A Game Theory Reward Model for Federated Learning with Probabilistic Verification

Gennaro Auricchio
gennaro.auricchio@unipd.it
University of Padua
Padua, Italy

Harry J. Clough
hc2405@bath.ac.uk
University of Bath
Bath, United Kingdom

Christopher Ho
cs2001h@gmail.com
University of Bath
Bath, United Kingdom

Kaigui Bian
bkg@pku.edu.cn
Peking University
Beijing, China

Changyu Dong
changyu.dong@gzhu.edu.cn
Guangzhou University
Guangzhou, China

Kan Yang
Kan.Yang@memphis.edu
University of Memphis
Memphis, Tennesee, USA

Jie Zhang
jz2558@bath.ac.uk
University of Bath
Bath, United Kingdom

Abstract

In Federated Learning, a Central Node (CN) coordinates a group of agents to collectively train a shared neural network. However, due to the inherent information asymmetry, some agents may behave as free riders and exploit the system by reaping rewards or by passively benefiting from the common model without contributing to the training process. Proof-of-Training (PoT) effectively allows the CN to verify that an agent has completed training honestly and correctly. However, this method incurs high costs, including proof generation by the agent, communication expenses, and proof verification by the CN. Conducting Proof-of-Training in each FL round is impractical due to these expenses. To enhance verification efficiency, a feasible strategy is to conduct probabilistic verification, where only a subset of agents is sampled for verification in each FL round. This paper aims to design a new incentive mechanism to motivate the agents behave honestly and potentially mitigate free riders. Our model hinges on two parameters: (i) the reward allocated to the local trainers, namely R, and (ii) a probability vector, denoted as $\vec{p}$, indicating the likelihood of subjecting each agent to PoT scrutiny. We show that it is possible to characterize a set of parameters R and $\vec{p}$ that minimizes the total CN cost and makes the routine Individually Rational and Incentive Compatible, so that every agent will actively train their local model. Finally, we validate our model through extensive experiments. Our findings show that our characterization of the best reward and validation scheme is correct as they minimize the cost of the training routine without compromising the convergence speed. All our experiments are conducted on various datasets, demonstrating the wide applicability of our results.

The Sixth International Conference on Distributed Artificial Intelligence, Dec 18th – Dec 22nd, 2024, Singapore

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

CCS Concepts

• **Theory of computation → Multi-agent learning**.

Keywords

Federated Learning, Incentive Schemes, Free-rider Problem

ACM Reference Format:
Gennaro Auricchio, Harry J. Clough, Christopher Ho, Kaigui Bian, Changyu Dong, Kan Yang, and Jie Zhang. 2018. A Game Theory Reward Model for Federated Learning with Probabilistic Verification. In *Proceedings of (The Sixth International Conference on Distributed Artificial Intelligence).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Federated Learning (FL) is a novel approach to machine learning that allows different data owners to train a neural network without sharing information with competitors [41] and simplifies the data transmission process required to aggregate datasets owned by different companies [34]. The idea behind any FL routine is simple, instead of collecting the data from different agents, the CN collect the parameters of a model that every single agent trains using their own data. After collecting the parameters, the CN combines them and form a new set of parameters, which is broadcasted back to the agents. The agents will then train again on the generated model and generate a new set of parameters, which are sent back to the CN, starting the procedure all over again. Despite the idea being simple, it allows every agent to not share its own dataset, it reduces the transmission cost, and the procedure is protected from leaks thanks to secure aggregation protocols [2, 29].

Even though this decentralized approach to machine learning works well in theory, it shows some flaws when implemented in practice. Indeed, FL operates under the assumption that all participating agents will cooperate with the CN to facilitate the efficacy of the global model. However, this assumption may not always hold true in real-world scenarios. Training a model takes time and resources

that must be paid by the agent performing the training, which can lead agents to behave as *free riders*. In FL, a *free rider* is an agent that does not actively contribute to the learning process of the global model while profiting from having access to the global model and/or passively benefits from the incentive scheme. This behaviour induces an interesting social aspect to the FL model, which has lead the scientific community to analyze the problem from a game-theoretical viewpoint, [17, 18, 28, 35, 44]. In this framework, agents and CN are regarded as agents that aim to achieve their maximum utility by calibrating their level of participation. The CN must address several key objectives: maintaining agent engagement in the training routine, discouraging free-rider behavior, minimizing costs, and upholding privacy and security throughout the process. To accomplish these aims, the CN can leverage methods like Proof-of-Training (PoT) that allows the CN to discern whether an agent is engaging in free-rider behavior without compromising the overall privacy of the FL process [27, 33]. Proof-of-Training effectively allows the CN to verify that an agent has completed training honestly and correctly. However, this method incurs high costs, including proof generation by the agent, communication expenses, and proof verification by the CN. Conducting PoT in each FL round is impractical due to these expenses. To enhance verification efficiency, a feasible strategy is to conduct probabilistic verification, where only a subset of agents is sampled for verification in each FL round. In this paper, we present a novel game-theoretical reward scheme for FL with probability verification.

Our Contribution

In this paper, we present a novel game-theoretical reward scheme for FL with probability verification. Our main contributions are summarized as follows

(1) In Section 3, we propose a game theory reward model for FL with a probabilistic verification method. In our model, the CN is in charge of determining two parameters during each training round: (i) the reward scheme, which is determined by the total reward, namely R, that is distributed among the participating agents, and (ii) the verification scheme, characterized by a vector $\vec{p} = (p_1, p_2, \ldots, p_n) \in [0, 1]^n$, wherein p_i represents the probability of subjecting agent i to PoT scrutiny. We posit that the utility of the CN hinges on two crucial factors: the aggregate size of data held by participating agents and the total cost incurred by the reward and verification scheme. Once the CN selects R and $\vec{p}$, every agent decides whether is in their interest to take part in the training process and whether it is more convenient to act as a free rider depending on the utility entailed to these behaviours.

(2) In Section 4, we study the equilibrium of this game and fully characterize the reward and verification schemes that keep honest agents engaged with the training routine, deter them from acting as free riders, and maximize the utility of the CN. Moreover, we characterize the optimal amount of data that the agents should be used to maximize the utility of the CN.

