中国科学 : 信息科学 2025 年 第 55 卷 第 11 期: 2704–2721
SCIENTIA SINICA Informationis
人工智能应用中的数据安全专刊 · 论文

基于更新残差的差分隐私联邦遗忘学习机制

王腾[1], 翟林东[1], 禹勇[2*], 杨腾飞[1], 张雪锋[1], 任雪斌[3]

1. 西安邮电大学网络空间安全学院, 西安 710121
2. 陕西师范大学人工智能与计算机学院, 西安 710119
3. 西安交通大学计算机科学与技术学院, 西安 710049
* 通信作者. E-mail: yuyong@snnu.edu.cn

收稿日期: 2025–05–30; 修回日期: 2025–08–01; 接受日期: 2025–09–09; 网络出版日期: 2025–11–07

国家自然科学基金 (批准号: U24B20149, U23A20302, 62272385, 62172329, 62311540156, 62102311, 62202377)、陕西省重点研发计划 (批准号: 2025CY-YBXM-069)、陕西省重点研发计划重点产业创新链项目 (批准号: 2024GX-ZDCYL-01-09) 和陕西省科学技术协会青年人才托举计划项目 (批准号: 20240116) 资助

摘要 联邦遗忘学习 (federated unlearning) 能够从已学习的模型中删除某些特定客户端数据及这些数据对联邦学习全局模型的影响, 从而支持赋予 “被遗忘权”. 实现遗忘最直接的方法是从头开始训练模型, 但高额的计算成本限制了重训练的可行性. 现有的联邦遗忘方法大都通过移除目标客户端历史贡献或采用梯度上升逐步调整模型来达到遗忘目标. 然而, 存储历史梯度和性能恢复训练均面临较高的存储和通信开销, 限制了联邦遗忘效率. 此外, 联邦遗忘学习的隐私风险问题也有待进一步研究. 因此, 本文提出基于更新残差的差分隐私联邦遗忘学习机制 FedUR, 实现了隐私保护、模型效用与遗忘效率三者之间的有效平衡. 具体地, FedUR 基于差分隐私范式使得遗忘模型与重训练模型在统计意义上不可区分, 提供隐私保护增强的联邦遗忘学习. FedUR 创新性地利用更新残差来量化遗忘客户端对全局模型的历史影响, 通过移除所有历史更新残差的加权和来实现快速遗忘, 无需依赖额外的模型恢复训练过程, 有效降低了存储开销并提升遗忘效率. 此外, FedUR 还集成了数据重要性采样和周期加权聚合策略, 以减轻数据异构性对模型性能的不利影响, 同时降低存储和通信开销. 实验结果表明, FedUR 机制在提供隐私保护的同时具有较高的遗忘效率和较好的模型效用.

关键词 联邦学习, 差分隐私, 联邦遗忘, 更新残差, 模型效用

1 引言

联邦学习 (federated learning, FL) 允许多个参与者之间协作训练一个模型且无需共享各自的隐私数据, 已经成为人工智能领域广泛使用的分布式训练框架 [1, 2], 有效解决了数据隐私保护、跨设备协同学习、计算资源优化和数据孤岛等问题. 尽管如此, 联邦学习模型可以记住有关训练数据的信息, 仍带来前所未有的数据安全和隐私泄露风险 [3, 4]. 当前, 数据安全与隐私受到国内外政界、工业界等的广泛重视, 针对个人数据隐私的监管措施也更加严格. 近年来, 欧盟《通用数据保护条例》(GDPR) [5]、《加

引用格式: 王腾, 翟林东, 禹勇, 等. 基于更新残差的差分隐私联邦遗忘学习机制. 中国科学: 信息科学, 2025, 55: 2704–2721, doi: 10.1360/SSI-2025-0243
Wang T, Zhai L D, Yu Y, et al. Differentially private federated unlearning mechanism based on update residuals. Sci Sin Inform, 2025, 55: 2704–2721, doi: 10.1360/SSI-2025-0243

州消费者隐私法案》(CCPA)[6]、我国《个人信息保护法》等相继出台, 赋予个人数据 “被遗忘权” (right to be forgotten), 即根据请求删除个人数据的权利. 因此, 联邦遗忘学习 (federated unlearning, FU)[7] 能够在联邦学习中, 从已学习的模型中删除某些特定数据及其对联邦学习模型的影响, 以满足人们从已训练模型中删除个人特定知识和信息的需求与权利, 并保护个人隐私.

