软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007622] [CSTR: 32375.14.jos.007622]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于安全多方计算的隐私保护异构联邦学习参与方选择*

刘腾飞[1], 杨安家[1], 翁 健[1], 陈泯融[2], 刘 逸[1], 曾 璜[3]

[1](暨南大学 网络空间安全学院, 广东 广州 511443)

[2](华南师范大学 计算机学院, 广东 广州 510631)

[3](空军预警学院, 湖北 武汉 430019)

通信作者: 杨安家, E-mail: anjiayang@gmail.com

摘 要: 联邦学习允许众多客户端利用其本地数据联合训练模型而不暴露各方的真实数据, 与传统机器学习方法相比, 避免了数据迁移导致的数据泄露和滥用问题. 然而, 在实际应用中, 客户端可能具有异构的数据分布和系统功能, 这会导致模型性能和训练效率下降. 通过选择一个“良好的”客户端子集作为联邦学习参与方可以有效提高全局模型的性能和收敛速度. 而一些研究者发现, 恶意敌手可以利用客户端的本地训练损失或梯度等一些相关信息来推断其隐私数据, 目前的异构联邦学习参与方选择方案并没有应对这种隐私泄露风险的解决方法. 为此, 设计了一种基于安全多方计算的隐私保护异构联邦学习参与方选择协议, 利用 3PC 秘密共享技术来保证训练过程的数据隐私以及联合模型的准确度, 同时还提出了一个安全 top-k 搜索协议来避免参与方选择过程中泄露任何隐私信息. 对协议的安全性进行分析, 证明了该协议可以满足安全需求, 并且开展相关实验, 实验结果表明相比于未使用隐私保护的异构联邦学习方案, 所提出方案的各方的平均计算与通信时间开销仅增加了 2.09%.

关键词: 隐私保护; 联邦学习; 参与方选择; 数据异构性; 安全多方计算

中图法分类号: TP18

中文引用格式: 刘腾飞,杨安家,翁健,陈泯融,刘逸,曾璜.基于安全多方计算的隐私保护异构联邦学习参与方选择.软件学报.http://www.jos.org.cn/1000-9825/7622.htm

英文引用格式: Liu TF, Yang AJ, Weng J, Chen MR, Liu Y, Zeng H. Privacy-preserving Participant Selection Based on Secure Multi-party Computation in Heterogeneous Federated Learning. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7622.htm

Privacy-preserving Participant Selection Based on Secure Multi-party Computation in Heterogeneous Federated Learning

LIU Teng-Fei[1], YANG An-Jia[1], WENG Jian[1], CHEN Min-Rong[2], LIU Yi[1], ZENG Huang[3]

[1](College of Cyber Security, Jinan University, Guangzhou 511443, China)

[2](School of Computer Science, South China Normal University, Guangzhou 510631, China)

[3](Air Force Early Warning College, Wuhan 430019, China)

Abstract: Federated learning enables numerous clients to collaboratively train a model using their local data without exposing the raw data of individual parties, thereby avoiding data leakage and misuse caused by data migration in traditional machine learning paradigms. However, in practical scenarios, clients often exhibit heterogeneous data distributions and diverse system capabilities, which degrade model performance and training efficiency. Selecting a high-quality subset of clients as participants in federated learning can effectively improve global model performance and accelerate convergence. Nevertheless, existing studies show that malicious adversaries can exploit information such as local training losses or gradients to infer sensitive private data, while current participant selection strategies for

* 基金项目: 国家科技创新 2030—重大项目 (2021ZD0112802); 国家自然科学基金 (62332007, 62572214, 62072215, U23A20303, U22B2028, 62302194, 62573201); 西藏自治区科技重大专项 (XZ202201ZD0006G); 深圳市科技重大专项 (KJZD20240903101108011); 广州市基础与应用基础项目 (2024A03J0405, 2024A04J3458); 广东省数据安全与隐私保护重点实验室 (2023B1212060036)
收稿时间: 2025-02-19; 修改时间: 2025-09-23, 2025-11-20; 采用时间: 2025-12-28; jos 在线出版时间: 2026-04-29

heterogeneous federated learning fail to adequately address such privacy leakage risks. To tackle this challenge, this study designs a privacy-preserving participant selection protocol for heterogeneous federated learning based on secure multi-party computation. By leveraging three-party computation (3PC) secret-sharing techniques, the proposed protocol ensures data privacy during training while maintaining the accuracy of the jointly trained model. Furthermore, a secure top-*k* search protocol is introduced to prevent privacy leakage during the participant selection phase. The security of the proposed protocol is formally analyzed, demonstrating that the required security properties are satisfied. Experimental results indicate that, compared with heterogeneous federated learning schemes without privacy preservation, the proposed approach increases the average computational and communication overhead across all parties by 2.09%.

Key words: privacy-preserving; federated learning (FL); participant selection; data heterogeneity; secure multi-party computation

联邦学习 (federated learning, FL) 允许中央服务器和持有训练数据的客户端联合训练出模型而不暴露各客户端的真实数据. 与传统机器学习方法相比, FL 避免了数据迁移导致的数据泄露和滥用问题[1], 因此逐渐受到研究者的青睐. 然而, 在实际应用中, 客户端的数据可能是非独立同分布的, 已经有研究证明这种数据异构性会导致模型漂移, 最终降低全局模型性能. 另一方面, 客户端在设备计算能力、网络带宽等系统功能方面存在差异, 因此联邦学习模型训练过程在系统性能方面也会出现显著的异构性, 导致客户端很难同时执行联邦学习训练或测试, 进而降低模型训练效率. 且当 FL 涉及大量客户端时, 这种系统异构性的影响会被进一步放大. 此外, 同时聚合如此多客户端上传的模型, 计算与通信开销也会非常大.

实际上, 目前已有研究者通过选择一部分客户端参与每轮训练或测试来处理前面提到的问题. 文献 [2] 中设置了一个协调器随机选择客户端作为参与方集合执行联邦学习训练, 但这种方法并没有充分考虑客户端之间数据异构性与系统异构性[3]对联邦学习模型性能和训练效率的影响. 针对数据异构性问题, 文献 [4−6] 使用客户端数据的统计信息来映射数据质量, 并选择数据质量较高的客户端作为参与方; 还有一些工作[7,8]则从客户端所拥有的系统资源方面进行考虑, 比如网络带宽, 处理器资源等, 会优先选择训练轮次物理时耗较少的客户端参与训练. 也有一些工作[9−11]会综合以上两方面因素进行考虑来选择每一轮的 FL 参与方.

然而, 现有的异构联邦学习参与方选择方案并没有考虑对参与方模型训练的中间结果进行隐私保护, 比如梯度、模型参数等仍是以明文形式上传至服务器. 已经有研究发现, 攻击者可以从客户端的本地训练损失或梯度中恢复出客户端的部分隐私训练数据[12−14]. 而随着联邦学习攻击方法研究的不断深入, 参与方选择过程中涉及的中间结果都可能会暴露客户端隐私数据的相关信息. 这种隐私泄露可能会对客户端造成严重损失, 并打击各参与方执行联邦学习的积极性. 因此, 通过设计一种隐私保护联邦学习参与方选择方案, 保证在训练过程中仅公开每轮参与聚合的客户端集合, 不会公开任何其他信息, 就可以有效防止攻击者对客户端隐私数据进行推理攻击.

设计满足上述要求的隐私保护异构联邦学习参与方选择方案并不是一件简单的事情. 尽管已经有相关工作使用密码原语解决 FL 中隐私问题的方法, 例如同态加密[15]、混淆电路[16]和差分隐私[17], 但这些方法都停留在独立同分布联邦学习的研究中. 同态加密和混淆电路很难应用于更加复杂的异构联邦学习环境中的计算任务, 因为它们需要大量计算和存储资源, 而异构环境下客户端的计算资源存在差异, 若所有客户端都必须执行复杂的密态计算, 设备较差的客户端难以有效参与联邦聚合过程, 最终会成为整个联合训练中的瓶颈, 导致隐私保护 FL 效率低下. 使用差分隐私可以提高模型训练效率, 因为只需要少量的计算就可以生成随机扰动, 但会对模型准确率产生影响. 尽管 FL 的应用十分广泛, 但面对数据异构性与系统异构性的威胁时, 仍缺乏能够同时解决准确率、效率和数据隐私问题的综合性方案, 需要进一步考虑如何针对性地将隐私保护技术融入异构联邦学习参与方选择方案中.

安全多方计算[18,19]技术不需要高昂的加密解密成本, 就可以使参与方协作执行复杂计算, 而不泄露各方隐私数据, 在保证高通信吞吐量的同时不会影响计算精度, 因而被广泛应用于独立同分布下的隐私保护联邦学习方案中. 然而, 在处理联邦学习数据异构性时, 引入的参与方选择策略非常依赖客户端本地数据分布先验特征, 即异构性优化过程的中间结果会隐含参与方的隐私数据特征. 目前异构环境下的联邦学习方案并没有考虑对这些中间结果进行隐私保护.

针对上述这些问题, 本文设计了一种基于安全多方计算的隐私保护异构联邦学习参与方选择方案, 该方案允许聚合服务器每一轮选择最优的客户端子集作为联邦学习参与方, 且可以有效防范恶意敌手根据训练过程中的公

开信息推理出客户端的隐私数据. 本文贡献总结如下.

● 提出一种基于 3PC 秘密共享的隐私保护异构 FL 参与者选择方案. 在不损害隐私的情况下, 该协议安全地实现了高效的参与者选择策略, 并且保证了 FL 模型性能和训练效率.

● 提出一种基于 3PC 秘密共享的安全 top-k 搜索协议, 通过该协议可以使得各方只能获得选择的参与方集合, 而不会暴露其他任何隐私信息. 这是目前其他隐私保护方案还未实现的安全性要求.

● 基于 UC 框架证明协议的安全性, 表明该协议在半诚实威胁模型下不会向服务器泄露任何隐私信息. 此外还对方案进行实验评估, 结果表明, 所提协议相比于未使用隐私保护技术的异构联邦学习协议, 计算和通信的平均时间开销仅增加了 2.09%, 并且不会影响模型训练的准确度.

1 相关工作

本节概述了联邦学习领域关于参与方选择和隐私保护技术相关工作.

参与方选择对于整个联邦学习模型的性能和效率提升有着十分重要的影响. 目前已有多个联邦学习参与方选择方案被提出. 最早由 Brendan 等人[2]提出了一种随机抽样客户端参与训练的联邦学习参与方选择方案, 但后续的实验证明这种方法会使模型准确率下降. 随着研究不断深入, 参与方选择方案逐渐侧重于优化模型性能和优化训练效率这两个方面.

优化模型性能主要通过利用本地数据的统计特征信息来计算客户端的统计效用, 统计效用一定程度上可以反映客户端的数据质量, 数据质量高的客户端可以提高模型的准确率. Huang 等人[20]设计了一种基于训练损失的联邦学习指标评估方法以选择合适的客户端. Cho 等人[21]用理论证明了有偏选择损失函数值大的客户端参与训练能够提升联邦模型的收敛速度, 并基于此提出了一种基于损失函数感知的客户端选择算法. 杜甜等人[22]提出了一种基于云边模型解耦的联邦学习框架, 通过为客户端动态选择最优个性化层来应对数据异构性.

优化训练效率方面则主要考虑客户端为联邦学习任务提供的除本地数据之外的系统资源, 如处理器性能、网络带宽等. 优先考虑系统资源较好的客户端, 可以有效防止因聚合服务器长时间同步阻塞导致的训练轮次和物理时耗增加. FedMCCS[7]通过客户端的设备资源判断客户端是否能够执行 FL 任务, 并选择合适的客户端作为 FL 参与方, 以达到最佳的训练效率. FedCS[8]将基站作为服务器, 探讨了单个蜂窝网络中的客户端选择问题, 基站通过广播收集所有客户端的系统资源特征并基于此估算客户端的模型上传下载时间和训练时间, 并以最大期望训练时间作为阈值, 在轮次总时间不超过阈值的情况下尽可能多地选择客户端.

