基于激励机制的联邦学习优化算法

田有亮[1,2,3]，吴柿红[1,2]，李沓[1,2]，王林冬[1,2]，周骅[4]

（1. 贵州大学公共大数据国家重点实验室，贵州 贵阳 550025；2. 贵州大学计算机科学与技术学院，贵州 贵阳 550025；
3. 贵州大学密码学与数据安全研究所，贵州 贵阳 550025；4. 贵州大学大数据与信息工程学院，贵州 贵阳 550025）

摘 要：针对联邦学习的训练过程迭代次数多、训练时间长、效率低等问题，提出一种基于激励机制的联邦学习优化算法。首先，设计与时间和模型损失相关的信誉值，基于该信誉值，设计激励机制激励拥有高质量数据的客户端加入训练。其次，基于拍卖理论设计拍卖机制，客户端通过向雾节点拍卖本地训练任务，委托高性能雾节点训练本地数据从而提升本地训练效率，解决客户端间的性能不均衡问题。最后，设计全局梯度聚合策略，增加高精度局部梯度在全局梯度中的权重，剔除恶意客户端，从而减少模型训练次数。

关键词：联邦学习；激励机制；信誉值；拍卖策略；聚合策略

中图分类号：TN92

文献标志码：A

DOI: 10.11959/j.issn.1000−436x.2023095

Federated learning optimization algorithm based on incentive mechanism

TIAN Youliang[1,2,3], WU Shihong[1,2], LI Ta[1,2], WANG Lindong[1,2], ZHOU Hua[4]

1. State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China
2. College of Computer Science and Technology, Guizhou University, Guiyang 550025, China
3. Institute of Cryptography & Data Security, Guizhou University, Guiyang 550025, China
4. College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China

Abstract: Federated learning optimization algorithm based on incentive mechanism was proposed to address the issues of multiple iterations, long training time and low efficiency in the training process of federated learning. Firstly, the reputation value related to time and model loss was designed. Based on the reputation value, an incentive mechanism was designed to encourage clients with high-quality data to join the training. Secondly, the auction mechanism was designed based on the auction theory. By auctioning local training tasks to the fog node, the client entrusted the high-performance fog node to train local data, so as to improve the efficiency of local training and solve the problem of performance imbalance between clients. Finally, the global gradient aggregation strategy was designed to increase the weight of high-precision local gradient in the global gradient and eliminate malicious clients, so as to reduce the number of model training.

Keywords: federated learning, incentive mechanism, reputation value, auction strategy, aggregation strategy

---

收稿日期：2023−02−27；**修回日期**：2023−04−06

基金项目：国家重点研发计划基金资助项目（No.2021YFB3101100）；国家自然科学基金资助项目（No.U1836205, No.62272123）；贵州省高层次创新型人才基金资助项目（黔科合平台人才[2020]6008）；贵阳市科技计划基金资助项目（筑科合[2021]1-5, 筑科合[2022]2-4）；贵州省科技计划基金资助项目（黔科合平台人才[2020]5017, 黔科合支撑[2022]一般 065）；贵州大学人才引进基金资助项目（贵大人基合字[2015]-53）

Foundation Items: The Key Research and Development Program of China (No.2021YFB3101100), The National Natural Science Foundation of China (No.U1836205, No.62272123), Project of High-Level Innovative Talents of Guizhou Province (No.[2020]6008), Science and Technology Program of Guiyang (No.[2021]1-5, No.[2022]2-4), Science and Technology Program of Guizhou Province (No.[2020]5017, No.[2022]065), Guizhou University Talent Introduction Research Fund (No.GDRJHZ[2015]-53)

0 引言

机器学习在工业生产[1]、自动驾驶[2-3]、医疗卫生[4]、零售业等行业中得到了广泛应用。为了在充分利用各个企业的数据进行模型训练从而得到更精确结果的同时保护数据的隐私，联邦学习应运而生。联邦学习作为解决数据孤岛问题的有效方法，不需要把数据进行汇集就可以把存储在不同设备上的数据充分利用起来进行模型训练[5]。

联邦学习虽然具有很多优势，但是也不可避免地存在一些问题，例如，联邦学习训练轮次多、训练时间长，为提高训练效率，目前存在一些方案利用客户端的自利性设计激励机制，激励拥有高质量数据的客户端加入训练[6-7]，从而提升模型训练效率。其次，传统的联邦学习在客户端执行本地训练，但客户端之间性能不均衡，会导致客户端完成本地训练的时间差别大、到达服务器的局部模型的时间间隔大，从而造成设备之间相互等待，增加训练时间[8]，这被称为“流浪者”问题。为解决“流浪者”问题，有方案提出部分模型聚合方案，在该方案中，服务器在每轮训练中仅等待适当数量的设备模型，不需要等待所有局部模型[8]。最后，人们广泛接受的联邦平均（FedAvg）算法[9]没有考虑到各个客户端在训练中的贡献，从而导致训练需要进行多轮才能达到预期精度，这在一定程度上增加了训练时间，降低了模型训练效率。在实际应用中，任务发布者对模型训练的时间不是无限包容的，对于急需获得训练结果的任务发布者来说，超出时间训练得到的结果是没有意义的。因此优化联邦学习训练模型、提升训练效率是有必要的。

本文通过设计激励机制、拍卖方案和聚合策略提升模型训练效率。考虑到模型精度的动态性，基于合约理论设计信誉值计算机制，根据客户端每轮训练的具体贡献给予客户端奖励；同时，基于雾节点低时延高存储[10]的特点，设计拍卖方案，激励客户端把训练任务委托给高性能雾节点，从而提升本地训练效率；最后，增加高精度局部梯度在全局梯度中的占比，从而实现模型的优化。

本文主要贡献如下。

1) 设计与时间和模型损失相关的信誉值计算机制。基于该信誉值建立奖励机制，激励拥有高质量数据且高性能的客户端加入训练，只有使用高质量数据高效训练客户端才能获得高奖励。

2) 为解决具有高质量数据的客户端性能低、时延长的问题，基于拍卖理论设计拍卖策略，客户端通过向高性能雾节点拍卖本地训练任务，委托雾节点训练本地数据从而提升本地训练效率，解决客户端间的性能不均衡问题。

3) 设计全局梯度聚合策略，增加高精度局部梯度在全局梯度中的权重，剔除恶意客户端，从而减少模型训练次数。

1 相关工作

2016 年，谷歌[9]给出了联邦学习的定义，即联邦学习是机器学习的一种分布式训练范式，其不需要把数据进行汇集，在每一轮训练中仅传递模型之间的参数就可实现模型训练。联邦学习的这一优势使它在医疗保健[3]、工业制造[1]、自动驾驶[11]等领域中得到广泛应用，用于解决数据分散且私密的问题。虽然它有很多优点，但是也不可避免地存在一些问题[12-13]。

