Digital Communications and Networks xxx (xxxx) xxx

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

A game-theoretic approach for federated learning: A trade-off among privacy, accuracy and energy

Lihua Yin [a], Sixin Lin [a], Zhe Sun [a,*], Ran Li [a], Yuanyuan He [b], Zhiqiang Hao [c]

[a] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, 510006, China
[b] School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China
[c] China Industrial Control Systems Cyber Emergency Response Team, Beijing, 100040, China

ARTICLE INFO

Keywords:
Federated learning
Privacy preservation
Energy optimization
Game theory
Distributed communication systems

ABSTRACT

Benefiting from the development of Federated Learning (FL) and distributed communication systems, large-scale intelligent applications become possible. Distributed devices not only provide adequate training data, but also cause privacy leakage and energy consumption. How to optimize the energy consumption in distributed communication systems, while ensuring the privacy of users and model accuracy, has become an urgent challenge. In this paper, we define the FL as a 3-layer architecture including users, agents and server. In order to find a balance among model training accuracy, privacy-preserving effect, and energy consumption, we design the training process of FL as game models. We use an extensive game tree to analyze the key elements that influence the players' decisions in the single game, and then find the incentive mechanism that meet the social norms through the repeated game. The experimental results show that the Nash equilibrium we obtained satisfies the laws of reality, and the proposed incentive mechanism can also promote users to submit high-quality data in FL. Following the multiple rounds of play, the incentive mechanism can help all players find the optimal strategies for energy, privacy, and accuracy of FL in distributed communication systems.

1. Introduction

The spread of distributed communication technologies has changed the patterns of many application fields, including e-health [1,2], Internet of Things (IoT) [3,4], vehicle networking [5,6]. Distributed devices and total connections are growing rapidly. According to a report by global mobile trading body GSMA, total connections will reach nearly 25 billion by 2025, up from 12 billion in 2019 [7]. Also, ubiquitous sensors continuously generate and collect massive amounts of data. It results in the development of intelligent applications based on machine learning and deep learning, while bringing great convenience to users [8,9]. Among them, Federated Learning (FL) is a typical distributed system paradigm that allows users distributed in different locations to cooperate in training a global model. However, advanced techniques lead to more energy consumption due to the separate computation and communication overheads in multiple locations [10].

Meanwhile, additional security and privacy preservation mechanism further exacerbate the energy consumption in distributed communication systems. It leads to three optimization goals of model accuracy, privacy preservation and energy consumption. Green communication technology is aimed at achieving energy and environmental protection by designing energy-sensitive distributed communication networks [11]. Most technologies are proposed to reduce energy consumption by saving unnecessary energy overhead. However, as these optimization techniques mature, the effectiveness of new energy-saving technologies becomes less apparent. Energy consumption optimization targets has shifted from optimization of technical details to energy distribution. Allocating energy to optimize model accuracy, privacy preservation and energy consumption is inescapable. There are a few global optimization approaches that consider the above three goals. Most existing work focuses on two sub-topics: the trade-off between privacy preservation and model accuracy, the optimization of energy consumption and model accuracy.

Existing work on the first topic can be roughly divided into two categories: reinforcement learning-based methods and game theory-based methods. Deep reinforcement learning methods aim to find the optimal noise parameters in the training models. Lian et al. [12] designed an FL system by applying Differential Privacy (DP) to the selected layer

* Corresponding author.
E-mail address: sunzhe@gzhu.edu.cn (Z. Sun).

https://doi.org/10.1016/j.dcan.2022.12.024
Received 30 September 2021; Received in revised form 13 December 2022; Accepted 31 December 2022
Available online xxxx
2352-8648/© 2023 Chongqing University of Posts and Telecommunications. Publishing Services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Please cite this article as: L. Yin et al., A game-theoretic approach for federated learning: A trade-off among privacy, accuracy and energy, Digital Communications and Networks, https://doi.org/10.1016/j.dcan.2022.12.024

parameters and found the optimal parameters to add noise. Such approaches tend to focus on specific technologies, rather than addressing the conflict between accuracy and privacy preservation from a generic perspective. Another approaches are to find a suitable trade-off through game theory. Liu et al. [13] modeled the relationship between uploaded sensing data and task performance, and found the optimal privacy strategy for users through a stackelberg game. This approach helped the platform collect as much high-quality sensor data as possible and reduce the leakage of sensitive user privacy. The methods presented above only address the subproblem of trade-off between privacy preservation and model accuracy. Optimization of energy consumption is not considered in the model training.

Energy optimization methods for training models are currently available as follows. One is to improve deep learning techniques to reduce energy consumption. Panda et al. [14] proposed a conditional deep learning approach for improving image recognition. It concatenated a linear network of output neurons for each convolutional layer, and then determined whether the classification task could be completed at the current stage to reduce the energy consumption. Other methods are focused on finding the equilibrium strategy through game-theoretic approaches. Zou et al. [15] modeled the energy consumption and Quality of Service (QoS) in FL, and obtained the evolutionary stable strategy. As a result, they found the optimal strategy for all players. However, these methods ignore privacy leakage. Model accuracy, privacy preservation and energy saving are three essential aspects to achieve widespread adoption of deep learning applications in distributed communication systems.

In this paper, we model the process of FL under a 3-layer distributed communication system scenario, which comprises three players: users, agents, and servers. Unlike traditional FL, where we introduce agents to reduce the computational and energy pressure of terminal devices. In addition, we help users understand their own privacy policies in FL and assume all players are rational and selfish. We consider the energy consumption and benefits of all players, as well as privacy issues in training. We also analyze the payoff for all players in single and repeated games, and find the optimal strategy and constraint for each one. In summary, we make the following contributions.

- We formalize FL in distributed communication systems as an extensive game tree through considering the sequential ordering of users, agents and servers in training. By simulating the decisions of each player, we design a punishment mechanism to help players find a balance between privacy preservation, model accuracy, and energy consumption.
- We analyze the key factors affecting the payoff of three players in the single game, such as the loss of privacy leaking, the energy consumption in communication, and the reward for training contributions. On this basis, we construct the pay-off function of each player and derive the Nash equilibrium.
- We propose social norms that satisfy the requirement of privacy preservation, energy consumption, and model accuracy in the repeated game. The numerical simulation experimental results show that our incentive mechanism works better for users who are more concerned about privacy leakage and energy consumption in FL.

The rest of the paper is organized as follows. In Section 2, we present the existing work related to the trade-off between privacy preservation and accuracy, and energy consumption optimization. In Section 3, we describe the system model and the responsibility of all players. In Section 4, we define the payoff function of each player in single game and derive the Nash equilibrium. We also introduce the discount to provide a long-term analysis in the repeated game. Section 5 describes the numerical simulation experimental results. And we conclude this paper in Section 6.

2. Related work

2.1. The trade-off between privacy preservation and accuracy in FL

An advantage of FL is that it does not need to send raw data. However, malicious attackers can infer user privacy through membership inference attacks. To reduce privacy leakage in FL, Hao et al. [16] developed a privacy-enhanced FL method by embedding gradient ciphertext into augmented learning with errors. Since DP does not require the massive computational overhead like traditional encryption methods, it can be widely used in FL to protect the user privacy. Truex et al. [17] combined DP to develop protocols while providing the provable privacy authorization in FL. Wei et al. [18] designed a framework for FL which could add artificial noise and satisfy DP before aggregation. However, when user privacy is guaranteed, the effective information delivery is reduced. It results that the QoS is difficult to meet user demand.

Based on the decentralized nature of the blockchain, Lu et al. [19] proposed a data sharing mechanism which integrated FL into the authorized blockchain by designing a collaborative architecture. In Ref. [17], the authors introduced the security multi-party computation in FL to improve model accuracy. Duan et al. [20] used multiple schedulers to design a self-balancing FL framework, and reduced the training errors to improve accuracy. In Ref. [21], the authors set a threshold and used it to identity the correlation of client-side updates and training data, while improving the model accuracy. Based on the same purpose of solving federation related client selection, Nagalapatti et al. [22] designed a Shapley value based the federated averaging algorithm to improve the accuracy.

2.2. The trade-off between privacy preservation and accuracy based game theory

The methods mentioned above focus only on specific technologies to guarantee privacy and accuracy, so that game theory is introduced to find a balance between privacy preservation and accuracy. Based on whether they have consistent goals and interests or not, game theory can be split into cooperative and non-cooperative games.

In cooperative games, privacy leakage is positively correlated with social networks benefits. Also, each user has the requirement to protect their privacy as much as possible. Du et al. [23] designed an evolutionary game to encourage users to adjust privacy preservation according their benefits. It could help social network managers find an evolutionary stable strategy. Meanwhile, the user data is not isolated but linked to each other in social networks. The level of privacy preservation for users depends on their own and neighbors' selections. Wu et al. [24] proposed a dynamic game model based on the influence of neighbors' selections, and evaluated the real privacy preservation effect of user.

In non-cooperative games, there is a conflict of interests between malicious attackers and users. Xiao et al. [25] applied Markov chain in stackelberg game and proposed a reinforcement learning algorithm to help them find the Nash equilibrium. The conflict also exists in data owners and data demanders. Sfar et al. [26] designed a Markov game based on data transaction process, and found the best transaction strategy to maximize the benefits of all players. To balance user privacy and data application, Xu et al. [27] analyzed the changes of user privacy budget and model accuracy, while modeling as a dynamic game with incomplete information. In Ref. [28], Hu et al. designed a two-stage Stackelberg game for On-Device Federated Learning (ODFL) to improve the QoS. They selected the users who were most likely to provide reliable data. Sun et al. [29] introduced concept drift and found the stable strategy based on evolution game, which could provide trade-off between QoS and privacy in adversarial training. These game models focus solely on cooperation or non-cooperation. Jin et al. [30] designed the attack and

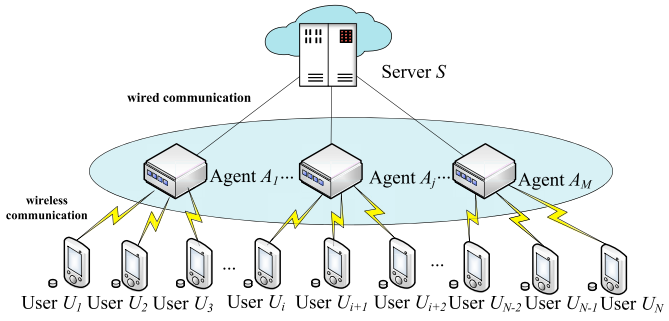

Fig. 1. An illustration of the large-scale FL.

defense models for intrusion detection system, and proposed a two-layer game to analyze the utility-privacy trade-off and calculated the optimal strategy.

2.3. Energy consumption optimization in FL

Model training in FL is implemented through multiple iterations in a distributed communication system. The communication and computing costs overhead increases as the number of iterations increases. However, these methods presented above consider the trade-off without energy consumption optimization.

To improve the performance and efficiency of FL, energy consumption optimization has been extensively studied in recent years. Zhou et al. [31] designed a framework that could calculate the optimal data online scheduling, while reducing the cost of energy consumption and computation. In Ref. [21], the authors cut down on energy consumption by identifying data and finding the relevance with client-side model updates. Yang et al. [32] constructed the completion time minimization problem in an iterative algorithm, while calculating the time allocation and power control to decrease the energy consumption.

The iterations of client-server also increase the energy consumption communication. Luo et al. [33] designed a sampling-based algorithm to select the number of clients and its local iterations in each training round, while reducing the communication costs and improve model convergence speed. Hamer et al. [34] provided an ensemble algorithm for FL, and the communication cost was independent of ensemble size. In Ref. [35], Elgabli et al. proposed an analog federated alternating direction method of multipliers, and used a single channel to upload and aggregate the models over-the-air. In order to balance the energy consumption of computing and communication, the authors [36] derived a convergence bound and introduced the bound to design a compression control scheme. Wei et al. [37] proposed a Communication Rounds Discounting (CRD) method to find a trade-off between the computational complexity and the convergence performance, and finally optimized the communication energy consumption.

These methods only consider energy consumption optimization without the trade-off between privacy and model accuracy. That is the reason why we propose a game-theoretic approach among privacy, accuracy and energy. In this paper, we formalize each player's decisions and design their payoff functions. Then we introduce an extensive game tree to derive the Nash equilibrium. Also, we make the long-term analysis in a repeated game, while calculating the social norms and finding the optimal strategies for all players in FL.

3. System model and problem statement

In this section, we formally define our FL model, energy consumption model and problem statement.

3.1. FL model

Integration of FL's incentive mechanism and mobile edge computing is becoming a trend [38]. As shown in Fig. 1, we consider a large-scale FL illustration that consists of 3 kinds of participants. We denote them as a set $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_N\}$ of users, a set $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_M\}$ of agents, and a set $\mathcal{S} = \{\mathcal{S}\}$ of a server, respectively. All participants cooperatively perform an FL algorithm to train a global model, as shown in Algorithm 1. Each user $\mathcal{U}_i$ can communicate with the server $\mathcal{S}$ through its agent $\mathcal{A}_j$, and has a local dataset with $\mathcal{D}_{\mathcal{U}_i} = \{\langle \mathcal{D}_{\mathcal{U}_i}1, \mathcal{D}_{\mathcal{U}_i}2, \ldots, \mathcal{D}_{\mathcal{U}_i\mathcal{D}_K} >, \mathcal{L}_{\mathcal{U}_i}\}$. Each data sample is defined as $\mathcal{D}_{\mathcal{U}_i k}$ and the number of samples is $\mathcal{D}_K$. $\mathcal{L}_{\mathcal{U}_i}$ is a label represents the association degree between dataset $\mathcal{D}_{\mathcal{U}_i}$ and FL task $\mathcal{T}$. In each iteration, the current global model is downloaded from the server $\mathcal{S}$ to each user $\mathcal{U}_i$ by its agent $\mathcal{A}_j$, and each user $\mathcal{U}_i$ trains the local model with its dataset $\mathcal{D}_{\mathcal{U}_i}$. Finally, the user updated model is sent back to the server $\mathcal{S}$ by the aggregation of agent $\mathcal{A}_j$, where it is aggregated and updated to the current global model.