还有一些参与方选择方案会综合考虑前面提到的两个因素, 以此来平衡训练效率和模型精度的提升. Oort[9]允许中心服务器根据客户端的本地训练损失和训练时间来评估客户端的效用值, 然后根据效用值进行选择. FedBalancer[23]在利用本地训练的损失来评估客户端的统计效用同时, 自适应地设置截止期限, 并阻止训练速度较慢的客户端加入联合训练. PyramidFL[24]则改进了 Oort 的客户端效用评估方法, 对客户端训练数据规模和迭代轮次进行细粒度优化, 取得了显著的提升.

但上述参与方选择方案对于客户端数据隐私保护方面的考虑仍有欠缺, 近年来出现的一些针对客户端的隐私攻击[12–14,25,26]已经可以通过分析客户端本地训练损失或梯度来推测出客户端的隐私数据相关信息. 目前已有一些隐私保护技术可以对聚合过程进行隐私屏蔽, 例如文献 [27–29] 采用同态加密技术对模型参数或梯度进行加密, 并支持密文下聚合梯度; Wei 等人[30]利用差分隐私对梯度进行屏蔽, 同时保证屏蔽后的梯度在聚合后能将噪声降低到可以接受的范围内; Zeng 等人[31]提出了一个基于承诺与零知识证明的隐私保护可审核联邦学习框架, 在该框架下也可以执行联邦学习参与方选择; 穆旭彤等人[32]基于轻量级安全两方计算技术, 设计了安全 QR 分解协议与安全的矩阵分解协议, 在此基础上实现了兼顾数据隐私性与计算高效性的鲁棒联邦学习, 但该协议在针对异构环境下的联邦学习性能较差.

总的来说, 目前已有的隐私保护框架并不能很好地适配异构环境下的参与方选择方案. 一方面, 许多参与方选择方案使用的评估函数复杂且非线性, 包含大量的除法、排序、最大最小值等操作, 这些中间计算的结果在密码

学层面都可能被敌手利用进行推理攻击 (例如两方的评估结果进行比较后公开还是不公开, 公开的话这个比较结果泄露会不会带来其他隐私信息泄露的可能; 如果不公开, 后续的计算例如排序该如何继续进行), 这些隐私保护框架在应用到异构环境下的联邦学习时并没有考虑这些情况. 另一方面, 前面提到异构环境下客户端的计算资源存在差异, 若使用的隐私保护技术要求客户端执行复杂的密态计算, 设备性能较差的客户端就难以有效参与联邦聚合, 并且会成为整个联合训练中的瓶颈.

2 基础知识

本节主要介绍所提出的隐私保护联邦学习参与方选择方案涉及的联邦学习的基本概念和密码原语. 并在表 1 中定义相关符号及其含义.

表 1 本文符号说明

符号	含义
$\mathbb{C}=\{C_1,C_2,\ldots,C_N\}$	客户端集合, 包含 N 个客户端
$\mathbb{P}$	当前轮被选中的联邦学习参与方集合
K	每一轮被选中的联邦学习参与方数量
ϵ	探索因子, 每一轮根据效用值排序选择的参与方的比例
T	每一轮预设置的阈值时间
Δ	阈值时间的调整步长
α	惩罚因子
B_i	客户端 C_i 的训练样本集
I_i	客户端 C_i 的本地迭代轮次
$[\cdot]$	3PC秘密共享份额
$\langle\cdot\rangle$	2PC加法秘密共享份额
I_{base}	每轮预设置的客户端本地迭代轮次

2.1 联邦学习

联邦学习[33]是一种新兴的分布式学习, 旨在提高协作训练的效率、隐私和可扩展性. 在联邦学习中, 训练过程发生在分布式客户端而不是中央服务器上. 客户端能够联合训练全局模型, 而无需直接与聚合服务器共享其本地数据. 在每次全局迭代中, 每个参与的客户端都会进行本地训练并将训练的本地模型上传到服务器, 服务器以同步方式汇总所有接收的模型来更新全局模型, 并将其广播给所有客户端以供下一轮的协作训练, 直到模型收敛或达到预先设置的训练轮次. 具体来说, 假设联邦学习中将有 N 个客户端参与, 每个客户端 i 将使用本地数据集 D_i 训练模型 W_i, 并计算:

$$\Delta W_i^t \leftarrow f(W^{t-1},D_i)-W^{t-1}.$$

然后将 ΔW_i^t 上传到聚合服务器上. 其中 $f(\cdot,\cdot)$ 返回的是客户端使用本地数据集 D_i 在第 t 轮训练的模型, W^{t-1} 是联邦轮次为 $t-1$ 时的全局模型. 而聚合服务器对上传的模型进行求和平均, 可以表示为:

$$W^t \leftarrow W^{t-1}+\frac{1}{N}\sum_{i=1}^{N}\Delta W_i^t.$$

2.2 安全多方计算

安全多方计算允许多方在不泄露私有数据的情况下协作执行运算. 其中每一方都持有私有数据的一部分, 称为秘密份额, 各方在秘密份额的基础上独立或交互式地执行一系列计算, 最终共同得到一个计算结果. 本文选择在环 $\mathbb{Z}_{2^l}$ 上执行秘密份额之间的计算, 使用三方的秘密共享方案 (后面统称 3PC 秘密共享)[34], 我们用 $[\cdot]$ 来表示在 $\mathbb{Z}_{2^l}$ 上的算术秘密共享份额, 用 $[\cdot]^{\mathrm{B}}$ 来表示在 $\mathbb{Z}_2$ 上的布尔秘密共享份额. 令服务器 S_1、S_2 和 S_3 作为 3PC 的 3 个计算方. 对于秘密值 $x\in\mathbb{Z}_{2^l}$, 持有方已经生成了 x 的 3 个加法秘密份额 (x_1,x_2,x_3), 且满足 $x=(x_1+x_2+x_3) \bmod 2^l$. 秘密值持有方将份额发送给服务器, 每个服务器获得 3 个份额中的 2 个, 即 S_i 得到 3PC 秘密份额 $[x]_i=(x_i,x_{(i+1) \bmod 3})$, 其

中 $i \in \{1,2,3\}$; 算术秘密共享份额与布尔秘密共享份额之间的相互转换可参考文献 [35], 后面只以算术秘密共享进行举例讲解. 下面我们介绍 3PC 秘密共享中的一些操作.

● 重构秘密值 $\prod_{\text{Rec}}$: 当向服务器 S_i 重构秘密值 x 时, 其他两方 $S_{(i+1) \bmod 3}$ 和 $S_{(i+2) \bmod 3}$ 将 $x_{(i+2) \bmod 3}$ 发送给 S_i. S_i 首先检查另外两方发送的份额是否一致, 如果一致, 就重构出秘密值 $x = (x_i + x_{(i+1) \bmod 3} + x_{(i+2) \bmod 3}) \bmod 2^l$; 如果不一致, 就中止协议执行.

● 3PC 秘密份额加法 $[x]+[y]$: 每个服务器 S_i 持有秘密值 x 和 y 的秘密份额 $[x]$ 和 $[y]$, 想要计算 $x+y$ 的 3PC 秘密份额, 可以直接在本地计算 $[x+y]_i = (x_i + y_i, x_{i+1} + y_{i+1})$.

● 常数加法: 三服务器持有 x 的秘密份额 $[x]$ 和一个常数 c, 可以直接在本地计算 $[x+c]$, 过程如下: S_1 计算 $[x+c]_1 = (x_1 + c, x_2)$, S_2 计算 $[x+c]_2 = (x_2, x_3)$, S_3 计算 $[x+c]_3 = (x_3, x_1 + c)$.

● 常数乘法: 三服务器持有 x 的秘密份额 $[x]$ 和一个常数 c, 想要计算 $[c \cdot x]$, 每个服务器 S_i 在本地计算: $[c \cdot x] = (c \cdot x_i, c \cdot x_{i+1})$.

● 3PC 秘密份额乘法: 每个服务器持有秘密值 x 和 y 的秘密份额 $[x]$ 和 $[y]$, 想要计算 $x \cdot y$ 的秘密份额 $[x \cdot y]$, 过程如下: 首先服务器 S_i 获取关于 0 的一个加法 S_i 秘密份额 α_i, 满足 $\alpha_1 + \alpha_2 + \alpha_3 = 0$. 接着 S_1 计算 $z_1 = x_1 y_1 + x_1 y_2 + x_2 y_1 + \alpha_1$, 并发送给 S_3; S_2 计算 $z_2 = x_2 y_2 + x_2 y_3 + x_3 y_2 + \alpha_2$, 并发送给 S_1; S_3 计算 $z_3 = x_3 y_3 + x_3 y_1 + x_1 y_3 + \alpha_3$, 并发送给 S_2. 最后每个服务器获得 $[x \cdot y]_i = (z_i, z_{i+1})$.

关于零加法秘密份额 $(\alpha_1, \alpha_2, \alpha_3)$ 的生成过程如下: 首先每个服务器 S_i 选择一个随机密钥 $K_i \in \{0,1\}^\kappa$, 然后将其发送给 S_{i-1}, 其中 S_0 即为 S_3. 然后每一方计算 $\rho_i = F_{K_i}(cnt)$, $\rho_{(i+1) \bmod 3} = F_{K_{(i+1) \bmod 3}}(cnt)$, 其中公开的伪随机函数 $F_K(\cdot)$: $\{0,1\}^\kappa \to \mathbb{Z}_{2^l}$, cnt 是一个公共计数器, 每次使用之后会递增. 最后每一方计算 $\alpha_i = \rho_i - \rho_{(i+1) \bmod 3}$.

我们的协议还需要一些可信函数 $\mathcal{F}_{\text{rand}}$、$\mathcal{F}_{\text{coin}}$、$\mathcal{F}_{\text{zero}}$ 和 $\mathcal{F}_{\text{comparison}}$, 下面我们将介绍这些函数以及实现这些函数的安全协议.

● $\mathcal{F}_{\text{rand}}$: 生成关于随机值 $r \in \mathbb{Z}_{2^l}$ 的 3PC 秘密共享份额 $[r]$. 首先每个服务器 S_i 选择一个随机密钥 $K_i \in \{0,1\}^\kappa$, 然后将其发送给 S_{i-1}, 其中 S_0 即为 S_3. 然后每一方计算 $r_i = F_{K_i}(cnt)$, $r_{(i+1) \bmod 3} = F_{K_{(i+1) \bmod 3}}(cnt)$, 其中公开的伪随机函数 $F_K(\cdot)$: $\{0,1\}^\kappa \to \mathbb{Z}_{2^l}$, cnt 是一个公共计数器, 每次使用之后会递增. 最后 S_i 设置 $[r]_i = (r_i, r_{i+1})$.

● $\mathcal{F}_{\text{coin}}$: 使各方得到一个相同的随机值 $r \in \mathbb{Z}_{2^l}$. 可以让各服务器先调用 $\mathcal{F}_{\text{rand}}$ 得到一个随机值 r 的 3PC 秘密共享份额, 然后再让所有服务器调用 $\prod_{\text{Rec}}$ 重构出 r 即可.

● $\mathcal{F}_{\text{zero}}$: 我们遵照文献 [36] 来实例化零秘密共享份额的生成. 各服务器得到关于 0 的一个 3PC 秘密共享份额 $[0]$. 与前面 3PC 秘密共享乘法中生成 0 的加法秘密份额 $(\alpha_1, \alpha_2, \alpha_3)$ 的方法一致. 在得到 $(\alpha_1, \alpha_2, \alpha_3)$ 之后, 每个服务器 S_i 将 α_i 发送给 S_{i-1}, 其中, S_0 即为 S_3. 每个服务器 S_i 设置 $[0]_i = (\alpha_i, \alpha_{i+1})$.

● $\mathcal{F}_{\text{comparison}}$: 各服务器持有秘密值对 (x,y) 的 3PC 份额 $([x],[y])$, 如果 $x < y$, 则各服务器获得 $b = 1$ 的共享份额 $[b]^B$, 否则得到 $b = 0$ 的共享份额. 实现该函数的安全协议参考文献 [35], 这里不再提供实例化.