(3) Lastly, in Section 5, we validate our results through extensive numerical experiments. All our results confirm that carefully picking a verification scheme and the reward allows the CN to efficiently train the model while minimizing the overall cost induced by the routine. The positive results come without compromising the convergence speed of the global model. Finally, it is worth stressing that our experiments are run on several different datasets, showcasing how broad the applicability of our results is.

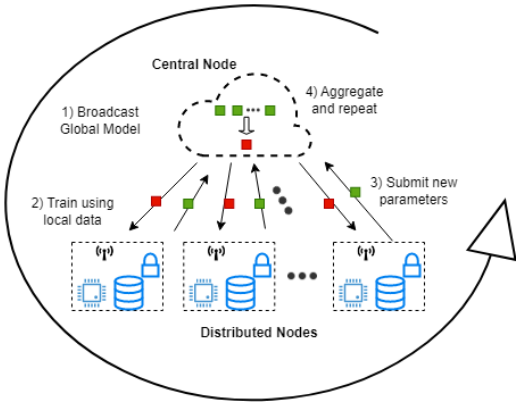

Figure 1: General training structure in FL.

2 Preliminaries

In this section, we recall the basic notions on Federated Learning and Proof-of-Training verification scheme.

2.1 Federated Learning

In the context of FL, the CN undertakes the task of training the model by aggregating the models updated by agents. Each agent possesses its own dataset denoted as X_i, with the size of the dataset represented as $n_i = |X_i|$. The cumulative size of all datasets is defined as $\mathcal{N} = \sum_i n_i$.

Federated average (FedAvg) is an approach to implementing federated learning to train a model distributively while preserving data-privacy. It assumes N participants (distributed nodes) who respectively own private local data-sets $X_1, X_2, \ldots, X_N$ and a model owner (CN) that coordinates the training. Each round, the CN broadcasts a global model with parameters θ_{global} to the distributed nodes. The distributed nodes will then train this model on their local data and submit their new parameters θ_i to the CN. The CN averages these parameters and updates the global model, concluding one round of training in FL. This process is illustrated in Figure 1.

To maintain data confidentiality, the CN employs secure data aggregation, which means the CN cannot access the specific model parameter values of each agent. Secure techniques, such as encryption and privacy-preserving methods, are used to protect individual parameters during transmission and aggregation. Techniques such as secure multi-party computation (SMPC) or homomorphic encryption are commonly utilized, allowing the CN to obtain only the encrypted and aggregated results without directly accessing the individual agent's parameters.

2.2 Free riders and PoT detection

In FL, free rider behavior refers to the behavior of agents participating in FL who choose not to actively contribute their local model

updates but instead rely on updates from other agents to benefit from model improvements and rewards while avoiding their own computation costs. This behavior is considered selfish and unfair as it relies on the contributions of others without making any contribution themselves. When agents choose free rider behavior, they do not contribute meaningful updates to the global model training. This can result in degraded performance or slower convergence of the global model. Agents that actively participate in FL and contribute their model updates bear more computation costs, while free rider agents avoid these costs. This leads to an unfair burden on some agents, potentially reducing their participation and willingness to cooperate.

For this reason, several researchers have explored various approaches to address the free rider phenomenon in Federated Learning and other domains [4, 14, 26]. Many methods aim to detect free-riding agents using tools from anomaly detection systems [11, 12, 36]. While these methods have proven reliable, they are often tailored to address specific types of attacks. The effectiveness of these detection systems depends significantly on how free riders generate the parameters or gradients they submit to the central node. An alternative approach involves designing incentive schemes that reward honest participants and penalize dishonest ones. However, this requires the Central Node to assess the behaviour of each agent. To achieve this, cryptographic tools such as Proof-of-Training (PoT) routines can be employed. PoT allows the Central Node to detect free rider behaviour without explicitly revealing the actual model parameters of each agent, preserving privacy and confidentiality losses. Agents who fail to provide valid proof or exhibit inconsistent responses can be identified as potential free riders, allowing the central node to take appropriate actions to mitigate their behaviour. Despite these benefits, PoT routines entail a high cost, making them not feasible to the Central Node to run them on every agent. Our solution to this issue is to implement a probabilistic version of such routines in which only part of the trainers are subject to scrutiny, leading to an overall lower cost. This paper aims to design a new incentive mechanism to motivate the agents behave honestly and potentially mitigate free riders.

3 A Game Model for Federated Learning with Probabilistic Proof-of-Training on Free Riders

In this section, we develop a game-theoretical framework that captures the dynamics between the agents and the CN. In our formalism, the CN has to reward every agent that was not caught to be a free rider at the end of the round. The overall routine can then be phrased as a game in which the participating agents try to maximize their own utility.[1] In what follows, we detail the objectives and strategies available to the agents and to the CN.

CN's strategy. In our model, the CN is in charge of determining the reward and the verification scheme of the game. We assume that the CN decides the total reward to distribute across the agents at every round, namely $R_t > 0$. We also assume that the process of running the Proof-of-Training (PoT) routine incurs a cost for every data entry of every agent, denoted as V. For the sake of simplicity, we focus on this class of verification costs because they are standard in most Proof-of-Training schemes (e.g. each scheme that leverages a Zero-Knowledge proof routine [11]). Notice however that this framework and the techniques we use can be adapted to handle any cost that can be represented as a polynome of n. Therefore, the cost of verifying an agent using a dataset composed of $n_i^{(t)}$ elements is $Vn_i^{(t)}$. If one or more agents are caught submitting false information to the CN, they are excluded from participating in any learning round, and their promised reward is waived. Notice that, according to our formalism, Free Riders that are not caught by the verification scheme can participate to the next training round.

The CN must publicly announce the total promised reward for round t, denoted as R_t, and the verification probability scheme p, which are defined as follows:

- The **total reward** at time t, denoted with $R_t \in [0, +\infty)$, is the amount of resources that the CN uses to compensate the local trainers. We assume that the training cost to each agent i is proportional to the size of their dataset, $n_i^{(t)}$, and hence divide the reward between agents proportionally to this value. That is, the reward to an agent at round t is $R_t \frac{n_i^{(t)}}{\mathcal{N}}$.
- The **verification scheme**, which is a function $p : \mathbb{N}^m \to [0,1]^m$. The output of the function, *i.e.* $p(\vec{n})$, is a vector whose entries correspond to the probability $p_i(\vec{n}) \in [0,1]$ that agent i will be subjected to verification by the CN. We want p to not depend on the index i but only on the agents' report, so that $p_i(\vec{n}) = p_j(\vec{n})$ if and only if $n_i^{(t)} = n_j^{(t)}$. In particular, p is an anonymous function.

