.

The platform payment strategy is based on Q-learning, where $Q(\boldsymbol{s},p)$ is set as the Q function of the platform under the action p of state $\boldsymbol{s}$. According to ξ-greedy strategy, with $0 < \xi \le 1$, the platform selects the action with the highest Q value with the probability of $1-\xi$, and randomly selects the other actions with probability ξ. Payment strategy $p^{(t)}$ is expressed by the following formula:

$$\Pr\left(\boldsymbol{p}^{(t)} = \boldsymbol{p}^*\right) = \begin{cases} 1-\xi, \boldsymbol{p}^* = \arg\max_{p_j^{(t)} \in \boldsymbol{p}} Q(\boldsymbol{s},\boldsymbol{p}) \\ \frac{\xi}{J^j - 1}, \text{ otherwise} \end{cases} \tag{34}$$

However, we consider to make the largest cumulative reward. In the initial stage of learning, increasing the number of

"exploration" can understand the environment better so that greater reward can be achieved, while in the later stage, in order to keep previous reward, we need to increase the number of "exploitation" so as to better fit our model. In this way, we can also achieve a better balance between "exploration" and "exploitation", so ξ varies according to the following equation:

$$\xi = \xi_{start} - \frac{(\eta_{start} - \eta_{end}) * learning\_step}{annealing\_step} \tag{35}$$

where ξ_{start}, ξ_{end} and $annealing\_step$ are constants, and $learning\_step$ changes with the number of iterations. When ξ_{start} reduces to ξ_{end}, ξ does not change.

The platform observes $\varepsilon^{(t)}$ of sensing data calculated by Eq. (32), and obtains the next platform state $\boldsymbol{s}^{(t+1)}$. The value of $\boldsymbol{s}$ is represented by the highest $V(\boldsymbol{s})$ in the state. The platform updates its Q function by:

$$\begin{aligned} Q(\boldsymbol{s}^{(t)}, \boldsymbol{p}^{(t)}) \leftarrow (1-\alpha) Q(\boldsymbol{s}^{(t), \boldsymbol{p}^{(t)}}) \\ + \alpha \left(u(\boldsymbol{\varepsilon}^{(t)}, \boldsymbol{p}^{(t)}) + \delta V(\boldsymbol{s}^{(t+1)}) \right), \end{aligned} \tag{36}$$

$$V\left(\boldsymbol{s}^{(t)}\right) \leftarrow \max_{p^{(t)} \in \boldsymbol{p}^J} Q\left(\boldsymbol{s}^{(t)}, \boldsymbol{p}^{(t)}\right), \tag{37}$$

where $\delta \in (0, 1]$ represents the weight of future payment that exceeds the current payment. Fig. 7 shows the state transition of the platform. And Algorithm 1 presents this process.

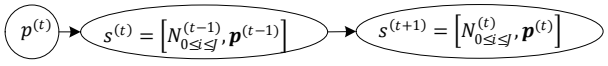


Fig. 7: State transition of Q-learning for platform payment.

Algorithm 1 Payment Based on Q-Learning (PQ)

Initialize α, δ, ξ_{start}, ξ_{end}, $annealing\_step$, $learning\_step$, $\boldsymbol{s}^0$=$\boldsymbol{0}$, $Q(\boldsymbol{s}, \boldsymbol{p})$=0, and $V(\boldsymbol{s})$=0, $\forall \boldsymbol{p}, \boldsymbol{s}$;
for $t = 1, 2, 3, \ldots$ **do**
  Choose $\boldsymbol{p}^{(t)}$ via Eq. (34);
  Update ξ via Eq. (35);
  for $i = 1, 2, \ldots, M$ **do**
    Evaluate x_i via Definition 3;
  end for
  Receive $u_s(\boldsymbol{s}, \boldsymbol{p})$ via Eq. (21);
  Observe $\boldsymbol{s}^{(t+1)}$;
  Update $Q\left(\boldsymbol{s}^{(t)}, \boldsymbol{p}^{(t)}\right)$ via Eq. (36);
  Update $V\left(\boldsymbol{s}^{(t)}\right)$ via Eq. (37);
end for

When storing the state-action values of platform, we use Q-table which is a two-dimensional matrix. In this way, it is necessary to maintain Q-table at all time in order to obtain the optimal action-state pairs.

Lemma 1. *As the number of participants increases, the size of Q-table increases exponentially.*

Proof. See Appendix D. □

B. PPL Based on Q-Learning

An MDP can also formulate the participants' PPLs decision process. Therefore, the participants can also use Q-learning to execute decision. For participant i, the state observed by the participant in time slot t is composed of his PPL and platform payment in the previous state, i.e., $\mathbf{s}_i^{(t)} = [\varepsilon_i^{(t-1)}, \boldsymbol{p}_i^{(t-1)}] \in \boldsymbol{s}_i$, where $\boldsymbol{s}_i$ is the state space of participant i. Fig.8 shows the state transition for participant i's PPL.

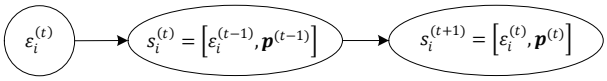


Fig. 8: State transition of Q-learning based participant i's PPL.

Let $Q_i(\boldsymbol{s}_i, \varepsilon_i)$ denote the value function of participant i, and $V_i(\boldsymbol{s}_i)$ be the state-action function. The Q-function is updated by:

$$\begin{aligned} Q_i(\boldsymbol{s}_i^{(t)}, \varepsilon_i^{(t)}) \leftarrow (1-\alpha) Q_i(\boldsymbol{s}_i^{(t)}, \varepsilon_i^{(t)}) \\ + \alpha \left(u_i(\boldsymbol{s}_i^{(t)}, \varepsilon_i^{(t)}) + \delta V(\boldsymbol{s}_i^{(t+1)}) \right), \end{aligned} \tag{38}$$

$$V\left(\boldsymbol{s}_i^{(t+1)}\right) \leftarrow \max_{\varepsilon_i} Q_i\left(\boldsymbol{s}_i^{(t)}, \varepsilon_i^{(t)}\right). \tag{39}$$

Participant i uses ξ-greedy policy to choose his PPL as

$$\Pr\left(\varepsilon_i^{(t)} = \varepsilon^*\right) = \begin{cases} 1-\xi, \varepsilon^* = \arg\max_{p_j \in \boldsymbol{P}} Q_i\left(s_i^{(t)}, \varepsilon_i^{(t)}\right) \\ \frac{\xi}{J^j - 1}, \text{otherwise} \end{cases} \tag{40}$$

And ξ changes as well as via Eq. (35), participant i's PPL strategy with Q-learning is written by Algorithm 2.