3 问题描述

3.1 系统模型

假设由开发人员确定 FL 任务和参与方选择标准. FL 训练部分主要包含聚合服务器 (S_1, S_2, S_3) 和多客户端 $(\mathbb{C})$ 两大实体.

● 聚合服务器 (S_1, S_2, S_3): 用于在每一轮联邦学习训练开始之前选择参与方, 以及聚合分布式客户端上传的本地模型. 为了简化 FL 场景, S_1、S_2 和 S_3 均作为分布式聚合服务器, 每个服务器仅获得来自客户端上传的敏感数据的秘密份额, 而无法知晓敏感数据的完整信息. 多服务器架构在安全多方计算中很常用, 这种设置能够安全地存储加密数据, 并与隐私保护子协议进行交互.

● 多客户端 $(\mathbb{C})$: 每个客户端都持有私有数据, 在完成本地训练后将敏感数据转换为 3 个秘密份额, 并发送给对应的服务器; 每个服务器在各自秘密份额的基础上进行聚合, 得到新的全局模型的秘密份额, 再发送给客户端;

客户端从服务器下载得到全部的 3 个全局模型秘密份额, 并在本地重构得到完整的全局模型, 在此模型上开始下一轮的训练.

● 可信的第三方协调器 ($\mathcal{T}$): 用于协助聚合服务器完成参与方选择, 输出每轮选择的参与方集合. 我们假设每个服务器 S_i 与协调器 $\mathcal{T}$ 之间都持有会话密钥 t_i 用于加密通信.

接下来将介绍我们整个系统的工作原理, 如图 1 所示.

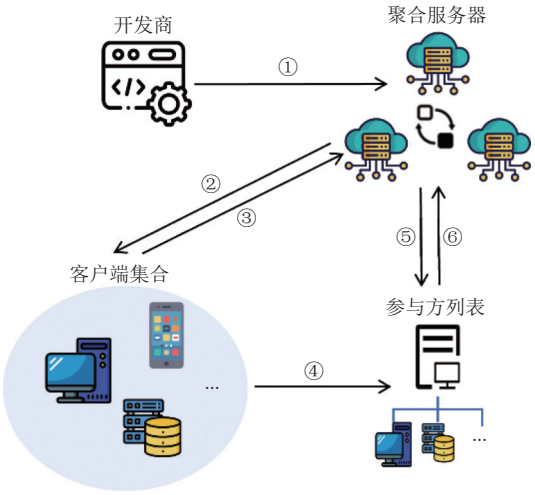

图 1 系统模型示意图

① 开发人员向各服务器提交 FL 任务以及参与方选择标准.

② 聚合服务器向各客户端发送新一轮参与方选择信号, 并根据参与方选择标准开始搜集所需的客户端信息.

③ 客户端在接收到信号后, 将服务器所需的所有信息以 3PC 秘密共享份额的形式上传到对应的服务器.

④ 各服务器在接收到秘密份额后参照选择标准, 并调用安全协议获取综合效用排名靠前的客户端集合, 将集合中的客户端作为这一轮参与方.

⑤ 服务器将最新的全局模型份额发送给各客户端.

⑥ 参与方在本地恢复出全局模型, 并开始新一轮的训练, 将训练后的模型生成秘密份额发送给服务器. 服务器在接收到本地模型份额之后完成这一轮的聚合.

3.2 敌手模型

由于服务器持有客户端敏感数据的份额, 一旦其中任意两方进行串通, 就可以直接恢复出客户端上传的完整数据, 因此我们的安全模型假设服务器为诚实但好奇的, 即服务器 S_1、S_2 和 S_3 将完全遵照安全协议执行, 但会从公开的以及持有的信息中尝试获取客户端的隐私数据. 这种设置在实际应用中也是可以实现的, 例如设置 3 个服务器分别来自不同的服务提供商, 这样两两之间进行串通的概率就会非常低. 此外, 我们假设协调器是一个可信第三方, 不会与任何一个服务器进行串通.

4 隐私保护参与方选择方案设计

本节将介绍所提方案的各个子模块, 首先介绍联邦学习参与方选择策略, 然后从聚合服务器和分布式客户端两个方面来阐述如何动态调整选择策略, 最后介绍如何将安全多方计算技术融入整个联邦学习参与方选择过程中.

4.1 基于效用函数的联邦学习参与方选择

本文基于先进的参与者选择框架 PyramidFL[24]设计了一个 MPC 友好的参与者选择方案. 该方案将客户端之间的数据异构性和系统异构性综合考虑, 并且用统计效用函数和系统效用函数分别量化这两个因素对最终模型的影响. 使用这两个效用指标可以评估客户端对最终模型的贡献大小, 进而可以依此对客户端进行排名, 排名靠前的

客户端会优先考虑选为本轮联合训练的参与方. 由于在训练前期阶段并不知道所有客户端的效用, 以及客户端的效用还会随模型训练的进行而变化. 因此还需要选取以前未参与或较少参与训练的客户端来参与聚合, 选择的比例用探索因子 ϵ 来控制. 设定初始探索因子 $\epsilon=0.9$, 每轮迭代将 ϵ 乘以 0.98, 最终逐渐降低到 0.2. 所设计方案还允许客户端自适应地调整本地训练迭代轮次来提高己方的综合效用, 以提升下一轮被选中为参与方的概率并促进全局模型的收敛速度. 下面将详细介绍我们的联邦学习参与方选择策略.

假设客户端集合为 $\mathbb{C}$. 为了选择客户参加第 r 轮 (其中 $r>1$) 训练, 每个客户端 i 在本地计算统计效用:

$$U_i^r=|B_i|\sqrt{\frac{1}{|B_i|}\sum_{b\in B_i}Loss(b)^2},$$

并将其上传到聚合服务器, 其中 $Loss(b)$ 为样本 b 的训练损失, U_i^r 为客户端的统计效用. 此外, 服务器还应记录这轮系统信息, 包括用户 i 这一轮训练所需时间 t_i (包括本地模型训练时间和数据上传时间). 同时服务器还可以根据过去一段轮次的累计统计效用是否下降来动态调整预设置的每轮持续时间 T: 假设以每 W 轮为一个时间窗口, 当 $r>2W$ 时, 如果不等式 $\sum_i U_i(r-2W:r-W)>\sum_i U_i(r-W:r)$ 成立, 说明过去几轮的累计统计效用在减少, 服务器应适当增加下一轮阈值时间 T (即 $T=T+\Delta$) 以提高统计效用. 其中, $\sum_i U_i(r-W:r)$ 表示从第 $r-W$ 轮到第 r 轮所用参与方的累计统计效用.

服务器根据客户端上传的统计效用以及该客户端的这一轮总耗时 t_i 来计算综合效用 $Util_i=U_i^r\times\left(\frac{T}{t_i}\right)^{1(T<t_i)\times\alpha}$, 其中 $\left(\frac{T}{t_i}\right)^{1(T<t_i)\times\alpha}$ 代表该客户端的系统效用; $1(T<t_i)$ 只输出判断结果 0 和 1, 如果括号内条件满足, 则输出 1, 否则输出 0; α 为惩罚因子, 其取值范围为 $[0,+\infty)$, 用来惩罚每轮参与者选择过程中严重拖慢整个训练过程的客户端, α 越大, 训练较快的客户端被选中的概率就越大, 以此来提升每轮的训练效率. 本方案参考 PyramidFL[24]选择 $\alpha=2$. 客户端总耗时越长, 系统效用就会越小, 被选中的概率就会降低, 该指标可以反映客户端系统性能对整个模型的贡献.

最后服务器按照综合效用值对 $\mathbb{C}$ 中客户端进行排序, 并依照排序选择排名最靠前的 $\epsilon\cdot K$ 个客户端, 即为集合 $\mathbb{C}^*$, 并从剩余客户端集合中随机抽取 $(1-\epsilon)\cdot K$ 个客户端得到这一轮的联邦学习参与方集合 $\mathbb{P}$.

然而, 由于每个客户端训练设备的差异, 每一轮所消耗时间 t_i 的差别会非常大, 如果将客户端按照其消耗时间 t_i 进行排序, 可以发现较快的客户端在完成训练并上传至服务器后, 必须等待较慢的客户端也完成同样的任务才能收到聚合后的全局模型以开启下一轮训练, 对于较快的客户端来说存在较长的空窗期被浪费. 方案将客户端一轮所消耗的时间细分为本地模型训练时间 t_i^{comp} 和模型参数上传时间 t_i^{comm}, 为了减少被浪费的时间成本, 可以让客户端自适应调整这两部分所需时间. 本文参考文献 [24] 中提出的联邦学习参与方选择方案, 从两个方面来对每轮训练过程进行动态调整: 一方面聚合服务器通过各个客户端的训练时间 t_i, 以及该轮所有客户端的效用值总和与上一轮效用值总和的对比来调整阈值时间 T; 另一方面每个客户端根据阈值时间 T 与置信度因子 β_i 来调整本地训练轮次 I_i. 具体来说, 每个参与者可以进一步调整其本地训练迭代轮次 I_i:

$$I_i=\left(\beta_i\times\frac{\min(T-t_i,0)}{t_i^{\text{comp}}}+1\right)\times I_{\text{base}},$$

其中, T 和 I_{base} 表示当前轮次预设置的阈值时间与训练轮次; β_i 是根据上一轮 t_i 设置的置信度因子, 其计算公式为: $\beta_i=\mathrm{e}^{-\gamma\cdot\theta_i}$, 其中 γ 是预先设置好的衰减系数, θ_i 为客户端 i 上次参与聚合的轮次到当前轮所经历的迭代轮数. 根据这两个值可以调整本轮的训练数据集大小:

$$|B_i|=Batch\_size\times I_i.$$

经过调整后, 客户端 i 本轮所消耗的时间变为 $t_i=t_i^{\text{comp}}+t_i^{\text{comm}}$ 会进一步减少, 在下一轮的综合效用值会变大, 被选中为参与方的概率也会提高. 同时, 当 $t_i\leqslant slant\ T$ 时, 由于客户端耗时满足阈值要求, 此时也可以选择保持基础迭代轮次 I_{base} 不变, 避免因减少迭代轮次导致的本地模型精度损失.

4.2 基于安全多方计算的隐私保护异构联邦学习参与方选择协议

尽管 PyramidFL[24]对联邦学习参与者选择的效用评估方法进行了优化改进, 能够更好地满足异构联邦学习环

境的要求, 但整个方案仍没有考虑对参与方模型训练的中间结果进行隐私保护, 比如上传至服务器的梯度、模型参数等, 仍是以明文形式上传至服务器. 而现有研究[12−14]已经证明了攻击者可以从客户端的本地训练损失或梯度中恢复出客户端的部分隐私训练数据. 此外, PyramidFL 方案中客户端的统计效用值可以反映客户端的隐私信息, 例如其数据分布情况, 这些数据同样通过明文形式上传至服务器. 随着联邦学习攻击方法研究的不断深入, 这些参与方选择过程中涉及的中间结果都可能会暴露客户端隐私数据的相关信息, 最终造成客户端的严重损失. 因此, 本文利用密码原语设计了一个隐私保护参与者选择协议, 该协议在保护客户端隐私数据与模型梯度的同时, 允许聚合服务器选择合适的 FL 参与者集合 $\mathbb{P}$, 可以有效防止攻击者利用公开的中间信息对客户端隐私数据进行推理攻击. 首先在本节中介绍 3 个安全组件, 它们是本协议实现隐私保护的重要组成部分.

● 2PC 秘密共享[37]: 即两服务器之间的加法秘密共享, 将 x 的 2PC 秘密份额表示为 $\langle x\rangle$, 假设 S_i 和 S_j 持有关于 x 的 2PC 份额 $\langle x\rangle_i$ 和 $\langle x\rangle_j$, 则满足 $\langle x\rangle_i+\langle x\rangle_j=x$. 本方案需要服务器之间进行 3PC 秘密共享份额与 2PC 秘密共享份额之间的相互转换.