Putting it all together, the cost of the CN at round t is as follows

$$c_{CN}(t) = R_t + V \sum_{i \in [m]} n_i^{(t)} p(n_i^{(t)}). \tag{1}$$

We measure the training quality of the CN model through a concave function g, which denotes the utility accrued by the CN upon receiving the parameters of agents utilizing a combined dataset comprising $\mathcal{N}_t$ elements. We assume that g is concave to reflect the diminishing impact of adding data as the total dataset size increases. This characteristic is commonly referred to as *diminishing marginal utility*. Therefore, the utility of the CN at round t is defined as

$$u_{CN}(t) = g(\mathcal{N}_t) - \left(R_t + V \sum_{i \in [m]} n_i^{(t)} p(n_i^{(t)})\right) \tag{2}$$

where $\mathcal{N}_t = \sum_i n_i^{(t)}$ is the total number of data used by local trainers during training round t. The functional in (2) balances between the training-efficiency of the neural network and the total cost of keeping agents honestly engaged with the training process.

Agents' strategies. Each combination of $\{(R_t, \vec{p})\}_{t \in \mathbb{N}}$ determines a game for every round t. In our model, every agent can either train honestly or act as a free rider. Since free riders are noxious to the model, the CN employs a probabilistic Proof-of-Training (PoT) detection to remove them. Before each round, the CN announces a probability, denoted as $p_{i,t}$, representing the likelihood of agent i being detected. If an agent is detected as a free rider, the CN withdraws the initially promised reward and removes them from the set of trainers.

[1] For this reason, from now on, we will refer to the FL routine or to the FL game interchangeably.

Table 1: Table of the most used variables and their meaning.

m	number of local trainers
X_i	data set owned by agent i
$n_i^{(t)}$	data size of agent i at time round t
$\mathcal{N}_t$	size of all the data owned by the trainers
R_t	total reward promised at round t
V	cost per data of running the PoT
α	cost per data of training a local model
β	transmission cost
$\vec{p}$	vector representing the verification scheme

- **Honest Agents.** An agent acting honestly actively trains their local model as intended in the FL routine prescribed by the CN. Since they train honestly, their utility in round t does not depend on the verification scheme, and thus is defined as
$$u_i(t) = \frac{n_i^{(t)}}{\mathcal{N}_t} R_t - \alpha n_i^{(t)} - \beta,$$
where, (i) R_t represents the total reward allocated by the CN to the agents, distributed in proportion to $n_i^{(t)}$; (ii) α denotes the cost for training their respective model; and (iii) β represents the communication cost the agent incurs when they transfer the models to the CN.
- **Free Riders.** An agent acting as a free rider during round t choose not to participate in local training and instead aim to receive rewards by submitting randomly generated models/gradients. Since they do not engage in training, their only cost is the one induced by the parameters transmission. Finally, notice that, since every free rider has a probability $p_i(\vec{n})$ to be detected, its utility at the t round is
$$u_i(t) = (1 - p_i(\vec{n})) \frac{n_i^{(t)}}{\sum_i^n n_i^{(t)}} R_t - \beta.$$

Each agent is motivated by self-interest, so they will honestly train if the following two conditions are met:

(a) The procedure is *individually rational*, that is every agents benefit from participating to the FL routine. In our framework, this is expressed by the condition
$$\frac{n_i^{(t)}}{\sum_i^m n_i^{(t)}} R_t - \alpha n_i^{(t)} - \beta \geq 0$$
for every round t.

(b) The procedure is *incentive compatible*, that is acting as a free rider is less rewarding than behaving truthfully. In our framework, this is expressed by the condition
$$(1 - p(n_i^{(t)})) \frac{n_i^{(t)}}{\sum_i^m n_i^{(t)}} R_t - \beta \leq \frac{n_i^{(t)}}{\sum_i^m n_i^{(t)}} R_t - \alpha n_i^{(t)} - \beta,$$
for every round t.

Our aim is to establish a reward and verification scheme that induces an individually rational and incentive compatible routine that minimises the total cost of the CN.

In Table 1 we report all the main variables of our model.

4 Equilibrium Analysis

In this section we characterize the round reward amount R_t and the verification probability function p that minimizes the CN's cost while keeping the agents honestly engaged with the training process.

Since the function g does not depend on t, the optimal reward and verification scheme for a given round t does not depend on t. For this reason, throughout this section, **we simplify the notation by dropping the index** t. To find the optimal solution, we proceed as follows. First, we fix the total amount of data owned by the local trainers, namely $\mathcal{N}$. For every $\mathcal{N}$, we find the best possible R and p entailed to $\mathcal{N}$. Once we express the optimal R and p as a function of $\mathcal{N}$, we retrieve the complete solution.

Given a fixed $\mathcal{N}$, maximizing the utility defined in (2) is the same as minimizing the cost $c_{CN}(t)$ defined in (1), with respect to R and p. We therefore obtain the following mathematical programming problem

$$\min_{R,p} R + V \sum_{i=1}^{m} p_i(\vec{n}) n_i, \tag{3}$$
$$\text{s.t.} \quad \forall i \in [m]$$
$$(1 - p_i(\vec{n})) \frac{n_i}{\sum_i^m n_i} R - \beta \leq \frac{n_i}{\sum_i^m n_i} R - \alpha n_i - \beta, \tag{4}$$
$$\frac{n_i}{\sum_i^m n_i} R - \alpha n_i - \beta \geq 0, \tag{5}$$

where, constraints (4) enforce that the dominant strategy of local trainers is to behave as honest agents (Incentive Comaptibility) and (5) enforces that every agent takes part in the training procedure willingly (Individual Rationality).

THEOREM 4.1. *Let $\vec{n}$ be the vector containing the agents datasizes. For every round t, the solution of* (3) *are as follows*

$$R(\vec{n}) = \max\left\{\sqrt{V\alpha}, \frac{\beta}{\min_{i\in[m]} n_i} + \alpha\right\}\mathcal{N} \quad \textit{and}$$
$$p_i(\vec{n}) = \frac{\alpha}{\max\{\sqrt{V\alpha}, \frac{\beta}{\min_{i\in[m]} n_i} + \alpha\}},$$

where $\mathcal{N} = \sum_i^m n_i$.