首先，传统的联邦学习假设客户端自愿奉献自己的数据加入训练[14]，然而，由于客户端的自利性，拥有高质量数据的客户端不愿意参与模型训练，这将会影响训练的精度与轮次，降低模型训练效率。为解决这个问题，Kang 等[15]提出使用主观逻辑计算各个客户端执行任务后的信誉值，基于合约理论设计激励机制，通过奖励贡献多的客户端获得更多的奖励，进而激励拥有高质量数据的客户端加入训练，由于合约理论的激励机制设计的合约是提前规定的，客户端只能选择是否接受合约，缺乏一定的灵活性，同时，基于主观逻辑模型设计信誉值存在主观性判断因素，没有量化的评价标准。Zeng 等[16]针对 Kang 等[15]存在的提前规定的合约不灵活性问题，提出多维采购拍卖方案，使客户端有更多机会提交任何资源组合和预期付款。Deng 等[17]针对 Kang 等[15]使用主观逻辑模型设计信誉值存在的问题，使用模型质量参数来计算信誉值，从而判断客户端的可靠性，提高信誉值在系统中的价值。不同于 Zeng 等[16]使用多维拍卖实现对客户端的选择，即选择拥有高质量数据的客户端加入训练，不同于 Deng 等[17]仅使用模型质量参数衡量信誉值，本文使用拍卖方式选择高性能雾节点为客户端进行本地训练，在进行信誉值计算时不仅考虑到当前轮次的数据质量，也兼顾历史训练的数据质量以及客户端执行相似任务获得的信誉值对当前任务信誉值的影响。

其次，客户端之间的性能不均衡使本地训练时间差大，从而导致局部梯度到达服务器的时间间隔大，客户端间相互等待，从而降低模型训练速度，这是并行计算中普遍存在的问题[18-19]。为缩短客户端间的等待时间，Niu 等[20]提出异步解决方案，在收到单个客户端的局部梯度后，立即更新全局梯度，并把该全局梯度传递给各个客户端，进而解决客户端性能不均衡的问题，然而当客户端间的数据分布不一致时，训练的结果将不正确。为解决 Liu 等[8]中的问题，Nishio 等[21]提出联邦学习的客户端选择（FedCS）协议过滤掉低性能客户端，但是该方案会与高性能客户端训练结果过度拟合。此外，Liu 等[8]提出聚合陈旧与常规模型以加快收敛速度，但是要求客户端在每轮训练的同时上传模型参数和梯度参数，这意味着传输的数据量是 FedAvg[9] 的 2 倍，因此数据传输时间将会增加。为了不增加数据传输时间，同时解决客户端间性能不均衡的问题，Liu 等[8]提出部分模型平均（FedPA）算法，服务器只聚合适当数量的客户端局部梯度，但是聚合的客户端数量针对不同的训练任务会有所变化，因此需要额外的工作确定参与聚合的客户端数量。

最后，传统的联邦学习使用 FedAvg[9]的方式聚合，忽略了使用高质量数据进行训练的客户端在训练中的贡献，从而增加训练的轮次，为解决该问题，Deng 等[17]提出具有质量意识的联邦学习（FAIR）方案，使用与数据质量相关的参数作为梯度聚合的权重，增加高质量数据的局部梯度在全局梯度中的占比，减少模型训练的轮次。不同于文献[9]的平均梯度聚合算法和文献[17]的聚合算法，本文使用模型质量及模型训练时间作为客户端贡献的衡量标准，设计聚合策略降低模型训练的轮次。

然而，目前没有方案同时从以上 3 个方面入手设计方案减少模型训练轮次，缩短一次迭代时间，从而实现联邦学习模型的优化和模型训练效率的提升。本文从以上 3 个方面考虑，通过设计信誉值[22]，激励具有高质量数据的客户端加入训练，设计聚合策略从而减少训练轮次，利用拍卖机制使低性能客户端可以通过委托低时延高性能雾节点进行训练，从而解决客户端性能不均衡的问题，提升本地训练效率。

2　基础知识

2.1　联邦学习

联邦学习可以实现各个客户端不需要汇集数据就可用客户端的数据进行模型训练[23]。具体操作是各个客户端从服务器下载全局参数并使用自己的本地数据进行模型训练，把训练得到的局部梯度上传至中央服务器，中央服务器聚合各个客户端的局部梯度后把全局梯度传送给客户端进行下一轮的模型训练，直到模型收敛。

神经网络可以表示为 $f(x,w)=y'$，其中，x 为客户端的输入，w 为模型参数，y' 为使用参数 w 和函数 f 得到的输出。整个训练集表示为 $D=\{\langle x_i,y_i\rangle,i=1,2,\cdots,T\}$，其中，$T$ 表示数据条数。数据集 D 的平均损失函数[24]为

$$\xi_f(D,w)=\frac{1}{|D|}\sum_{(x_i,y_i)\in D}\xi_f(x_i,y_i,w) \tag{1}$$

其中，对于特定的损失函数 $\xi_f(x,y,w)$，有 $\xi_f(x,y,w)=l\big(y,f(x,w)\big)=l(y,y')$。

联邦学习的训练目标是通过改变 w 最小化损失函数，其每轮迭代的计算式[24]为

$$w^{j+1}=w^j-\lambda\Delta\xi_f(D^j,w^j) \tag{2}$$

其中，λ 表示学习率，w^{j+1} 表示第 $j+1$ 轮训练后的模型参数，D^j 表示从数据集 D 中随机选择的子集。

在服务器端对每个客户端进行平均梯度聚合，计算式[8]为

$$w^j=\sum_{i=1}^{N}\frac{|D_i|}{|D|}w_i^j \tag{3}$$

其中，w_i^j 表示第 j 轮训练后客户端 i 的模型参数；D_i 表示客户端 i 的本地数据集，$i=1,2,\cdots,N$，$D=\{D_1,D_2,\cdots,D_N\}$ 表示联合数据集，N 表示客户端的数量。

2.2　设计目标

定义 1　个人理性。如果每个节点在每一轮训练中的收益非负，则该机制是个人理性的。

定义 2　真实性。形式上，对于每个节点，机制的真实出价 b_i^j 等于节点的学习成本 l_i^j，如果在每次竞拍中，节点不能通过不真实的出价（$\neg b_i^j>l_i^j$，$\neg b_i^j$ 表示不真实的出价）来获得更高的收益，则机制具有真实性[17]。