● 3PC 秘密共享份额转换为 2PC 秘密共享份额 $[x]\xrightarrow{i,j}\langle x\rangle_{i,j}$ [38]: 假设 3 个服务器持有客户端上传的数据 x 的 3PC 秘密份额 $[x]$. 可以将 $[x]$ 转换为任意两个服务器 (假设为 (S_i,S_j)) 之间的 2PC 秘密份额. 下面分情况进行讨论.

1) 如果 $j=i+1$, $i\in\{1,2\}$ 时, S_i 计算 $\langle x\rangle_i=x_i+x_{i+1}$, S_j 计算 $\langle x\rangle_j=x_{(j+1)\bmod 3}$.

2) 当 S_1 和 S_3 之间进行转换时, S_1 计算 $\langle x\rangle_1=x_1+x_2$, S_3 计算 $\langle x\rangle_3=x_3$.

● 2PC 秘密共享份额转换为 3PC 秘密共享份额 $\langle x\rangle_{i,j}\xrightarrow{i,j}[x]$ [38]: 假设除了服务器 S_k 的另外两个服务器希望将它们之间的 2PC 秘密共享份额转换为 3PC 秘密共享份额 $[x]$, 以 $S_k=S_1$ 为例, 即将 S_2 和 S_3 之间的 2PC 份额进行转换, S_2 持有份额 x_2, S_3 持有份额 x_3, 满足 $x_2+x_3=x$, 整个转换过程如下.

1) 首先 S_1 和 S_2 之间协商得到随机值 $r_{1,2}$, S_1 和 S_3 之间协商得到随机值 $r_{1,3}$, S_1 设置 $[x]_1=(r_{1,2},r_{1,3})$.

2) S_2 计算 $x_2-r_{1,2}$, 并将其发送给 S_3, S_3 计算 $x_3-r_{1,3}$ 并将其发送给 S_2.

3) S_2 设置 $[x]_2=(r_{1,2},x_2-r_{1,2}+x_3-r_{1,3})$, S_3 设置 $[x]_3=(x_2-r_{1,2}+x_3-r_{1,3},r_{1,3})$.

根据第 4.1 节可知本文的参与方选择方案需要挑选综合效用值排名靠前的客户端作为下一轮的联邦学习参与方, 因此设计了一个隐私保护 top-k 搜索协议, 该协议将一个 3PC 秘密共享数组作为输入, 输出一个包含数组中前 k 个最大元素的秘密共享份额子集, 且不会暴露集合内秘密份额之间的大小关系, 整个协议如协议 1 所示.

协议 1. 隐私保护 top-k 搜索协议 $\prod_{\text{top-}k}([V],k)$.

输入: 服务器持有一组元素的份额 $[V]=\{[v_1],[v_2],\ldots,[v_n]\}$, 需要搜索的最大元素的数量 k;
输出: 服务器输出包含最大的前 k 个最大元素的秘密份额的集合 $\Phi_{\text{top-}k}$, 以及其对应的客户端集合 $\mathbb{C}_E$.

1. S_1、S_2 和 S_3 调用 $\prod_{\text{shuffle}}([V])$ 将输入的秘密份额数组 $[V]$ 进行混洗, 得到新的份额数组 $[V']=[v'_1,v'_2,\ldots,v'_n]$.
2. 初始时各服务器将 $[V']$ 中的所有元素按顺序放入集合 Φ 中.
3. 重复下列过程直到满足 $\left|\Phi_{\text{top-}k}\right|=k$:
   a) 各服务器协商出一个基准份额的序号 $r\in|\Phi|$.
   // 集合分割
   b) 服务器调用安全比较协议 $\prod_{\text{SCmp}}$ 将 Φ 中元素的份额与 $[v'_r]$ 进行比较, 并输出比较结果 $\left[\vec{b}\right]\in\{0,1\}^{|\Phi|}$.
   c) 各方对比较结果份额 $\left[\vec{b}\right]$ 执行秘密值重构协议 $\prod_{\text{Rec}}$, 得到 $\vec{b}$.
   d) For $j\in|\Phi|$:
      i. 如果 $b_j=1$, 即 $v'_j\geqslant v'_r$, 则将 $\left[v'_j\right]$ 放入集合 Φ_1 中.
      ii. 如果 $b_j=0$, 即 $v'_j\leqslant v'_r$, 则将 $\left[v'_j\right]$ 放入集合 Φ_0 中.
   e) 如果 $|\Phi_1|\leqslant k-\left|\Phi_{\text{top-}k}\right|$, 则将 Φ_1 中的份额合并入 $\Phi_{\text{top-}k}$ 中, 并设置 $\Phi=\Phi_0$, 否则设置 $\Phi=\Phi_1$.
4. S_1 向 $\mathcal{T}$ 加密传输己方的 $[V]$ 和 $[V']$, $\Phi_{\text{top-}k}$ 和随机排列 π_1、π_3, S_2 向 $\mathcal{T}$ 加密传输 π_2. 最终 $\mathcal{T}$ 根据 $\pi_1\circ\pi_2\circ\pi_3$ 排列恢复出集合 $\Phi_{\text{top-}k}$ 对应的客户端集合 $\mathbb{C}_E$, 并广播集合 $\mathbb{C}_E$.

首先, 调用安全混洗协议 $\prod_{\text{shuffle}}$ 对输入数组进行随机排列, 得到混洗后的新数组. 每一对服务器 (S_i, S_{i+1}) $i \in \{1,2,3\}$, 其中 S_4 (即为 S_1) 之间协商出一个随机排序序列 π_i 用于后续重新排列各自的秘密份额. 然后每个服务器将新数组中的所有份额按顺序放入集合 Φ 中, 并设置一个空集合 $\Phi_{\text{top-}k}$ 来放置前 k 个最大元素的份额. 接下来基于快速排序的思想递归寻找前 k 个最大元素: 每一轮迭代都随机选择混洗数组中的一个元素作为基项, 然后通过执行文献 [35] 中的安全比较协议 $\prod_{\text{SCmp}}$ 将基项元素与集合中的其他元素进行比较, 并重构出比较结果. 虽然服务器知道了混洗数组中的比较结果, 但因为混洗数组中的元素是经过随机顺序打乱的, 因此服务器获得的只是混洗后数组之间的大小关系, 依然无法推断原始数组中各元素之间的大小关系.

然后, 依据比较结果将集合 Φ 划分为两个集合 Φ_0 和 Φ_1, 前者包含小于基项的元素, 后者包含大于基项的元素. 如果 $|\Phi_1| > k - \left|\Phi_{\text{top-}k}\right|$, 则意味着需要的 top-$k$ 个元素的集合是 Φ_1 的子集, 这时就用 Φ_1 中的元素来替代 Φ 中的元素, 然后递归执行前面的划分过程; 如果 $|\Phi_1| < k - \left|\Phi_{\text{top-}k}\right|$, 则说明 Φ_1 中的元素都在前 k 个最大元素中, 可以将 Φ_1 中的元素都添加到 $\Phi_{\text{top-}k}$ 中, 并用 Φ_0 中的元素来替代 Φ 中的元素以开始下一轮的递归划分. 整个递归过程直到满足 $|\Phi_1| = k - \left|\Phi_{\text{top-}k}\right|$ 为止.

接下来将介绍如何实现对秘密共享数组的安全混洗 $\prod_{\text{shuffle}}([V])$. 如协议 2 所示, 整个协议旨在通过随机排列秘密地打乱一个数组 $[V]$, 并且新数组中的秘密份额与原始数组中的秘密份额也会有所不同, 这样每个服务器就无法知晓具体的随机排列. 首先在步骤 1 中, 三服务器的每一对之间都协商出一个随机排列, 然后服务器 S_1 和 S_2 首先将输入的 3PC 秘密共享数组 $[V]$ 转换为 2PC 秘密共享数组, 然后双方使用协商好的随机排列 π_1 打乱原始 2PC 数组, 并加入零秘密共享 $\langle\vec{0}\rangle$ 将秘密份额重新随机化得到 $\langle V_1\rangle$, 然后发送给服务器 S_3. 随后 S_2 和 S_3 在混洗 2PC 秘密共享数组上按照协商好的随机排列 π_2 打乱顺序, 并重新随机化得到 $\langle V_2\rangle$, 然后发送给 S_1. 最后 S_1 和 S_3 再按照随机排列 π_3 进行顺序打乱并重新随机化得到 $\langle V_3\rangle$, 并将 $\langle V_3\rangle$ 转换为 3PC 秘密共享数组 $[V']$. $[V']$ 即 $[V]$ 经过排列顺序 $\pi_1 \circ \pi_2 \circ \pi_3$ 得到的新的 3PC 秘密共享数组, 而每个服务器只知晓 3 个随机排列中的 2 个排列, 因此在不与其他服务器协作的情况下无法推断出 $[V]$ 和 $[V']$ 的对应关系. 而协调器 $\mathcal{T}$ 虽然知道随机排列, 但由于未参与比较过程, 仅通过一方的秘密份额也无法推断出各方秘密值的大小关系.

协议 2. 隐私保护混洗协议 $\prod_{\text{shuffle}}([V])$.

输入: 各服务器持有的一组元素的份额 $[V] = \{[v_1], [v_2], \ldots, [v_n]\}$;
输出: 服务器输出以私有随机排列顺序对 $[V]$ 进行混洗的新份额 $[V']$.

1. 每一对服务器 (S_i, S_{i+1}) $(i \in \{1,2,3\}$, 其中 S_4 即为 $S_1)$ 之间协商出一个随机排序序列 π_i 和 3PC 的零秘密共享份额 $\left[\vec{0}\right]_i$.
2. 首先服务器 S_1 和 S_2 将 3PC 份额转换为 2PC 份额 $[V] \xrightarrow{1,2} \langle V_{1,2}\rangle$, $\left[\vec{0}\right] \xrightarrow{1,2} \left\langle\vec{0}\right\rangle_{1,2}$, 依据 π_1 各自在本地混洗和重随机化份额, 得到新份额 $\langle V_1\rangle_{1,2} = \langle\pi_1(V)\rangle + \left\langle\vec{0}\right\rangle_{1,2}$; 并且 S_1 将自己的份额 $\langle V_1\rangle_1$ 发送给 S_3.
3. 接着服务器 S_2 和 S_3 依据 π_2 各自在本地混洗和重随机化份额, 得到新份额 $\langle V_2\rangle_{2,3} = \langle\pi_2(V_1)\rangle + \left\langle\vec{0}\right\rangle_{2,3}$; 并且 S_2 将 $\langle V_2\rangle_2$ 发送给 S_1.
4. 服务器 S_3 和 S_1 依据 π_3 各自在本地混洗和重随机化份额, 得到新份额 $\langle V_3\rangle_{3,1} = \langle\pi_3(V_2)\rangle + \left\langle\vec{0}\right\rangle_{3,1}$.
5. S_1、S_2 和 S_3 将 2PC 份额转换为 3PC 份额 $\langle V_3\rangle_{3,1} \xrightarrow{3,1} [V']$, 最终输出完成混洗的份额 $[V']$.

下面将各安全组件与第 4.1 节的异构联邦学习参与方选择方案结合, 来介绍整个联邦学习过程. 在第 r 轮 $(r > 1)$ 训练开始之前, 服务器需要进行参与方选择: 客户端每一轮在训练过程中顺便收集训练损失信息 $Loss(b)$, 并根据第 4.1 节计算出统计效用值 U^{r-1}, 再为每个服务器生成对应的秘密份额 $[U^{r-1}]$; 服务器收集上一轮训练中各客户端的统计效用秘密份额 $[U^{r-1}]$, 并依据第 4.1 节中的计算方法在本地计算综合效用的秘密份额 $\left[Util^{r-1}\right]$. 各服务器调用协议 $\prod_{\text{top-}k}\left(\left[Util^{r-1}\right], \epsilon{\times}k\right)$ 获取排名靠前的 $\epsilon{\times}k$ 个客户端集合 $\mathbb{C}^*$, 再随机选取 $(1-\epsilon){\times}k$ 个客户端并入 $\mathbb{C}^*$, 得到第 r 轮的参与方集合 $\mathbb{P}$ 并公布给客户端. 且服务器根据相邻窗口期内的累计统计效用确定这一轮的阈值时间 T_r.