PROOF. Let $\mathcal{N}$ be the sum of all the datasets used by the local trainers. First, we notice that $p_i(\vec{n})$ is subject only to the Incentive Compatibility constraint (4), that is
$$(1 - p_i(\vec{n})) \frac{n_i}{\sum_i^m n_i} R - \beta \leq \frac{n_i}{\sum_i^m n_i} R - \alpha n_i - \beta$$
$$-p_i(\vec{n}) \frac{R}{\sum_i^m n_i} \leq -\alpha$$
that is
$$p_i(\vec{n}) \geq \frac{\alpha \mathcal{N}}{R}. \tag{6}$$
Since we want to minimize the objective in (3), we obtain that the best verification scheme is given by the formula $p_i(\vec{n}) = \frac{\alpha\mathcal{N}}{R}$. If we plug this relation in (3), we are able to express the objective value of the problem as a function of R, that is
$$R + V \sum_{i=1}^{m} \frac{\alpha \mathcal{N}}{R} n_i. \tag{7}$$

To find the best value of R, we take the derivative of (7) and study the stationary points. A simple computation allows us to infer that the best reward total to distribute at every time round is

$$R = \mathcal{N}\sqrt{V\alpha}.$$

Notice, however, that depending on α, V and β, the value we obtained might not satisfy the IR constraint (5), which reads as

$$\frac{n_i}{\sum_i^m n_i} R \geq \alpha n_i + \beta,$$

for every $i \in [m]$. We then infer $R \geq \max_i\{\alpha + \frac{\beta}{n_i}\}\mathcal{N}$. Therefore we conclude that $R = \max\{\sqrt{V\alpha}, \alpha + \frac{\beta}{\min_{i\in[m]} n_i}\}\mathcal{N}$. By plugging the latter identity into $p_i(\vec{n}) = \frac{\alpha\mathcal{N}}{R}$, we conclude the proof. □

Notice that the probability of selecting one agent does not depend on n_i as long as

$$\sqrt{V\alpha} \geq \alpha + \beta$$

holds, since $\alpha+\beta \geq \alpha+\frac{\beta}{n_i}$ for every $n_i \in \mathrm{N}$. Intuitively, this condition tells us that if the verification cost is too high, the best strategy that the CN has is to check every agent with equal probability. Moreover, given $\mathcal{N}$, the vector $\vec{n}$ that minimizes the reward to share is such that $n_i = \frac{\mathcal{N}}{m}$ for every $i \in [m]$, *i.e.* every agent trains on the same amount of data. To conclude, we show that given the best R and p as a function of $\mathcal{N}$, we can define the best possible value of $\mathcal{N}$ by taking the derivative of

$$\begin{aligned}\mathcal{N} \to & g(\mathcal{N}) - \max\left\{\sqrt{V\alpha}, \alpha + \frac{\beta}{n_i}\right\}\mathcal{N} \\ & - V\sum_i n_i \frac{\alpha}{\max\{\sqrt{V\alpha}, \alpha + \frac{\beta}{n_i}\}}\end{aligned}$$

and set it to be equal to zero.

THEOREM 4.2. *Let $\sqrt{V\alpha} \geq \alpha + \beta$ hold and let g be a concave differentiable function. Then, we have that the best number of data that the CN needs to train the model is*

$$\mathcal{N} = (g')^{-1}(2\sqrt{V\alpha}).$$

PROOF. Since g is concave by hypothesis, we have that the function

$$\mathcal{N} \to g(\mathcal{N}) - \left(\sqrt{V\alpha}\mathcal{N} + V\sum_{i\in[m]} n_i^{(t)}\sqrt{\frac{\alpha}{V}}\right) = g(\mathcal{N}) - 2\sqrt{V\alpha}\mathcal{N}$$

is concave with respect to $\mathcal{N}$ as well, hence it admits a unique maximizer. By taking the derivative of the CN's utility and by setting it equal to zero we have

$$g'(\mathcal{N}) - 2\sqrt{V\alpha} = 0,$$

which concludes the proof. □

5 Experiments

In this section, we run extensive numerical experiments to validate our findings. In particular, we consider the case in which the CN needs to train a classifier using a Convolutional Neural Network. The scope of our experiments is threefold:

(i) First, we want to confirm that the probability p defined in Theorem 4.1 leads to the lowest cost for the CN;
(ii) Second, we want to evaluate the convergence speed and the cost the CN incurs when we include our verification scheme;
(iii) Third, we want to assess to which extent changing the data used to perform the learning routing affects the results.

In what follows, we outline the specifics of our experiments, i.e. how we model the agents, the parameters of the training rounds, and the specifics of the neural network we train. We denote with $\mathcal{D}$ the global dataset, i.e. the set containing all the data own by the agents.

Data Allocation: We denote with m the number of agents taking part in the training routine, be them Honest Agents or Free Riders. The CN is allocated data in the same manner as the agents. We refer to the CN and agents collectively as *nodes*. For a fixed global dataset $\mathcal{D}$, we assume that every node has the same number of elements of $\mathcal{D}$, moreover we assume that the data owned by each node does not change throughout the rounds. Every node is thus allocated $\frac{1}{m+1}$ of the global dataset $\mathcal{D}$, namely $\mathcal{D}_i$ for the agents and $\mathcal{D}_{\mathrm{CN}}$ for the CNs. Each $\mathcal{D}_i$ and $\mathcal{D}_{\mathrm{CN}}$ is disjoint from the others, so that the each element of $\mathcal{D}$ is available to one and only one node. We do this by removing values selected randomly one-by-one from $\mathcal{D}$ until each node has the specified amount of data. Once allocated its data, each node performs a stochastic 90-10 train-test split on the data. The testing data is kept separate for all future rounds and only used to collect performance metrics. We ran experiments where each $\mathcal{D}_\rangle$ could vary from each-other, but found its affect on performance was negligible compared to other factors; in particular, the ratio of total data owned by honest agents to that owned by free-riders. We therefore chose not to include those experiments in this paper.

The Agents: At the beginning of every training round, each agent decides whether they join the training round or not by computing their expected utility after the CN announces the Reward R and the Verification Scheme p'. The reward is calculated according to Theorem 4.1, and the Verification Scheme is our dependant variable.