3　系统模型

系统模型如图 1 所示，主要由 5 个部分组成，分别为服务器、雾节点、客户端、区块链和任务发布者。

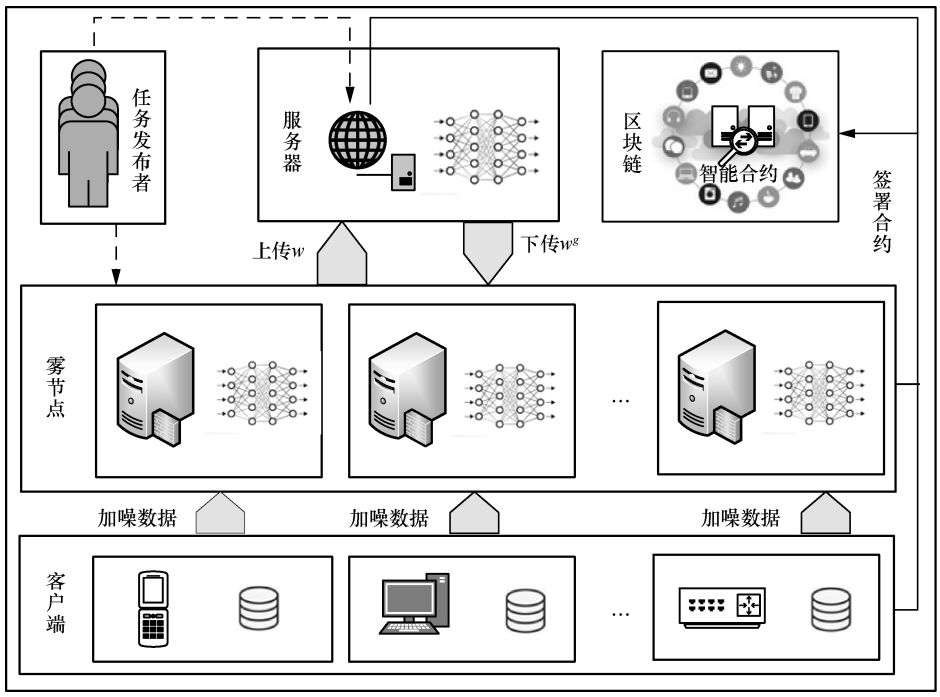

图 1　系统模型

1) 服务器。执行局部梯度聚合任务，并把训练得到的全局梯度传送至雾节点进行下一轮的训练，同时与客户端签署参与模型训练的合约。

2) 雾节点。执行本地训练任务，并把训练得到的局部模型上传至服务器，参与本地训练任务的竞拍，在拍卖获胜后与客户端签署本地训练任务委托合约。

3) 客户端。收集本地数据，发布本地训练任务拍卖，确定竞拍获胜的雾节点，与竞拍获胜的雾节点签署本地训练任务委托合约，并把参与训练的数据传送给竞拍获胜的雾节点。为防止数据的隐私泄露，客户端需对本地数据进行加噪后再向雾节点传送训练数据，同时与服务器签署参与模型训练的合约。

4) 区块链。部署智能合约，对客户端的信誉值进行管理，根据参与者在训练中的表现分发奖励。

5) 任务发布者。发布训练任务。

4　算法设计

为了提升训练的效率，设计与时间和模型损失相关的信誉值计算策略，并设计激励机制激励拥有高质量数据的客户端加入训练。为了提升客户端的本地训练效率，设计拍卖机制，提供客户端通过向雾节点拍卖本地训练任务从而提升本地训练效率的机会。最后，通过设计聚合算法，增加使用高质量数据进行训练的客户端在全局梯度中的占比，实现训练效率的提升。表 1 为本文涉及的参数和参数描述。本文方案由如下 6 个部分组成。

表 1　本文涉及的参数和参数描述

参数	描述
data_size	本地训练数据的大小
P_i	雾节点 i 竞拍的价格
T_i	雾节点 i 一次迭代的时间
c_n	执行一个数据样本的 CPU 周期数
f_i	设备 i 的频率
$E_i^{\text{cmp}}(f_i)$	客户端 i 用于计算的能量消耗
loss_i^n	客户端 i 在第 n 轮训练后的模型损失
loss_avg^n	第 n 轮训练后的平均模型损失
d	任务发布者与客户端签署合约需要交纳的押金
r	客户端与雾节点签署合约需要交纳的押金
task_x	任务 x
curret_task	当前任务
w_i^j	客户端 i 的第 j 轮训练梯度
w^j	第 j 轮训练的全局梯度
T_{ex}	任务发布者期望的总训练时间
T_r	实际训练的总时间
t_i^j	客户端 i 在第 j 轮训练所用的时间

1) 任务发布。任务发布者发布训练任务，并发布奖励规则和资源要求（包含加入训练的本地数据大小、本地训练的时间阈值、数据类型、训练最终需要达到的精度、期望的训练时间）；客户端根据自己的实际情况判断是否能从中获利并选择是否签署合约，签署合约的双方需交纳押金。

2) 本地数据加噪。为保护本地数据隐私，客户端根据自己的风险承受能力对本地数据进行加噪[21]，隐私预算越大，隐私保护程度越低，但是会获得更高的精度；隐私预算越小，隐私保护程度越高，但是会获得更低的精度[25-26]。每个客户端具体的隐私预算设置不是本文重点，因此不对数据的加噪细节进行详细阐述，只是使用相同的隐私预算值对数据进行加噪。

3) 本地训练任务拍卖。客户端向雾节点发布本地训练任务，雾节点参与竞拍，客户端选择能使其获得更高收益的雾节点执行本地训练任务。

4) 模型训练。训练开始前，客户端向竞拍获胜的雾节点传递训练数据，雾节点训练本地数据并上传局部梯度，在服务器对局部梯度进行聚合并向雾节点传递全局梯度，雾节点更新模型参数并进行下一轮训练。

5) 信誉值更新。每轮训练后，根据信誉值计算规则重新计算与时间和模型损失相关的信誉值。

6) 奖励结算。根据奖励机制计算每轮训练客户端的奖励，同时结算雾节点与客户端的奖励。

4.1 信誉系统

区别于传统的联邦学习信誉值计算机制，本文使用时间作为信誉值计算的输入，增加客户端对训练时长的重视，提升模型训练的效率，由此，本文的信誉值由 2 个部分组成：与模型损失相关的信誉值及与时间相关的信誉值。

模型损失在机器学习中经常作为学习准则与优化问题相联系，即通过最小化模型损失求解和评估模型。模型损失反映了训练结果的优劣，因此使用模型损失作为衡量训练数据质量的标准。客户端 i 在第 n 轮的数据质量计算式为

$$q_i^n = \frac{\text{loss\_avg}^n}{\text{loss}_i^n} \tag{4}$$

为使客户端在每一轮训练中都使用高质量数据进行训练，本文设计的与模型损失相关的信誉值（即损失信誉值）不仅受该轮数据质量的影响，还受其之前训练的数据质量的影响，训练轮次越新，对损失信誉值的影响越大，客户端 i 在第 n 轮训练后的信誉值为

$$r\_a_i^n = \sum_{j=1}^{n} w_i^j q_i^j \tag{5}$$

其中，$\sum_{j=1}^{n} w_i^j = 1$，且 $w_i^1 < w_i^2 < \cdots < w_i^n$。

此外，客户端的损失信誉值也受其在执行相似任务时获得的信誉值的影响，由于任务具有时效性，相近时间内执行的相似任务对当前任务更有价值，时效性更高[15]，当前任务的信誉值受该任务信誉值的影响越大。因此，本文使用执行相似任务与执行当前任务的时间差作为相似任务的实效系数，用于计算损失信誉值，即在第 n 轮训练结束时客户端 i 的信誉值为

$$r\_a_i^n = \sum_{j=1}^{n} w_i^j q_i^j + w_i^0 \frac{\sum_{x\in U} \frac{1}{T_x^{'}} \text{sim}(\text{curret\_task}, \text{task}_x) r\_a_i}{\sum_{x\in U} \frac{1}{T_x^{'}} \text{sim}(\text{curret\_task}, \text{task}_x)} \tag{6}$$