所有服务器将最新的全局模型份额分发给各客户端, 客户端在本地恢复出模型的秘密值. 接着被选择的参与

方根据阈值时间 T_r 确定本地训练迭代轮次 $I_i^r=\left(\beta\times\frac{\min(T_r-t_i^{r-1},0)}{t_i^{\text{comp}}}+1\right)\times I_{\text{base}}$, 并在此基础上训练模型和计算训练样本集大小 $\left|B_i^r\right|=Batchsize\times I_i^r$, 将更新后的参数生成相应的秘密份额, 上传至对应的服务器. 服务器接收到参与方的模型份额后, 各自在本地完成聚合得到全局模型的份额, 并开始准备下一轮训练. 隐私保护联邦学习参与方选择协议如协议 3 所示.

协议 3. 隐私保护异构联邦学习参与方选择协议.

输入: 客户端集合 $\mathbb{C}$, 联邦学习参与方集合大小 K, 探索因子 ϵ, 服务器阈值步长调整值 Δ, 惩罚因子 α, 置信因子 β, 本地训练基础迭代轮次 I_{base};

输出: 参与方集合 $\mathbb{P}$.

// 初始化被探索的客户端集合、效用值排序集合以及每轮耗时阈值

1. $\mathbb{C}_E\leftarrow\varnothing$; $\mathbb{R}\leftarrow\varnothing$; $T\leftarrow\Delta$;

// 服务器端执行客户端探索与选择

2. 当训练轮次 $r=1$ 时:

3. 首轮服务器根据各客户端的训练耗时收集排名前 K 位的客户端作为首轮的参与方集合 $\mathbb{P}$.

4. 当训练轮次 $r>1$ 时:

5. 客户端 $i\in\mathbb{C}$ 收集上一轮训练损失信息, 计算相应的统计效用 $U_i^r=\left|B_i^{r-1}\right|\sqrt{\frac{1}{\left|B_i^{r-1}\right|}fd\sum_{b\in B_i^{r-1}}Loss(b)^2}$, 并为三服务器生成对应的统计效用份额 $[U_i^r]$. 服务器 S_j 收集上一轮训练中统计效用的份额以及各客户端的耗时 $\mathbb{F}_j^{\text{stat}}=[U^r]_j$, $\mathbb{F}_j^{\text{sys}}=t^{r-1}$, $j\in\{1,2,3\}$.

// 服务器根据过去几轮累计统计效用是否下降来调整耗时阈值

6. 服务器执行协议 $\prod_{\text{SCmp}}\left(\sum_{r-W}^{r}[U^r],\sum_{r-2W}^{r-W}[U^r]\right)$, 并恢复出比较结果 c_r; 如果 $c_r=1$, 则得到: $T_r\leftarrow T_{r-1}+\Delta$.

// 计算这一轮的综合效用值

7. for 客户端 $i\in\mathbb{C}$:

8. $[Util_i^r]_j=\left([U_i^r]_j\times\left(\frac{T_r}{t_i^{r-1}}\right)^{1\left(T_{r-1}<t_i^{r-1}\right)\times\alpha}\times 2^d\right)\bmod 2^l$.

// 按照综合效用排名前 $\epsilon\times K$ 个客户端作为部分参与方

9. 服务器和协调器联合执行协议 $\prod_{\text{top-}k}([Util^r],\epsilon\times K)$ 得到 $\mathbb{C}_E$.

// 剩下的 $(1-\epsilon)\times K$ 个参与方随机选择

10. $\mathbb{P}=\mathbb{C}_E\cup RandomSelect(\mathbb{C}_E,(1-\epsilon)\times K)$.

11. 服务器发布 $\mathbb{P},T_r$, 并将最新的全局模型份额分发给集合 $\mathbb{P}$ 中的客户端.

12. 客户端 $i\in\mathbb{P}$ 在本地恢复出全局模型, 并计算本地训练迭代轮次 $I_i^r=\left(\beta\times\frac{\min(T_r-t_i^{r-1},0)}{t_i^{\text{comp}}}+1\right)\times I_{\text{base}}$ 和训练样本集大小 $\left|B_i^r\right|=Batchsize\times I_i^r$, 接着启动本轮训练并将结果模型生成 3PC 份额 $[W_i^r]$ 上传至对应服务器.

13. 服务器在本地对所有参与方上传的模型份额进行聚合 $\left[W_g^r\right]=\frac{\sum_{i\in\mathbb{P}}D_i\cdot[W_i^r]}{\sum_{i\in\mathbb{P}}D_i}$, 得到新一轮的全局模型份额.

14. 当 r 达到预设轮次时, 训练结束, 服务器恢复出全局模型 W_g^r 并将模型发布; 否则返回到步骤 4 继续新一轮的参与方选择.

5 安全性分析

为了证明安全多方计算框架下本文的客户端选择过程在半诚实威胁模型下是安全的, 即每个服务器都完全遵

守协议执行且不会串通, 同时又对客户端的真实数据非常好奇, 本文采用基于现实世界和理想世界的模拟证明, 即服务器在协议执行期间接收到的消息视图都可以被模拟, 这意味着服务器在协议过程接收到的消息视图中没有获取任何关于客户端上传数据的额外信息.

首先对于 3PC 秘密共享份额转换为 2PC 秘密共享份额 $[x] \xrightarrow{i,j} \langle x\rangle_{i,j}$, 以 $j=i+1,\ i\in\{1,2\}$ 的情况来举例, S_i 计算 $\langle x\rangle_i = x_i + x_{i+1}$, S_j 计算 $\langle x\rangle_j = x_{(j+1) \bmod 3}$; S_i 和 S_j 都是在本地利用持有的 3PC 份额生成新的 2PC 份额, 彼此之间并没有进行任何交互. 显然在半诚实安全模型下, 这种转换是满足安全性要求的.

对于 2PC 秘密共享份额转换为 3PC 秘密共享份额 $\langle x\rangle_{i,j} \xrightarrow{i,j} [x]$ [38]: 假设除了服务器 S_k 的另外两个服务器希望将它们之间的 2PC 秘密共享份额转换为 3PC 秘密共享份额 $[x]$, 以 $S_k = S_1$ 为例, 即将 S_2 和 S_3 之间的 2PC 份额进行转换, S_2 持有份额 x_2, S_3 持有份额 x_3, 满足 $x_2 + x_3 = x$. 假设 S_1 希望获得 x 相关信息, 由于 S_1 和 S_2 之间协商得到随机值 $r_{1,2}$, S_1 和 S_3 之间协商得到随机值 $r_{1,3}$, S_1 只能设置己方份额 $[x]_1 = (r_{1,2}, r_{1,3})$, 显然 S_1 无法获取任何关于 x 的信息. 假设 S_2 希望获得 x_3 的相关信息, 在半诚实安全模型下各方诚实地执行协议, S_1 和 S_2 之间协商得到随机值 $r_{1,2}$, S_1 和 S_3 之间协商得到随机值 $r_{1,3}$, 在 S_2 看来都是均匀随机值, 因此从 S_3 收到的 $x_3 - r_{1,3}$ 在 S_2 看来也是均匀随机的, 而 S_2 无法获取 $r_{1,3}$ 的具体值, 因此无法从 $x_3 - r_{1,3}$ 中获得任何关于 x_3 的信息. 针对 S_3 的安全性证明显然与 S_2 同理, 因此可以证明在半诚实安全模型下, 这种份额转换是满足安全性要求的.

然后证明协议 $\prod_{\text{shuffle}}$ 和 $\prod_{\text{top-}k}$ 的安全性, 最后再证明整个方案的安全性. 首先定义 $\prod_{\text{shuffle}}$ 和 $\prod_{\text{top-}k}$ 对应的理想世界函数 $\mathcal{F}_{\text{shuffle}}$ 和 $\mathcal{F}_{\text{top-}k}$, 如协议 4 和 5 所示.

协议 4. 理想函数 $\mathcal{F}_{\text{shuffle}}$. 假设 S_1、S_2 和 S_3 持有 3PC 秘密共享数组 $[V] = ([v_1], [v_2], \ldots, [v_n])$, 并将秘密共享数组输入理想函数 $\mathcal{F}_{\text{shuffle}}$, 最终 S_1、S_2 和 S_3 从 $\mathcal{F}_{\text{shuffle}}$ 得到输出 $[V'] = ([v'_1], [v'_2], \ldots, [v'_n])$, 且 $[V]$ 与 $[V']$ 满足如下条件.

1) $[V']$ 是 $[V]$ 的重新随机化.

2) $[V']$ 与 $[V]$ 元素之间按照某个的随机排列相互对应.

协议 5. 理想函数 $\mathcal{F}_{\text{top-}k}$. 假设 S_1、S_2 和 S_3 持有 3PC 秘密共享数组 $[V] = ([v_1], [v_2], \ldots, [v_n])$, 并将秘密共享数组输入理想函数 $\mathcal{F}_{\text{top-}k}$, 最终 S_1、S_2 和 S_3 从 $\mathcal{F}_{\text{top-}k}$ 得到输出 $[V'] = ([v'_1], [v'_2], \ldots, [v'_k])$, 且 $[V]$ 与 $[V']$ 满足: $[V']$ 中的元素是 $[V]$ 中前 k 个最大元素的重新随机化.

下面证明 $\prod_{\text{shuffle}}$ 的安全性, 将借助定理 1 进行证明.

定理 1. 协议 2 的 $\prod_{\text{shuffle}}$ 在半诚实威胁模型下不会将隐私数据 V 及随机排列顺序泄露给服务器.

证明: 以 S_1 为例, 假设 S_1 希望从接收到的消息中获取有关数组 $V = (v_1, v_2, \ldots, v_n)$ 的信息以及打乱后的 $[V'] = ([v'_1], [v'_2], \ldots, [v'_n])$ 和 $[V] = ([v_1], [v_2], \ldots, [v_n])$ 的对应关系, 模拟器 $\mathcal{P}$ 将模拟与 S_1 的交互过程.

(1) 首先在协议 2 的第 1 步中, $\mathcal{P}$ 选择两个随机排列 π'_1 和 π'_3 发送给 S_1, 并与 S_1 之间协商出一个随机向量 $\vec{R}_1 = \{r_1^1, r_2^1, \ldots, r_n^1\}$, 作为 2PC 零秘密共享 $\langle\vec{0}'\rangle_{1,2} = (\vec{R}_1 - \vec{R}_1)$.

(2) 在协议 2 的第 2 步中 $\mathcal{P}$ 模拟 S_2 与 S_1 进行交互, 其中 S_1 遵照协议计算 $\langle V_1\rangle_1 = \langle\pi_1(\langle V\rangle_1)\rangle + \langle\vec{0}\rangle_{1,2}$, $\mathcal{P}$ 选择包含 n 个随机值的向量 $\langle V'\rangle_2 = (u_1^1, u_2^1, \ldots, u_n^1)$ 来模拟 $\langle V\rangle_2$; 随后 S_1 将自己在本地计算的新份额 $\langle V_1\rangle_1$ 发送给 $\mathcal{P}$.

(3) 然后在协议 2 的第 3 步中, $\mathcal{P}$ 从 $\mathbb{Z}_{2^{l+s}}$ 中采样一组随机值 $\langle V'_2\rangle_2 = (u_1^2, u_2^2, \ldots, u_n^2)$ 来模拟 S_1 从 S_2 那里接收到的消息, 并发送给 S_1.

(4) 最后在协议 2 的第 4 步中, $\mathcal{P}$ 仿照步骤 (1) 与 S_1 之间协商出一个随机向量 $\vec{R}_2 = \{r_1^2, r_2^2, \ldots, r_n^2\}$, 作为 2PC 零秘密共享 $\langle\vec{0}'\rangle_{1,3} = (\vec{R}_2, -\vec{R}_2)$, S_1 依据 π'_3 模拟计算出 $\langle V'_3\rangle_1 = \pi'_3(\langle V'_2\rangle_1) + \langle\vec{0}'\rangle_{1,3}$.