Algorithm 1 FL-Algorithm

Input: the initial Global-FL model $G_{\mathcal{S}}$
Output: the updated Global-FL model $G'_{\mathcal{S}}$

Users: Local-FL model training

while $\mathcal{U}_i \in \mathcal{U}_1, \mathcal{U}_2, \cdots, \mathcal{U}_N$ **do**
  Initialize the model $L_{\mathcal{U}_i}(\omega) = G_{\mathcal{S}}$
  Decide the noise intensity $\epsilon_{\mathcal{U}_i}$
  Compute gradient $\min_\omega f_{\mathcal{U}_i}(\omega, \mathcal{D}_{\mathcal{U}_i}) =$
  $\frac{1}{\mathcal{D}_k} \sum_{k=1}^{\mathcal{D}_k} Loss_{\mathcal{T}}(\omega, \mathcal{D}_{\mathcal{U}_i k})$
  Add noise to the gradient $\tilde{g} = g + \varepsilon_{\mathcal{U}_i}$
  Update the parameters $\omega' = \omega - \alpha\tilde{g}$
end while

Agents: Local-FL models aggregating

while $\mathcal{A}_j \in \mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_M$ **do**
  $P_{\mathcal{A}_j} = G_{\mathcal{S}} + \frac{1}{N_{\mathcal{U}_i}} \sum_{i=1}^{N_{\mathcal{U}_i}} L_{\mathcal{U}_i}$
end while

Server: Agent-FL models aggregating

$G'_{\mathcal{S}} = G_{\mathcal{S}} + \frac{1}{M_{\mathcal{A}_j}} \sum_{j=1}^{M_{\mathcal{A}_j}} P_{\mathcal{A}_j}$ **return** $G'_{\mathcal{S}}$

Hereinafter, we define the FL model trained by each user's dataset $\mathcal{D}_{\mathcal{U}_i}$ as Local-FL model. Aggregated by each agent $\mathcal{A}_j$ and server $\mathcal{S}$, the FL models are called the Agent-FL model and Global-FL model, respectively.

3.1.1. Local-FL model

At the beginning of training, each user $\mathcal{U}_i$ receives a Global-FL model and determines the noise intensity $\epsilon_{\mathcal{U}_i}$ that satisfies Local Differential Privacy (LDP), for it should make a trade-off between privacy preservation and QoS. For instance, going to a restaurant may not be as sensitive as going to a hospital for most users. Also, the noise $\epsilon_{\mathcal{U}_i}$ is added to the gradients.

We set a vector θ to denote the parameters of Global-FL model, while the Global-FL model is defined as $G_{\mathcal{S}}(\theta)$ at the current iteration. To simplify the notation, it is denoted by $G_{\mathcal{S}}$. The Local-FL model of each user $\mathcal{U}_i$ can be expressed as

$$L_{\mathcal{U}_i}(\omega) = G_{\mathcal{S}} \tag{1}$$

where ω is a vector that denote the parameters of Local-FL model.

The total loss function for each Local-FL model is defined as

$$f_{\mathcal{U}_i}(\omega, \mathcal{D}_{\mathcal{U}_i}) = \frac{1}{\mathcal{D}_k} \sum_{k=1}^{\mathcal{D}_k} Loss_{\mathcal{T}}(\omega, \mathcal{D}_{\mathcal{U}_i k}) \tag{2}$$

where $Loss_{\mathcal{T}}$ is the loss function of FL task $\mathcal{T}$. The optimization problem can be formulated as

$$\min_{\omega} f_{\mathcal{U}_i}(\omega, \mathcal{D}_{\mathcal{U}_i}) = \frac{1}{\mathcal{D}_k} \sum_{k=1}^{\mathcal{D}_k} Loss_{\mathcal{T}}(\omega, \mathcal{D}_{\mathcal{U}_i k}) \tag{3}$$

We compute the gradient g by solving equation (3), so that the gradient g with noise $\epsilon_{\mathcal{U}_i}$ is described as

$$\tilde{g} = g + \epsilon_{\mathcal{U}_i} \tag{4}$$

The updated vector of Local-FL model is calculated by

$$\omega' = \omega - \alpha \tilde{g} \tag{5}$$

where α is the learning rate, and the updated Local-FL model is expressed as $L_{\mathcal{U}_i}(\omega')$. It is denoted by $L_{\mathcal{U}_i}$ and the QoS is defined as $QoS_{L_{\mathcal{U}_i}}$.

To measure the privacy sensitivity degree of each user dataset $\mathcal{D}_{\mathcal{U}_i}$, we design the formula

$$\mathcal{S}ens(\mathcal{D}_{\mathcal{U}_i}) = \sum_{t=0}^{\infty} \sum_{\omega^s \in W_s} (\epsilon_{\mathcal{U}_i})^t \left| P_r\left[D^t = \omega_s \middle| D^0 = \omega_0\right] - P_r[D^t = \omega_s] \right| \tag{6}$$

where Pr represents the probability [39]. D^t and D^0 indicate the vector ω updating at time t and time 0, respectively. W_s is a set of vectors.

3.1.2. Agent-FL model

The Agent-FL model is aggregated from its user group, and it can be expressed as

$$P_{\mathcal{A}_j} = G_{\mathcal{S}} + \frac{1}{N_{\mathcal{U}_i}} \sum_{i=1}^{N_{\mathcal{U}_i}} L_{\mathcal{U}_i} \tag{7}$$

where $N_{\mathcal{U}_i}$ is the number of received Local-FL models. $L_{\mathcal{U}_i}$ denotes the updated Local-FL model from the user $\mathcal{U}_i$ at current iteration.

To measure the QoS of Agent-FL model, we use the average contribution degree ρ_1 $(0 < \rho_1 < 1)$ instead of the specific value for all users in the group [29].

We denote the following function

$$QoS_{P_{\mathcal{A}_j}}(\rho_1, r_{\mathcal{U}_i}) = 1 - (1 - \rho_1)^{r_{\mathcal{U}_i}} \tag{8}$$

where $r_{\mathcal{U}_i}$ is the number of the users who make contributions.

3.1.3. Global-FL model

The Global-FL model is aggregated from its agent groups models, and it can be expressed as

$$G'_{\mathcal{S}} = G_{\mathcal{S}} + \frac{1}{M_{\mathcal{A}_j}} \sum_{j=1}^{M_{\mathcal{A}_j}} P_{\mathcal{A}_j} \tag{9}$$

where $M_{\mathcal{A}_j}$ is the number of received Agent-FL models. $P_{\mathcal{A}_j}$ denotes the updated Agent-FL model from the agent $\mathcal{A}_j$ at current iteration.

To measure the QoS of Global-FL model, we denote the following function

$$QoS_{G_{\mathcal{S}}}(\rho_2, r_{\mathcal{A}_j}) = 1 - (1 - \rho_2)^{r_{\mathcal{A}_j}} \tag{10}$$

where ρ_2 $(0 < \rho_2 < 1)$ is the average contribution degree of Global-FL model QoS, and $r_{\mathcal{A}_j}$ is the number of the agents who make contributions in the agent groups of server $\mathcal{S}$.

3.2. Energy consumption model

We mainly analyze two aspects of the energy consumption in FL: local computing and communication computation. Note that we do not consider the energy usage of devices operations and the energy consumption of models downloads in FL. This is because the energy usage of devices operations has little impact on the FL energy consumption compared with the model aggregation and uploading. Also, the high bandwidths in data broadcasting make the energy consumption of models downloads negligible for the agents and server. Hereinafter, we formalize the energy consumption.

3.2.1. Computational energy consumption

For each user $\mathcal{U}_i$, the local computation is focused on the Local-FL model training with its local dataset $D_{\mathcal{U}_i}$.

To compute the local iterations, we design the following formula

$$T\left(QoS_{P_{\mathcal{A}_j}}, QoS_{L_{\mathcal{U}_i}}\right) = \frac{\zeta \cdot \log\left(\frac{1}{QoS_{P_{\mathcal{A}_j}}}\right)}{1 - QoS_{L_{\mathcal{U}_i}}} \tag{11}$$

where $\zeta (0 < \zeta < 1)$ is a constant [40], and the local computation energy of each user $\mathcal{U}_i$ can be expressed as

$$B_{\mathcal{U}_i} = \kappa_{\mathcal{U}_i} C_{\mathcal{D}_{\mathcal{U}_i k}} \mathcal{D}_K T\left(QoS_{P_{\mathcal{A}_j}}, QoS_{L_{\mathcal{U}_i}}\right) f_{\mathcal{U}_i}^2 \tag{12}$$

where $\kappa_{\mathcal{U}_i}$ is the effective switched capacitance depending on the chip architecture of each user's device [32]. $C_{\mathcal{D}_{\mathcal{U}_i k}}$ is the number of CPU cycles to compute one data sample $\mathcal{D}_{\mathcal{U}_i k}$. $f_{\mathcal{U}_i}$ is the computation capacity measured by the number of CPU cycles per second at each user $\mathcal{U}_i$.

For each agent $\mathcal{A}_j$, the local computation energy is mainly on the Local-FL models aggregation. It can be described as

$$B_{\mathcal{A}_j} = \kappa_{\mathcal{A}_j} C_{L_{\mathcal{U}_i}} N_{\mathcal{U}_i} f_{\mathcal{A}_j}^2 \tag{13}$$

where $\kappa_{\mathcal{A}_j}$ is the effective switched capacitance of each agent's device, and $C_{L_{\mathcal{U}_i}}$ is the number of CPU cycles to compute one Local-FL model. $f_{\mathcal{A}_j}$ is the computation capacity of each agent $\mathcal{A}_j$.

Similarly, the local computational energy of server $\mathcal{S}$ can be expressed as

$$B_{\mathcal{S}} = \kappa_{\mathcal{S}} C_{P_{\mathcal{A}_j}} M_{\mathcal{A}_j} f_{\mathcal{S}}^2 \tag{14}$$

where $\kappa_{\mathcal{S}}$ is the effective switched capacitance of server's device, and $C_{P_{\mathcal{A}_j}}$ is the number of CPU cycles to compute one Agent-FL model $P_{\mathcal{A}_j}$. $f_{\mathcal{S}}$ is the computation capacity of server $\mathcal{S}$.

3.2.2. Communication energy consumption

It is reasonable that the higher $QoS_{G_{\mathcal{S}}}$ requires more rounds of communication as shown in equation (11), so that we define the communication expenditure between each user $\mathcal{U}_i$ and agent $\mathcal{A}_j$ as follows

$$C_{\mathcal{U}_i} = T_{\mathcal{U}_i} \cdot \left(1 + QoS_{L_{\mathcal{U}_i}}\right) \tag{15}$$

where $T_{\mathcal{U}_i}$ is the time that the user $\mathcal{U}_i$ takes to communicate with the agent $\mathcal{A}_j$ for model parameters exchange at current iteration.

Similarly, the communication expenditure between each agent $\mathcal{A}_j$ and server $\mathcal{S}$ is expressed as follows

$$C_{\mathcal{A}_j} = T_{\mathcal{A}_j} \cdot \left(1 + QoS_{P_{\mathcal{A}_j}}\right) \tag{16}$$

where $T_{\mathcal{A}_j}$ is the time it takes to communicate with the server $\mathcal{S}$ for model parameters exchange at current iteration.

In the literature [40], $T_{\mathcal{U}_i}$ and $T_{\mathcal{A}_j}$ can be calculated by the following formula

Table 1
The frequently used notations in FL model and energy consumption model.

Notations	Description
$\mathcal{D}_{\mathcal{U}_i}$	The dataset of the user $\mathcal{U}_i$
$\mathcal{D}_K$	The number of the dataset $\mathcal{D}_{\mathcal{U}_i}$
$L_{\mathcal{U}_i}$	The Local-FL model of the user $\mathcal{U}_i$
$P_{\mathcal{A}_j}$	The Agent-FL model of the agent $\mathcal{A}_j$
$G_{\mathcal{S}}$, $G_{\mathcal{S}}^{'}$	The initial and updated Global-FL model of the server $\mathcal{S}$, respectively
$\epsilon_{\mathcal{U}_i}$	The noise intensity on the local training decided for the user $\mathcal{U}_i$
g	The gradient solved from the loss function optimization
$\tilde{g}$	The gradient added noise $\epsilon_{\mathcal{U}_i}$
ω, ω'	The vector that represents the initial and updated Local-FL model parameters, respectively
α	The learning rate of the Local-FL model
$\mathcal{S}ens(\mathcal{D}_{\mathcal{U}_i})$	The privacy sensitivity degree of the user $\mathcal{U}_i$
$N_{\mathcal{U}_i}$	The number of received Local-FL model for the agent $\mathcal{A}_j$
ρ_1	The average contribution degree of the user group for the agent $\mathcal{A}_j$
$r_{\mathcal{U}_i}$	The number of the users who make contributions on the user group for the agent $\mathcal{A}_j$
$QoS_{P_{\mathcal{A}_j}}(\rho_1, r_{\mathcal{U}_i})$	The QoS of the Agent-FL model
$M_{\mathcal{A}_j}$	The number of received Agent-FL model for the server $\mathcal{S}$
ρ_2	The average contribution degree of the agent group for the server $\mathcal{S}$
$r_{\mathcal{A}_j}$	The number of the agents who make contributions on the user group for the server $\mathcal{S}$
$QoS_{G_{\mathcal{S}}}(\rho_2, r_{\mathcal{A}_j})$	The QoS of the Global-FL model
$T\left(QoS_{P_{\mathcal{A}_j}}, QoS_{L_{\mathcal{U}_i}}\right)$	The local iterations of the Local-FL model $L_{\mathcal{U}_i}$
$B_{\mathcal{U}_i}$, $B_{\mathcal{A}_j}$, $B_{\mathcal{S}}$	The computational energy consumption of user $\mathcal{U}_i$, agent $\mathcal{A}_j$, and server $\mathcal{S}$, respectively
$C_{\mathcal{U}_i}$, $C_{\mathcal{A}_j}$	The communication energy consumption of user $\mathcal{U}_i$ and agent $\mathcal{A}_j$, respectively

$$T = \frac{e_{size}}{B\log_2\left(1 + \frac{p|G|^2}{\mathcal{N}}\right)} \tag{17}$$

where e_{size} is the size of model, and B is the allocated bandwidth. p is the transmission power. $|G|$ is the channel gain and $\mathcal{N}$ is the spectral density of Gaussian noise.