然而, 高效的联邦遗忘学习方法是一个具有挑战性的问题. 最简单直接的联邦遗忘方法是在剩余数据上重新进行联邦学习 (retraining from scratch), 训练一个不含遗忘数据的新模型. 然而, 这种方法在实际操作中并不现实. 联邦学习中的客户端通常是动态变化的, 服务器无法召回已经退出训练的客户端参与重训练. 并且重训练过程需要消耗大量的计算资源, 导致联邦遗忘效率低下[7]. 因此, 高效的联邦遗忘学习方法要求能够从已训练的联邦学习模型中删除数据, 即避免重训练. 此外, 联邦遗忘方法还需要保证遗忘的有效性, 即在联邦学习过程中, 训练数据的影响在多轮迭代中已经隐式地嵌入到全局模型中[8], 故联邦遗忘需要能够移除训练数据的所有增量效应.

为了缓解上述问题, 众多研究者对高效的联邦遗忘方案展开了深入探索. 研究工作 [9,10] 均利用联邦学习训练中存储的历史更新知识加速重训练的过程, 以克服重训练带来的高计算开销. Liu 等[11] 提出了根据对角经验 Fisher 信息矩阵来定制快速的重训练方法, 以完全擦除已训练模型中的数据样本. Tao 等[12] 利用总变异稳定性来衡量模型参数对数据集变化的敏感性, 以实现快速精确的联邦遗忘. 尽管这些方法显著提升了重训练效率, 但仍然需要大量计算开销. 为进一步提升联邦遗忘效率, Wu 等[13] 提出了基于知识蒸馏的联邦遗忘方法, 通过从全局模型中减去客户端历史更新并通过蒸馏恢复遗忘模型的性能, 实现了高效的联邦遗忘. Zhao 等[14] 提出了利用动量退化 (momentum degradation, MoDe) 来擦除联邦学习模型中的隐式知识, 并基于内存引导对模型进行微调以提升遗忘模型的效用. Li 等[15] 提出了基于子空间的联邦遗忘方法, 通过在垂直于剩余客户端的输入空间的正交子空间中进行梯度上升来训练全局模型, 以消除遗忘客户端的贡献, 能够有效提升遗忘模型的性能且无需额外的存储. 此外, Pan 等[16] 进一步提出了基于梯度上升的联邦遗忘方案, 通过优化梯度上升损失函数并执行正交最速下降来实现联邦遗忘并保持模型整体性能.

除了关注联邦遗忘方法的有效性和效率, 联邦遗忘学习的隐私风险问题也逐渐受到众多研究者的关注. 尽管联邦遗忘的原始目的是保护数据隐私, 但最近研究发现联邦遗忘方案本身可能会以意想不到的方式危及隐私[17~19]. 联邦遗忘前后两个模型在输出分布、参数、梯度更新等方面的差异会导致遗忘数据的额外信息泄漏, 表现为成员推理攻击、模型窃取攻击等. 因此, 一些研究工作利用差分隐私 (differential privacy, DP)[20] 来保证遗忘模型不会透露关于任何训练样本的隐私, 基本思想是保证满足 DP 的遗忘模型与重训练模型在统计意义上是不可区分的, 即近似遗忘. 但大部分基于 DP 的机器遗忘算法[21,22] 都是针对经典的机器学习场景. 针对联邦学习场景, Zhang 等[23] 提出了基于差分隐私的联邦遗忘算法, 引入了梯度残差来量化增量效应, 以便从全局模型中删除梯度残差的加权和来消除遗忘客户端的影响, 并通过调整高斯 (Gauss) 噪声使得遗忘模型和重训练模型在统计意义上不可区分, 从而为联邦遗忘过程提供隐私保护. Jiang 等[24] 进一步引入自适应差分隐私机制, 设计了面向联邦遗忘的隐私分配策略以平衡隐私保护水平和遗忘性能, 同时减少了存储和通信成本.