Every honest agent trains its own neural network for a full epoch each round, mimicking the behaviour of people truthfully engaging with the training routine. Every Free Rider taking part in the training round resubmits the global model sent by the CN after they applying one of the following two operations: (i) they add some Gaussian noise to each weight in the network, or (ii) they perform a delta-weight attack, which generates the fake updates by combining two previously received global model [23].

Since our model has two different types of Free Riders, we assume that half of the free-riders attack the routine by performing attack (i), and the other half attacks the routine by performing attack (ii). Which type of attack is performed by the Free Rider is determined during the initialisation of the problem and does not change throughout the training rounds.

The Specifics of the Convolution Neural Network: We conduct our experiments on three standard open datasets: MNIST [21], FASHION-MNIST [39], and CIFAR-10 [20]. We train a basic Convolutional Neural Network, with 9 hidden layers and around 100 000 parameters. Indeed, we use the same model architecture for each dataset and change only the input layer to match the dimensions of the input data, hence the differences in parameter numbers.

The Federated Learning Routine: Each Federated Learning routine is composed of 10 rounds, each corresponding to an epoch of training. After a round is completed, the CN gathers all the model trained by the agents participating in the training routine and aggregates them by taking the mean of each weight in the network across all models. The CN then performs its own round of training on the aggregated model, using its own dataset $\mathcal{D}_{\text{CN}}$. We repeat this process for different values of p in order to assess how implementing the verification scheme affects the participation of the Free Riders and the overall cost of the training process.

The CN's Utility: The cost to the CN was defined in Equation 1. However, this cost is the expected cost to the CN, rather than the actualised cost. In order to compute the cost in practice, we replace the verification probability term $p(n_i^{(t)})$ with a binary function $q(n_i^{(t)})$ which is 1 if the agent i was verified on round t, and 0 otherwise. Additionally, we add a term to include the cost of the CN's training which depends on its datasize, namely $X_{\text{CN}}^{(t)}$. This term was omitted from Equation 1 as it is a constant value, and therefore has no effect on the minimisation. We can therefore express the actualised cost function as

$$c'_{\text{CN}}(t) = R_t + \sum_i^m y_i^{(t)} q(n_i^{(t)}) V + \alpha X_{\text{CN}}^{(t)}. \tag{8}$$

Notice that, up to a constant, the expected value of (8) is equal to the original cost presented in 1.

Trials: We repeat each experiment 10 times. All data presented is the mean over these 10 trials.

5.1 Parameters

In what follows, without incurring in any loss of generality we set α to be equal to 1. Now notice that for a given V, β will change the result if and only if $\beta > n_i\sqrt{V} - 1$, i.e. if and only if the cost of transferring the model is greater than the cost of training it. In practice, the cost of training a neural network will almost always exceed the cost of transferring its weights, and this is one of the assumptions behind the concept of Federated Learning itself. We can therefore arbitrarily fix the value of β to some value $\leq n_i\sqrt{V} - 1$.

This leaves us with V, the number of honest agents (m_{H}), and the number of free-riders (m_{FR}) as the main parameters for our experiments. The values of the parameters used to generate the figures in this section are shown in Table 2. Additional combinations of parameters, not directly discussed in the following section, are listed in Table 3. All of our experimental results validated our theory, however those listed in Table 2 were found to be the most illustrative of our findings.

The dependant variable of our experiments is the validation probability. We use the notation p' to refer to this variable, and p to refer to the optimal value obtained from Theorem 4.1.

5.2 Discussion of Results

In this section we comment on our results and discuss the trends we see in the data across the different metrics we collected: the *Loss* of the model, the *Accuracy* of the model, and the *Cost* incurred by the CN. All metrics are collected at the end of each epoch, after the CN had combined the models are performed its own round of training. We use the results from the MNIST dataset in the graphs in this section to illustrate our findings. However, across the three datasets we used in our experiments, all three of our experiments showed the same trends, if to slightly different extents. This is demonstrated by Figure 2.

Table 2: Values of parameters used to generate the results plotted in figs. 3 to 6.

Parameter	m_{H}	m_{FR}	α	β	V
Value	5	15	1	5000	100

Table 3: Additional values of parameters tested, not shown in figures.

Parameter	m_{H}	m_{FR}	V
Values	5, 10	2, 5, 10, 15	4, 16, 100

The results across the different metrics we measured fell into two behavioural groups: $p' < p$ and $p' \geq p$. For our parameters, this is $p' < 0.1$ and $p' \geq 0.1$. The main difference between these groups is that, as predicted by Theorem 4.1, the free-riders stop participating at $p' \geq 0.1$, and thus stop stealing reward and poisoning the training.

Loss and Accuracy: We use categorical cross-entropy as our loss function, while the accuracy is measured as the ratio of the number of successes over the number of trials, that is

$$Acc = \frac{\text{number of successes}}{\text{number of trials}}.$$

As it is shown in Figures 4 and 3, for $p' < 0.1$, the models perform worse in both metrics. The influence of free-riders reduces accuracy and increase loss at all epochs, and decreases the convergence speed.

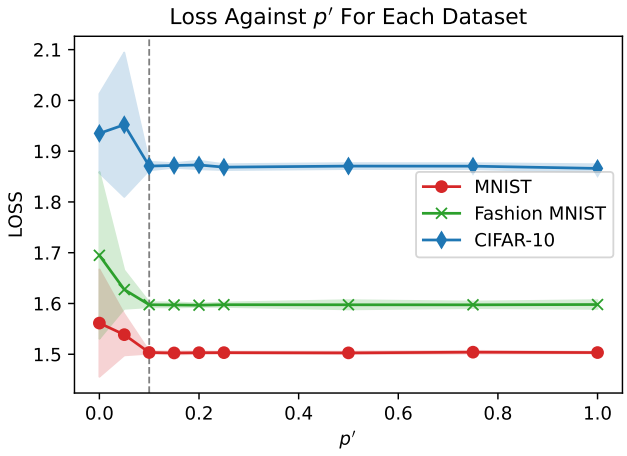

Figure 2: Comparison of mean loss of our models trained on three datasets for nine epochs. The shaded regions show a 95% confidence interval. $p' = p$ is indicated by the dashed line.