(5) 第 5 步中 $\mathcal{P}$ 调用理想函数 $\mathcal{F}_{\text{shuffle}}$, 并将输出 $[V^p]_1$ 发送给 S_1, 来模拟 S_1 期望接收到的消息.

对于模拟器 $\mathcal{P}$ 与服务器 S_1 的交互可知, S_1 得到的现实世界视图和输出的联合分布为:

$$REAL_{\prod_{\text{shuffle}}} = \left\{\pi_1, \pi_3, \langle\vec{0}\rangle_{1,2}, \langle V_2\rangle_2, \langle\vec{0}\rangle_{1,3}, [V']_1\right\}.$$

理想世界的联合分布为:

$$IDEAL_{\mathcal{F}_{\text{shuffle}}} = \left\{\pi'_1, \pi'_3, \langle\vec{0}'\rangle_{1,2}, \langle V'_2\rangle_2, \langle\vec{0}'\rangle_{1,3}, [V^p]_1\right\}.$$

可知 S_1 都诚实地执行协议时, 对于 S_1 来说这两个世界的联合分布是计算不可区分的. 显然 S_2 和 S_3 的证明同理. 因此证明了 $\prod_{\text{shuffle}}$ 协议在半诚实威胁模型下安全地实现了理想函数 $\mathcal{F}_{\text{shuffle}}$, 定理 1 证毕.

接下来借助定理 2 来证明 $\prod_{\text{top-}k}$ 的安全性.

定理 2. 协议的 $\prod_{\text{top-}k}$ 在半诚实威胁模型下不会将隐私数据 V 及 V 中元素的大小关系暴露给服务器.

证明: 以 S_1 为例, 假设 S_1 希望从接收到的消息中获取有关数组 V 的信息以及 V 中元素的大小关系. 对于所有服务器来说模拟证明都是相同的, 因此以 S_1 为例, 模拟器 $\mathcal{P}$ 将模拟与 S_1 进行交互.

(1) 首先在协议 1 的第 1 步中, $\mathcal{P}$ 模拟 $\mathcal{F}_{\text{shuffle}}$, 从服务器 S_1 那里接收 $[V]_1$, 然后随机选择 n 个值发送给 S_1, 以此来模拟 $\mathcal{F}_{\text{shuffle}}$ 打乱的数组 $[V^p]_1 = v_1^p, v_2^p, \ldots, v_n^p$.

(2) 假设在执行第 i 轮迭代时如下.

1) 在协议 1 第 3 步的 a) 中, $\mathcal{P}$ 选择两组随机密钥 $L_2 = \{l_1^2, l_2^2, \ldots, l_n^2\}$, $L_3 = \{l_1^3, l_2^3, \ldots, l_n^3\}$, 将 L_2 发送给 S_1, 并从 S_1 那里获取 $K_1 = (k_1^1, k_2^1, \ldots, k_n^1)$. 然后 S_1 计算 $\rho_1 = F_{K_1}(cnt)$, $\rho_2' = F_{L_2}(cnt)$, 并设置 $\alpha_1 = \rho_1 - \rho_2'$; $\mathcal{P}$ 计算 ρ_2' 和 $\rho_3' = F_{L_3}(cnt)$, 以及 $\alpha_2 = \rho_2' - \rho_3'$, $\alpha_3' = \rho_3' - \rho_1$. S_1 将 α_1 发送给 $\mathcal{P}$, $\mathcal{P}$ 将 α_2' 发送给 S_1. 最终 S_1 设置 $\left[\vec{0}\right]_1 = \alpha_1 - \alpha_2'$, $\mathcal{P}$ 设置 $\left[\vec{0}'\right]_2 = \alpha_2' - \alpha_3'$, $\left[\vec{0}'\right]_3 = \alpha_3' - \alpha_1$.

2) 在协议 1 第 3 步的 b) 中, 由于 $[V]$ 中元素份额已经过顺序打乱且每一轮都进行了重新随机化, 所以第 3 步的 d) 中得到的 $\vec{b}$ 在 S_1 看来仍是一个随机值, 因此 $\mathcal{P}$ 可以随机选择 $|\Phi|$ 个随机比特作为比较结果份额 $\vec{b}' = \left(b_1', b_2', \ldots, b_{|\Phi|}'\right)$.

根据模拟器 $\mathcal{P}$ 与服务器 S_1 的交互可知, S_1 得到的现实世界视图和输出的联合分布为:

$$REAL_{\Pi_{\text{top-}k}} = \left\{[V']_1, K_2, \alpha_2, \left[\vec{b}\right]_1, \vec{b}\right\}.$$

理想世界的联合分布为:

$$IDEAL_{\Pi_{\text{top-}k}} = \left\{[V^p]_1, L_2, \alpha_2', \left[\vec{b}'\right]_1, \vec{b}'\right\}.$$

与定理 1 的证明同理, 可知当 S_1 诚实地执行协议时, 对于 S_1 来说两个世界的联合分布是计算不可区分的. 由此证明协议 1 的 $\prod_{\text{top-}k}$ 在半诚实威胁模型下可以安全地实现理想函数 $\mathcal{F}_{\text{top-}k}$.

最终本文所构造的隐私保护异构联邦学习参与方选择协议的安全性可以归结为各个底层构造模块的安全性. 运用组合定理, 依次调用安全子协议, 然后这些协议对组合函数进行安全的求值. 每个子协议以秘密共享值作为输入, 输出也是秘密共享. 输入数据和中间结果的隐私性得到保护. 因此本文的隐私保护异构联邦学习参与方选择方案在半诚实威胁模型下是安全的.

6 实验分析

本节实现了所构建的隐私保护 FL 参与方选择协议并将其用于联邦学习, 同时还进行了实验来评估所提方案的性能. 首先介绍实验设置, 然后展示隐私保护异构联邦学习参与方选择方案的性能.

6.1 实验设置

本文采用开源 Paillier 算法库和 C++实现协议, 再利用 Python 3.8.15、TensorFlow 2.11.1 等算法包实现联邦学习. 然后在运行 Ubuntu 22.04、Intel Core CPU i7-12700F 2.10 GHz 和 NVIDIA RTX A6000 GPU 的服务器上多次评估协议的有效性和效率, 得到平均指标.

为了进行基准测试和比较, 使用 4 层 CNN 架构进行客户端本地模型训练, 第 1 层为二维卷积层, 具有 3 个输入通道, 32 个输出通道, 卷积核尺寸为 3×3. 接下来是 ReLU 激活函数以及 2×2 的 Maxpool. 第 2 层同样为二维卷积层, 具有 32 个输入通道、64 个输出通道, 卷积核尺寸为 3×3, 并继续采用 ReLU 激活函数与 2×2 的 Maxpool. 第 3 层首先将特征图展平, 然后进入全连接层, 该层包含 128 个神经元, 采用 ReLU 激活函数, 所有网络层权重均使用 He 正态初始化, 以提升训练收敛稳定性. 最终一层为线性输出层, 包含 10 个输出单元, 用于类别预测, 对应 10 个分类类别.

本文在 MNIST、Fashion-MNIST 和 CIFAR-10 数据集上进行了实验. MNIST 和 Fashion-MNIST 数据集均包含 70 000 张 28×28 大小的灰度图像, 其中 60 000 张用于训练, 10 000 张用于测试, 区别在于 MNIST 数据集为手写的单个数字 0–9 的灰度图像, 而 Fashion-MNIST 为服装物品图像. CIFAR-10 数据集包含 50 000 张训练图像和 10 000

张测试图像, 由大小为 32×32 的彩色图像组成, 例如飞机、汽车, 共包括 10 个类别, 每个类别 6 000 张图像. 设置的模型训练参数如下: 训练轮数 $R = 500$, 每轮训练的 $epoch = 1$, 学习率 $\eta = 0.5$, 批量大小 $Batch = 64$.

为模拟数据异构环境, 我们设置客户端本地数据集标签重叠率低于 20%. 此外, 我们还保证客户端最大最小样本量差异在 3 倍以上, 以此实现非独立同分布 (Non-IID) 数据集的数量偏移要求. 为模拟系统异构环境, 我们参考文献 [39] 中的数据, 设置客户端 CPU 性能跨度为 1.5–4.0 GHz, 并根据文献 [40] 中提供的移动设备上的网络测量结果, 模拟具有不同网络吞吐量的客户端的通信时间消耗, 最佳的通信信道传输速度可比最差信道快 100 倍以上.

6.2 性能评估

首先, 使用未使用隐私保护的联邦学习参与方选择协议来对比测试本文协议的性能, 以分析其是否对模型训练有效, 同时不影响选择结果的准确性. 实验将数据集划分到 30 个客户端中. 然后让聚合服务器分别选择 8、12、16、20、24 个客户端在 3 个不同的数据集上进行模型训练. 这 5 个实例分别用 I-8、I-12、I-16、I-20、I-24 表示. 如图 2(a)–(c) 所示, 我们测试了在不同数据集上训练 500 轮时, 所提出的隐私保护方案和未使用 MPC 技术进行隐私保护的方案在模型准确率上的差异. 结果显示, 在 MNIST 数据集上, 隐私保护方案与明文方案的准确率差值均小于 0.61%; 在 Fashion-MNIST 数据集上, 隐私保护方案与明文方案的准确率差值均小于 1.02%; 在 CIFAR-10 数据集上, 隐私保护方案与明文方案的准确率差值均小于 0.82%. 可以看出所提出的隐私保护方案和未使用 MPC 技术进行隐私保护的方案在不同数据集以及在不同的参与方选择比例下都达到了相似训练效果. 为进一步拓展这一结论, 本文还测试在不同数据集上训练 750 轮时, 所提出的隐私保护方案和明文方案在模型准确率上的差异, 如图 2(d)–(f) 所示. 结果显示, 在 MNIST 数据集上, 隐私保护方案与明文方案的准确率差值均小于 0.64%; 在 Fashion-MNIST 数据集上, 隐私保护方案与明文方案的准确率差值均小于 0.98%; 在 CIFAR-10 数据集上, 隐私保护方案与明文方案的准确率差值均小于 0.56%. 实验结果更加证明了所提出的隐私保护方案对模型精度影响极小.