尽管现有研究从改进重训练流程、移除目标客户端贡献以及隐私保护等方向不断探索联邦遗忘机制, 但当面对异构数据的联邦学习场景时, 现有的联邦遗忘机制在隐私保护、模型效用、遗忘效率等方面仍面临以下挑战. (1) 忽视隐私保护: 当前多数联邦遗忘方法主要关注遗忘本身的实现[12,13], 而忽略了联邦遗忘过程中的隐私泄露风险. (2) 遗忘学习效率低: 基于快速重训练[10,11] 与移除目标客户端贡献[14,16] 的联邦遗忘学习方案往往依赖大量的计算资源和上百轮次的通信迭代以提升模型效用, 且需要保存大量历史参数或梯度信息, 因而具有较高的通信开销和存储开销. (3) 模型效用下降: 现有多数方案在遗忘后, 全局模型效用会明显下降, 尤其是在异构数据环境下甚至会出现灾难性遗忘. 尽管一些工作引入了恢复训练机制以弥补性能损失, 但这不仅进一步增加资源开销, 还可能导致 “遗忘回退”, 削弱原本的遗忘效果[16]. 综上所述, 现有联邦遗忘方法在隐私保护、模型效用与遗忘效率三者

之间尚未实现良好的平衡.

针对上述问题, 本文设计了基于更新残差的差分隐私联邦遗忘学习机制——FedUR, 致力于在隐私保护、模型效用与遗忘效率三者之间实现良好平衡. FedUR 机制在联邦学习阶段引入了重要性采样策略和周期加权聚合策略, 克服数据异构性对模型效用的影响并有效降低了通信开销. FedUR 机制集成了基于更新残差的联邦遗忘学习过程, 通过从全局模型中移除所有历史更新残差的加权和来消除遗忘客户端的影响, 且能够有效避免异构数据环境中的灾难性遗忘. 由于 FedUR 执行基于差分隐私的本地模型训练, 因此不仅增强数据隐私保护, 并为遗忘模型与重训练模型提供统计意义上的不可区分性, 以防止成员推理等攻击. 本文的主要贡献如下.

(1) 本文提出了满足差分隐私的联邦遗忘学习机制 FedUR, 通过在客户端执行基于差分隐私的本地模型训练, 实现遗忘模型与重训练模型在统计意义上不可区分, 从而为联邦遗忘学习提供严格隐私保证.

(2) 本文设计了基于重要性采样和周期加权聚合的联邦学习过程, 根据样本对于模型训练的稳定性选择重要数据, 并采取周期加权聚合提升模型对本地数据的学习能力, 不仅极大降低了通信和存储开销, 还有效提升了异构数据环境中的联邦学习模型效用.

(3) 本文提出了基于更新残差的联邦学习遗忘方法, 采用更新残差来量化要遗忘客户端的增量效应, 通过移除所有历史更新残差的加权和来实现联邦遗忘, 无需保存完整的历史梯度信息且不依赖额外的通信或恢复训练过程, 显著降低了存储开销并提升了遗忘效率.

(4) 本文在多个基准数据集上进行了广泛实验验证, 结果表明遗忘机制 FedUR 能够在保证遗忘效果的同时保持良好的模型性能, 且能快速完成联邦遗忘过程, 具有较好的实用性.

2　相关工作

现有的联邦遗忘算法根据设计原理可分为 3 种类别, 包括基于重训练、移除目标客户端贡献以及逐步模型调整的遗忘方法.