Algorithm 2 PPL Based on Q-Learning (PPQ)

Initialize α, δ, η_{start}, η_{end}, $annealing\_step$, $learning\_step$, $\boldsymbol{s}_i^0$=$\boldsymbol{0}$, $Q_i(\boldsymbol{s}_i, \varepsilon_i)$=0, and $V_i(\boldsymbol{s}_i)$=0, $\forall \boldsymbol{s}_i, \varepsilon_i$;
for $t = 1, 2, 3, \ldots$ **do**
  Choose $\varepsilon_i^{(t)}$ via Eq. (40);
  Update ξ via Eq. (35);
  Upload data x_i and PPL ε_i to platform;
  Evaluate x_i via Definition 3;
  Receive $u_i(\varepsilon_i, p_{\varepsilon_i})$ via Eq. (19);
  Observe $\boldsymbol{s}_i^{(t+1)}$;
  Update $Q_i\left(\boldsymbol{s}_i^{(t)}, \varepsilon_i^{(t)}\right)$ via Eq. (38);
  Update $V_i\left(\boldsymbol{s}_i^{(t)}\right)$ via Eq. (39);
end for

C. Payment Based on DQN

According to Lemma 1, when the number of participants increases to a certain level, it is difficult to simply rely on Q-table because the size of Q-table increases exponentially, so we adopt DQN to reduce the size of Q-table and obtain more comprehensive global information. More specifically, DQN

uses a convolution neural network (CNN) to approximate Q-function, i.e., $Q(\boldsymbol{s},\boldsymbol{p};\theta) \approx Q(\boldsymbol{s},\boldsymbol{p})$. And the Q-function is updated as follows:

$$Q(\boldsymbol{s},\boldsymbol{p}) = \mathbb{E}_{\boldsymbol{s}'\in\boldsymbol{s}}\left[u' + \gamma \max_{\boldsymbol{p}'\in\boldsymbol{p}} Q\left(\boldsymbol{s}',\boldsymbol{p}'\right)\right], \tag{41}$$

where γ is the discount factor.

We use $\varphi^{(t)}$ to denote the state sequence in time slot t which includes the recent W+1 states and W payment actions, i.e., $\varphi^{(t)} = \left\{\boldsymbol{s}^{(t-W)}, \boldsymbol{p}^{(t-W)}, \cdots, \boldsymbol{s}^{(t-1)}, \boldsymbol{p}^{(t-1)}, \boldsymbol{s}^{(t)}\right\}$. The platform experience in time slot t is denoted by $\boldsymbol{e}^{(t)} = \left\{\varphi^{(t)}, \boldsymbol{p}^{(t)}, u^{(t)}, \varphi^{(t+1)}\right\}$. The experiences are stored in memory pool. In our dynamic payment-PPL game, the memory pool only stores the latest related experiences to save memory space, i.e., $\mathcal{D} = \left\{\boldsymbol{e}^{(d)}\right\}_{1\le d\le D}$.

In our proposed DQN-base payment-PPL game system as shown in Fig. 9, our proposed CNN includes 2 convolution (Conv) layers, 2 Batch Normalization (BN) layers, and 2 full connected (FC) layers. Both of the Conv layers use rectified linear units (ReLUs) as the activation functions. The parameters of the layers are summarized in TABLE II. The state sequence $\varphi^{(t)}$ is input to the CNN with a 12×10 matrix. In time slot t, the platform obtains $\theta^{(t)}$ by minimizing the mean-squared error with learning rate ξ, and uses the loss function as follows:

$$L\left(\theta^{(t)}\right) = \mathbb{E}_{\varphi,\boldsymbol{p},u_s,\varphi'}\Big[\Big(u_s^{(t)} + \gamma \max_{\boldsymbol{p}'\in\boldsymbol{p}^N} Q(\varphi',\boldsymbol{p}';\theta^{(t-1)}) - Q(\varphi,\boldsymbol{p};\theta^{(t)})\Big)^2\Big]. \tag{42}$$

Thus

$$\nabla_{\theta^{(t)}} L\left(\theta^{(t)}\right) = -\mathbb{E}_{\varphi,\boldsymbol{p},u_s,\varphi'}\Big[\Big(u_s^{(t)} + \gamma \max_{\boldsymbol{p}'\in\boldsymbol{p}^N} Q(\boldsymbol{s}^{t+1},\boldsymbol{p}';\theta^{(t-1)}) - Q(\boldsymbol{s},\boldsymbol{p};\theta^{(t)})\Big)\nabla_{\theta^{(t)}} Q(\boldsymbol{s},\boldsymbol{p};\theta^{(t)})\Big]. \tag{43}$$

The platform repeats stochastic gradient descent (SGD) algorithm in each time slot to update the CNN parameters by randomly selecting an experience from memory pool. The payment decision algorithm based on DQN is summarized in Algorithm 3. And our proposed privacy-preserving data aggregation Game is presented in Algorithm 4.