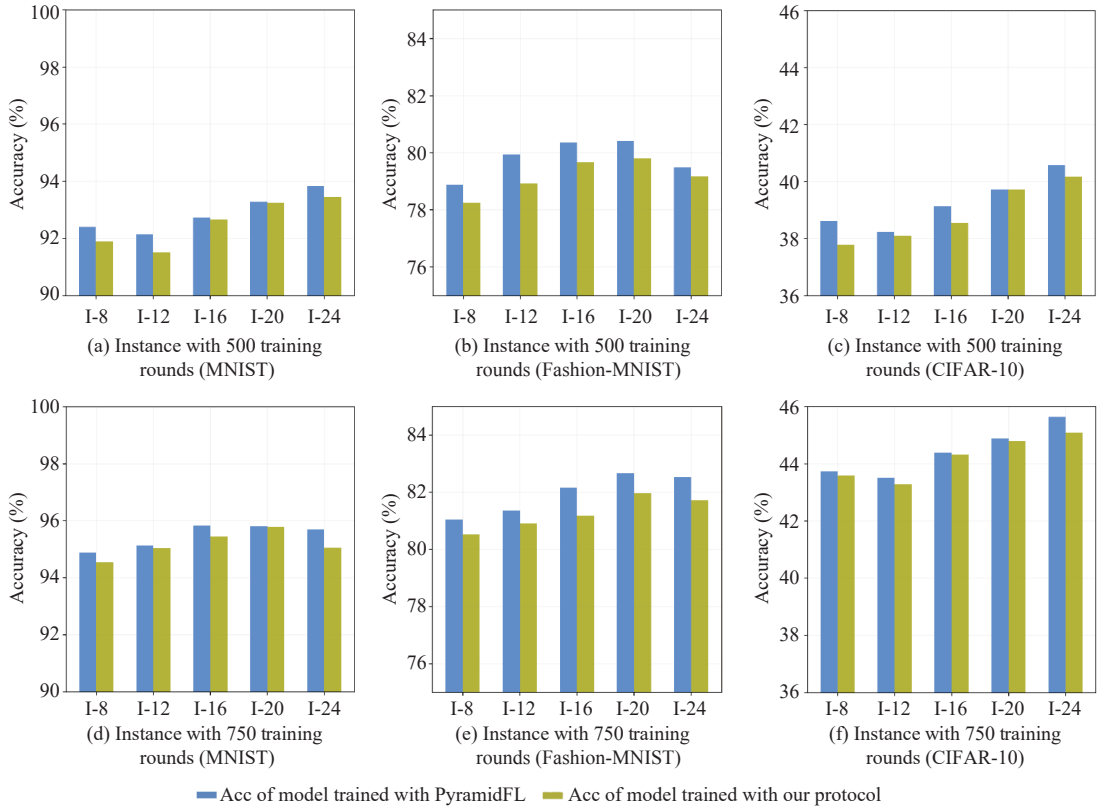

图 2 本文协议与明文训练下的模型准确率比较

准确率略有差异的主要原因在于数据转换带来的精度损失: 在协议执行过程中, 浮点型秘密值被转换为定点整数形式的随机秘密份额, 这一过程导致秘密值的小数部分被截断, 进而引发了精度误差. 如图 3 所示, 测试了协议在不同截断比特设置下对客户端选择的准确率影响, 即比较隐私保护情况下的参与方选择结果与明文下的参与方选择结果的差异. 结果显示, 当截断比特为 1 时, 代表定点数中只有 1 bit 用来表示浮点数秘密值中的小数, 显然在大多数情况下, 1 bit 不能完全表示小数, 很多小数因截断而被丢弃, 因此计算结果的准确率也是最低的. 当截断比特为 8 时, 准确率已经达到 95% 以上. 当截断比特为 16 时, 测试准确率接近 100%, 这也证明对准确率损失的分析是合理的.

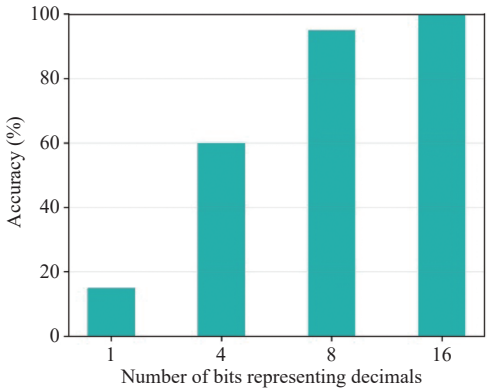

图 3　隐私保护参与方选择协议在不同截断比特下的准确率

接着讨论隐私保护协议的高效性. 首先测试不同参与方选择比例下, 未使用隐私保护技术的完整联邦学习训练 (包括明文下的参与方选择过程) 与使用隐私保护技术的协议 3 分别执行 500 轮所需时间与通信开销对比, 不同数据集下的时间开销如图 4(a)–(c) 所示, 通信开销如图 5(a)–(c) 所示. 图 4(a) 和图 5(a) 显示在 CIFAR-10 数据集上, 引入隐私保护方案的平均时间开销仅增加 2.09%, 而引入隐私保护方案的平均通信开销约为明文方案的 3.23×; 图 4(b) 和图 5(b) 显示在 Fashion-MNIST 数据集上, 引入隐私保护方案的平均时间开销增加 2.32%, 而引入隐私保护方案的平均通信开销约为明文方案的 3.24×; 图 4(c) 和图 5(c) 显示在 MNIST 数据集上, 引入隐私保护方案的平均时间开销增加 2.73%, 而引入隐私保护方案的平均通信开销约为明文方案的 3.21×. 此外, 实验还分别测试当客户端数量为 20、30、40、50 时, 每轮选择的参与方比例固定时, 未使用隐私保护技术的完整联邦学习训练 (包括明文下的参与方选择过程) 与使用隐私保护技术的协议 3 分别执行 500 轮所需时间与通信开销对比. 经过实验验证, 不同的选择比例并不会影响模型的收敛速度, 因此, 这里仅展示当每轮选择的参与方比例为 60% 时的实验结果. 不同数据集下的时间开销如图 4(d)–(f) 所示, 通信开销如图 5(d)–(f) 所示. 图 4(d) 和图 5(d) 显示在 CIFAR-10 数据集上, 引入隐私保护方案的平均时间开销仅增加 1.09%, 而引入隐私保护方案的平均通信开销约为明文方案的 3.32×; 图 4(e) 和图 5(e) 显示在 Fashion-MNIST 数据集上, 引入隐私保护方案的平均时间开销增加 1.27%, 而引入隐私保护方案的平均通信开销约为明文方案的 3.29×; 图 4(f) 和图 5(f) 显示在 MNIST 数据集上, 引入隐私保护方案的平均时间开销增加 1.33%, 而引入隐私保护方案的平均通信开销约为明文方案的 3.30×. 此外, 我们还发现随着联邦学习客户端数量的增加, 各方的计算时间和通信开销也在增加. 这是因为当客户端总数增加时, top-k 的数量随之线性增加, 导致协议运行时间和通信开销增加.

为了更直观地展示隐私保护过程中联邦学习参与方选择协议的性能, 即根据所有客户端的效用值进行 top-k 计算过程的时间与通信开销. 由于每一轮的 top-k 计算的时间与通信开销基本相同, 因此, 实验测试了在客户端数量为 30, 每轮客户端选择数量为 16 以及不同安全参数的条件下, 执行一次客户端选择过程 (即测试隐私保护的 top-k 搜索协议) 所需要的平均通信与计算开销, 协议执行开销的具体值如表 2 所示. 可以看出本文设计的协议是高效的, 支持各方以较少的时间与通信开销执行隐私保护协议.

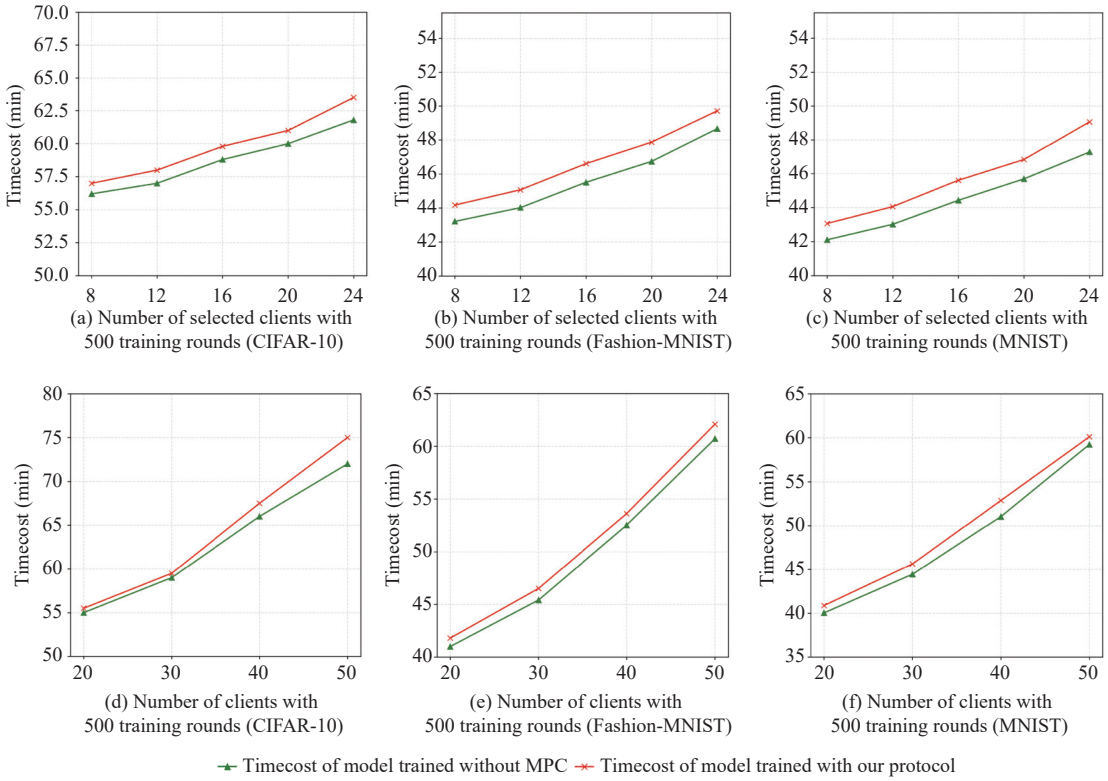

图 4　使用本文隐私保护协议的模型训练与明文下的模型训练的时间开销对比

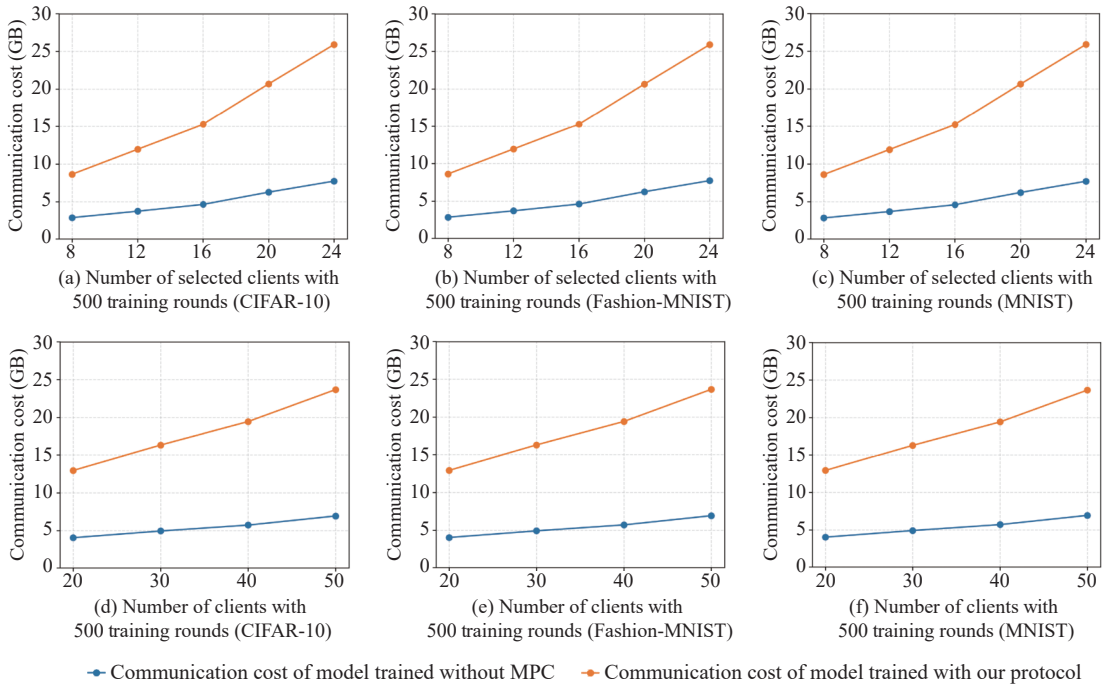

图 5　使用本文隐私保护协议的模型训练与明文下的模型训练的通信开销对比

表 2 隐私保护联邦学习参与方选择协议在不同安全参数下的执行成本

开销	$s=8$	$s=16$	$s=32$	$s=64$
时间成本 (ms)	5.072	5.8617	6.2714	6.6403
通信成本 (KB)	18.46	33.27	58.84	116.82

7 总 结

本文设计了一个隐私保护的联邦学习模型训练参与方选择协议, 支持聚合服务器和协调器输出一组有利于提升模型准确率和训练效率的客户端集合作为每轮联邦学习参与方. 在参与方选择过程中利用 3PC 秘密共享来保护客户端的隐私数据, 并设计了一种隐私保护 top-k 搜索协议来防止参与方选择过程的隐私泄露, 使得聚合服务器和协调器只能得到参与方集合, 而不会获取任何其他的衍生信息. 本文还基于 UC 安全框架对所提出的协议进行安全性分析. 此外, 实验表明本文协议与明文设置下的模型训练效率相当, 保证了协议的开销是可以接受的. 未来会在协议中引入验证机制来适应恶意安全需求, 以处理存在恶意服务器不遵守协议执行的情况.