In summary, the notations of the FL model and energy consumption model are denoted in Table 1.

3.3. Problem statement

The advantage of FL is that participants have no requirements to provide raw data. Also, it splits the original traditional centralized deep learning model training task into multiple submodels, which means the participants have to perform model training tasks and pay corresponding energy consumption costs. However, malicious attackers can infer private information from training data, such as gradient inversion attacks and member inference attacks. That is the reason why most rational and selfish users are reluctant to participate in training. On the other hand, it complicates the issues that users make the considerations when energy consumption is intertwined with privacy leakage and model accuracy. Meanwhile, the requirements of QoS and low-energy aggregation consumption for agents and server should be considered in FL.

Therefore, we use the idea of maximizing utility for all players with the aim of finding the incentive mechanism [41]. We observe the strategies changes from a dynamic dimension through the multiple iterations of FL training. We design a game-theoretic approach among privacy, accuracy and energy to find the optimal strategies for all players.

4. Game model and analysis

In this section, we analyze the game model and design the payoff functions around privacy leakages, related rewards, and energy consumptions for all players.

4.1. Players descriptions

As shown in Fig. 2, we use a 3-layer architecture as our prototype for game models and simulate the training process for large-scale data in FL. We assume that all players are selfish and rational in the game model, and it is comprised of three players: users, agents, and server. We also suppose there is a social externality in the game model that the actions of all players affect each other, so that their payoffs $PO_{\mathcal{U}_i}$, $PO_{\mathcal{A}_j}$, $PO_{\mathcal{S}}$ are mutually constrained. In the meantime, the higher payoff strategies for each user $\mathcal{U}_i$ and agent $\mathcal{A}_j$ are easily spread in FL training. Their specific responsibilities are described as follows:

Users: Each user $\mathcal{U}_i$ selects the agent $\mathcal{A}_j$ based on the nearest transmission distance, and receives a Global-FL model $G_{\mathcal{S}}$ if it determines to participate in FL. Then it makes a trade-off between privacy sensitivity degree $\mathcal{S}ens(\mathcal{D}_{\mathcal{U}_i})$ and QoS $QoS_{L_{\mathcal{U}_i}}$ to decide the noise intensity $\epsilon_{\mathcal{U}_i}$, while training the Local-FL model$L_{\mathcal{U}_i}$ with its local dataset $D_{\mathcal{U}_i}$. After the local training, it sends the Local-FL model to the agent $\mathcal{A}_j$ and queries for the contribution degree $\rho_{\mathcal{U}_i}$. The payoff $PO_{\mathcal{U}_i}$ depends on the reward $Q_{\mathcal{U}_i}$, the subsidy $F_{\mathcal{U}_i}$, the privacy sensitivity degree $\mathcal{S}ens(\mathcal{D}_{\mathcal{U}_i})$, the energy expenditure $B_{\mathcal{U}_i}$ and $C_{\mathcal{U}_i}$.

Agents: Each agent $\mathcal{A}_j$ has its own user group and communicates with the server $\mathcal{S}$. It sends the Global-FL model $G_{\mathcal{S}}$ to each user $\mathcal{U}_i$ at the beginning. It is responsible for aggregating Local-FL models on its group and sending the updated Agent-FL model $P_{\mathcal{A}_j}$ to the server $\mathcal{S}$. It can provide the contribution queries in each iteration when it decides to

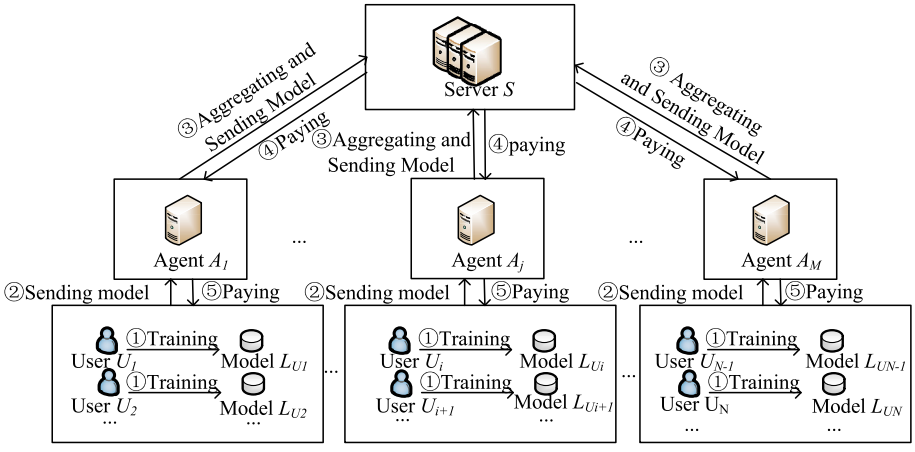

Fig. 2. The game model of 3-layer FL.

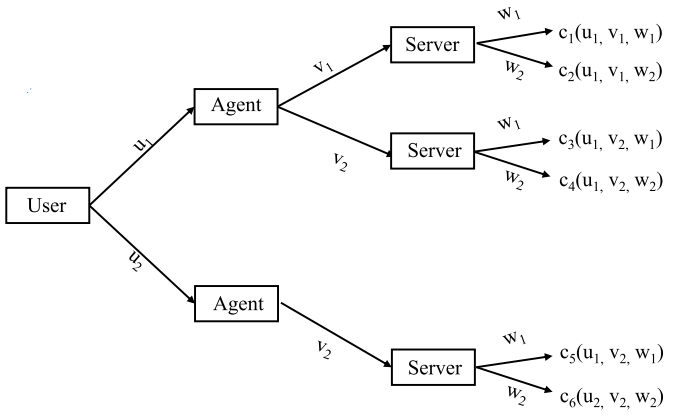

Fig. 3. The FL extensive game tree.

aggregate and update the Agent-FL model $P_{\mathcal{A}_j}$ in current iteration. Note that the agent $\mathcal{A}_j$ cannot participate in the FL training unless the user $\mathcal{U}_i$ updates the local model with its dataset $D_{\mathcal{U}_i}$. The payoff $PO_{\mathcal{A}_j}$ depends on the reward $Q_{\mathcal{A}_j}$, the subsidy $F_{\mathcal{A}_j}$, the energy expenditure $B_{\mathcal{A}_j}$ and $C_{\mathcal{A}_j}$.

Server: The server $\mathcal{S}$ has many users and agents' groups, and it provides Global-FL model $G_{\mathcal{S}}$ in each iteration. It is responsible for aggregating Agent-FL models and updating the Global-FL model $G_{\mathcal{S}}$ when it decides to participate in the FL training. The payoff $PO_{\mathcal{S}}$ depends on the reward $Q_{\mathcal{S}}$, the cost of paying $Q_{\mathcal{U}_i}$ and $Q_{\mathcal{A}_j}$, and the energy expenditure $C_{\mathcal{S}}$.

Hereinafter, we perform the single game analysis and repeated game analysis to observe how all players' strategies and payoffs change with iterations. We also find the optimal strategies and social norms for all players.

4.2. Single game model and analysis

In this section, we adopt the extensive game model to analyze the strategies among three players in a single interaction.

4.2.1. Single game model

We use the extensive game to make a single game analysis, and the extensive game tree is expressed as a three-tuple $\varphi = \{O, H, C\}$ as shown in Fig. 3. The set of players is $O = \{user, agent, server\}$. The action set of each player is $H = \{u_1, u_2, v_1, v_2, w_1, w_2\}$. We discuss u_1 and u_2 that the user $\mathcal{U}_i$ respectively trains the Local-FL model and not. v_1 and v_2 that the agent $\mathcal{A}_j$ respectively aggregates the Agent-FL model to the server $\mathcal{S}$ and not. w_1 and w_2 that the server $\mathcal{S}$ respectively completes the Global-FL model updating and not. The set of players' all strategies is $C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$. For example, if all players decide to participate in FL, the strategy is $c_1 = \{u_1, v_1, w_1\}$.

To better motivate each user $\mathcal{U}_i$ to use high-quality data for FL training, the server will provide reward $Q_{\mathcal{U}_i}$ to each user $\mathcal{U}_i$. It can be formulated as $Q_{\mathcal{U}_i} = k_2 Q(QoS_{L_{\mathcal{U}_i}} + \mathcal{L}_{\mathcal{U}_i})$, where $Q(QoS_{L_{\mathcal{U}_i}} + \mathcal{L}_{\mathcal{U}_i})$ is the profit of model QoS $QoS_{L_{\mathcal{U}_i}}$ and data quality $\mathcal{L}_{\mathcal{U}_i}$ $k_2(k_2 > 0)$ is the coefficient reflecting the positive impact of it. The rewards of each agent $\mathcal{A}_j$ and server $\mathcal{S}$ in terms of updating model are $Q_{\mathcal{A}_j} = \lambda Q_{\mathcal{U}_i}(\lambda > 0)$ and $Q_{\mathcal{S}} = \beta Q_{\mathcal{U}_i}(\beta \gg \lambda)$, where λ and β are the coefficients that represent the proportion of user's reward $Q_{\mathcal{U}_i}$. In addition, the subsidy $F_{\mathcal{U}_i} = k_3 T_{\mathcal{U}_i}$ for the user $\mathcal{U}_i$ is provided when the server $\mathcal{S}$ does not update the Global-FL model. The subsidy $F_{\mathcal{A}_j} = k_4 T_{\mathcal{A}_j}$ for the agent $\mathcal{A}_j$ is provided at the same state. k_3 and k_4 are the coefficients reflecting the subsidy ratio of their energy consumption. We also set the $-F_{\mathcal{A}_j}$ as a punishment for each agent $\mathcal{A}_j$ when it refuses to update Agent-FL model.

4.2.2. Payoff functions

In this section, we analyze the payoff functions of all players. We use u, v and w to represent the probabilities of participating in FL training for each user $\mathcal{U}_i$, agent $\mathcal{A}_j$ and server $\mathcal{S}$, respectively.

Based on the above analysis, we define the payoff function for each user $\mathcal{U}_i$ as

$$PO_{\mathcal{U}_i} = \begin{cases} Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i} & \text{if } u = 1, v = 1 \text{ and } w = 1 \\ F_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i} & \text{if } u = 1, v = 1 \text{ and } w = 0 \\ F_{\mathcal{U}_i} - T_{\mathcal{U}_i} & \text{if } u = 1, v = 0 \\ 0 & \text{else} \end{cases} \tag{18}$$

where $k_1(k_1 > 0)$ is the coefficient reflecting the negative impact of privacy leakage on user's payoff. $T_{\mathcal{U}_i}$ is the total energy consumption of local computing $B_{\mathcal{U}_i}$ and communication computing $C_{\mathcal{U}_i}$.

The payoff function for each user $\mathcal{A}_j$ as

$$\mathrm{PO}_{\mathrm{A_j}} = \begin{cases} \lambda Q_{u_i} - T_{A_i} & \text{if } u = 1, v = 1 \text{ and } w = 1 \\ F_{A_j} - T_{A_j} & \text{if } u = 1, v = 1 \text{ and } w = 0 \\ -F_{A_j} & \text{if } u = 1, v = 0 \text{ and } w = 1 \\ 0 & \text{else} \end{cases} \tag{19}$$

where $T_{\mathcal{A}_j}$ is the total energy consumption of local computing $B_{\mathcal{A}_j}$ and communication computing $C_{\mathcal{A}_j}$.

The payoff function for the server $\mathcal{S}$ as

$$PO_S = \begin{cases} (\beta - 1 - \lambda) Q u_i - T_S & \text{if } u = 1, v = 1 \text{ and } w = 1 \\ -F_{A_j} - F_{u_i} & \text{if } u = 1, v = 1 \text{ and } w = 0 \\ F_{A_j} - F_{u_i} - T_S & \text{if } u = 1, v = 0 \text{ and } w = 1 \\ -F_{u_i} & \text{if } u = 1, v = 0 \text{ and } w = 0 \\ -T_S & \text{if } u = 0, v = 0 \text{ and } w = 1 \\ 0 & \text{else} \end{cases} \tag{20}$$

where $T_{\mathcal{S}}$ is the total energy consumption of local computing $B_{\mathcal{S}}$ and communication computing $C_{\mathcal{S}}$.

In summary, the payoffs of the players in every strategy $c_k(k = 1, 2, \ldots 6)$ are shown in Table 2. $h_{\mathcal{U}_i}(c_k)$, $h_{\mathcal{A}_j}(c_k)$ and $h_{\mathcal{S}}(c_k)$ represent the specific payoff of the user $\mathcal{U}_i$, the agent $\mathcal{A}_j$ and the server $\mathcal{S}$ $h(c_k)$ is a set of them.

4.2.3. The solution of the nash equilibrium

To solve the Nash equilibrium for each player, we express $P(C_k)(k = 1, 2, \ldots 6)$ as the probability of each strategy, which can be represented as the function of $u(0 \le u \le 1)$, $v(0 \le v \le 1)$, $w(0 \le w \le 1)$. For example, $P(C_1)$ is uvw and $P(C_2)$ is $uv(1 - w)$.

The mathematical expectation E of each player's payoff can be calculated by the following formula

$$E = \sum_{k=1}^{6} h(C_k) P(C_k) \tag{21}$$

We define the expectations of the user $\mathcal{U}_i$, the agent $\mathcal{A}_j$ and the server $\mathcal{S}$ as $E_{\mathcal{U}_i}$, $E_{\mathcal{A}_j}$, $E_{\mathcal{S}}$. The user's expectation is described as

$$E_{\mathcal{U}_i} = (Q_{\mathcal{U}_i} - F_{\mathcal{U}_i}) uvw - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) uv + (F_{\mathcal{U}_i} - T_{\mathcal{U}_i}) u \tag{22}$$

The agent's expectation is described as

Table 2
The payoff of the players in different strategies.