Algorithm 3 Payment Based on DQN (PDQN)

Initialize α, γ, $\boldsymbol{p}$, $D = 36$, $W = 10$, $\mathcal{D} = \emptyset$, ξ_{start}, ξ_{end}, $annealing\_step$, $\hat{N}_i = 0$;
Initialize DQN with random weight θ and structure with Table II;
for $t = 1, 2, 3, \ldots$ **do**
  $\boldsymbol{s}^{(t)} = [\hat{N}_{0\le i\le J}^{(t-1)}, \boldsymbol{p}^{(t-1)}]$;
  if $t \le W$ **then**
    Select $\boldsymbol{p}^{(t)} \in \boldsymbol{p}_{0\le\boldsymbol{p}^{(t)}\le N}$ at random;
  else
    Obtain $\varphi^{(t)} = \left\{\boldsymbol{s}^{(t-W)}, \boldsymbol{p}^{(t-W)}, \cdots, \boldsymbol{s}^{(t-1)}, \boldsymbol{p}^{(t-1)}, \boldsymbol{s}^{(t)}\right\}$ with weight $\theta^{(t)}$;
    Obtain $Q(\boldsymbol{p})$;
    Select $\boldsymbol{p}^{(t)}$ via the ξ-greedy algorithm;
    Update ξ via Eq. (35);
  end if
  Calculate payment list $\boldsymbol{p}^{(t)}$ with $p^{(t)}$;
  Broadcast the recruit message with $\boldsymbol{p}^{(t)}$;
  while Receiving sensing data and PPL from participant i **do**
    Evaluate x_i via Definition 3 and pay participant i with $\boldsymbol{p}^{(t)}(\varepsilon_i^{(t)})$;
  end while
  Obtain $u_s^{(t)}(\boldsymbol{s},\boldsymbol{p}^{(t-1)})$;
  for $0 \le i \le J$ **do**
    Calculate $\hat{N}_i^{(t)}$ via Eq. (32);
  end for
  $\mathcal{D} \leftarrow \mathcal{D} \bigcup \boldsymbol{e}^{(t)}$;
  for $d = 1, 2, \cdots, D$ **do**
    Randomly select $\boldsymbol{e}^{(d)}$ from $\mathcal{D}$;
    $Q^{(d)} \leftarrow u^{(d)} + \gamma \max_{\boldsymbol{p}'} Q\left(\boldsymbol{s}^{(d+1)},\boldsymbol{p}';\theta^{(t)}\right)$;
  end for
  Calculate $\theta^{(t)}$ via Eq. (42);
  Update the CNN weight with $\theta^{(t)}$ using SGD algorithm;
end for

VI. Performance Evaluation

Simulations have been carried out to evaluate the performance of our privacy-preserving data aggregation game in crowdsensing.

A. Parameter Settings

System parameter setting: We set the number of participants M to $[60, 120, 180, 240, 300]$, $J = 10$, i.e., $\varepsilon_i \in \boldsymbol{\varepsilon} = \{-1, 0, 1, \cdots, 10\}$, and the corresponding participants' cost is $[1.0, 0, 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1]$, $N = 24$, $y_{11} = 1.0$, $y_{N1} = 5.6$, confidence level $\delta = 0.95$, R=1e5.

Algorithm parameter setting: The learning rate of Q-learning is 0.2. The learning rate of CNN is derived when the loss value is minimal after convergence, as shown in Fig. 10. We can find that the learning rate is between 0.2 and 0.6, in our simulations, we set the learning rate to 0.3, and ξ_{start}=0.3, ξ_{end} = 0.1, $anneal\_step$=1000. The unit cost of privacy is randomly selected from 1 to 2. Each simulation is carried out 500 times, and the average result is obtained.

B. Participants' Performance

We compare DQN where the platform uses DQN algorithm and the participants use Q-learning algorithm due to insufficient computation resource of smart devices, Q-learning where both platform and participants use Q-learning algorithm, and Random where the platform pays to the participants randomly without considering their PPLs.

As shown in Fig. 11(a), we can see that the average utility of participants using DQN is 3.17% more than that of Q-learning, and 24.33% more than that of Random. We can also observe the DQN converges much quickly than Q-learning. It is because DQN applies CNN to map state-action pairs in order to accelerate learning speed.

Fig. 11(b) uses boxplot to represent the PPL distribution of participants, we can see that the PPLs using both DQN and

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/JIOT.2019.2957400, IEEE Internet of Things Journal