其中，$w_i^0 < w_i^1 \cdots < w_i^n$，$T_x^{'}$ 表示执行相似任务与执行当前任务的时间差，U 表示与当前任务相似的任务集合，$\sum_{j=1}^{n} w_i^j = 1$，$\text{sim}(h,i)$ 表示任务 h 与任务 i 的相似度。本文使用修正余弦函数衡量该相似度[27-28]，即

$$\text{sim}(h,i) = \frac{\sum_{j\in\Gamma} \left(r_{h,j}^t - \bar{r}_h^t\right)\left(r_{i,j}^t - \bar{r}_i^t\right)}{\sqrt{\sum_{j\in H} \left(r_{h,j}^t - \bar{r}_h^t\right)^2} \sqrt{\sum_{j\in I} \left(r_{i,j}^t - \bar{r}_i^t\right)^2}} \tag{7}$$

其中，$0 < \text{sim}(h,i) < 1$，H 和 I 分别表示执行任务 h 与任务 i 的客户端集合，$\Gamma = H \cap I$ 表示同时执行任务 h 与任务 i 的客户端集合，$\bar{r}_h^t$ 和 $\bar{r}_i^t$ 分别表示在 t 时刻执行任务 h 与任务 i 的所有客户端的平均信誉值，$r_{h,j}^t$ 和 $r_{i,j}^t$ 分别表示在 t 时刻执行任务 h 与任务 i 的客户端 j 的信誉值。

为防止恶意客户端在训练的某一轮次中加入低质量数据影响训练结果，通过设计信誉值计算策

略提高客户端使用低质量数据训练时信誉值的降低速度，减缓客户端使用高质量数据训练数据时信誉值的增长速度。设定数据质量阈值 $q_{\min}$，与模型损失相关的信誉值更新式为

$$r\_a_i^n = w_i^0 \frac{\sum_{x\in U}\frac{1}{T_x^{'}}\mathrm{sim}(\mathrm{curret\_task},\mathrm{task}_x)r\_a_i}{\sum_{x\in U}\frac{1}{T_x^{'}}\mathrm{sim}(\mathrm{curret\_task},\mathrm{task}_x)} + \begin{cases} \sum_{j=1}^{n} w_i^j q_i^j, & q_i^n \geqslant q_{\min} \\ q_i^n \sum_{j=1}^{n} w_i^j q_i^j, & q_i^n < q_{\min} \end{cases} \tag{8}$$

与时间相关的信誉值（即时间信誉值）同样考虑当客户端的训练时间高于阈值 $t_{\max}$ 时，提高时间信誉值的下降速度。由于时间信誉值的计算需要考虑任务发布者期望的总训练时间，而不同任务发布者期望的训练总时间有所差异，因此时间信誉值不考虑客户端执行其他任务时的表现。具体计算式为

$$r\_t_i^n = \begin{cases} \sum_{j=1}^{n} w\_t_i^j \frac{T}{t_i^j}, t_i^n \leqslant t_{\max} \\ \frac{t_{\max}}{t_i^n}\sum_{j=1}^{n} w\_t_i^j \frac{T}{t_i^j}, t_i^n > t_{\max} \end{cases} \tag{9}$$

其中，$\sum_{j=1}^{n} w\_t_i^j = 1, w\_t_i^1 < w\_t_i^2 < \cdots < w\_t_i^n$，$T = \frac{T_{\mathrm{ex}}}{n}$ 为任务发布者期望的每轮训练时间。式(9)中的信誉值越小，说明一次迭代的时间越长。

总信誉值与时间信誉值和损失信誉值相关，只有当 $r\_a_i^n$ 和 $r\_t_i^n$ 都高时总信誉值才最优，存在一方信誉值低则总信誉值低。具体计算式[27,29]为

$$r\_at_i^n = r\_a_i^n r\_t_i^n \tag{10}$$

4.2 高质量数据激励机制

每轮本地迭代时间[30]为

$$T_i = \frac{c_i D_i}{f_i} \tag{11}$$

其中，$c_i D_i$ 为客户端 i 运行一次本地迭代所需的 CPU 周期数。

用于本地训练的能量消耗为[30]

$$E_i^{\mathrm{cmp}}(f_i) = \frac{\alpha_i}{2} c_i D_i f_i^2 \tag{12}$$

由式(11)和式(12)可得

$$E_i^{\mathrm{cmp}}(f_i) = \frac{\alpha_i}{2} c_i D_i f_i^2 = \frac{\alpha_i}{2}\frac{c_i^3 D_i^3}{T_i^2} \tag{13}$$

根据个人理性定义可知，客户端只在其收益非负时才愿意加入训练，所以满足

$$U_i = P_i - E_i^{\mathrm{cmp}}(T_i) > 0 \tag{14}$$

客户端挑选高质量数据参与模型训练，使用模型的总信誉值作为支付函数的输入，由式(13)可知，本地训练消耗的能量与训练时间的平方成反比，所以客户端 i 在第 n 轮训练后的支付函数为

$$P_i(r\_a_i^n, r\_t_i^n) = \begin{cases} \psi(r\_t_i^n)^2 r\_a_i^n, & r\_t_{\min} \leqslant r\_t_i^n \text{且} \\ & r\_a_{\min} \leqslant r\_a_i^n \\ 0, & r\_t_i^n < r\_t_{\min} \text{或} \\ & r\_a_i^n < r\_a_{\min} \end{cases} \tag{15}$$

其中，$r\_t_{\min}$ 为容忍的最小时间信誉值，当该值高于该信誉值时客户端将获得奖励，否则客户端将不会获得奖励；$r\_a_{\min}$ 为容忍的最小模型损失信誉值，只有当信誉值不低于该值时，客户端才能获得奖励。最终只有当客户端训练的损失值低且花费的训练时间短时才能获得更高的奖励。

针对客户端的总体表现，对训练轮次减少使总体时间缩短的客户端 i 给予奖励，具体为

$$P_i^{'}(r\_at_i) = \phi(T_{\mathrm{ex}} - T_r) r\_at_i \tag{16}$$

其中，ϕ 为系数，r_at_i 为最后一轮训练后客户端 i 的信誉值。

客户端 i 的总收益为

$$\mathrm{Utility}_i = \sum_{n=1}^{N} P_i(r\_a_i^n, r\_t_i^n) + P_i^{'}(r\_at_i) - \sum_{n=1}^{N} E_{i,n}^{\mathrm{cmp}}(f_i) \tag{17}$$