基于重训练的遗忘算法的核心思路是优化重训练过程, 以实现联邦遗忘. 然而, 传统的重训练方法通常计算密集, 要求消耗大量的计算和时间资源. 因此, 针对这一问题, 许多基于重训练的遗忘算法提出了相应的改进方案. Bourtoule 等 [25] 提出了 SISA 遗忘框架, 该方法通过将训练数据划分为多个分片, 并设置模型检查点, 从中间状态快速重训练模型. 同样基于这一思路, Tao 等 [12] 提出了快速遗忘算法 FATS, 该算法通过引入稳定性控制机制并进行回溯重训练, 实现高效的联邦遗忘. Cao 等 [10] 提出了 FedRecover 算法, 通过利用训练过程中保存的历史信息估算客户端的更新, 从而加速遗忘过程. Liu 等 [11] 提出了基于对角 FIM 和泰勒 (Taylor) 一阶展开的高效重训练方法, 进一步提升了重训练效率. 此外, Liu 等 [9] 提出了客户端级数据删除算法, 利用中央服务器存储每个客户端的历史提交信息, 并通过参数校准加速重训练过程. 尽管这些基于重训练的方案大大提升了重训练的效率, 但依然需要消耗大量的计算资源, 并且往往伴随着较高的存储开销, 因而实用性较低.

移除目标客户端贡献的遗忘方法通过量化目标客户端在联邦学习中的贡献, 并通过调整全局模型的参数来消除其影响. Wang 等 [26] 提出了一种基于 TF-IDF 通道评分与剪枝的联邦遗忘方法, 通过剪除对目标类别判别性强的通道来实现遗忘. Guo 等 [27] 通过在正则化线性模型中引入一次牛顿 (Newton) 更新步骤, 消除被删除样本对模型参数的主要影响, 并通过随机扰动掩盖, 实现遗忘效果. 类似地, Golatkar 等 [28] 基于局部二次近似, 利用 Hessian 与梯度的乘积来刻画遗忘数据对模型的贡献, 并通过在全局模型中执行反牛顿步来清除遗忘数据的影响. Sekhari 等 [29] 采用牛顿法优化经验损失, 设计了特定的样本删除策略, 以删除目标数据的影响. Wu 等 [13] 提出了一种基于知识蒸馏的联邦遗忘方法, 通过从全局模型中减去客户端历史更新并通过蒸馏恢复模型性能, 实现高效遗忘. Zhang 等 [23] 通过保存每一轮的历史梯度, 量化联邦学习的增量效应, 以擦除遗忘客户端的影响. 移除目标客户端贡

献的方法大多通过数学手段量化目标数据的贡献, 并通过在全局模型中移除其影响来实现遗忘, 通常不需要大量的计算资源且时间消耗较少. 但是, 这类方法往往会引入额外的存储开销.

逐步模型调整的核心思想是通过不断训练, 使全局模型逐渐远离目标客户端的数据, 从而逐步实现遗忘. Ginart 等 [30] 针对 k 类均值聚类开发了一种删除算法, 通过随机化算法的输出, 使得模型在目标数据上的表现与从头重训练相当. Jiang 等 [24] 引入了适应性差分隐私和双层选择机制, 不仅提高了遗忘效率, 同时保护了用户隐私. Zhao 等 [14] 提出了通过动量退化的方法 MoDe, 实现目标客户端数据的逐步遗忘. Li 等 [15] 采用损失函数的逆运算, 降低遗忘客户端对模型性能的影响, 从而逐渐实现遗忘. Pan 等 [16] 提出了改进的梯度上升遗忘方法 FedOSD, 通过改进梯度上升损失公式并执行正交最速下降, 实现了遗忘目标客户端数据的同时保持模型整体性能. 一般而言, 通过逐步调整模型的策略实现遗忘需要较长的训练时间, 且调整后的模型效用会下降, 通常需要设计相应的恢复算法来逐步恢复模型的效用, 因而遗忘效率较低.