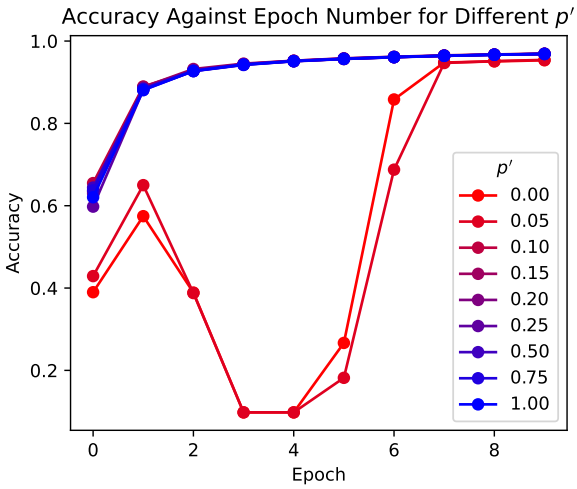

Figure 3: Plot of average accuracy of our models trained on the MNIST dataset.

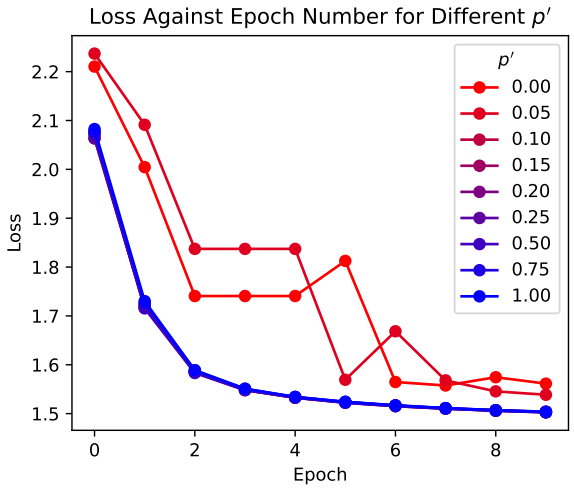

Figure 4: Plot of average loss of our models trained on the MNIST dataset.

We generally see that as p' decreases, the performance further decreases, as free-riders are less likely to be detected and removed from the round. The $p' \geq 0.1$ group all perform virtually the same, with any difference being statistically insignificant. This is expected, as there are no free-riders participating to influence training.

Cost: The cost to the CN each round is calculated according to Equation 8. In these results section we specifically consider the cumulative cost – the cumulative cost on round t is $\sum_{j=1}^{t} c'_{\text{CN}}(j)$. As demonstrated in Figure 5, the two groups are very distinct from one-another in terms of cost to the CN. It is clear the $p' < p$ group is significantly more expensive to the CN, as the large number of free-riders require more reward-payouts. In this group, lower values of p' cost slightly more on average than higher values, as the free-riders are less likely to be detected and removed. Although, this is slightly diminished by the increased cost of running the PoT routine. The $p' \geq p$ group displays a greater spread of costs, with a higher p' directly corresponding to a greater cost. This is due to the cost of running additional PoT verifications, despite no free-riders actually participating.

The findings of our study are comprehensively summarized in Figure 6. The results indicate that, first and foremost, preventing free-riders from participating is the most critical factor. Our theorems accurately predict the optimal value of p' that minimizes both cost and loss, and, although not depicted in the graph, also maximizes accuracy. This validation underscores the robustness of our theoretical model. Furthermore, our analysis demonstrates that increasing p' beyond p results in higher costs without any corresponding improvement in loss or accuracy. These insights highlight the importance of carefully selecting p' to achieve the best balance between cost efficiency and performance.

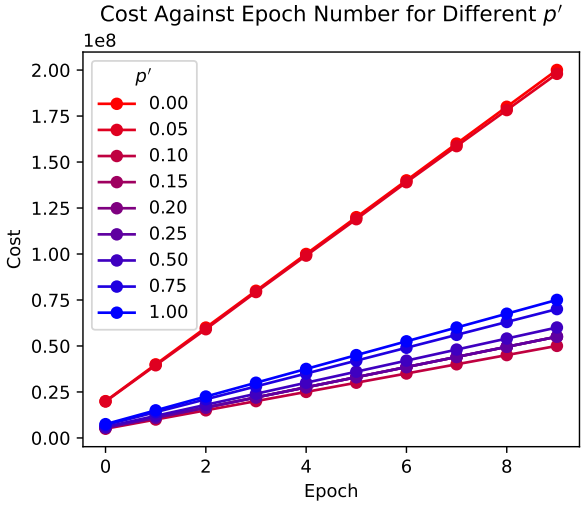

Figure 5: Plot of average cumulative cost of our models trained on the MNIST dataset.

6 Related Works

FL has gained significant attention in the field of AI, despite being a relatively new area of research [25]. Prior works have contributed extensively to the understanding of FL, with a primary focus on cross-device settings [41, 42]. There have been a few studies, that have enriched the diversity of research on FL by considering cross-silo settings, [43]. Through these diverse investigations, the significance of pairing FL with an incentive mechanisms has been consistently underscored. In this section, we aim to provide the reader a comprehensive understanding of FL while incorporating relevant findings from prior works.

Incentive mechanisms. In [6], the authors proposed a general game-theoretical framework that considers three key self-interested actors in FL: data providers, users, and the FL model owner. The authors highlight the problem of information asymmetry, where the

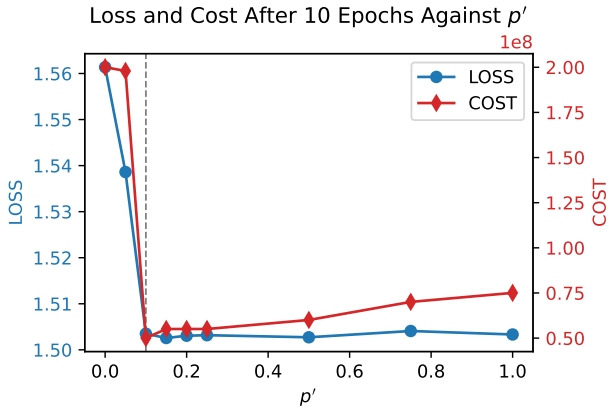

Figure 6: Plot of how both the average cumulative cost and loss vary after 10 rounds of training for different values of p' on the MNIST dataset. $p' = p$ is indicated by the dashed line.

model owner lacks complete information about the data providers due to their private attributes, such as data and cost. This information asymmetry makes the design of effective incentive mechanisms for FL challenging. Moreover, incentive mechanisms for FL should possess certain desirable properties such as incentive compatibility, individual rationality, and fairness. Indeed, traditional incentive mechanisms are inadequate for FL due to the unique characteristics of the FL setting [45]. As a result, incentive mechanisms need to be specifically tailored for FL. For this reason, various alternative incentive mechanisms have been proposed, such as employing auctions mechanisms [5, 13, 43], Stackelberg games [45], and Contract Theory [15]. Perhaps the most promising approach is the lightweight and incentive-compatible model introduced in [43]. The authors extend a multi-dimensional procurement auction proposed in [3], resulting in a mechanism that incentivizes participants to contribute honestly. Experimental results in their study demonstrate a reduction of 51.3% in training rounds, on average, for datasets such as MNIST, Fashion MNIST, CIFAR-10, and HPNews. Despite these positive results, their study has been condacted without considering possible free rider attacks, [8, 23].