具体激励算法如算法 1 所示，为防止拥有低质量数据以及故意拖延训练时间的恶意客户端加入训练并对模型的精度和效率造成不利影响，本文设计了阈值与惩罚机制，即签署参与训练合约的客户端需要交纳押金，若客户端在训练期间加入低质量数据或故意拖延训练时间，影响模型训

练的精度或效率，则该客户端将会被没收押金且不会获得额外奖励，这确保了任务发布者的开销最小化，同时迫使拥有低质量数据的恶意客户端不敢加入训练，也保证了训练过程中客户端不敢拖延训练时间。

算法 1　激励算法

初始化　训练轮次 N，$n=1$，合约押金 d，总收益 $P_z=0$

1) 任务发布者发布任务与合约，对于客户端 i:

2) 客户端 i 与任务发布者在 t 时刻前签署合约并交纳押金 d

3) 筛选符合要求的客户端加入训练并为不符合要求的客户端退回押金

4) while $n \leqslant N$

5)　　if　$r\_t_i^n < r\_t_{\min} \| r\_a_i^n < r\_a_{\min}$

6)　　没收客户端 i 的押金 d，并把客户端 i 从训练模型中剔除

7)　　else

8)　　训练模型

9)　　end if

10)　　获得一次迭代时间并计算局部梯度和平均梯度损失

11)　　计算并更新客户端 i 的信誉值 $r\_a_i^n$，$r\_t_i^n$，$r\_at_i^n$

12)　　计算客户端 i 的收益 $P_z=P_z+P_i(r\_a_i^n, r\_t_i^n)$

13)　$n++$

14)退回押金并结算奖励 $P_z=P_i'(r\_at_i)+P_z$

15) end while

4.3　拍卖策略

雾节点可提供本地化服务，从而实现低时延和上下文感知[31]。为提升本地训练速度，解决客户端性能不均衡、设备计算能力有限和电池导致模型参与者之间相互等待的问题，使用雾节点执行本地训练任务，提升本地训练的效率，降低通信时延，具体拍卖过程如图 2 所示。然而，由于雾节点也具有自利性，不可能无偿为客户端训练本地数据，因此，本文为客户端设计拍卖策略，通过把训练任务委托给雾节点，让高性能雾节点为客户端执行训练任务，从而降低客户端之间等待的时间，提升一次迭代的速度。由于雾节点性能的稳定性，因此拍卖只需在训练前执行一次。具体过程如下。

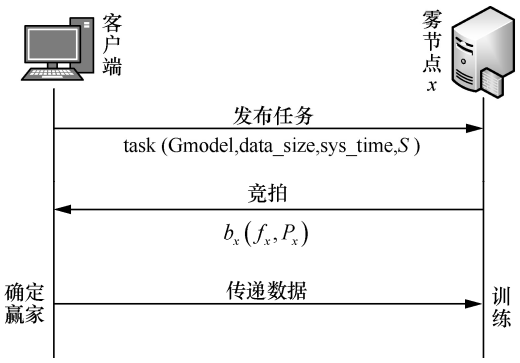

图 2　拍卖过程

1) 发布任务

每个客户端在确定能加入训练后，在训练开始前向雾节点发布本地训练任务 task(Gmodel,data_size, sys_time,S)，训练任务包括全局模型 Gmodel、训练数据量 data_size、竞拍结束时间 sys_time、聚合服务器 S。同时发布雾节点参与竞拍需提供的信息 $b(f,P)$，即频率 f 和支付函数 p。

2) 竞标

达到要求的雾节点 x 参与竞拍，根据训练开销提交竞拍 $b_x(f_x,P_x)$。雾节点不能以更高的竞标价获得本地训练任务，且根据个人理性定义，雾节点不可能在收益低于其自身训练开销时接受训练任务，所以竞拍胜者的竞拍价是唯一的。

3) 确定赢家

客户端 i 根据雾节点 x 的出价计算自身可获得收益，即

$$\text{Utility}_i = P_i(r\_a_i, r\_t_x) - P_x \tag{18}$$

每个客户端按收益最大化原则选择一个雾节点代替其训练，频率高、支付最低的雾节点将成为最终赢家执行本地训练任务。此时，客户端的信誉值取系统当前对应客户端的信誉值，并选择能使其获得最大收益的雾节点作为竞拍获胜者执行训练任务。同时，雾节点需要与客户端签署委托合约并交纳押金，训练完成后则退还押金，一旦雾节点违背承诺不及时完成训练任务，押金就会传递给客户端以弥补雾节点违背承诺给客户端带来的损失。

4) 传递数据进行训练

雾节点接近边缘设备是一种在网络边缘提供计算服务的云扩展[32]，所以客户端向雾节点传送数据的开销在本文中不进行详细考虑。客户端从本地选择数据量为 data_size 的高质量数据，并对数据进行加噪后传递给雾节点。之后，本地训练任务由雾节点完成，具体如算法 2 所示。

算法 2 拍卖算法

初始化 训练轮次 N，$n=1$，合约押金 r，总收益 $P_z=0$

① 客户端发布本地训练任务 task

② 雾节点 i 在 sys_time 内决定是否参与竞拍并提交竞价 $b_i(f_i,P_i)$

③ 客户端根据所有雾节点的竞价计算自身可能获得的收益并获得收益集

④ 客户端从收益集中选择使其收益最大的雾节点参与本地训练任务

⑤ 客户端与竞拍成功的雾节点 i 签署委托合约并交纳押金 r

⑥ 客户端向竞拍成功的雾节点传递数据量为 data_size 的数据

⑦ while $n \leqslant N$

⑧ 训练本地数据

⑨ if 雾节点按时完成本地训练

⑩ 计算雾节点 i 的收益 $P_z=P_z+P_i$

⑪ else

⑫ 扣押雾节点押金，重新挑选雾节点进行本地训练

⑬ end if

⑭ $n++$

⑮ 结算雾节点奖励并退还押金 r

⑯ end while

定理 1 拍卖是激励相容且真实的。

证明 由于雾节点竞拍成功后需要交纳押金才可进行训练，如果雾节点谎报自己的频率，训练结果出现偏差，雾节点的押金就会被没收。根据个人理性定义，雾节点不会谎报自己的频率。雾节点有出更高的价格 $P_i^{"}$ 获得更高收益的动机，然而在拍卖中不真实的出价将会失去获胜的机会，即 $P_i^{"}>P_i$，有 $P_i(r\_a_i^n,r\_t_i^n)-P_i>P_i(r\_a_i^n,r\_t_i^n)-P_i^{"}$。因此雾节点会真实出价，证毕。