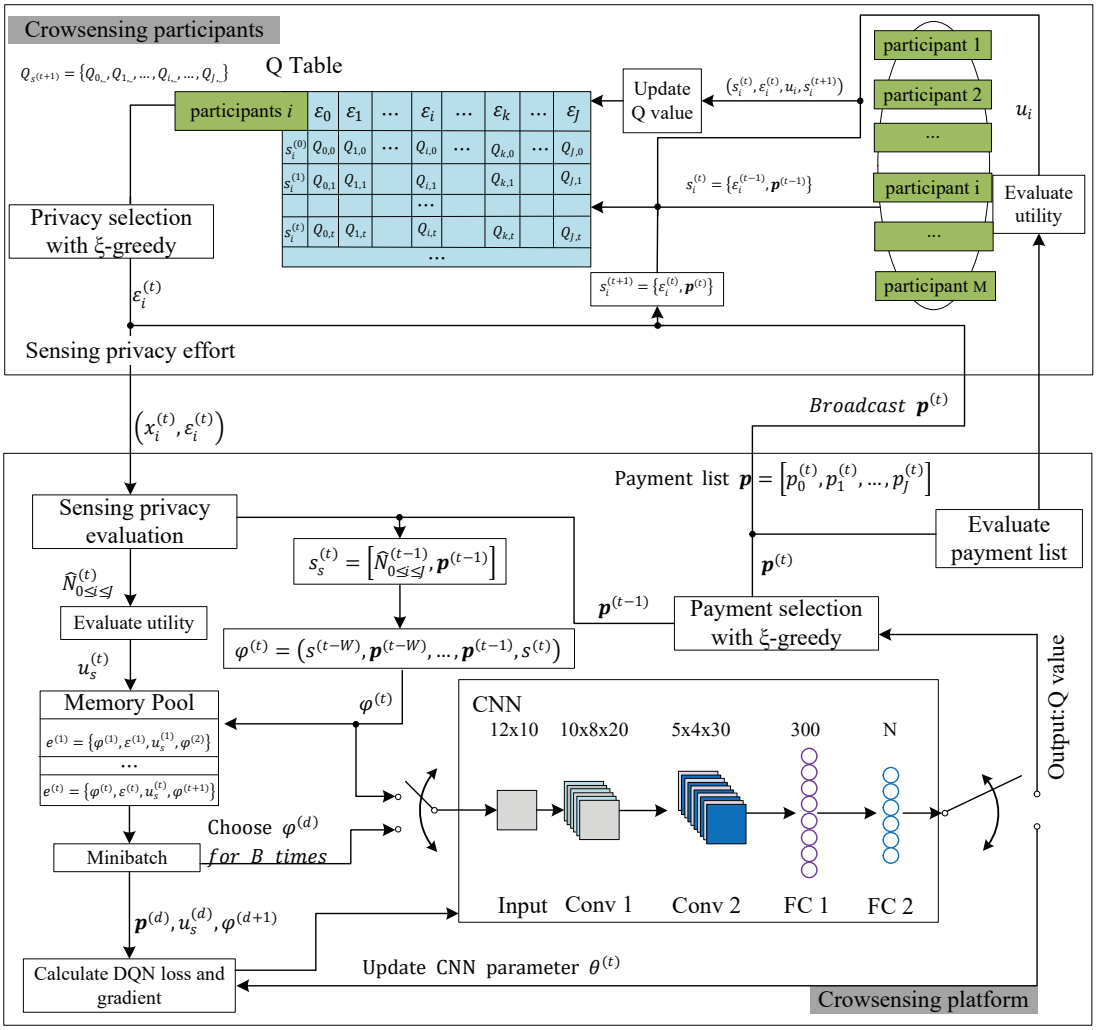


Fig. 9: Illustration of DQN-based Payment-PPL game.

TABLE II: CNN parameters in Algorithm 3.

Layer	Conv 1	BN 1	Conv 2	BN 2	FC 1	FC 2
Input	12 × 10	10 × 8 × 20	10 × 8 × 20	5 × 4 × 30	600	300
Filter size	5 × 5	/	3 × 3	/	/	/
Stride	2	/	/	1	/	/
Padding	1	/	1	/	/	/
Number of filters	20	/	30	/	300	N
Activation	ReLU	/	ReLU	/	ReLU	/
Output	10 × 8 × 20	10 × 8 × 20	5 × 4 × 30	5 × 4 × 30	300	N

Q-learning are higher than those using Random, due to the fact that both DQN and Q-learning adopt learning strategies to make decisions, leading to that participants are more inclined to decrease their PPLs in order to obtain more payment. We can also observe that with the increase of the number of participants, the range using DQN is narrower than that using Q-learning. It is because as the number of participants increases, the state and action set is large, the learning speed of Q-learning reduces, resulting in inaccurate PPL estimation compared with DQN.

Fig. 11(c) shows that with the increase of the number of participants, the average utility of participants using Q-learning and Random keeps stable, the stable utility of Random is because the payment is randomly selected based on the participants' PPLs, while the stable utility of Q-learning is because the payment is chosen based on the participants' PPLs, and the number of participants does not affects the payment selection. We observe that the average utility of participants using DQN increases as the number of participants increases, it is because as the number of participants increases, the size of state-action set increases, DQN uses CNN to accelerate learning speed and improves payment matching accuracy, resulting in higher average utility. More specifically, the average utility of participants using DQN increases 1.37% than that of Q-learning and 21.46% than that of Random.

Algorithm 4 Privacy-Preserving Data Aggregation Game in Crowdsensing

Input: Task type τ;
Output: Sensing data x;
1: Recruit users to participate in sensing tasks;
2: Get $\boldsymbol{p}$ via Algorithm 1 or Algorithm 3;
3: **if** $\tau == \tau_1$ **then**
4: **for** $i = 1$ to V **do**
5: Participant i chooses PPL ε_i via Algorithm 2;
6: Obtain the number of valid data via Eq. (16);
7: **end for**
8: Obtain participant i's sensing data x_i via Eq. (17);
9: return x_i;
10: **else**
11: At time slot $t \in T$;
12: **for** $i = 1$ to U **do**
13: Participant i chooses PPL ε_i via Algorithm 2;
14: Obtain the number of valid data via Eq. (16);
15: **end for**
16: Obtain time i's sensing data x_t via Eq. (18);
17: return x_t;
18: **end if**

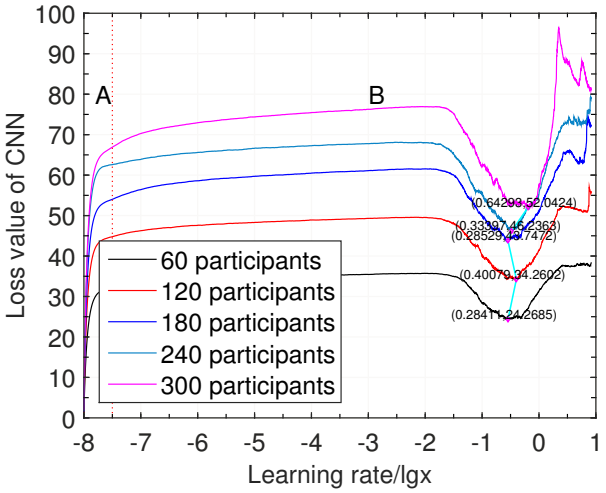


Fig. 10: The CNN loss value and learning rate.

C. Platform Performance

As shown in Fig. 12(a), the average aggregated error using DQN reduces by 4.31% compared with Q-learning and 12.88% compared with Random. We also observe that the average aggregated error using DQN converges faster than using Q-learning due to the employment of CNN to accelerate learning speed. Similarly, Fig 12(b) depicts that, the average utility of platform is the largest when DQN is used, with 4.58% more than Q-learning and 15.39% more than Random. And the convergence of DQN is faster than that of Q-learning.

Fig. 12(c) shows that as the number of participants increases, the average aggregated error decreases according to Eq. (15). More specifically, the average aggregated error using DQN decreases by 2.67% compared with that using Q-learning, and the average aggregated error using Q-learning decreases by 12.40% compared with that using Random.

We observe in Fig. 12(d) that with the increase of the number of participants, the average utility of platform increases. It is because the aggregated error decreases as the number of participants increases, resulting in the benefit of platform increases, while the average payment to the participants does not change much as shown in Fig. 11(c). More specifically, the average utility of platform using DQN is 2.70% more than that using Q-learning and 12.84% more than that using Random.

Fig. 12(e) shows the change of loss value in the CNN learning process. It can be seen that the loss value increases with the increase of the number of participants, it is because the utility of platform increases as the number of participants, according to Eq. (42), the loss value increases. However, with the increase of the number of iterations, the loss value is stable, leading to stable payment.