Another approach to design meaningful incentive schemes, consists in assessing which parameters are actively improving the global model allows to fairly distribute rewards and keep the local trainers engaged in the training process. A large portion of the currently proposed approaches rely on Shapley Values [31]; however, it is well-known the computation of Shapley Value is computationally expensive [9], leading to the exploration of approximation methods [32, 37]. Other promising approaches for contribution evaluation include utilizing deep reinforcement learning within a Stackelberg game model to learn from historical training records [45] and incorporating reputation as a factor [15].

Free rider attacks. Free riders attacks were firstly studied in [8], where the authors showed that aggregating free rider weights significantly reduces convergence speed of the model. This pioneering work highlighted how important is to defend the FL structure from free rider attacks be them simple free rider attacks (returning global parameters) or disguised free rider attacks (adding noise to the contributions). The first mechanism for FL able to detect free riders was presented in [23]. Their approach utilized the DAGMM, an anomaly detection method [46]. In their work, they focus on two types of free-rider attacks: one that replaces the parameter weights from global models with random weights sampled from a uniform distribution, and one that generates weights by subtracting two parameter sets from previous global models.

Free Rider detection. There are several approaches to identify free riders. In [38], the authors tried to identify free riding nodes in a given graph by suitably weighting each node. In [24], the authors make use of a Deep autoencoding Gaussian mixture model (DAGMM) as an anomaly detector to decide whether agent is honestly training and which one is a free rider. This approach has been then improved [12] and enriched with a reputation and contribution system [36]. Lastly, we mention [16], in which each node is in charge of checking whether its neighborhood nodes are behaving as free riders or not. In this paper, we consider the case in which the Central Node runs an incentive scheme, to keep the agents engaged with the training procedure, and a probabilistic Proof-of-Training detection method (as in [27]) to prevent free riders from participating in the training routine.

7 Conclusion and Future Work

Addressing Free Riders' behaviours in cooperative mechanisms is an important issue in several applied areas. Motivated by the rising popularity of Federated Learning routines to train Neural Networks and their susceptibility to Free Riders, we designed a reward model endowed with probabilistic verification that keeps honest trainers engaged while preventing free riders from passively benefiting from the reward scheme. First, we have introduced a novel game-theoretical framework in which the Central Node can leverage Probabilistic Proof-of-Training (PoT) to effectively detect free riders. We have then thoroughly described the agents' utilities and the cost of the Central Node and characterized when the reward and verification scheme induces an Individually Rational and Incentive Compatible routine. Consequentially, we retrieved the schemes that maximize the utility of the Central Node. Finally, we have assessed our theoretical results by running several numerical experiments. Our empirical validation underscores the practical viability of the proposed model in addressing free rider phenomena within FL environments.

Moving forward, we aim to expand our study in several ways. First, further exploration into randomized incentive mechanisms could further lower the training cost of the CN. Another way to reduce the cost incurred by the Central Node would be to detect some atypical nodes using clustering methods, as done in [10, 19, 22]. This direction of study is extremely promising as fast clustering routines have been studied and developed in recent years [1, 7, 30, 40]. Additionally, we believe that investigating the scalability and robustness of our proposed framework across diverse FL settings and network topologies would further enhance our results. Lastly, it would be interesting to study a dynamic framework in which the agents and the neural network change depending on the different rounds.

References

[1] Gennaro Auricchio, Federico Bassetti, Stefano Gualandi, and Marco Veneroni. 2019. Computing Wasserstein barycenters via linear programming. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 16th International Conference, CPAIOR 2019, Thessaloniki, Greece, June 4–7, 2019, Proceedings 16*. Springer, 355–363.

[2] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (Dallas, Texas, USA) *(CCS '17)*. Association for Computing Machinery, New York, NY, USA, 1175–1191. https://doi.org/10.1145/3133956.3133982

[3] Yeon-Koo Che. 1993. Design competition through multidimensional auctions. *The RAND Journal of Economics* (1993), 668–680.

[4] Ming-Hsiang Chen, Shangzhi Charles Qiu, Min Wei, and Haiyu Huang. 2024. The free riding in hospitality corporate giving: Theoretical explanation and implications. *International Journal of Hospitality Management* 122 (2024), 103878.

[5] Mingshu Cong, Han Yu, Xi Weng, Jiabao Qu, Yang Liu, and Siu Ming Yiu. 2020. A VCG-based fair incentive mechanism for federated learning. *arXiv preprint arXiv:2008.06680* (2020).

[6] Mingshu Cong, Han Yu, Xi Weng, and Siu Ming Yiu. 2020. *A Game-Theoretic Framework for Incentive Mechanism Design in Federated Learning*. Springer International Publishing, Cham, 205–222. https://doi.org/10.1007/978-3-030-63076-8_15

[7] Marco Cuturi and Arnaud Doucet. 2014. Fast computation of Wasserstein barycenters. In *International conference on machine learning*. PMLR, 685–693.

[8] Yann Fraboni, Richard Vidal, and Marco Lorenzi. 2021. Free-rider attacks on model aggregation in federated learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1846–1854.

[9] Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*. PMLR, 2242–2251.

[10] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems* 33 (2020), 19586–19597.

[11] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. 2016. {ZKBoo}: Faster {Zero-Knowledge} for Boolean Circuits. In *25th usenix security symposium (usenix security 16)*. 1069–1083.

[12] Hai Huang, Borong Zhang, Yinggang Sun, Chao Ma, and Jiaxing Qu. 2022. Delta-DAGMM: A Free Rider Attack Detection Model in Horizontal Federated Learning. *Security and Communication Networks* 2022, 1 (2022), 8928790.