4.4 聚合策略

传统的联邦学习采用梯度平均的方式计算全局梯度[9,24]，该方式忽略了高精度局部梯度在全局梯度的贡献。为解决这个问题，Li 等[11]提出使用信誉值作为全局梯度聚合的权重，但是该信誉值的计算以聚合后的全局梯度为参考，局部梯度与全局梯度越接近则信誉值将越高，此时高质量数据训练的局部模型同样会远离全局梯度[17]，计算得到的信誉值将不能反映其贡献程度。针对此问题，本文设计了聚合策略。由于训练的模型损失函数值反映了数据的质量，模型损失函数值越大，训练得到的模型越远离预期模型；模型损失函数值越小，训练得到的模型越接近预期模型[33-34]。通过使用模型损失函数值作为全局梯度聚合的权重，增加高精度局部模型在全局模型中的占比，如式(19)所示。

$$w^n=\sum_{i=1}^{I}\frac{q_i^n\left|D_i^n\right|}{|D|}w_i^n \tag{19}$$

其中，I 为客户端数量。聚合算法如算法 3 所示。

算法 3 聚合算法

初始化 模型参数 w，训练轮次 N，学习率 λ，$n=1$

1) while $n \leqslant N$ 对于客户端

2) 雾节点训练本地模型，计算模型损失并上传至服务器

3) 服务器计算全局梯度 w^n 返回雾节点

4) $n++$

5) end while

5 实验分析

5.1 信誉值评估

设置任务发布者每次迭代的期待时间 $T_{\mathrm{ex}}=0.4\,s$，本文方案中的模型损失、一次迭代的时间以及系统信誉值之间的关系如图 3 所示。由图 3 可知，系统信誉值随模型损失和一次迭代的时间的降低而上升，只有当模型损失和一次迭代的时间都最低时，系统信誉值才能达到最大值。模型损失和一次迭代的时间中存在一方增加，系统信誉值都不能达到最优，即本文提出的信誉值计算方案可衡量模型训练中模型的精度和训练时间，使用该信誉值作为对客户端进行奖励的衡量标准具有可行性。

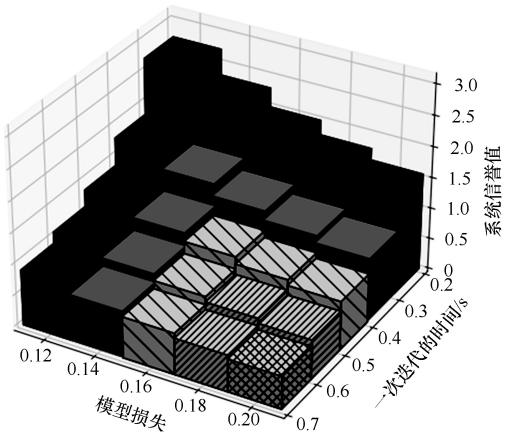

图 3 模型损失、一次迭代的时间以及系统信誉值之间的关系

5.2　实验评估

实验使用 100 台客户端（具有 1.80 GHz CPU 和 8 GB RAM 的 64 位个人计算机）参与训练，同时使用高性能的计算机（具有 3.00 GHz CPU 和 8 GB RAM 的 64 位个人计算机）充当雾节点，同时使用 LeNet 对 MNIST 和 CIFAR-10 数据集进行实验。

1) 激励机制性能

为测试激励机制的有效性，使用平均梯度聚合算法对全局梯度进行聚合，并与文献[17]的 FAIR 方案进行对比，测试 20 轮训练后方案所能达到的精度，结果如表 2 所示。由表 2 可知，本文设计的激励机制与 FAIR 方案在训练相同轮次时的精度相近。同时，使用平均梯度聚合算法进行 20 轮训练，对 2 种方案需要花费的时间进行测试，如表 3 所示。由于本文设计的激励方案与客户端的性能相关，在数据质量与客户端性能双重因素的影响下，本文方案的开销低于 FAIR 方案。

表 2　20 轮训练后方案所能达到的精度

方案	MNIST	CIFAR-10
本文方案	96.3%	96.1%
FAIR 方案	51.8%	51.8%

表 3　20 轮训练后方案所需要的时间

方案	MNIST/s	CIFAR-10/s
本文方案	421.4	391.58
FAIR 方案	611.4	568.85

2) 训练轮次

将本文方案分别与文献[9]中的 FedAvg 方案和文献[17]中的 FAIR 方案进行对比，结果分别如图 4 和图 5 所示。由图 4 可知，使用 MNIST 进行训练时（指定训练精度需达到 96%），本文方案所需要的训练轮次最少。由图 5 可知，使用 CIFAR 数据集进行训练时（指定训练精度需达到 60%），该结论同样成立。如表 4 所示，相比于 FedAvg 方案，本文方案在数据集 MNIST 和 CIFAR-10 训练轮次减少 10%以上。这是因为本文设计的聚合算法增加了高质量数据拥有者的局部梯度在全局梯度中的占比，从而减少了模型训练轮次。

表 4　达到指定精度所需的训练轮次

方案	MNIST（96%）/轮	CIFAR-10（60%）/轮
本文方案	16	86
FedAvg 方案	20	98
FAIR 方案	17	91

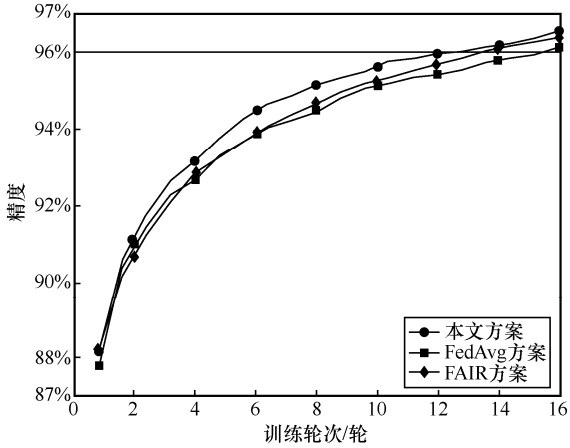

图 4　MNIST 数据集达到指定精度的训练轮次

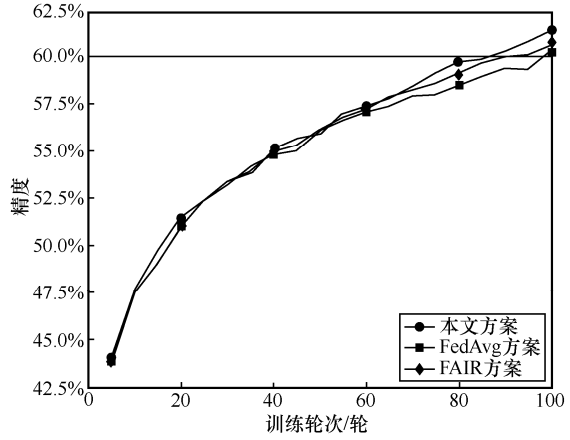

图 5　CIFAR-10 数据集达到指定精度的训练轮次

3) 一次迭代的时间

本节使用 MNIST 数据集和 CIFAR-10 数据集进行训练，将本文方案与 FedAvg 方案和 FAIR 方案对比，结果如图 6 和图 7 所示。本文方案一次迭代的时间在 16 s 上下浮动，明显低于 FedAvg 方案和 FAIR 方案一次迭代的时间。这是因为雾节点性能高，从而缩短了本地训练时间，降低了局部梯度到达服务器的时间差。