Fig. 12(f) shows the payment distribution of platform. Being consistent with PPL distribution of participants in Fig. 11(b), with the increase of the number of participants, the range using DQN is narrower than that using Q-learning. It is because as the number of participants increases, the state and action set is large, DQN improves the learning speed, resulting in more accurate payment estimation compared with Q-learning.

VII. Conclusion

In this paper, we have formulated a payment-PPL game and derived the NE of the static game, where the platform chooses a specified payment according to each participant's PPL. In the dynamic payment-PPL game, a Q-learning algorithm is used to derive the payment-PPL strategy without knowing system model. We then employ a deep reinforcement learning technique, i.e., DQN to accelerate learning speed especially when the size of state-action pairs is large. We have carried out extensive simulations to demonstrate that compared with Q-learning algorithm, our proposed DQN based algorithm has greater utilities of both platform and participants and less data aggregation error.

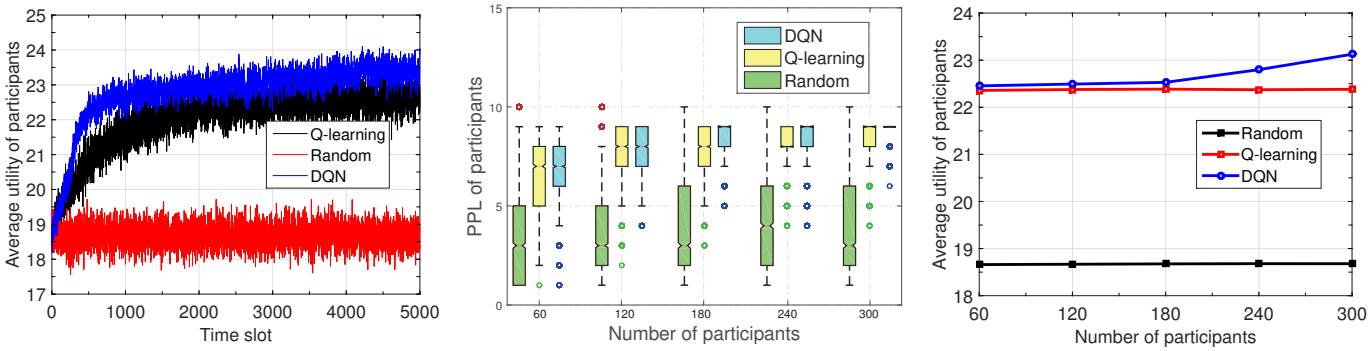
Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61972049 and No. 61602038, and Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, NJUPT under Grand No. BDSIP1908.

Appendix A
Proof of Theorem 1

Proof. Given two adjacent data set D_1 and D_2 with only one different element, let $\boldsymbol{x}_i \in D_1$ and $\boldsymbol{x}'_i \in D_2$. And, known by Eq. (1)

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/JIOT.2019.2957400, IEEE Internet of Things Journal



(a) The average utility of participants. $M = 300$

(b) The distribution of participants' actions.

(c) The average utility of participants with different numbers of participants.

Fig. 11: Performance of participants.

$$\begin{aligned}\frac{\Pr(\boldsymbol{x}_i = \boldsymbol{x})}{\Pr(\boldsymbol{x}_i' = \boldsymbol{x})} &= \frac{\Pr([x_{1,i}, x_{2,i}, \ldots, x_{V,i}] = \boldsymbol{x})}{\Pr([x_{1,i}', x_{2,i}', \ldots, x_{V,i}'] = \boldsymbol{x})} \\ &= \prod_{j=1}^{V} \frac{\Pr(x_{j,i})}{\Pr(x_{j,i}')} \\ &= \prod_{j=1}^{V} e^{\frac{|x_{j,i}'| - |x_{j,i}|}{b}} \\ &\overset{a}{\leq} \prod_{j=1}^{V} e^{\frac{|x_{j,i}' - x_{j,i}|}{b}},\end{aligned} \tag{44}$$

where the elements in $\boldsymbol{x}$ are all the same, inequality a holds because of triangle-inequality. Further, we can get

$$\frac{\Pr(\boldsymbol{x}_i = \boldsymbol{x})}{\Pr(\boldsymbol{x}_i' = \boldsymbol{x})} \overset{c}{\leq} e^{\sum_{i=1}^{V} \frac{1}{b}} = e^{\sum_{i=1}^{V} \varepsilon_i}, \tag{45}$$

where inequality c holds because of Definition 1. According to Definition 2, the time-invariant task satisfies $\sum_{i=1}^{V} \varepsilon_i$-differential privacy. Therefore, Theorem 1 is proved. □

Appendix B
Proof of Theorem 2

Proof. Given two data sets D_1 and D_2 where k-th element is different, $\boldsymbol{x}_i^{(t)} \in D_1$ and $\boldsymbol{x}_i'^{(t)} \in D_2$. Then, we have

$$\begin{aligned}\frac{\Pr(\boldsymbol{x}_i^{(t)} = \boldsymbol{x})}{\Pr(\boldsymbol{x}_i^{(t)\prime} = \boldsymbol{x})} &= \frac{\Pr(x_{j,i}^{(t)} | x_{j,i}^{(t)} \in \boldsymbol{x}_i^{(t)})}{\Pr(x_{j,i}^{(t)\prime} | (x_{j,i}^{(t)\prime} \in \boldsymbol{x}_i^{(t)\prime})} \\ &= \frac{\prod_{j=1}^{U} \Pr(x_{j,i}^{(t)})}{\prod_{j=1}^{U} \Pr(x_{j,i}^{(t)\prime})} \\ &= \frac{\prod_{j=1}^{U} \Pr(x_{j,i}^{(t)})}{\Pr(x_{k,i}^{(t)\prime}) \prod_{j\neq k}^{U} \Pr(x_{j,i}^{(t)\prime})} \\ &= \frac{\Pr(x_{k,i}^{(t)})}{\Pr(x_{k,i}^{(t)\prime})} \leq e^{\frac{|x_{k,i}^{(t)\prime} - x_{k,i}^{(t)}|}{b}} \overset{d}{\leq} e^{\frac{b}{l}}.\end{aligned} \tag{46}$$