[13] Yutao Jiao, Ping Wang, Dusit Niyato, Bin Lin, and Dong In Kim. 2020. Toward an automated auction framework for wireless federated learning services market. *IEEE Transactions on Mobile Computing* 20, 10 (2020), 3034–3048.

[14] William B Joyce. 1999. On the free-rider problem in cooperative learning. *Journal of Education for Business* 74, 5 (1999), 271–274.

[15] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. 2019. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal* 6, 6 (2019), 10700–10714.

[16] Murat Karakaya, İbrahim Körpeoğlu, and Özgür Ulusoy. 2008. Counteracting free riding in Peer-to-Peer networks. *Computer Networks* 52, 3 (2008), 675–694.

[17] Sai Praneeth Karimireddy, Wenshuo Guo, and Michael I Jordan. 2022. Mechanisms that incentivize data sharing in federated learning. *arXiv preprint arXiv:2207.04557* (2022).

[18] Latif U. Khan, Shashi Raj Pandey, Nguyen H. Tran, Walid Saad, Zhu Han, Minh N. H. Nguyen, and Choong Seon Hong. 2020. Federated Learning for Edge Networks: Resource Optimization and Incentive Mechanism. *IEEE Communications Magazine* 58, 10 (2020), 88–93. https://doi.org/10.1109/MCOM.001.1900649

[19] Yeongwoo Kim, Ezeddin Al Hakim, Johan Haraldson, Henrik Eriksson, José Mairton B da Silva, and Carlo Fischione. 2021. Dynamic clustering in federated learning. In *ICC 2021-IEEE International Conference on Communications*. IEEE, 1–6.

[20] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[22] Chengxi Li, Gang Li, and Pramod K Varshney. 2021. Federated learning with soft clustering. *IEEE Internet of Things Journal* 9, 10 (2021), 7773–7782.

[23] Jierui Lin, Min Du, and Jian Liu. 2019. Free-riders in federated learning: Attacks and defenses. *arXiv preprint arXiv:1911.12560* (2019).

[24] Jierui Lin, Min Du, and Jian Liu. 2019. Free-riders in federated learning: Attacks and defenses. *arXiv preprint arXiv:1911.12560* (2019).

[25] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[26] John McMillan. 1979. The free-rider problem: a survey. *Economic Record* 55, 2 (1979), 95–107.

[27] Truc Nguyen and My T Thai. 2023. Preserving privacy and security in federated learning. *IEEE/ACM Transactions on Networking* (2023).

[28] Shashi Raj Pandey, Nguyen H. Tran, Mehdi Bennis, Yan Kyaw Tun, Aunas Manzoor, and Choong Seon Hong. 2020. A Crowdsourcing Framework for On-Device Federated Learning. *IEEE Transactions on Wireless Communications* 19, 5 (2020), 3241–3256. https://doi.org/10.1109/TWC.2020.2971981

[29] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. 2018. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Transactions on Information Forensics and Security* 13, 5 (2018), 1333–1345. https://doi.org/10.1109/TIFS.2017.2787987

[30] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. 2012. Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision: Third International Conference, SSVM 2011, Ein-Gedi, Israel, May 29–June 2, 2011, Revised Selected Papers 3*. Springer, 435–446.

[31] Lloyd S Shapley and Martin Shubik. 1954. A method for evaluating the distribution of power in a committee system. *American political science review* 48, 3 (1954), 787–792.

[32] Tianshu Song, Yongxin Tong, and Shuyue Wei. 2019. Profit allocation for federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2577–2586.

[33] Haochen Sun and Hongyang Zhang. 2023. zkDL: Efficient Zero-Knowledge Proofs of Deep Learning Training. *arXiv preprint arXiv:2307.16273* (2023).

[34] Xuezhen Tu, Kun Zhu, Nguyen Cong Luong, Dusit Niyato, Yang Zhang, and Juan Li. 2022. Incentive Mechanisms for Federated Learning: From Economic and Game Theoretic Perspective. *IEEE Transactions on Cognitive Communications and Networking* 8, 3 (2022), 1566–1593. https://doi.org/10.1109/TCCN.2022.3177522

[35] Xuezhen Tu, Kun Zhu, Nguyen Cong Luong, Dusit Niyato, Yang Zhang, and Juan Li. 2022. Incentive Mechanisms for Federated Learning: From Economic and Game Theoretic Perspective. *IEEE Transactions on Cognitive Communications and Networking* 8, 3 (2022), 1566–1593. https://doi.org/10.1109/TCCN.2022.3177522

[36] Bo Wang, Hongtao Li, Ximeng Liu, and Yina Guo. 2023. Frad: Free-rider attacks detection mechanism for federated learning in aiot. *IEEE Internet of Things Journal* (2023).

[37] Guan Wang, Charlie Xiaoqian Dang, and Ziye Zhou. 2019. Measure contribution of participants in federated learning. In *2019 IEEE international conference on big data (Big Data)*. IEEE, 2597–2604.

[38] Yubao Wu, Ruoming Jin, Jing Li, and Xiang Zhang. 2015. Robust local community detection: on free rider effect and its elimination. *Proceedings of the VLDB Endowment* 8, 7 (2015), 798–809.

[39] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).

[40] Rui Xu and Donald Wunsch. 2005. Survey of clustering algorithms. *IEEE Transactions on neural networks* 16, 3 (2005), 645–678.

[41] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[42] Hao Yu, Sen Yang, and Shenghuo Zhu. 2019. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5693–5700.

[43] Rongfei Zeng, Shixun Zhang, Jiaqi Wang, and Xiaowen Chu. 2020. FMore: An Incentive Scheme of Multi-dimensional Auction for Federated Learning in MEC. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. https://doi.org/10.1109/icdcs47774.2020.00094

[44] Yufeng Zhan, Peng Li, Zhihao Qu, Deze Zeng, and Song Guo. 2020. A Learning-Based Incentive Mechanism for Federated Learning. *IEEE Internet of Things Journal* 7, 7 (2020), 6360–6368. https://doi.org/10.1109/JIOT.2020.2967772

[45] Yufeng Zhan, Peng Li, Zhihao Qu, Deze Zeng, and Song Guo. 2020. A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal* 7, 7 (2020), 6360–6368.

[46] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*.