	$c_1 = \{u_1, v_1, w_1\}$	$c_2 = \{u_1, v_1, w_0\}$	$c_3 = \{u_1, v_0, w_1\}$	$c_4 = \{u_1, v_0, w_0\}$	$c_5 = \{u_0, v_0, w_1\}$	$c_6 = \{u_0, v_0, w_0\}$
$h_{\mathcal{U}_i}(c_k)$	$Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}$	$F_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}$	$F_{\mathcal{U}_i} - T_{\mathcal{U}_i}$	$F_{\mathcal{U}_i} - T_{\mathcal{U}_i}$	0	0
$h_{\mathcal{A}_j}(c_k)$	$\lambda Q_{\mathcal{U}_i} - T_{\mathcal{A}_j}$	$F_{\mathcal{A}_j} - T_{\mathcal{A}_j}$	$-F_{\mathcal{A}_j}$	0	0	0
$h_{\mathcal{S}}(c_k)$	$\beta Q_{\mathcal{U}_i} - (1 + \lambda) Q_{\mathcal{U}_i} - T_{\mathcal{S}}$	$-F_{\mathcal{A}_j} - F_{\mathcal{U}_i}$	$F_{\mathcal{A}_j} - F_{\mathcal{U}_i} - T_{\mathcal{S}}$	$-F_{\mathcal{U}_i}$	$-T_{\mathcal{S}}$	0

$$E_{\mathcal{A}_j} = \lambda Q_{\mathcal{U}_i} uvw + F_{\mathcal{A}_j} uv - F_{\mathcal{A}_j} uw - T_{\mathcal{A}_j} uv \tag{23}$$

The server's expectation is described as

$$\begin{aligned} E_{\mathcal{S}} = {} & [(\beta - 1 - \lambda) Q_{\mathcal{U}_i} + F_{\mathcal{U}_i} - T_S] uv \\ & w + F_{\mathcal{A}_j} uw - F_{\mathcal{A}_j} uv - F_{\mathcal{U}_i} u + T_S vw - T_S w \end{aligned} \tag{24}$$

and the user's optimal strategy can be obtained by solving the following equation:

$$\frac{\partial E_{\mathcal{U}_i}}{\partial u} = (Q_{\mathcal{U}_i} - F_{\mathcal{U}_i}) vw - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) v + F_{\mathcal{U}_i} - T_{\mathcal{U}_i} = 0 \tag{25}$$

and the agent's optimal strategy can be obtained by solving the following equation:

$$\frac{\partial E_{\mathcal{A}_j}}{\partial v} = \lambda Q_{\mathcal{U}_i} uw + F_{\mathcal{A}_j} u - T_{\mathcal{A}_j} u = 0 \tag{26}$$

and the server's optimal strategy can be obtained by solving the following equation:

$$\frac{\partial E_{\mathcal{S}}}{\partial w} = [(\beta - 1 - \lambda) Q_{\mathcal{U}_i} + F_{\mathcal{U}_i} - T_S] uv + F_{\mathcal{A}_j} u + T_S v - T_S = 0 \tag{27}$$

We can get the following equation by equation (26).

$$\lambda Q_{\mathcal{U}_i} uw = T_{\mathcal{A}_j} u - F_{\mathcal{A}_j} u \tag{28}$$

We discuss equation (28) in two cases. Case 1: $u = 0$ and Case 2: $u \neq 0$.

Proposition 1. If case 1 is true, the optimal strategies of the user $\mathcal{U}_i$, the agent $\mathcal{A}_j$ and the server $\mathcal{S}$ are $u = 0$, $v = 0$ and $w = 0$.

Proof of Proposition 1: Case 1 is true means the user $\mathcal{U}_i$ refuses to participate in the FL training. Therefore, it has no loss of privacy $Sens(\mathcal{D}_{\mathcal{U}_i})$ and energy consumption $T_{\mathcal{U}_i}$, and does not get the reward $Q_{\mathcal{U}_i}$ and subsidy $F_{\mathcal{U}_i}$ provided by the server $\mathcal{S}$. That means the specific values of $Sens(\mathcal{D}_{\mathcal{U}_i})$, $T_{\mathcal{U}_i}$, $Q_{\mathcal{U}_i}$ and $F_{\mathcal{U}_i}$ are zero. Inserting them into equation (27), we can get the following equation:

$$\frac{\partial E_{\mathcal{S}}}{\partial w} = T_S v - T_S = 0 \tag{29}$$

We find that v can only be one if $v \neq 0$. It is contradicted with our assumptions as shown in Fig. 3 so that v is zero. Then equation (29) can be simplified as

$$\frac{\partial E_{\mathcal{S}}}{\partial w} = -T_S = 0 \tag{30}$$

The equation (30) means the total energy expenditure of the server $\mathcal{S}$ is zero. It reflects the server does not participate in the FL training, so that w is zero. Therefore, the optimal strategies of the user $\mathcal{U}_i$, the agent $\mathcal{A}_j$ and the server $\mathcal{S}$ are $u = 0$, $v = 0$ and $w = 0$.

Proposition 2. If case 2 is true, the optimal strategies of the user $\mathcal{U}_i$, the agent $\mathcal{A}_j$ and the server $\mathcal{S}$ are $u = \frac{(Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}) T_S}{(Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i}) F_{\mathcal{A}_j} + [(\beta - 1 - \lambda) Q_{\mathcal{U}_i} + F_{\mathcal{U}_i} - T_S](T_{\mathcal{U}_i} - F_{\mathcal{U}_i})}$, $v = \frac{T_{\mathcal{U}_i} - F_{\mathcal{U}_i}}{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i}}$, $w = 1$.

Proof of Proposition 2: Case 2 is true means the user $\mathcal{U}_i$ performs the FL training in current iteration. Therefore, it not only has the loss of privacy $Sens(\mathcal{D}_{\mathcal{U}_i})$ and energy consumption $T_{\mathcal{U}_i}$, but also can get the reward $Q_{\mathcal{U}_i}$ or subsidy $F_{\mathcal{U}_i}$ provided by the server $\mathcal{S}$. In the meantime, equation (28) can be simplified as

$$\lambda Q_{\mathcal{U}_i} w = T_{\mathcal{A}_j} - F_{\mathcal{A}_j} \tag{31}$$

We discuss equation (31) in two cases. Case 3: $\lambda = 0$ and Case 4: $\lambda \neq 0$.

(1) Case 3 is true means there is a conflict between the case and our regulation. We set $\lambda > 0$ when we define the reward of the agent $\mathcal{A}_j$ that $Q_{\mathcal{A}_j} = \lambda Q_{\mathcal{U}_i} (\lambda > 0)$. Thus, case 3 is invalid for equation (31) and λ can only be nonzero.

Table 3
The frequently used notations in game model and analysis.

Notations	Description
u, v, w	The probability participating in the FL training of user $\mathcal{U}_i$, agent $\mathcal{A}_j$ and server $\mathcal{S}$
k_1, k_2	The coefficient reflecting the impact of privacy sensitivity degree and the benefits of data
k_3, k_4	The coefficient reflecting the impact of subsidy ratio of user $\mathcal{U}_i$ and agent $\mathcal{A}_j$
$Q_{\mathcal{U}_i}$, $Q_{\mathcal{A}_j}$, $Q_{\mathcal{S}}$	The benefit of user $\mathcal{U}_i$, agent $\mathcal{A}_j$ and server $\mathcal{S}$
λ, β	The coefficient reflecting the proportion of user's benefits $Q_{\mathcal{U}_i}$
$PO_{\mathcal{U}_i}$, $PO_{\mathcal{A}_j}$, $PO_{\mathcal{S}}$	The payoff of user $\mathcal{U}_i$, agent $\mathcal{A}_j$ and server $\mathcal{S}$
$F_{\mathcal{U}_i}$, $F_{\mathcal{A}_j}$	The subsidy of user $\mathcal{U}_i$ and agent $\mathcal{A}_j$
$T_{\mathcal{U}_i}$, $T_{\mathcal{A}_j}$, $T_{\mathcal{S}}$	The total energy consumption of user $\mathcal{U}_i$, agent $\mathcal{A}_j$ and server $\mathcal{S}$
$E_{\mathcal{U}_i}$, $E_{\mathcal{A}_j}$, $E_{\mathcal{S}}$	The mathematical expectation of user $\mathcal{U}_i$, agent $\mathcal{A}_j$ and server $\mathcal{S}$
$\delta_{\mathcal{U}_i}$, $\delta_{\mathcal{A}_j}$, $\delta_{\mathcal{S}}$	The discount factor of user $\mathcal{U}_i$, agent $\mathcal{A}_j$ and server $\mathcal{S}$

(2) Case 4 is true means the agent $\mathcal{A}_j$ participates in the FL training at current iteration. Therefore, it not only has the energy consumption $T_{\mathcal{A}_j}$, but also can get the reward $Q_{\mathcal{A}_j}$ or subsidy $F_{\mathcal{A}_j}$ provided by the server $\mathcal{S}$. Also, we can know that $\lambda > 0$ is based on our regulation.

According to equation (31), the probability of the server $\mathcal{S}$ is

$$w = \frac{T_{\mathcal{A}_j} - F_{\mathcal{A}_j}}{\lambda Q_{\mathcal{U}_i}} \tag{32}$$

and we can find that $w \neq 0$ since $T_{\mathcal{A}_j} > F_{\mathcal{A}_j}$. Also, it is the stagnation point of $E_{\mathcal{A}_j}$. According to equation (31), we know that $\frac{\partial T_{\mathcal{A}_j}}{\partial v} \geq 0$ if $w \geq \frac{T_{\mathcal{A}_j} - F_{\mathcal{A}_j}}{\lambda Q_{\mathcal{U}_i}}$. It implies that $E_{\mathcal{A}_j}$ is non-decreasing and the value of $E_{\mathcal{A}_j}$ increases as w becomes larger. The range of w is $0 \leq w \leq 1$, so that $E_{\mathcal{A}_j}$ takes the maximum value when $w = \frac{T_{\mathcal{A}_j} - F_{\mathcal{A}_j}}{\lambda Q_{\mathcal{U}_i}} = 1$.

Inserting $w = 1$ into equation (25), we can get the following equation:

$$\frac{\partial E_{\mathcal{U}_i}}{\partial u} = [Q_{\mathcal{U}_i} - F_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i})] v + F_{\mathcal{U}_i} - T_{\mathcal{U}_i} = 0 \tag{33}$$

By solving equation (33), the probability of the agent $\mathcal{A}_j$ is $v = \frac{T_{\mathcal{U}_i} - F_{\mathcal{U}_i}}{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i}}$.

Inserting $v = \frac{T_{\mathcal{U}_i} - F_{\mathcal{U}_i}}{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i}}$ and $w = 1$ into equation (27), we can get the following equation:

$$\begin{aligned} & \left\{ [(\beta - 1 - \lambda) Q_{\mathcal{U}_i} + F_{\mathcal{U}_i} - T_s] \left(\frac{T_{\mathcal{U}_i} - F_{\mathcal{U}_i}}{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i}} \right) + F_{\mathcal{A}_j} \right\} u \\ & = \frac{(Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}) T_S}{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i}} \end{aligned} \tag{34}$$

By solving equation (34), the probability of the user $\mathcal{U}_i$ is $u = \frac{(Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}) T_S}{(Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i}) F_{\mathcal{A}_j} + [(\beta - 1 - \lambda) Q_{\mathcal{U}_i} + F_{\mathcal{U}_i} - T_S](T_{\mathcal{U}_i} - F_{\mathcal{U}_i})}$.

u represents the probability of the user $\mathcal{U}_i$ participating in FL. From the payoff functions we can find that not only are the user's payoff $PO_{\mathcal{U}_i}$ related to u, but the payoffs of the agent $\mathcal{A}_j$ and server $\mathcal{S}$ are also related to u. If the probability u is greater or more users choose to participate in the training, the reward for all players will increase. Proposition 2 shows that the key to FL is to promote user participation while providing high-quality data and more privacy information. In the meantime, the payoffs of all players include the expenditure of local and communication computing. For their selfishness and rationality, they try to reduce their own costs of energy consumption.

4.2.4. The analysis of the nash equilibrium

From the two theorems above, we find that whether the user $\mathcal{U}_i$ performs in the FL training is a key factor in the single game model.

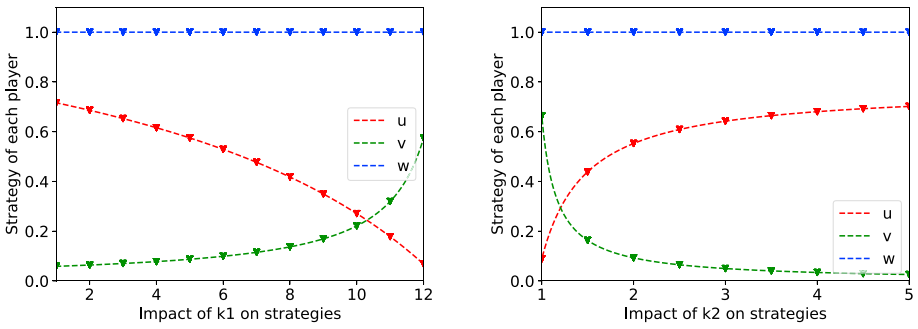

Fig. 4. Players' strategies. (a) Impact of k_1; (b) Impact of k_2.

$u = 0$ represents the user $\mathcal{U}_i$ refuses to participate in the FL training. In this case, neither the agent $\mathcal{A}_j$ nor the server $\mathcal{S}$ can benefit because the Local-FL model has not been updated. That means their rewards are zero and possibly negative. It is reasonable that the agent $\mathcal{A}_j$ and server $\mathcal{S}$ prefer not to update the models since they are selfish and rational in reality, which is consistent with $v = 0, w = 0$. The optimal strategy is $u = 0, v = 0, w = 0$.

$u \neq 0$ denotes the user $\mathcal{U}_i$ participating in the current iteration of FL training. During the process of FL training, the higher Local-FL model QoS $QoS_{\mathcal{L}_{\mathcal{U}_i}}$ means the result of the FL task $\mathcal{T}$ is close to the server's requirement in reality. It implies that the server expects the user uses high-quality data or more privacy information to train the Local-FL model, which results that more reward is required for the user $\mathcal{U}_i$ in the meantime. Then the Local-FL model is also aggregated and transmitted by the agent $\mathcal{A}_j$, so that it is reasonable that the server wishes to receive more Agent-FL models. The key to realizing this wish is to maximize the agent's payoff $PO_{\mathcal{A}_j}$, which is referred to the user's payoff $PO_{\mathcal{U}_i}$. This is consistent with $w = 1$ which can maximize the payoff $PO_{\mathcal{U}_i}$ as shown in Algorithm 2 in this case.