Inequality d holds because of Definition 1 and triangle-inequality. Due to $1 \leq k \leq U$,

$$\begin{aligned}\frac{\Pr(\boldsymbol{x}_i^{(t)} = \boldsymbol{x})}{\Pr(\boldsymbol{x}_i^{(t)\prime} = \boldsymbol{x})} &\leq e^{\min\left\{\varepsilon_i^{(t)} | \varepsilon_i^{(t)} \geq \max\limits_{1\leq k\leq U} \varepsilon_k^{(t)}\right\}} \\ &\leq e^{\max\limits_{1\leq i\leq U} \varepsilon_i^{(t)}}.\end{aligned} \tag{47}$$

According to Definition 2, the time-invariant task satisfies $\sum_{i=1}^{V} \varepsilon_i$-differential privacy. Therefore, Theorem 2 is proved. □

Appendix C
Proof of Theorem 3

Proof. Given two data sets D_1 and D_2 where m-th element is different, $\boldsymbol{x}_i^{(t)} \in D_1$ and $\boldsymbol{x}_i^{(t)\prime} \in D_2$. Then, we have

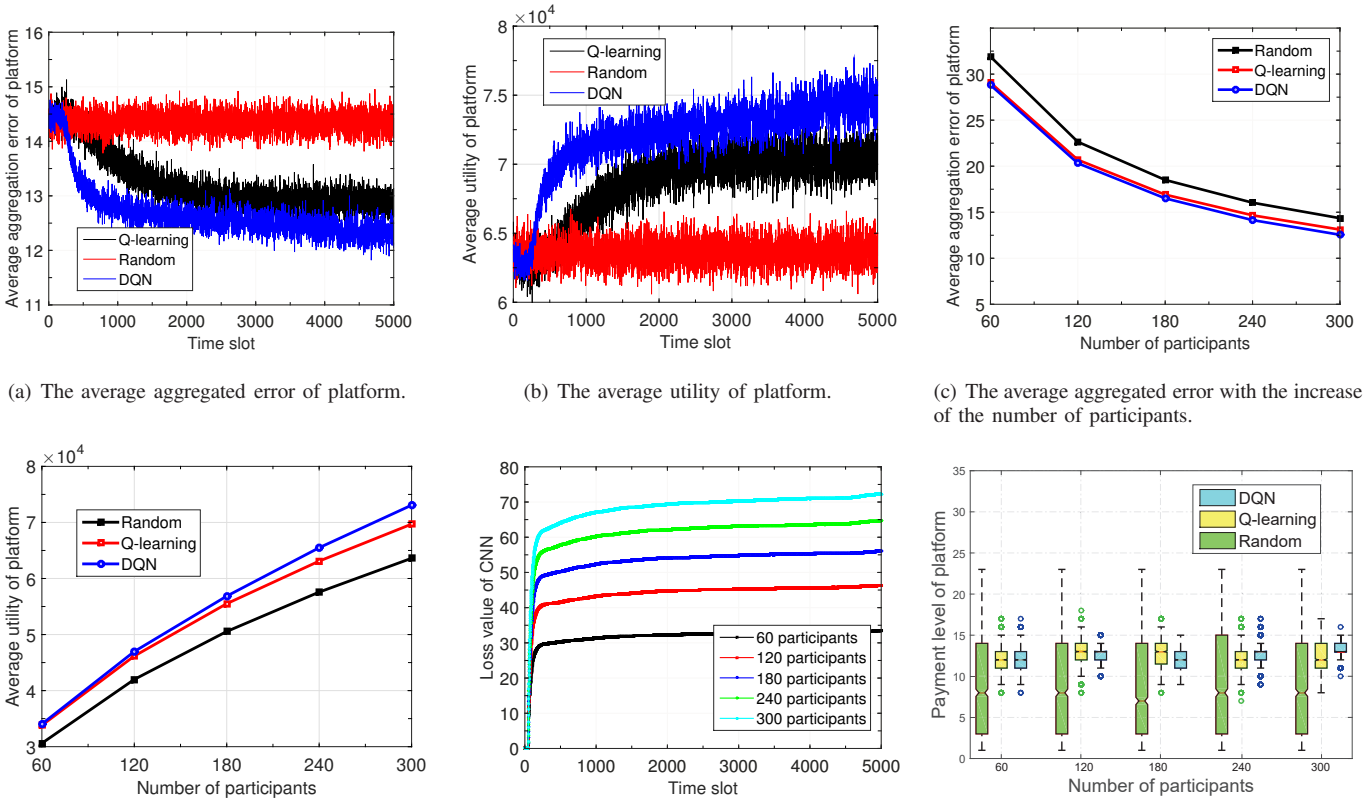
$$\frac{\Pr(\boldsymbol{x}_i^{(t)} = \boldsymbol{x})}{\Pr(\boldsymbol{x}_i^{(t)\prime} = \boldsymbol{x})} \leq e^{\max\limits_{1\leq k\leq U} \varepsilon_k^{(t)}}. \tag{48}$$

Furthermore, we consider a sensing task in period T, given two dat sets D_3 and D_4 which j-th element is different, $\boldsymbol{x}_i^{(T)} \in D_3$ and $\boldsymbol{x}_i^{(T)\prime} \in D_4$. Thus, we have

$$\begin{aligned}\frac{\Pr(\boldsymbol{x}_i^{(T)} = \boldsymbol{x})}{\Pr(\boldsymbol{x}_i^{(T)\prime} = \boldsymbol{x})} &= \frac{\Pr(x_i^{(t)} | x_i^{(t)} \in \boldsymbol{x}_i^{(T)})}{\Pr(x_i^{(t)\prime} | x_i^{(t)\prime} \in \boldsymbol{x}_i^{(T)\prime})} \\ &= \frac{\prod_{t=1}^{T} \Pr(x_i^{(t)})}{\prod_{t=1}^{N} \Pr(x_i^{(t)\prime})} \\ &= \frac{\prod_{t=1}^{T} \Pr(x_i^{(t)})}{\Pr(x_i^{(j)\prime}) \prod_{t\neq j}^{N} \Pr(x_i^{(t)\prime})} \\ &= \frac{\Pr(x_i^{(j)})}{\Pr(x_i^{(j)\prime})} \leq e^{\frac{|x_i^{(j)\prime} - x_i^{(j)}|}{b}} \overset{g}{\leq} e^{\frac{1}{b}}.\end{aligned} \tag{49}$$

Inequality g holds because of Definition 1 and triangle-inequality. Due to $1 \leq n \leq T$,

$$\begin{aligned}\frac{\Pr(\boldsymbol{x}_i^{(T)} = \boldsymbol{x})}{\Pr(\boldsymbol{x}_i^{(T)\prime} = \boldsymbol{x})} &\leq e^{\min\left\{\varepsilon_i^{(n)} | \varepsilon_i^{(n)} \geq \max\limits_{1\leq t\leq T} \varepsilon_i^{(t)}\right\}} \\ &\leq e^{\max\limits_{1\leq t\leq T} \varepsilon_i^{(t)}}.\end{aligned} \tag{50}$$



(a) The average aggregated error of platform.