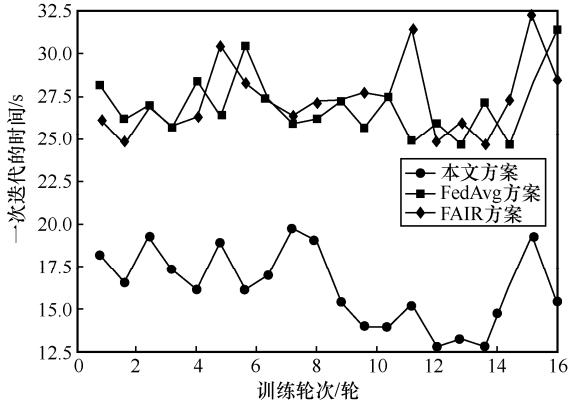

图 6　MNIST 数据集一次迭代的时间

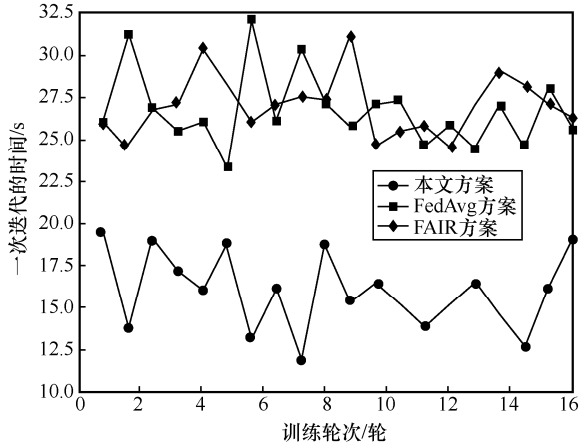

图 7 CIFAR-10 数据集一次迭代的时间

4) 训练总时间对比

完成指定精度的训练总时间如图 8 和表 5 所示。使用 MNIST 数据集进行训练，对本文方案、FedAvg 方案和 FAIR 方案在模型精度达到 96%时所用的训练总时间进行对比。由图 8 可知，本文方案的训练总时间最低，这是因为本文方案减少了模型训练轮次，缩短了本地训练时间和客户端间的等待时间，从而缩短了训练总时间。使用 CIFART-10 数据集进行训练，对本文方案、FedAvg 方案和 FAIR 方案在模型精度达到 60%时所用的训练总时间进行对比，该结论同样成立。每次任务客户端只需执行一次拍卖和本地数据委托，表 5 中加粗部分为拍卖开销和客户端向雾节点传送训练数据的开销。由表 5 可知，与 FedAvg 方案相比，本文方案在 2 个不同数据集下的训练总时间降低 30%以上；与 FAIR 方案相比，本文方案在 2 个不同数据集下的训练总时间降低 20%以上。

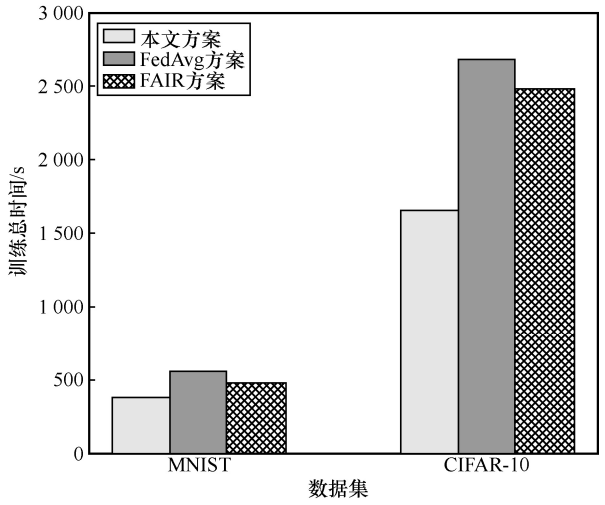

图 8 完成指定精度的训练总时间

表 5 训练总时间

方案	MNIST/s	CIFAR-10/s
本文方案	296.03+**71.43**	1 579.82+**71.43**
FedAvg 方案	550.37	2 682.26
FAIR 方案	467.92	2 490.69

5.3 性能对比

本文方案同时考虑了客户端激励、流浪者效应、聚合策略 3 个方面，设计方案使其解决客户端性能不均衡、客户端自利性和数据泄露问题，从而提升模型训练的效率。通过与 FedPA 方案、FedAvg 方案、FAIR 方案分别在以上 3 个方面进行对比（如表 6 所示），本文方案存在多方面优势：首先，针对拥有高质量数据的客户端不愿意加入训练，设计有效的激励机制；然后，使用拍卖算法解决流浪者效应；最后，通过聚合策略增加高质量数据拥有者的局部梯度在全局梯度中的占比，从而提高模型的整体性能。

表 6 性能对比

方案	客户端激励	流浪者效应	聚合策略
FedPA 方案	×	√	×
FedAvg 方案	×	×	×
FAIR 方案	√	×	√
本文方案	√	√	√

6 结束语

本文基于信誉值，通过设计拍卖机制、激励机制和聚合策略，激励拥有高质量数据的客户端加入训练，同时，通过激励客户端向雾节点拍卖训练任务解决流浪者效应问题。其次，设计与数据质量相关的聚合策略增加拥有高质量数据的客户端在训练中的贡献，从而减少模型训练轮次，缩短局部梯度到达服务器的时间差，最终缩短训练时间，提升模型效率。最后，分析证明本文方案的正确性，实验结果表明，与已有方案相比，本文方案的训练轮次减少，训练总时间降低。未来工作将致力于联邦学习的进一步优化。

参考文献：

[1] GE Z Q, SONG Z H, DING S X, et al. Data mining and analytics in the process industry: the role of machine learning[J]. IEEE Access, 2017, 5: 20590-20616.

[2] MOHASSEL P, ZHANG Y P. Secure ML: a system for scalable privacy-preserving machine learning[C]//Proceedings of 2017 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2017: 19-38.

[3] ZHANG Y, XU C X, LI H W, et al. HealthDep: an efficient and secure deduplication scheme for cloud-assisted eHealth systems[J]. IEEE Transactions on Industrial Informatics, 2018, 14(9): 4101-4112.

[4] AHISKA K, OZGOREN M K, LEBLEBICIOGLU M K. Autopilot design for vehicle cornering through icy roads[J]. IEEE Transactions on Vehicular Technology, 2018, 67(3): 1867-1880.