References

[1] Kairouz P, McMahan HB, Avent B, *et al*. Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 2021, 14(1-2): 1–210. [doi: 10.1561/2200000083]

[2] McMahan B, Moore E, Ramage D, Hampson S, Arcas BAY. Communication-efficient learning of deep networks from decentralized data. In: Proc. of the 20th Int'l Conf. on Artificial Intelligence and Statistics. Fort Lauderdale: PMLR, 2017. 1273–1282.

[3] Hsieh K, Phanishayee A, Mutlu O, Gibbons PB. The Non-IID data quagmire of decentralized machine learning. In: Proc. of the 37th Int'l Conf. on Machine Learning. PMLR, 2020. 408.

[4] Wang H, Kaplan Z, Niu D, Li BC. Optimizing federated learning on Non-IID data with reinforcement learning. In: Proc. of the 2020 IEEE INFOCOM Conf. on Computer Communications. Toronto: IEEE, 2020. 1698–1707. [doi: 10.1109/INFOCOM41043.2020.9155494]

[5] Wang JY, Joshi G. Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD. In: Proc. of the 2nd Conf. on Machine Learning and Systems. Stanford: mlsys.org, 2019. 212–229.

[6] Peng XC, Huang ZJ, Zhu YZ, Saenko K. Federated adversarial domain adaptation. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: OpenReview.net, 2020.

[7] AbdulRahman S, Tout H, Mourad A, Talhi C. FedMCCS: Multicriteria client selection model for optimal IoT federated learning. IEEE Internet of Things Journal, 2021, 8(6): 4723–4735. [doi: 10.1109/jiot.2020.3028742]

[8] Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge. In: Proc. of the 2019 IEEE Int'l Conf. on Communications (ICC). Shanghai: IEEE, 2019. 1–7. [doi: 10.1109/ICC.2019.8761315]

[9] Lai F, Zhu XF, Madhyastha HV, Chowdhury M. Oort: Efficient federated learning via guided participant selection. In: Proc. of the 15th USENIX Symp. on Operating Systems Design and Implementation (OSDI 21). USENIX Association, 2021. 19–35.

[10] Luo B, Xiao WL, Wang SQ, Huang JW, Tassiulas L. Tackling system and statistical heterogeneity for federated learning with adaptive client sampling. In: Proc. of the 2022 IEEE INFOCOM Conf. on Computer Communications. London: IEEE, 2022. 1739–1748. [doi: 10.1109/INFOCOM48880.2022.9796935]

[11] Zhang RL, Xu ZN, Yin H. Scout: An efficient federated learning client selection algorithm driven by heterogeneous data and resource. In: Proc. of the 2023 IEEE Int'l Conf. on Joint Cloud Computing (JCC). Athens: IEEE, 2023. 46–49. [doi: 10.1109/JCC59055.2023.00012]

[12] Nasr M, Shokri R, Houmansadr A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In: Proc. of the 2019 IEEE Symp. on Security and Privacy (SP). San Francisco: IEEE, 2019. 739–753. [doi: 10.1109/SP.2019.00065]

[13] Melis L, Song CZ, de Cristofaro E, Shmatikov V. Exploiting unintended feature leakage in collaborative learning. In: Proc. of the 2019 IEEE Symp. on Security and Privacy (SP). San Francisco: IEEE, 2019. 691–706. [doi: 10.1109/SP.2019.00029]

[14] Zhao B, Mopuri KR, Bilen H. iDLG: Improved deep leakage from gradients. arXiv:2001.02610, 2020.

[15] Acar A, Aksu H, Uluagac AS, Conti M. A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys, 2019, 51(4): 79. [doi: 10.1145/3214303]

[16] Carter H, Mood B, Traynor P, Butler K. Secure outsourced garbled circuit evaluation for mobile devices. Journal of Computer Security,

2016, 24(2): 137–180. [doi: 10.3233/JCS-150540]

[17] Xu RH, Baracaldo N, Zhou Y, Anwar A, Ludwig H. HybridAlpha: An efficient approach for privacy-preserving federated learning. In: Proc. of the 12th ACM Workshop on Artificial Intelligence and Security. London: ACM, 2019. 13–23. [doi: 10.1145/3338501.3357371]

[18] Shamir A. How to share a secret. Communications of the ACM, 1979, 22(11): 612–613. [doi: 10.1145/359168.359176]

[19] Liu Y, Lai JZ, Wang Q, Qin XR, Yang AJ, Weng J. Robust publicly verifiable covert security: Limited information leakage and guaranteed correctness with low overhead. In: Proc. of the 29th Int'l Conf. on Advances in Cryptology. Guangzhou: Springer, 2023. 272–301. [doi: 10.1007/978-981-99-8721-4_9]

[20] Huang HL, Shi W, Feng YH, Niu CY, Cheng GQ, Huang JC, Liu Z. Active client selection for clustered federated learning. IEEE Trans. on Neural Networks and Learning Systems, 2024, 35(11): 16424–16438. [doi: 10.1109/TNNLS.2023.3294295]

[21] Cho YJ, Wang JY, Joshi G. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. arXiv:2010.01243, 2020.

[22] Du T, Chen XX, Kou G, Zhao Y, Xu CQ. Clustered federated learning with cloud-edge personalized model decoupling. Chinese Journal of Computers, 2025, 48(2): 407–432 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2025.00407]

[23] Shin J, Li YC, Liu YX, Lee SJ. FedBalancer: Data and pace control for efficient federated learning on heterogeneous clients. In: Proc. of the 20th Annual Int'l Conf. on Mobile Systems, Applications and Services. Portland: ACM, 2022. 436–449. [doi: 10.1145/3498361.3538917]

[24] Li CN, Zeng X, Zhang M, Cao ZC. PyramidFL: A fine-grained client selection framework for efficient federated learning. In: Proc. of the 28th Annual Int'l Conf. on Mobile Computing and Networking. Sydney: ACM, 2022. 158–171. [doi: 10.1145/3495243.3517017]

[25] Ye JY, Maddi A, Murakonda SK, Bindschaedler V, Shokri R. Enhanced membership inference attacks against machine learning models. In: Proc. of the 2022 ACM SIGSAC Conf. on Computer and Communications Security. Los Angeles: ACM, 2022. 3093–3106. [doi: 10.1145/3548606.3560675]

[26] Fu C, Zhang XH, Ji SL, Chen JY, Wu JZ, Guo SQ, Zhou J, Liu AX, Wang T. Label inference attacks against vertical federated learning. In: Proc. of the 31st USENIX Security Symp. (USENIX Security 22). Boston: USENIX, 2022. 1397–1414.

[27] Zhao JQ, Zhu H, Wang FW, Lu RX, Liu Z, Li H. PVD-FL: A privacy-preserving and verifiable decentralized federated learning framework. IEEE Trans. on Information Forensics and Security, 2022, 17: 2059–2073. [doi: 10.1109/TIFS.2022.3176191]

[28] Zhou CY, Fu AM, Yu S, Yang W, Wang HQ, Zhang YQ. Privacy-preserving federated learning in fog computing. IEEE Internet of Things Journal, 2020, 7(11): 10782–10793. [doi: 10.1109/JIOT.2020.2987958]

[29] Hijazi NM, Aloqaily M, Guizani M, Ouni B, Karray F. Secure federated learning with fully homomorphic encryption for IoT communications. IEEE Internet of Things Journal, 2024, 11(3): 4289–4300. [doi: 10.1109/JIOT.2023.3302065]

[30] Wei K, Li J, Ding M, Ma C, Yang HH, Farokhi F, Jin S, Quek TQS, Poor HV. Federated learning with differential privacy: Algorithms and performance analysis. IEEE Trans. on information forensics and security, 2020, 15: 3454–3469. [doi: 10.1109/TIFS.2020.2988575]

[31] Zeng H, Zhang MT, Liu TF, Yang AJ. A federated learning framework with blockchain-based auditable participant selection. Computers, Materials & Continua, 2024, 79(3): 5125–5142. [doi: 10.32604/cmc.2024.052846]

[32] Mu XT, Cheng K, Song AX, Zhang T, Zhang ZW, Shen YL. Privacy-preserving federated learning resistant to Byzantine attacks. Chinese Journal of Computers, 2024, 47(4): 842–861 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2024.00842]

[33] Zhang RL, Du JH, Yin H. Client selection algorithm in cross-device federated learning. Ruan Jian Xue Bao/Journal of Software, 2024, 35(12): 5725–5740 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/7085.htm [doi: 10.13328/j.cnki.jos.007085]

[34] Chen YN, Tan WZ, Zhong YJ, Kang YL, Yang AJ, Weng J. Byzantine-robust and privacy-preserving federated learning with irregular participants. IEEE Internet of Things Journal, 2024, 11(21): 35193–35205. [doi: 10.1109/JIOT.2024.3434660]

[35] Mohassel P, Rindal P. ABY^3: A mixed protocol framework for machine learning. In: Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security. Toronto: ACM, 2018. 35–52. [doi: 10.1145/3243734.3243760]

[36] Boyle E, Gilboa N, Ishai Y, Nof A. Practical fully secure three-party computation via sublinear distributed zero-knowledge proofs. In: Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security. London: ACM, 2019. 869–886. [doi: 10.1145/3319535.3363227]

[37] Demmler D, Schneider T, Zohner M. ABY—A framework for efficient mixed-protocol secure two-party computation. In: Proc. of the 22nd Annual Network and Distributed System Security Symp. San Diego: The Internet Society, 2015.

[38] Le PH, Ranellucci S, Gordon SD. Two-party private set intersection with an untrusted third party. In: Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security. London: ACM, 2019. 2403–2420. [doi: 10.1145/3319535.3345661]

[39] Ignatov A, Timofte R, Kulik A, Yang S, Wang K, Baum F. AI benchmark: All about deep learning on smartphones in 2019. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision Workshop (ICCVW). Seoul: IEEE, 2019. 3617–3635. [doi: 10.1109/ICCVW.2019.00447]

[40] Huang JX, Chen C, Pei YT, Wang ZG, Qian ZY, Qian F, Tiwana B, Xu Q, Mao ZM, Zhang M, Bahl P. MobiPerf: Mobile network measurement system. 2011. https://mobiperf.com/file/doc.pdf

附中文参考文献

[22] 杜甜, 陈星延, 寇纲, 赵宇, 许长桥. 面向云边个性化模型解耦的聚类联邦学习方法. 计算机学报, 2025, 48(2): 407–432. [doi: 10.11897/SP.J.1016.2025.00407]

[32] 穆旭彤, 程珂, 宋安霄, 张涛, 张志为, 沈玉龙. 抗拜占庭攻击的隐私保护联邦学习. 计算机学报, 2024, 47(4): 842–861. [doi: 10.11897/SP.J.1016.2024.00842]

[33] 张瑞麟, 杜晋华, 尹浩. 跨设备联邦学习中的客户端选择算法. 软件学报, 2024, 35(12): 5725–5740. http://www.jos.org.cn/1000-9825/7085.htm [doi: 10.13328/j.cnki.jos.007085]

作者简介

刘腾飞, 硕士生, 主要研究领域为机器学习, 隐私计算.

杨安家, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为人工智能, 物联网, 数据安全和隐私.

翁健, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为公钥密码学, 云安全, 区块链.

陈泯融, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为智能计算, 物联网, 信息安全.

刘逸, 博士, 讲师, 主要研究领域为安全多方计算, 零知识证明, 定时密码学, 区块链应用.

曾璜, 硕士, 助理教师, 主要研究领域为安全多方计算, 隐私保护联邦学习.