(b) The average utility of platform.

(c) The average aggregated error with the increase of the number of participants.

(d) The average utility of platform with the increase of the number of participants.

(e) CNN loss value with the increase of the number of participants.

(f) The payment distribution of platform.

Fig. 12: Performance of platform.

According to inequality (47), we can further derive

$$\frac{\Pr\left(\boldsymbol{x}_i^{(T)}=\boldsymbol{x}\right)}{\Pr\left(\boldsymbol{x}_i^{(T)\prime}=\boldsymbol{x}\right)} \leq e^{\max\limits_{1\leq t\leq T}\max\limits_{1\leq k\leq U}\varepsilon_{k,i}^{(t)}}. \tag{51}$$

According to Definition 2, the time-invariant task satisfies $\sum_{i=1}^{V}\varepsilon_i$-differential privacy. Therefore, Theorem 3 is proved. □

Appendix D
Proof of Lemma 1

Proof. In a crowdsensing task, it is assumed that there are M participants, N payment levels of the platform, and J PPLs. For the platform, a payment is selected through Q-table, and a record is required for each payment. For the participants, we first consider that each participant will randomly select a PPL, and then each participant will have J choices, among which M participants have J^M choices, that is the number of $\hat{N}_j$, the size of Q-table is at most $N\times J^M$. This proves that, as M increases, the size of Q-table size increases exponentially. □

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile Crowdsensing: Current State and Future Challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.
[2] http://www.noisetube.net/.
[3] C. Cao, Z. Liu, M. Li, W. Wang, and Z. Qin, "Walkway Discovery from Large Scale Crowdsensing," in *Proc. of ACM IPSN*, pp. 13–24, 2018.
[4] http://www.fieldagent.net/.
[5] https://www.waze.com/.
[6] http://opensense.epfl.ch/.
[7] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li, "Jigsaw: Indoor Floor Plan Reconstruction via Mobile Crowdsensing," in *Proc. of ACM MobiCom*, pp. 2420–2428, 2014.
[8] X. Zheng, Z. Cai, and Y. Li, "Data Linkage in Smart IoT Systems: A Consideration from Privacy Perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.
[9] C. Dwork, "Differential Privacy," in *Encyclopedia of Cryptography and Security*, pp. 338–340, 2011.
[10] Z. Wang, X. Pang, Y. Chen, H. Shao, Q. Wang, L. Wu, H. Chen, and H. Qi, "Privacy-Preserving Crowd-Sourced Statistical Data Publishing with An Untrusted Server," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1356–1367, 2019.
[11] L. Zhu, M. Li, and Z. Zhang, "Secure Fog-Assisted Crowdsensing With Collusion Resistance: From Data Reporting to Data Requesting," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5473–5484, 2019.
[12] X. Wang, Z. Liu, X. Tian, X. Gan, Y. Guan, and X. Wang, "Incentivizing Crowdsensing With Location-Privacy Preserving," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6940–6952, 2017.
[13] Z. Wang, J. Hu, R. Lv, J. Wei, Q. Wang, D. Yang, and H. Qi, "Personalized Privacy-Preserving Task Allocation for Mobile Crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1330–1341, 2019.
[14] H. Wu, L. Wang, and G. Xue, "Privacy-Aware Task Allocation and Data Aggregation in Fog-Assisted Spatial Crowdsourcing," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2019.
[15] Y. Sei and A. Ohsuga, "Differential Private Data Collection and Analysis Based on Randomized Multiple Dummies for Untrusted Mobile Crowdsensing," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 926–939, 2017.
[16] F. Qiu, F. Wu, and G. Chen, "Privacy and Quality Preserving Multimedia Data Aggregation for Participatory Sensing Systems," *IEEE Transactions on Mobile Computing*, vol. 14, no. 6, pp. 1287–1300, 2015.

[17] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-Preserving Support Vector Machine Training over Blockchain-Based Encrypted IoT Data in Smart Cities," *IEEE Internet of Things Journal*, DOI: 10.1109/JIOT.2019.2901840.

[18] M. Shen, Y. Deng, L. Zhu, D. Xiaojiang, and N. Guizani, "Privacy-Preserving Image Retrieval for Medical IoT Systems: A Blockchain-Based Approach," *IEEE Network*, DOI: 10.1109/MNET.001.1800503.

[19] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep Learning Based Inference of Private Information Using Embedded Sensors in Smart Devices," *IEEE Network Magazine*, vol. 32, no. 4, pp. 8–14, 2018.

[20] 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, 2018.

[21] M. A. Alsheikh, D. Niyato, D. Leong, P. Wang, and Z. Han, "Privacy Management and Optimal Pricing in People-Centric Sensing," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 906–920, 2017.

[22] H. Jin, L. Su, and K. Nahrstedt, "CENTURION: Incentivizing Multi-requester Mobile Crowd Sensing," in *Proc. of IEEE INFOCOM*, pp. 1–9, 2017.

[23] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to Smartphones: Incentive Mechanism Design for Mobile Phone Sensing," in *Proc. of ACM Mobicom*, pp. 173–184, 2012.

[24] J. Nie, J. Luo, Z. Xiong, D. Niyato, and P. Wang, "A Stackelberg Game Approach Toward Socially-Aware Incentive Mechanisms for Mobile Crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 724–738, 2019.

[25] B. Cao, S. Xia, J. Han, and Y. Li, "A Distributed Game Methodology for Crowdsensing in Uncertain Wireless Scenario," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2019.