Algorithm 2 Maximize user's payoff $PO_{\mathcal{U}_i}$

if Not to participate in the FL training **then**
  $u = 0$
  $PO_{\mathcal{U}_i} = 0$
else
  $u = \frac{(Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i})T_{\mathcal{S}}}{(Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i})F_{\mathcal{A}_j} + [(\beta-1-\lambda)Q_{\mathcal{U}_i} + F_{\mathcal{U}_i} - T_s](T_{\mathcal{U}_i} - F_{\mathcal{U}_i})}$
  Decide the noise intensity $\epsilon_{\mathcal{U}_i}$
  Compute the privacy sensitivity degree $Sens(\mathcal{D}_{\mathcal{U}_i}) =$
  $\sum_{t=0}^{\infty} \sum_{\omega^s \in W_s} (\varepsilon_{\mathcal{U}_i})^t$
  $\left| P_r\left[D^t = \omega_s \mid D^0 = \omega_0\right] - P_r\left[D^t = \omega_s\right] \right|$
  Compute the local computation
  $B_{\mathcal{U}_i} = \kappa_{\mathcal{U}_i} C_{\mathcal{D}_{\mathcal{U}_i k}} \mathcal{D}_K T \left(QoS_{P_{\mathcal{A}_j}}, QoS_{L_{\mathcal{U}_i}}\right) f_{\mathcal{U}_i}^2$
  Compute the communication computation
  $C_{\mathcal{U}_i} = T_{\mathcal{U}_i} \cdot \left(1 + QoS_{L_{\mathcal{U}_i}}\right)$
  Compute the payoff $PO_{\mathcal{U}_i}$
  if the payoff is lower than 0 **then**
    adjust the noise intensity $\epsilon_{\mathcal{U}_i}$ or
    not to participate in next iteration
  elseRemain the noise intensity $\epsilon_{\mathcal{U}_i}$
  end if
end if

According to $w = 1$ and equation (33), we can find the optimal strategy is $u = \frac{(Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i})T_{\mathcal{S}}}{(Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i})F_{\mathcal{A}_j} + [(\beta-1-\lambda)Q_{\mathcal{U}_i} + F_{\mathcal{U}_i} - T_s](T_{\mathcal{U}_i} - F_{\mathcal{U}_i})}$, $v = \frac{T_{\mathcal{U}_i} - F_{\mathcal{U}_i}}{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i}}$, $w = 1$.

Since the range of probabilities is from zero to one, we can denote u and v by the following formula

$$\begin{cases} 0 \le u = \dfrac{(Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i})T_{\mathcal{S}}}{(Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i})F_{\mathcal{A}_j} + R_1 \cdot (T_{\mathcal{U}_i} - F_{\mathcal{U}_i})} \le 1 \\ 0 \le v = \dfrac{T_{\mathcal{U}_i} - F_{\mathcal{U}_i}}{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i}} \le 1 \end{cases} \tag{35}$$

where $R_1 = [(\beta - 1 - \lambda)Q_{\mathcal{U}_i} + F_{\mathcal{U}_i} - T_s]$.

So that

$$\begin{cases} Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i} > 0 \\ (\beta - 1 - \lambda)Q_{\mathcal{U}_i} + F_{\mathcal{U}_i} - T_{\mathcal{S}} > 0 \end{cases} \tag{36}$$

If the user $\mathcal{U}_i$ decides to perform in the FL training, its payoff $PO_{\mathcal{U}_i}$ is one of $Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}$, $F_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}$, $F_{\mathcal{U}_i} - T_{\mathcal{U}_i}$ and 0. Since $F_{\mathcal{U}_i} < T_{\mathcal{U}_i}$, $F_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}$, $F_{\mathcal{U}_i} - T_{\mathcal{U}_i}$ are both less than 0. If $Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}$ is less than 0, there exists the pure strategy Nash equilibrium instead of the mixed strategy Nash equilibrium. This is no accord to our assumptions. Thus, $Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}$ should be higher than zero. And the constrain is $Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - F_{\mathcal{U}_i} > 0$, because $F_{\mathcal{U}_i} < T_{\mathcal{U}_i}$.

Similarly, if the server $\mathcal{S}$ decides to participate in the FL training, its payoff $PO_{\mathcal{S}}$ is one of $(\beta - 1 - \lambda)Q_{\mathcal{U}_i} - T_{\mathcal{S}}$, $-F_{\mathcal{A}_j} - F_{\mathcal{U}_i}$, $F_{\mathcal{A}_j} - F_{\mathcal{U}_i} - T_{\mathcal{S}}$, $-F_{\mathcal{U}_i}$, $-T_{\mathcal{S}}$ and 0. Since $w = 1$, the payoff $PO_{\mathcal{S}}$ only can be $(\beta - 1 - \lambda)Q_{\mathcal{U}_i} - T_{\mathcal{S}}$ while it is the maximum payoff of server $\mathcal{S}$ according to the analysis above. If $(\beta - 1 - \lambda)Q_{\mathcal{U}_i} - T_{\mathcal{S}}$ is less than zero, there exists the pure strategy Nash equilibrium instead of the mixed strategy Nash equilibrium. Thus, $(\beta - 1 - \lambda)Q_{\mathcal{U}_i} - T_{\mathcal{S}}$ is higher than zero. And the constrain is $(\beta - 1 - \lambda)Q_{\mathcal{U}_i} + F_{\mathcal{U}_i} - T_{\mathcal{S}} > 0$.

4.3. Repeated game model and analysis

Note that FL is a repeated and limited iteration process, meaning that each player considers the overall payoff of learning when adopting strategies. The social externality also implies that the higher-payoff strategies are easily spread among players.

Therefore, we perform the repeated game model to analyze all players' strategies change with iterations. We consider many factors such as privacy leakage, reward and subsidy, QoS and energy consumption based on the single interaction. Also, we design the punishment mechanism to encourage more users according to the conclusion above, and the probability u is a key factor. We analyze the repeated game model in two cases: establishing a punishment mechanism and not.

4.3.1. Repeated game model

From the single game analysis, we find that there exists a pure strategy Nash equilibrium when $u = 0, v = 0, w = 0$. In the literature [42], we can know that the only subgame perfect Nash equilibrium solution for repeated game is for each player to adopt the Nash equilibrium strategy of the original game. Therefore, the corresponding outcome is that the user $\mathcal{U}_i$, the agent $\mathcal{A}_j$ and the server $\mathcal{S}$ are reluctant to participate in the FL training.

In addition, there is a mixed strategy Nash equilibrium when $u =$

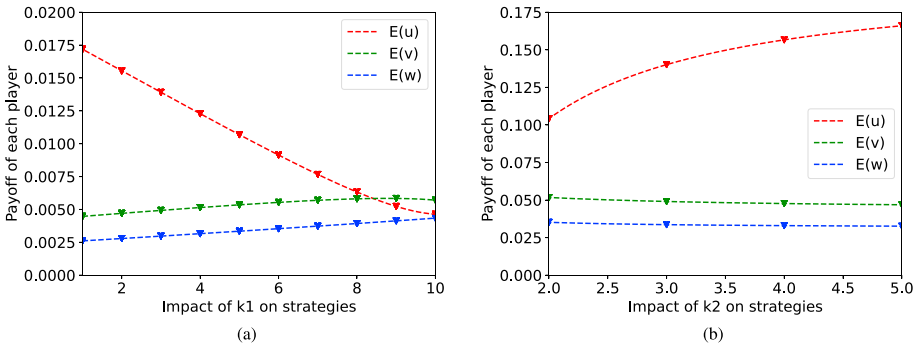

Fig. 5. Players' payoffs. (a) Impact of k_1; (b) Impact of k_2.

$\frac{(Q_{\mathcal{U}_i}-k_1 Sens(\mathcal{D}_{\mathcal{U}_i})-T_{\mathcal{U}_i})T_S}{(Q_{\mathcal{U}_i}-k_1 Sens(\mathcal{D}_{\mathcal{U}_i})-F_{\mathcal{U}_i})F_{\mathcal{A}_j}+[(\beta-1-\lambda)Q_{\mathcal{U}_i}+F_{\mathcal{U}_i}-T_S](T_{\mathcal{U}_i}-F_{\mathcal{U}_i})}$, $v = \frac{T_{\mathcal{U}_i}-F_{\mathcal{U}_i}}{Q_{\mathcal{U}_i}-k_1 Sens(\mathcal{D}_{\mathcal{U}_i})-F_{\mathcal{U}_i}}$, $w = 1$.

Once it comes to the repeated game, we need to weigh the benefits of different strategies over different stages. To resolve this issue, we introduce a discount factor δ to discount the future payoff to the current stage.

If the player's payoff in each stage is describe as $\eta_1, \eta_2, \ldots, \eta_y$, the total payoff can be denoted as

$$\eta = \eta_1 + \delta\eta_2 + \delta^2\eta_3 + \cdots + \delta^{y-1}\eta_y = \sum_{z=1}^{\infty}\delta^{z-1}\eta_z \tag{37}$$

We can fine that the discount factor reflects the player's payoff

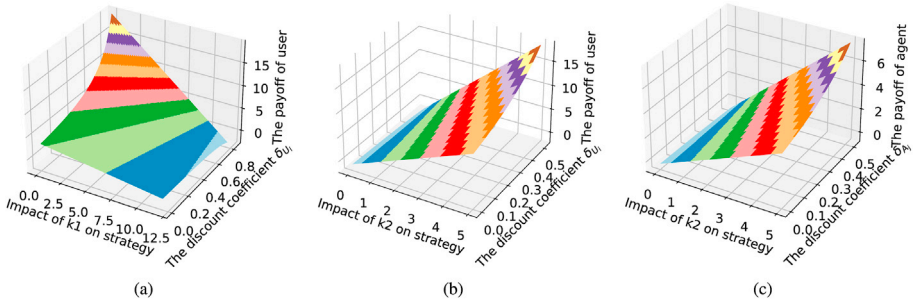

Fig. 6. Player's payoff without punishment mechanism. (a) Impact of k_1 and $\delta_{\mathcal{U}_i}$ on user's payoff $PO_{\mathcal{U}_i}$; (b) Impact of k_2 and $\delta_{\mathcal{U}_i}$ on user's payoff $PO_{\mathcal{U}_i}$; (c) Impact of k_2 and $\delta_{\mathcal{A}_j}$ on agent's payoff $PO_{\mathcal{A}_j}$.

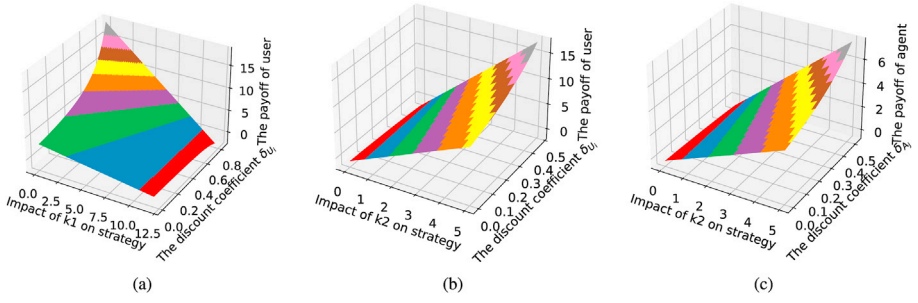

Fig. 7. Player's payoff with the punishment mechanism. (a) Impact of k_1 and $\delta_{\mathcal{U}_i}$ on user's payoff $PO_{\mathcal{U}_i}$; (b) Impact of k_2 and $\delta_{\mathcal{U}_i}$ on user's payoff $PO_{\mathcal{U}_i}$; (c) Impact of k_2 and $\delta_{\mathcal{A}_j}$ on agent's payoff $PO_{\mathcal{A}_j}$.

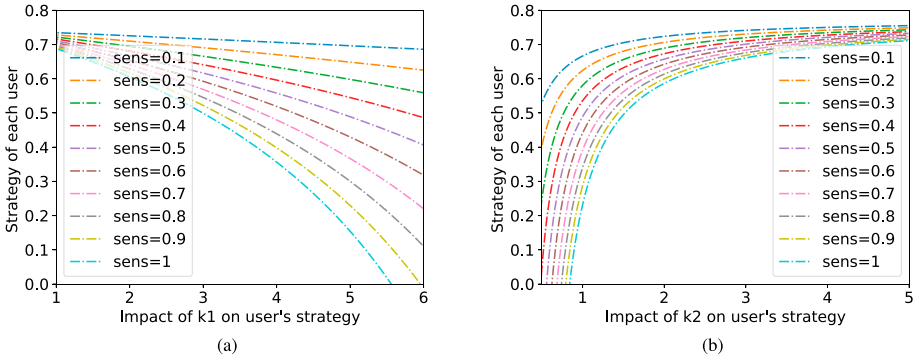

Fig. 8. User's payoff. (a) Impact of k_1 and $Sens(\mathcal{D}_{\mathcal{U}_i})$; (b) Impact of k_2 and $Sens(\mathcal{D}_{\mathcal{U}_i})$.

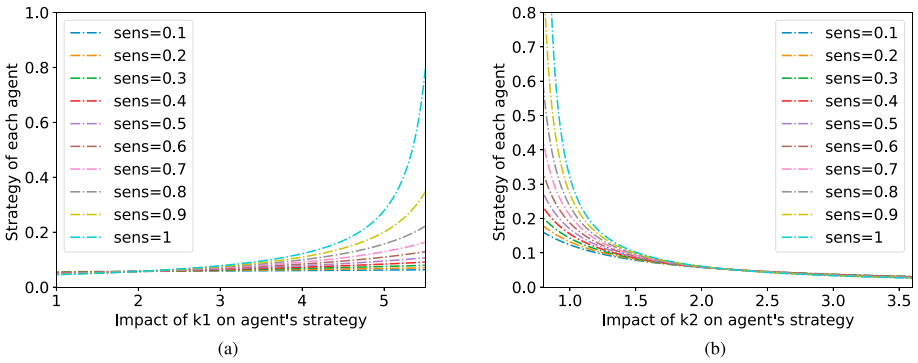

Fig. 9. Agent's payoff. (a) Impact of k_1 and $\mathcal{S}ens(\mathcal{D}_{\mathcal{U}_i})$; (b) Impact of k_2 and $\mathcal{S}ens(\mathcal{D}_{\mathcal{U}_i})$.