[5] YANG Q, LIU Y, CHEN T J, et al. Federated machine learning: concept and applications[J]. ACM Transactions on Intelligent Systems and Technology, 2019, 10(2): 1-19.

[6] PANDEY S R, TRAN N H, BENNIS M, et al. A crowdsourcing framework for on-device federated learning[J]. IEEE Transactions on Wireless Communications, 2020, 19(5): 3241-3256.

[7] ZHAN Y F, LI P, QU Z H, et al. A learning-based incentive mechanism for federated learning[J]. IEEE Internet of Things Journal, 2020, 7(7): 6360-6368.

[8] LIU J, WANG J H, PONG C, et al. FedPA: an adaptively partial model aggregation strategy in Federated Learning[J]. Computer Networks, 2021, 199: 108468.

[9] MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[J]. arXiv Preprint, arXiv: 1602.05629, 2016.

[10] BONOMI F, MILITO R, ZHU J, et al. Fog computing and its role in the Internet of things[C]//Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing. New York: ACM Press, 2012: 13-16.

[11] LI Y J, TAO X F, ZHANG X F, et al. Privacy-preserved federated learning for autonomous driving[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(7): 8423-8434.

[12] KIM H, PARK J, BENNIS M, et al. Blockchained on-device federated learning[J]. IEEE Communications Letters, 2020, 24(6): 1279-1283.

[13] 陈明鑫, 张钧波, 李天瑞. 联邦学习攻防研究综述[J]. 计算机科学, 2022, 49(7): 310-323.
CHEN M X, ZHANG J B, LI T R. Survey on attacks and defenses in federated learning[J]. Computer Science, 2022, 49(7): 310-323.

[14] LI Y P, COURCOUBETIS C, DUAN L J. Recommending paths: follow or not follow?[C]//Proceedings of IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2019: 928-936.

[15] KANG J W, XIONG Z H, NIYATO D, et al. Incentive mechanism for reliable federated learning: a joint optimization approach to combining reputation and contract theory[J]. IEEE Internet of Things Journal, 2019, 6(6): 10700-10714.

[16] ZENG R F, ZHANG S X, WANG J Q, et al. FMore: an incentive scheme of multi-dimensional auction for federated learning in MEC[C]//Proceedings of 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS). Piscataway: IEEE Press, 2021: 278-288.

[17] DENG Y H, LYU F, REN J, et al. FAIR: quality-aware federated learning with precise user incentive and model aggregation[C]//Proceedings of IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2021: 1-10.

[18] LIAN X R, HUANG Y J, LI Y C, et al. Asynchronous parallel stochastic gradient for nonconvex optimization[C]//Proceedings of the 28th International Conference on Neural Information Processing Systems. New York: ACM Press, 2015: 2737-2745.

[19] DAI W, KUMAR A, WEI J L, et al. High-performance distributed ML at scale through parameter server consistency models[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2015: 79-87.

[20] NIU F, RECHT B, RE C, et al. HOGWILD!: a lock-free approach to parallelizing stochastic gradient descent[J]. Advances in Neural Information Processing Systems, 2011, 24: 693-701.

[21] NISHIO T, YONETANI R. Client selection for federated learning with heterogeneous resources in mobile edge[C]//Proceedings of 2019 IEEE International Conference on Communications (ICC). Piscataway: IEEE Press, 2019: 1-7.

[22] 余佳仁, 田有亮, 林晖. 基于信誉管理模型的矿工类型鉴别机制设计[J]. 网络与信息安全学报, 2022, 8(1): 128-138.
YU J R, TIAN Y L, LIN H. Design of miner type identification mechanism based on reputation management model[J]. Chinese Journal of Network and Information Security, 2022, 8(1): 128-138.

[23] 王勇, 李国良, 李开宇. 联邦学习贡献评估综述[J]. 软件学报, 2023, 34(3): 1168-1192.
WANG Y, LI G L, LI K Y. Survey on contribution evaluation for federated learning[J]. Journal of Software, 2023, 34(3): 1168-1192.

[24] XU G W, LI H W, LIU S, et al. VerifyNet: secure and verifiable federated learning[J]. IEEE Transactions on Information Forensics and Security, 2020, 15: 911-926.

[25] WANG S Q, TUOR T, SALONIDIS T, et al. When edge meets learning: adaptive control for resource-constrained distributed machine learning[C]//Proceedings of IEEE 2018 IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2018: 63-71.

[26] ZHENG S Y, CAO Y, YOSHIKAWA M. Incentive mechanism for privacy-preserving federated learning[J]. arXiv Preprint, arXiv: 2106.04384, 2021.

[27] MIAO Y B, LIU Z T, LI H W, et al. Privacy-preserving Byzantine-robust federated learning via blockchain systems[J]. IEEE Transactions on Information Forensics and Security, 2022, 17: 2848-2861.

[28] XIONG L, LIU L. PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(7): 843-857.

[29] HUANG C Y, WANG Z Y, CHEN H X, et al. RepChain: a reputation-based secure, fast, and high incentive blockchain system via sharding[J]. IEEE Internet of Things Journal, 2021, 8(6): 4291-4304.

[30] TRAN N H, BAO W, ZOMAYA A, et al. Federated learning over wireless networks: optimization model design and analysis[C]//Proceedings of 2019 IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2019: 1387-1395.

[31] ABDALI T A N, HASSAN R, AMAN A H M, et al. Fog computing advancement: concept, architecture, applications, advantages, and open issues[J]. IEEE Access, 2021, 9: 75961-75980.

[32] ZHOU C Y, FU A M, YU S, et al. Privacy-preserving federated learning in fog computing[J]. IEEE Internet of Things Journal, 2020, 7(11): 10782-10793.

[33] WAN C B, JIN F S, QIAO Z, et al. Unsupervised active learning with loss prediction[J]. Neural Computing and Applications, 2023, 35(5): 3587-3595.

[34] FU A M, ZHANG X L, XIONG N X, et al. VFL: a verifiable federated learning with privacy-preserving for big data in industrial IoT[J]. IEEE Transactions on Industrial Informatics, 2022, 18(5): 3316-3326.

[作者简介]

田有亮（1982− ），男，贵州盘州人，博士，贵州大学教授、博士生导师，主要研究方向为博弈论、密码学与安全协议、大数据隐私保护。

吴柿红（1995– ），女，贵州福泉人，贵州大学硕士生，主要研究方向为联邦学习、密码学等。

李沓（1998– ），男，贵州盘州人，贵州大学博士生，主要研究方向为密码学与区块链技术。

王林冬（1997– ），男，浙江杭州人，贵州大学硕士生，主要研究方向为数字水印、信息安全等。

周骅（1978– ），男，江苏无锡人，博士，贵州大学副教授、硕士生导师，主要研究方向为物联网安全、硬件安全机制、电路与系统。