[26] Y. Zhan, Y. Xia, Y. Liu, F. Li, and Y. Wang, "Incentive-Aware Time-Sensitive Data Collection in Mobile Opportunistic Crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 7849–7861, 2017.

[27] Y. Zhan, C. H. Liu, Y. Zhao, J. Zhang, and J. Tang, "Free Market of Multi-Leader Multi-Follower Mobile Crowdsensing: An Incentive Mechanism Design by Deep Reinforcement Learning," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2019.

[28] C. Wang, C. Wang, Z. Wang, Y. Xiaojun, J. X. Yu, and B. Wang, "DeepDirect: Learning Directions of Social Ties with Edge-based Network Embedding," *IEEE Transactions on Knowledge and Data Engineering*, DOI: 10.1109/TKDE.2018.2877748.

[29] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, "Enabling Privacy-Preserving Incentives for Mobile Crowd Sensing Systems," in *Proc. of IEEE ICDCS*, pp. 344–353, 2016.

[30] Y. Wang, Z. Cai, X. Tong, Y. Gao, and G. Yin, "Truthful Incentive Mechanism With Location Privacy-Preserving for Mobile Crowdsourcing Systems," *Elsevier Computer Network*, vol. 135, pp. 32–43, 2018.

[31] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "INCEPTION: Incentivizing Privacy-preserving Data Aggregation for Mobile Crowd Sensing Systems," in *Proc. of IEEE MobiHoc*, pp. 341–350, 2016.

[32] T. Li, T. Jung, Z. Qiu, H. Li, L. Cao, and Y. Wang, "Scalable Privacy-Preserving Participant Selection for Mobile Crowdsensing Systems: Participant Grouping and Secure Group Bidding," *IEEE Transactions on Network Science and Engineering*, DOI: 10.1109/TNSE.2018.2791948.

[33] Z. Zhang, S. He, J. Chen, and J. Zhang, "REAP: An Efficient Incentive Mechanism for Reconciling Aggregation Accuracy and Individual Privacy in Crowdsensing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 12, pp. 2995–3007, 2018.

[34] Y. Liu, L. Hao, Z. Liu, K. Sharif, Y. Wang, and S. K. Das, "Mitigating Interference via Power Control for Two-Tier Femtocell Networks: A Hierarchical Game Approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 7194–7198, 2019.

Yang Liu (M'13) received the BE degree in electrical engineering and its automation and the ME degree in control theory and control engineering from Harbin Engineering University, Harbin, China, in 2008 and 2010, respectively, and the PhD degree in computer engineering at the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, in 2014. He is currently an associate professor at Beijing University of Posts and Telecommunications. His current research interests include wireless networking and mobile computing. He is a member of the IEEE and ACM.

Hongsheng Wang received the BE degree in computer science and technology from Shijiazhuang Tiedao University, Shijiazhuang, China, in 2018. He is currently pursuing the ME degree in computer technology from Beijing University of Posts and Telecommunications, Beijing, China. His research interests including machine learning.

Mugen Peng (M'05, SM'11, F'19) received the PhD degree in communication and information systems from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2005. Afterward, he joined BUPT, where he has been a Full Professor with the School of Information and Communication Engineering since 2012. During 2014 he was also an academic visiting fellow at Princeton University, USA. He leads a Research Group focusing on wireless transmission and networking technologies in BUPT. He has authored and coauthored over 90 refereed IEEE journal papers and over 300 conference proceeding papers. His main research areas include wireless communication theory, radio signal processing, cooperative communication, self-organization networking, heterogeneous networking, cloud communication, and Internet of Things.

Dr. Peng was a recipient of the 2018 Heinrich Hertz Prize Paper Award, the 2014 IEEE ComSoc AP Outstanding Young Researcher Award, and the Best Paper Award in the JCN 2016, IEEE WCNC 2015, IEEE GameNets 2014, IEEE CIT 2014, ICCTA 2011, IC-BNMT 2010, and IET CCWMC 2009. He is currently or have been on the Editorial/Associate Editorial Board of the IEEE Communications Magazine, IEEE ACCESS, IEEE Internet of Things Journal, IET Communications, and China Communications.

Jianfeng Guan (M'14) received the BE degree in telecommunication engineering from Northeastern University, Shenyang, China in 2004, and PhD degree in Communication and Information System from Beijing Jiaotong University, Beijing, China in 2010. He is currently an associate professor at Beijing University of Posts and Telecommunications. His research interests include future network architecture, network security and mobile Internet.

Jia Xu (M'15) received the M.S. degree in School of Information and Engineering from Yangzhou University, Jiangsu, China, in 2006 and the PhD. degree in School of Computer Science and Engineering from Nanjing University of Science and Technology, Jiangsu, China, in 2010. He is currently a professor in the School of Computer Science at Nanjing University of Posts and Telecommunications. He was a visiting Scholar in the Department of Electrical Engineering & Computer Science at Colorado School of Mines from Nov.2014 to May.2015. His main research interests include crowdsourcing, edge computing and wireless sensor networks.

Yu Wang (M'04, SM'12, F'18) is currently a Professor in the Department of Computer and Information Sciences at Temple University. Prior to joining Temple University, he was a Professor of Computer Science at the University of North Carolina at Charlotte (UNC Charlotte). He holds a Ph.D. from Illinois Institute of Technology, an MEng and a BEng from Tsinghua University, all in Computer Science. His research interest includes wireless networks, smart sensing, and mobile computing. His research has been continuously supported by US National Science Foundation and US Department of Transportation. He has published over 200 papers in peer reviewed journals and conferences, with four best paper awards. He has served as general chair, program chair, program committee member, etc. for many international conferences (such as IEEE IPCCC, ACM MobiHoc, IEEE INFOCOM, IEEE GLOBECOM, IEEE ICC). He has served as Editorial Board Member of several international journals, including IEEE Transactions on Parallel and Distributed Systems. He is a recipient of Ralph E. Powe Junior Faculty Enhancement Awards from Oak Ridge Associated Universities (2006), Outstanding Faculty Research Award from College of Computing and Informatics at UNC Charlotte (2008), and Fellow of IEEE (2018). He is also a senior member of ACM and a member of AAAS.