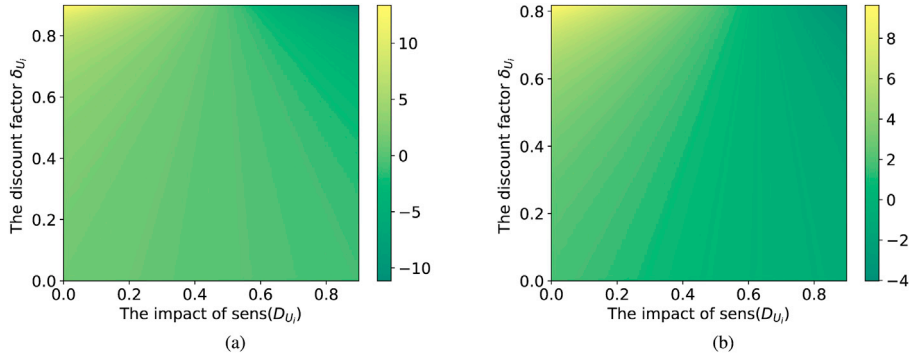

Fig. 10. Impact of $\mathcal{S}ens(\mathcal{D}_{\mathcal{U}_i})$ and $\delta_{\mathcal{U}_i}$ on user's payoff $PO_{\mathcal{U}_i}$. (a) without the punishment mechanism; (b) with the punishment mechanism.

preferences over time, and higher δ means the player pays more attention to the later payoff in the game model. On the contrary, lower δ indicates that the player focuses more on the current payoff. Hereinafter, we use $\delta_{\mathcal{U}_i}, \delta_{\mathcal{A}_j}, \delta_{\mathcal{S}}$ to mark the influence of δ on each user $\mathcal{U}_i$, each agent $\mathcal{A}_j$ and server $\mathcal{S}$. In the existence of mixed strategy Nash equilibrium, we can use equation (37) to calculate all players' payoff.

We also know the ideal social norm is that all players participate in the FL training in reality, with higher rewards, lower energy expenditure and loss of privacy.

If all players' actions correspond to the ideal social norm, their strategies paths can be described as

$$(1,1,1) \rightarrow (1,1,1) \rightarrow (1,1,1) \rightarrow \cdots \rightarrow end \tag{38}$$

Then the payoff functions for the user $\mathcal{U}_i$, the agent $\mathcal{A}_j$ and server $\mathcal{S}$ are

$$\begin{aligned} PO'_{\mathcal{U}_i} &= [Q_{\mathcal{U}_i} - k_1 \mathcal{S}ens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}](1 + \delta_{\mathcal{U}_i} + \delta^2_{\mathcal{U}_i} + \delta^3_{\mathcal{U}_i} + \cdots) \\ &= \frac{Q_{\mathcal{U}_i} - k_1 \mathcal{S}ens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}}{1 - \delta_{\mathcal{U}_i}} \end{aligned} \tag{39}$$

$$\begin{aligned} PO'_{\mathcal{A}_j} &= [\lambda Q_{\mathcal{U}_i} - T_{\mathcal{A}_j}](1 + \delta_{\mathcal{A}_j} + \delta^2_{\mathcal{A}_j} + \delta^3_{\mathcal{A}_j} + \cdots) \\ &= \frac{\lambda Q_{\mathcal{U}_i} - T_{\mathcal{A}_j}}{1 - \delta_{\mathcal{A}_j}} \end{aligned} \tag{40}$$

$$\begin{aligned} PO'_{\mathcal{S}} &= [(\beta - 1 - \lambda) Q_{\mathcal{U}_i} - T_{\mathcal{S}}](1 + \delta_{\mathcal{S}} + \delta^2_{\mathcal{S}} + \delta^3_{\mathcal{S}} + \cdots) \\ &= \frac{(\beta - 1 - \lambda) Q_{\mathcal{U}_i} - T_{\mathcal{S}}}{1 - \delta_{\mathcal{S}}} \end{aligned} \tag{41}$$

However, it is unrealistic that the players are rational and selfish. Also, the loss of privacy is positively correlated with the QoS. According to the analysis above, we propose a social norm that conforms to the player's attitude and real situation. The user $\mathcal{U}_i$ always performs in the FL training n_1−*th* consecutive times until its payoff $PO'_{\mathcal{U}_i}$ is less than expected, and then stops one iteration to adjust its strategy such as the noise intensity $\epsilon_{\mathcal{U}_i}$. The agent always aggregates and uploads the Agent-FL model n_2−*th* consecutive times until its payoff is less than expected, and then it stops one iteration to adjust the strategy such as the proportion λ. The server $\mathcal{S}$ updates the Global-FL model n_3−*th* consecutive times until its payoff is less than expected, and then it stops one iteration to adjust the strategy such as the provided reward $Q_{\mathcal{U}_i}$. Our social norm can be applied to any scenarios with different values of n_1, n_2 and n_3. To specifically analyze the repeated game model, we set $n_1 = 1$, $n_2 = 1$ and $n_3 = 1$.

4.3.2. Repeated game analysis

We mainly analyze the deviations on each user's payoff $PO'_{\mathcal{U}_i}$. If the player does not adhere to the ideal social norm, the following situations probably occur.

We reasonably assume that the first case

$$(1,1,1) \rightarrow (0,0,0) \rightarrow (0,0,0) \rightarrow \cdots \rightarrow end \tag{42}$$

and the payoff function for the user $\mathcal{U}_i$ is

$$PO_{\mathcal{U}_i}(1) = Q_{\mathcal{U}_i} - k_1 \mathcal{S}ens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i} \tag{43}$$

The second case is denoted as

$$(1,1,1) \rightarrow (1,0,0) \rightarrow (0,0,0) \rightarrow (0,0,0) \rightarrow \cdots \rightarrow end \tag{44}$$

and the payoff function for the user $\mathcal{U}_i$ is

$$PO_{\mathcal{U}_i}(2) = [Q_{\mathcal{U}_i} - k_1 \mathcal{S}ens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}] + (F_{\mathcal{U}_i} - T_{\mathcal{U}_i})\delta_{\mathcal{U}_i} \tag{45}$$

According to the analysis above, they have the following relationship

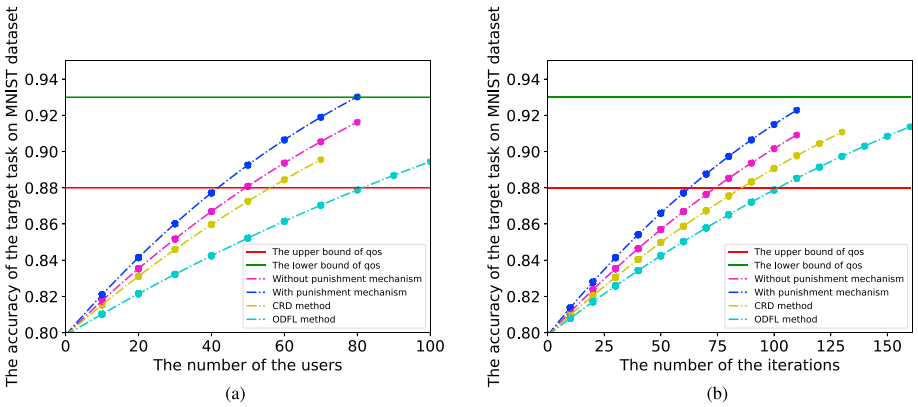

Fig. 11. Different strategies on the model qos of MNIST dataset. (a) impact of the number of users; (b) impact of the number of iterations.

$$\begin{cases} PO_{\mathcal{U}_i}(1) < PO'_{\mathcal{U}_i} \\ PO_{\mathcal{U}_i}(2) < PO'_{\mathcal{U}_i} \end{cases} \tag{46}$$

Inserting the specific value, we can know that

$$\begin{cases} R_2 < \dfrac{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}}{1 - \delta_{\mathcal{U}_i}} \\ R_2 + (F_{\mathcal{U}_i} - T_{\mathcal{U}_i})\delta_{\mathcal{U}_i} < \dfrac{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}}{1 - \delta_{\mathcal{U}_i}} \end{cases} \tag{47}$$

where $R_2 = Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}$. By solving equation (47), the discount factor $\delta_{\mathcal{U}_i}$ of the user $\mathcal{U}_i$ is $\delta_{\mathcal{U}_i} < \frac{F_{\mathcal{U}_i} + k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - Q_{\mathcal{U}_i}}{F_{\mathcal{U}_i} - T_{\mathcal{U}_i}}$.

Only when the discount factor $\delta_{\mathcal{U}_i}$ satisfies the above range, can we ensure that the user $\mathcal{U}_i$ does not actively deviate from social norm. It also ensures that the user $\mathcal{U}_i$ considers its payoff $PO'_{\mathcal{U}_i}$ with the privacy sensitivity degree $Sens(\mathcal{D}_{\mathcal{U}_i})$, QoS $L_{\mathcal{U}_i}$ and energy consumption $T_{\mathcal{U}_i}$. Therefore, the data privacy of the user can be protected while the energy consumption including local and communication computing can be saved.

Actually, the user $\mathcal{U}_i$ may forgo long-term payoff for short-term payoff to achieve hitchhiking purpose in reality. To solve this problem, we design the punishment mechanism to encourage more users to follow the social norm. We analyze a scenario where the user $\mathcal{U}_i$ submits the lower QoS $\mathcal{U}_i$ of $\mathcal{L}_{\mathcal{U}_i}$ to the agent, while it should be punished and have some restraint.

We add a new social norm based on real situation. We assume that the agent $\mathcal{A}_j$ will take a punishment mechanism that subtracts the user's reward $Q_{\mathcal{U}_i}$, if the contribution of Local-FL model ρ is lower than average contribution ρ_1 in the agent's group for n_3−*th* consecutive times. To analyze the situation specifically, we set $n_3 = 2$ and denote the

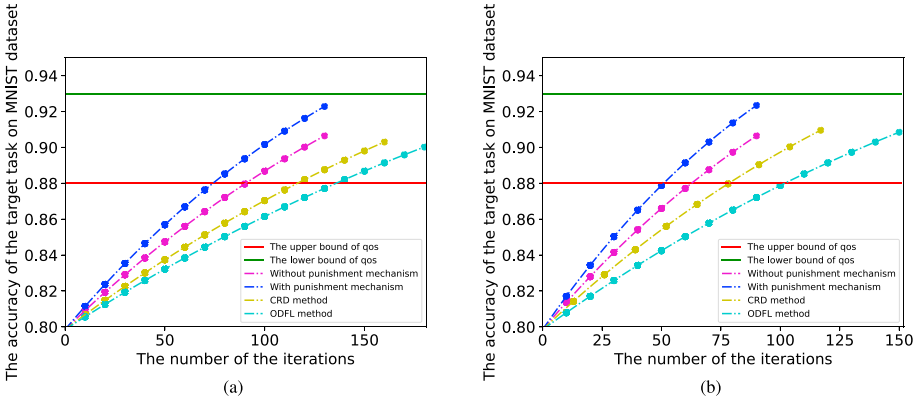

Fig. 12. Different strategies on the model qos of MNIST dataset. (a) $Sens(\mathcal{D}_{\mathcal{U}_i}) = 0.1$; (b) $Sens(\mathcal{D}_{\mathcal{U}_i}) = 0.5$.

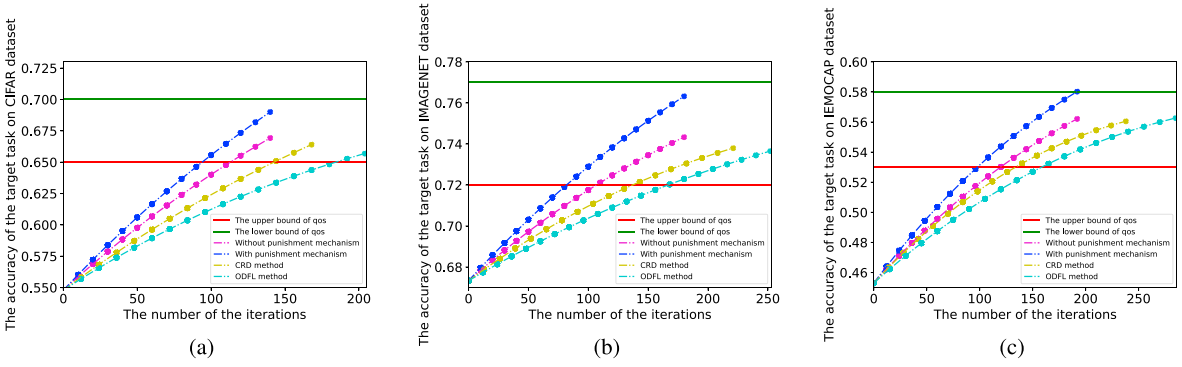

Fig. 13. Different strategies on the model qos of different datasets. (a) CIFAR-100 dataset; (b) ImageNet dataset; (c) IEMOCAP dataset.

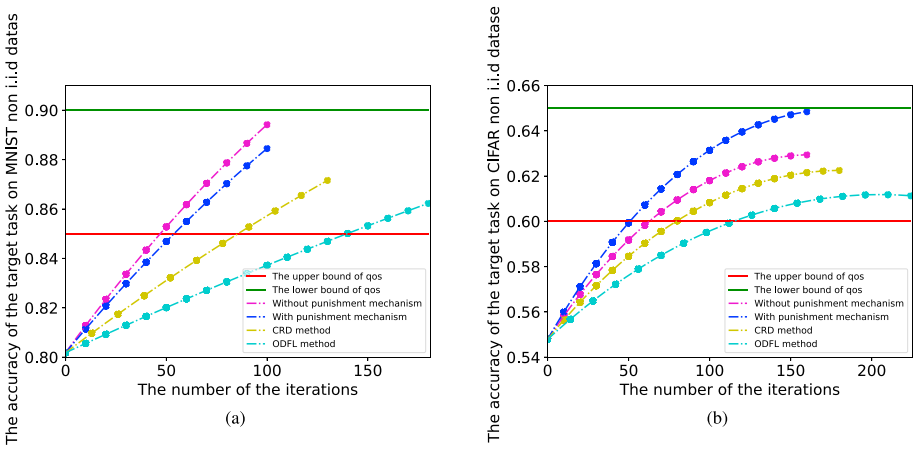

Fig. 14. Different strategies on the model qos of non-i.i.d datasets. (a) MNIST dataset; (b) CIFAR-100 dataset.

punishment cost as $PC_{\mathcal{U}_i}$ for the user $\mathcal{U}_i$.

In this case, the third case is

$$(1,1,1) \to (0,0,0) \to (1,1,1) \to (1,1,1) \overset{PC_{\mathcal{U}_i}}{\to} \cdots \to end \tag{48}$$

and the payoff function is

$$\begin{aligned} PO_{\mathcal{U}_i}(3) &= [Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}](1+\delta_{\mathcal{U}_i}^2) \\ &+[Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i} - PC_{\mathcal{U}_i}]\frac{\delta_{\mathcal{U}_i}^3}{1-\delta_{\mathcal{U}_i}} \end{aligned} \tag{49}$$

Since the relationship between $PO_{\mathcal{U}_i}(3)$ and $PO'_{\mathcal{U}_i}$ is $PO_{\mathcal{U}_i}(3) < PO'_{\mathcal{U}_i}$, the specific equation is represented as

$$\begin{aligned} &[Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}](1+\delta_{\mathcal{U}_i}^2) + [Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) \\ &-T_{\mathcal{U}_i} - PC_{\mathcal{U}_i}]\frac{\delta_{\mathcal{U}_i}^3}{1-\delta_{\mathcal{U}_i}} < \frac{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}}{1-\delta_{\mathcal{U}_i}} \end{aligned} \tag{50}$$

By solving from equation (50), we can find that when $\frac{PC_{\mathcal{U}_i}}{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}} < \frac{1}{4}$, the discount factor $\delta_{\mathcal{U}_i}$ is $\delta_{\mathcal{U}_i} < \frac{\left(1-\sqrt{1-\frac{4\cdot PC_{\mathcal{U}_i}}{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}}}\right) * [Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}]}{2\cdot PC_{\mathcal{U}_i}}$.

The user $\mathcal{U}_i$ does not actively deviate from social norm only when the discount factor δ_U satisfies the range. Therefore, not only can the data privacy be protected and the energy consumption be saved, but also the QoS will update more stably compared with no punishment mechanism from a long-term perspective.

In summary, the notations of the game model are denoted in Table 3.

4.4. Incentive mechanism design

From the repeated game, we can obtain social norms that satisfy the Nash equilibrium. Since users are rational, they will decide whether to participate in the FL training based on their payoffs. That is, they use Algorithm 2 to calculate their own discount factor $\delta_{\mathcal{U}_i}$ to maximize their own payoffs while ensuring the privacy of the individual is compromised to an acceptable degree. In considering whether the addition of punishment mechanisms is allowed in real applications, our scenarios can be classified as follows.

(a) The social norm without the punishment mechanism.

Case 1. $\delta_{\mathcal{U}_i} < \frac{F_{\mathcal{U}_i} + k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - Q_{\mathcal{U}_i}}{F_{\mathcal{U}_i} - T_{\mathcal{U}_i}}$: The users are willing to participate in the FL training and paying more attention to the whole training process. Also, the users tend to provide the high-quality data on the premise of ensuring privacy.

Case 2. $\delta_{\mathcal{U}_i} \geq \frac{F_{\mathcal{U}_i} + k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - Q_{\mathcal{U}_i}}{F_{\mathcal{U}_i} - T_{\mathcal{U}_i}}$: The users are reluctant to participate in the FL training since his/her payoff is deviated from his/her expectation.

(b) The social norm with the punishment mechanism.

Case 3. $\delta_{\mathcal{U}_i} < \frac{\left(1-\sqrt{1-\frac{4\cdot PC_{\mathcal{U}_i}}{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}}}\right) * [Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}]}{2\cdot PC_{\mathcal{U}_i}}$: The punishment mechanism will supervise user's behavior. As a result, the user maintains more rational behavior for the long-term FL training. It also can motivate more users to use more high-quality data on the FL training.

Case 4. $\delta_{\mathcal{U}_i} \geq \frac{\left(1-\sqrt{1-\frac{4\cdot PC_{\mathcal{U}_i}}{Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}}}\right) * [Q_{\mathcal{U}_i} - k_1 Sens(\mathcal{D}_{\mathcal{U}_i}) - T_{\mathcal{U}_i}]}{2\cdot PC_{\mathcal{U}_i}}$: The users refuse to participate because its payoff is lower than its expectation.

Based on the above social norms analysis, we can design the incentive mechanisms to encourage more users to provide high-quality data in the FL training, resulting in a more stable FL system for all players.

5. Numerical and simulation experiments

In this section, we conduct the numerical analysis and simulation experiments to validate the proposed game models. All experiments are implemented in Python 3.8 and run on Nvidia Tesla GPU, with 16 GB of Graphics Card.

5.1. Numerical analysis

For our simulations, we deploy $N = 100$ users, $M = 5$ agents with one server in the FL system. For Local-FL model, the QoS $QoS_{L_{\mathcal{U}_i}}$ is from 0.2 to 0.6 uniformly. The privacy sensitivity degree is set as $Sens(\mathcal{D}_{\mathcal{U}_i}) = 0.3$ for each user $\mathcal{U}_i$ [42]. The effective switched capacitance is $\kappa = 10^{-28}$ [43], the parameter is $C = 10^4$ and the computation capacity is $f = 2$ GHz [32] for each player. The allocated bandwidth is $B = 30$ MHz, which is uniform for each user $\mathcal{U}_i$ and agent $\mathcal{A}_j$. The model size is $e_{size} = 30$ kbits, the transmit power is 10 dB and the spectral density of Gaussian noise is $N = -175$ dBm/Hz.

5.1.1. Numerical analysis of the single game

From the Nash equilibrium and restraints in section 4, we find that the possibilities u, v of each user $\mathcal{U}_i$ and agent $\mathcal{A}_j$ are related to k_1, k_2. Therefore, we analyze how the strategies of each player change as the coefficients vary.

Fig. 4(a) shows the relationship between the probability u, v and the coefficient k_1 of each user $\mathcal{U}_i$, and the higher k_1 reflects the smaller u. It is reasonable that the larger $Sens(\mathcal{D}_{\mathcal{U}_i})$ means each user's data includes more sensitive information and it is reluctant to participate in the

training. In addition, each agent $\mathcal{A}_j$ is willing to participate because the profit $Q(QoS_{L_{\mathcal{U}_i}} + \mathcal{L}_{\mathcal{U}_i})$ of model is higher than before in reality. It means that the benefit of each agent increases in this case, so that the probability v also increases.

Fig. 4(b) shows the relationship between the probability u, v and the coefficient k_2 of each user $\mathcal{U}_i$. We can find that the higher k_2 results in a higher u and a smaller v. The phenomenon is caused because the higher k_2 can produce more positive impact for each user $\mathcal{U}_i$, which increases its benefit $Q_{\mathcal{U}_i}$ and motivates more users to use high-quality data to participate in FL training because of the external sociality. It also results that more Local-FL models are aggregated and the energy consumption will grow for the agent $\mathcal{A}_j$. Thus, u increases but v decreases in this case.

The payoffs for the three players, from Fig. 5(a)-5(b), show that no matter how the coefficients k_1 and k_2 change, their payoffs are almost unchanged. This is because the players' payoffs are so close to constant that they have no incentive to adjust their strategy when there is an optimal strategy. The results also reveal the meanings of the Nash equilibrium.

5.1.2. Numerical analysis of the repeated game

Based on the Nash equilibrium in the single game, we divide the repeated game into two situations where each user $\mathcal{U}_i$ is punished in section 4. Under the $\delta_{\mathcal{U}_i}$ range, we analyze how changes in $\delta_{\mathcal{U}_i}$ and k_1, k_2 affecting the payoff $PO_{\mathcal{U}_i}$.

Fig. 6(a) and Fig. 7(a) show the relationship among user's payoff $PO_{\mathcal{U}_i}$, the coefficient that represents the privacy leakage impact k_1 and the discount factor $\delta_{\mathcal{U}_i}$. It is obvious that the user $\mathcal{U}_i$ who pays more attention to latter gets a higher payoff than before. That means the user $\mathcal{U}_i$ takes into account that FL is a long-term process and not focused on current payoff. Compared with Figs. 6(a) and Figure 7(a), we also can find that the payoff change with punishment mechanism is less than no punishment mechanism. It is because the user $\mathcal{U}_i$ is rational and considers the punishment mechanism effects while deciding on a particular strategy. That means establishing a punishment mechanism is necessary, and ensuring that the user's behavior fits within social norms.

Figs. 6(b) and Figure 7(c) show the relationship among each agent's and user's payoff, the coefficient that represents the profit $Q_{\mathcal{U}_i}$ impact k_2 and the discount factor $\delta_{\mathcal{U}_i}$. Compared with Figs. 6(b), 7(b) and 6(c) and Figure 7(c), there is a common phenomenon that the player with more concern for long-term training process has higher payoff. It indicates the player to maximize its payoff will take measures that focus on long-term training in reality. In addition, the payoff trends to change slowly with punishment mechanism, which results in a stable FL system with social norms.

Those experiments prove that the proposed social norms can make players pay more attention to the training latter, and motivate the players to engage in positive FL training with high-quality data while reducing the negative impact and positive impact. In the meantime, establishing a punishment mechanism is important in encouraging FL users to be more concerned about whether their behaviors are consistent with social norms. Their payoffs can be maximized only when their behaviors are "rational". Establishing a punishment mechanism also provides a stable environment for FL system and it is meaningful for the training process.

5.1.3. Numerical analysis of the privacy sensitivity degree

We conduct the experiments with different privacy sensitivity degree $Sens(\mathcal{D}_{\mathcal{U}_i})$ to observe the strategy of each user $\mathcal{U}_i$ and agent $\mathcal{A}_j$, and consider the coefficients k_1 and k_2 change in the FL training.

As shown in Fig. 8(a) and (b), we can find a larger $Sens(\mathcal{D}_{\mathcal{U}_i})$ curve with faster descent speed, which only matches smaller k_1. In the meantime, higher privacy sensitivity degree $Sens(\mathcal{D}_{\mathcal{U}_i})$ requires a higher k_2 to improve the user's payoff $PO_{\mathcal{U}_i}$, so as to ensure it is willing to participate in the FL training. It is realistic that the user is reluctant to lose more private information in FL training, and refuses the privacy information that is disclosed by the Local-FL model to have a significant negative impact on their payoff. Correspondingly, that means it requires smaller $Sens(\mathcal{D}_{\mathcal{U}_i})$ and k_1, higher k_2 in the game model. For each agent $\mathcal{A}_j$, the higher $Sens(\mathcal{D}_{\mathcal{U}_i})$ can promote its activity to participate in the FL training. It is reasonable that the agent's benefit is positively correlated with the profit of $Sens(\mathcal{D}_{\mathcal{U}_i})$. From Fig. 9(a), the higher k_1 has a greater impact of the user's privacy and can provide more profit.

We also observe the user's payoff $PO_{\mathcal{U}_i}$ changes with different privacy sensitivity degrees $Sens(\mathcal{D}_{\mathcal{U}_i})$ and the discount factor $\delta_{\mathcal{U}_i}$. It is obvious that the user's payoff $PO_{\mathcal{U}_i}$ decrease with the increasing of $Sens(\mathcal{D}_{\mathcal{U}_i})$ as shown in Fig. 10(a) and Fig. 10(b). Also, the user who focuses more on the training latter has a higher payoff than others. In addition, the users's payoff $PO_{\mathcal{U}_i}$ with the punishment mechanism outperforms than no punishment mechanism. That is because our punishment mechanism regulates user behavior and keeps them within a more rational range.

5.2. Simulation evaluation

We evaluate our method on four different datasets, which are MNIST [44], CIFAR-100 [45], ImageNet [46] and IEMOCAP [47]. MNIST is a handwritten numbers dataset containing 60000 training images and 10000 testing images. CIFAR-100 contains 50000 training images and 10000 testing images, while ImageNet contains 14,197,122 color images. We identify animals as our task in this experiment. And IEMOCAP is a multimodal emotion analysis dataset containing approximately 12 h of audiovisual data. Specifically, we use the ResNet-110 [48] to train MNIST and CIFAR-100 with small sample sizes, and ResNet-50 [48] to train ImageNet and IEMOCAP with large sample sizes.

Baseline: Since existing gaming approaches in federated learning fall into two main groups: One is to encourage users to trade-off accuracy and privacy through incentive mechanisms, resulting in more active participation in federated learning [28,49]. The other is to optimize the iterations of federation learning and then find the optimal iterations to save more energy [37]. To verify effectiveness, we compared our method with ODFL and CRD strategies, representative of two categories. ODFL [28] is a strategy that motivates more users to participate in the training through designing a two-stage Stackelberg game. CRD [37] is a communication optimization strategy that can find a trade-off between the computational complexity and the convergence performance.

5.2.1. Computational energy consumption

The number of users involved in FL training determines the number of local model training in this whole system, which indirectly affects the computational energy consumption. To more clearly observe the relationship between the number of participating users and model accuracy on MNIST, we fix the privacy sensitivity degree $Sens(\mathcal{D}_{\mathcal{U}_i}) = 0.3$ and the number of iterations $I = 80$. As shown in Fig. 11(a), our method requires only 80 participating users to obtain a high model accuracy. ODFL method encourages more users to participate in the learning and ignores the local model with lower QoS does not contribute enough to the global model. A large number of low-QoS participating users are required to train their own local models, which can make the computational energy consumption even larger. The CRD method is not optimized at the user incentive level, so it may result in an insufficient number of participating users, which leads to low overall FL model accuracy.

Our proposed method encourages users to submit high-quality data, thereby achieving higher accuracy rates with less user involvement. This means that fewer users perform local model computations throughout the FL training, thus reducing computational energy consumption. Specifically, punishment mechanisms require higher QoS for data and require fewer participating users compared to non-punishment methods, thus reducing computational energy consumption.

5.2.2. Communication energy consumption

In addition to the number of participating users affecting the communication energy between users and agents, the number of itera-

tions is also a significant factor in the global communication energy consumption. We set a fixed privacy sensitivity degree $Sens(\mathcal{D}_{\mathcal{U}_i}) = 0.3$ and the number of users $N = 60$ to see more clearly the impact of the iteration number on the accuracy of the model. As shown in Fig. 11(b), the ODFL approach mainly motivates more users to participate in training and does not optimize the model convergence speed. Thus, its model accuracy improves slowly with the number of iterations and grows slowly with later iterations. The CRD method allows users to perform multiple training rounds locally before uploading to the server for aggregation. The CRD method requires only approximately 130 rounds of iterations to achieve comparable model accuracy of 160 rounds for the ODFL method. The CRD method reduces the number of iterations, thus reducing communication energy consumption.

Our method further reduces the number of iterations through a 3-layer architecture with multiple rounds of training in both the user and agent layers, respectively. It requires only about 110 rounds to obtain the accuracy of the CRD method. The method with punishment mechanism encourages users to submit data with higher QoS, making the model accuracy converge faster. This results in fewer iteration rounds and greater savings in communication energy consumption.

We also evaluate the impact of various privacy sensitivities $Sens(\mathcal{D}_{\mathcal{U}_i}) = 0.1, 0.3$, and 0.5 on the number of iteration rounds. As shown in Figs. 12(a) and 11(b) and 12(b), we find that the model converges faster when the privacy sensitivity degree is greater. The reason is that users can tolerate greater privacy leakage and then choose less noise $\varepsilon_{\mathcal{U}_i}$, which can improve the accuracy of Local-FL model and thus the convergence speed. In other words, a higher level of privacy tolerance can save more communication energy consumption.

5.2.3. Generality

To verify the generality of our approach, we conduct extended experiments on different datasets. As shown in Fig. 13, we obtained curves similar to MNIST on CIFAR-100, ImageNet and IEMOCAP datasets. This means that our incentive mechanism is still applicable for other deep learning tasks. The designed incentives encourage users who are satisfied with the social norms to participate in the training of FL, resulting in a balance among model accuracy, privacy preservation, and energy consumption.

All of the above experiments are implemented on i.i.d data, and each dataset is shuffled and partitioned into 100 clients. Similar to literature [50], the non i.i.d data indicates that the client holds the data from only a few limited classes. For MNIST dataset, we sort the data by its label, and then divide it into 300 shards of size 200, and assign each of 100 clients 3 shards. For CIFAR-100 dataset, we sort and divide it into 200 shards of size 250, and assign each of 100 clients 2 shards. We set the learning rate as 0.05 and 0.03, respectively.

To observe the performance on non-i.i.d data, we test our method in comparison to CRD and ODFL on MNIST and CIFAR-100 datasets. As shown in Fig. 14(a) and Fig. 14(b), our method both leads to an improvement of 5.97% and 4.78% on the accuracy when we fix the number of iterations. And by comparing Figs. 14(a), 11(b) and 14(a) and Fig. 13(a), we also find that the gap between our method and baselines on non-i.i.d data is greater than i.i.d data. This phenomenon is theoretical since our incentive mechanism is calculated from the three-player interaction scenario and maximize all players' payoffs. Our agent plays a supervisory role in addition to computing power. It prevents clients who are unable to achieve average accuracy for consecutive training rounds from participating in the next training round, mitigating the problem of slow model convergence due to data size and distribution.

6. Conclusion

In this paper, we use a game-theoretic approach to simulate all players' decision-making processes, and help them balance their privacy preservation effects, model accuracy, and energy consumption. Based on the sequential order of all players in the FL training, we model each player's decision by introducing an extensive game tree. We also construct the payoff functions for all players, and derive their Nash equilibrium to find the optimal strategies. In addition, we observe the trend of payoffs in repeated game and propose the social norms that can help users to achieve a balance of privacy, accuracy and energy. Our numerical simulation results demonstrate that the punishment mechanism motivates players to pay more attention to the longer-term payoffs instead of current payoffs, and enables global optimization of federated learning.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research is sponsored by the National Key R&D Program of China (No. 2018YFB2100400), the National Natural Science Foundation of China (No. 62002077, 61872100), the Major Research Plan of the National Natural Science Foundation of China (92167203), the Guangdong Basic and Applied Basic Research Foundation (No. 2020A1515110385), the China Postdoctoral Science Foundation (No. 2022M710860), the Zhejiang Lab (No. 2020NF0AB01), Guangzhou Science and Technology Plan Project (202102010440).

References

[1] J. Zhang, Y. Zhang, X. Xu, Pyramid u-net for retinal vessel segmentation, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 1125–1129.
[2] B. Shickel, P.J. Tighe, A. Bihorac, P. Rashidi, Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis, IEEE J. Biomed. Health Inform. 22 (5) (2017) 1589–1604.
[3] J. Xiong, J. Ren, L. Chen, Z. Yao, M. Lin, D. Wu, B. Niu, Enhancing privacy and availability for data clustering in intelligent electrical service of iot, IEEE Internet Things J. 6 (2) (2019) 1530–1540.
[4] L. Yin, J. Feng, H. Xun, Z. Sun, X. Cheng, A privacy-preserving federated learning for multiparty data sharing in social iots, IEEE Trans. Netw. Sci. Eng. 8 (3) (2021) 2706–2718.
[5] Z. Tian, X. Gao, S. Su, J. Qiu, X. Du, M. Guizani, Evaluating reputation management schemes of internet of vehicles based on evolutionary game theory, IEEE Trans. Veh. Technol. 68 (6) (2019) 5971–5980.
[6] Y. Liu, H. Yu, S. Xie, Y. Zhang, Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks, IEEE Trans. Veh. Technol. 68 (11) (2019) 11158–11168.
[7] GSMA, Iot connections forecast: the impact of covid-19. https://data.gsmaintelligence.com/research/research/research-2020/iot-connections-forecast-the-impact-of-covid-19.
[8] M. Li, Y. Sun, H. Lu, S. Maharjan, Z. Tian, Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems, IEEE Internet Things J. 7 (7) (2019) 6266–6278.
[9] H. Dong, S. Yu, C. Wu, Y. Guo, Semantic image synthesis via adversarial learning, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5706–5714.
[10] S. Murugesan, Harnessing green it: principles and practices, IT professional 10 (1) (2008) 24–33.
[11] A.P. Bianzino, L. Chiaraviglio, M. Mellia, J.-L. Rougier, Grida: green distributed algorithm for energy-efficient ip backbone networks, Comput. Network. 56 (14) (2012) 3219–3232.
[12] Z. Lian, W. Wang, C. Su, Cofel: communication-efficient and optimized federated learning with local differential privacy, in: ICC 2021-IEEE International Conference on Communications, IEEE, 2021, pp. 1–6.
[13] Y. Liu, H. Wang, M. Peng, J. Guan, Y. Wang, An incentive mechanism for privacy-preserving crowdsensing via deep reinforcement learning, IEEE Internet Things J. 8 (10) (2020) 8616–8631.
[14] P. Panda, A. Sengupta, K. Roy, Energy-efficient and improved image recognition with conditional deep learning, ACM J. Emerg. Technol. Comput. Syst. 13 (3) (2017) 1–21.
[15] Y. Zou, S. Feng, D. Niyato, Y. Jiao, S. Gong, W. Cheng, Mobile device training strategies in federated learning: an evolutionary game approach, in: 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, 2019, pp. 874–879.

[16] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, S. Liu, Efficient and privacy-enhanced federated learning for industrial artificial intelligence, IEEE Trans. Ind. Inf. 16 (10) (2019) 6532–6542.
[17] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, Y. Zhou, A hybrid approach to privacy-preserving federated learning, in: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, 2019, pp. 1–11.
[18] K. Wei, J. Li, M. Ding, C. Ma, H.H. Yang, F. Farokhi, S. Jin, T.Q. Quek, H.V. Poor, Federated learning with differential privacy: algorithms and performance analysis, IEEE Trans. Inf. Forensics Secur. 15 (2020) 3454–3469.
[19] Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang, Blockchain and federated learning for privacy-preserved data sharing in industrial iot, IEEE Trans. Ind. Inf. 16 (6) (2019) 4177–4186.
[20] M. Duan, D. Liu, X. Chen, Y. Tan, J. Ren, L. Qiao, L. Liang, Astraea: self-balancing federated learning for improving classification accuracy of mobile deep learning applications, in: 2019 IEEE 37th International Conference on Computer Design (ICCD), IEEE, 2019, pp. 246–254.
[21] W. Luping, W. Wei, L. Bo, Cmfl: mitigating communication overhead for federated learning, in: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2019, pp. 954–964.
[22] L. Nagalapatti, R. Narayanam, Game of Gradients: Mitigating Irrelevant Clients in Federated Learning, arXiv:2110.12257.
[23] J. Du, C. Jiang, K.-C. Chen, Y. Ren, H.V. Poor, Community-structured evolutionary game for privacy protection in social networks, IEEE Trans. Inf. Forensics Secur. 13 (3) (2017) 574–589.
[24] X. Wu, T. Wu, M. Khan, Q. Ni, W. Dou, Game theory based correlated privacy preserving analysis in big data, IEEE Trans. Big Data 7 (4) (2017) 643–656.
[25] L. Xiao, Y. Li, G. Han, H. Dai, H.V. Poor, A secure mobile crowdsensing game with deep reinforcement learning, IEEE Trans. Inf. Forensics Secur. 13 (1) (2017) 35–47.
[26] A.R. Sfar, Y. Challal, P. Moyal, E. Natalizio, A game theoretic approach for privacy preserving model in iot-based transportation, IEEE Trans. Intell. Transport. Syst. 20 (12) (2019) 4405–4414.
[27] L. Xu, C. Jiang, Y. Qian, J. Li, Y. Zhao, Y. Ren, Privacy-accuracy trade-off in differentially-private distributed classification: a game theoretical approach, IEEE Trans. Big Data 7 (4) (2021) 770–783.
[28] R. Hu, Y. Gong, Trading data for learning: incentive mechanism for on-device federated learning, in: GLOBECOM 2020-2020 IEEE Global Communications Conference, IEEE, 2020, pp. 1–6.
[29] Z. Sun, L. Yin, C. Li, W. Zhang, A. Li, Z. Tian, The qos and privacy trade-off of adversarial deep learning: an evolutionary game approach, Comput. Secur. 96 (2020) 101876.
[30] R. Jin, X. He, H. Dai, On the tradeoff between privacy and utility in collaborative intrusion detection systems-a game theoretical approach, in: Proceedings of the Hot Topics in Science of Security: Symposium and Bootcamp, 2017, pp. 45–51.
[31] Z. Zhou, S. Yang, L. Pu, S. Yu, Cefl: online admission control, data scheduling, and accuracy tuning for cost-efficient federated learning across edge nodes, IEEE Internet Things J. 7 (10) (2020) 9341–9356.
[32] Z. Yang, M. Chen, W. Saad, C.S. Hong, M. Shikh-Bahaei, Energy efficient federated learning over wireless communication networks, IEEE Trans. Wireless Commun. 20 (3) (2020) 1935–1949.
[33] B. Luo, X. Li, S. Wang, J. Huang, L. Tassiulas, Cost-effective federated learning in mobile edge networks, IEEE J. Sel. Area. Commun. 39 (12) (2021) 3606–3621.
[34] J. Hamer, M. Mohri, A.T. Suresh, Fedboost: a communication-efficient algorithm for federated learning, in: International Conference on Machine Learning, PMLR, 2020, pp. 3973–3983.
[35] A. Elgabli, J. Park, C.B. Issaid, M. Bennis, Harnessing wireless channels for scalable and privacy-preserving federated learning, IEEE Trans. Commun. 69 (8) (2021) 5194–5208.
[36] L. Li, D. Shi, R. Hou, H. Li, M. Pan, Z. Han, To talk or to work: flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices, in: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, IEEE, 2021, pp. 1–10.
[37] K. Wei, J. Li, M. Ding, C. Ma, H. Su, B. Zhang, H.V. Poor, User-level privacy-preserving federated learning: analysis and performance optimization, IEEE Trans. Mobile Comput. 21 (9) (2022) 3388–3401.
[38] X. Tu, K. Zhu, N.C. Luong, D. Niyato, Y. Zhang, J. Li, Incentive mechanisms for federated learning: from economic and game theoretic perspective, IEEE Trans. Cogn. Commun. Netw. 8 (3) (2022) 1566–1593.
[39] W. Wang, Q. Zhang, A stochastic game for privacy preserving context sensing on mobile phone, in: IEEE INFOCOM 2014-IEEE Conference on Computer Communications, IEEE, 2014, pp. 2328–2336.
[40] S.R. Pandey, N.H. Tran, M. Bennis, Y.K. Tun, A. Manzoor, C.S. Hong, A crowdsourcing framework for on-device federated learning, IEEE Trans. Wireless Commun. 19 (5) (2020) 3241–3256.
[41] K.L. Ng, Z. Chen, Z. Liu, H. Yu, Y. Liu, Q. Yang, A multi-player game for studying federated learning incentive schemes, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 5279–5281.
[42] S. Wang, L. Li, W. Sun, J. Guo, R. Bie, K. Lin, Context sensing system analysis for privacy preservation based on game theory, Sensors 17 (2) (2017) 339.
[43] Y. Mao, J. Zhang, K.B. Letaief, Dynamic computation offloading for mobile-edge computing with energy harvesting devices, IEEE J. Sel. Area. Commun. 34 (12) (2016) 3590–3605.
[44] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[45] A. Krizhevsky, G. Hinton, Learning Multiple Layers of Features from Tiny Images, Department of Computer Science, University of Toronto, 2009.
[46] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a Meeting Held December 3-6, 2012, Lake Tahoe, Nevada, United States, 2012, pp. 1106–1114.
[47] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J.N. Chang, S. Lee, S.S. Narayanan, Iemocap: interactive emotional dyadic motion capture database, Comput. Humanit. 42 (4) (2008) 335–359.
[48] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[49] M. Wu, D. Ye, J. Ding, Y. Guo, R. Yu, M. Pan, Incentivizing differentially private federated learning: a multidimensional contract approach, IEEE Internet Things J. 8 (13) (2021) 10639–10651.
[50] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.