3　预备知识

3.1　联邦学习和联邦遗忘学习

联邦学习是一种分布式机器范式, 旨在多个客户端间协调训练模型, 而无需将数据集中存储在中央服务器 [31]. 每个客户端 c_i 利用本地数据集 D_i 进行模型更新, 然后将更新后的模型参数发送给服务器 S, 服务器通过联邦平均算法聚合这些参数, 从而更新全局模型. 其中 FedAvg [32] 是最经典的联邦学习算法之一, 通过平均聚合实现联邦学习. 假设有 n 个客户端, 第 i 个客户端 c_i 第 $t-1$ 轮的本地模型为 w_i^{t-1}, 则第 t 轮的模型为 $w_i^t = w_i^{t-1} - \eta\nabla\mathcal{L}_i(w_i^{t-1})$, 其中 $\mathcal{L}_i(w_i^{t-1})$ 表示客户端 c_i 在第 $t-1$ 轮次的损失函数, η 为学习率. 服务器 S 通过加权平均聚合各个客户端模型参数更新全局模型, 即 $w_t = \frac{1}{n}\sum_{i=1}^{n} w_i^t$, 其中 n 表示参与训练的客户端数量.

假设在经过 t 轮联邦学习之后, 客户端 c_i 提出了对其私有数据 D_i 的遗忘请求, 并执行一个联邦遗忘学习算法 $\mathcal{M}_U$ 来满足该请求. 遗忘算法 $\mathcal{M}_U$ 被应用到全局模型 w_t 上, 而该全局模型是经过包括客户端 c_i 在内的所有客户端训练得到的. 执行完遗忘算法可以得到遗忘模型 w_u, 即 $w_u = \mathcal{M}_U(w_t)$. 这里 $\mathcal{M}_U(\cdot)$ 可以定义为一个函数, 确保遗忘模型 w_u 与没有客户端 c_i 参与的重训练全局模型 $\tilde{w}_t$ 在性能上几乎相当.

联邦遗忘学习根据遗忘内容不同, 可分为客户端遗忘、样本遗忘和类别遗忘. 客户端遗忘是联邦学习中特有的遗忘方式, 即某个客户端申请遗忘后, 希望模型能够移除该客户端本地所持有的所有样本, 即 $D_i = \mathcal{S}_i$. 若 $\mathcal{S}_i \subset D_i$, 代表客户端希望删除其所拥有数据的部分指定样本, 称之为样本遗忘. 如果要移除联邦学习中类别 $\mathcal{C}$ 的所有本地样本, 即 $\mathcal{S}_i = \{x_i, \mathcal{C}\}_i^{n_i^{\mathcal{C}}}$, 其中 $n_i^{\mathcal{C}}$ 表示客户端 c_i 中标签为 $\mathcal{C}$ 的样本数量, 则这类遗忘类型成为类别遗忘. 本文主要针对客户端遗忘进行研究.

3.2　差分隐私

差分隐私 [20] 是当前广泛被认可的隐私保护范式, 旨在确保数据集中的个体数据隐私不被泄露. 差分隐私的核心方法是通过向查询结果添加噪声, 确保删除或添加的单个数据不会显著影响查询结果, 从而避免泄露数据集中任何记录的信息.

定义1 ((ϵ,δ)- 差分隐私)　设 $\mathcal{M}: \mathcal{D} \to \mathcal{R}$ 是一个随机机制, 定义域为 $\mathcal{D}$, 值域为 $\mathcal{R}$, 如果对于任意两个相邻输入 $d, d' \in \mathcal{D}$ 和任意的输出 $O \in \mathcal{R}$, 满足以下条件, 则称 $\mathcal{M}$ 满足 (ϵ,δ)- 差分隐私:

$$\Pr[\mathcal{M}(d) \in O] \leqslant e^{\epsilon}\Pr[\mathcal{M}(d') \in O] + \delta, \tag{1}$$

其中 ϵ 为隐私预算. 隐私预算越小, 对数据的隐私保护程度越高.

敏感度 Δ 是衡量机制 $\mathcal{M}$ 对单个数据的变化的敏感程度, 定义了在最坏情况下某个数据的改变对结果产生的最大影响. 在差分隐私中, 敏感度是决定需要添加噪声强度的关键因素.

定义2 (敏感度)　设 $\mathcal{M}:\mathcal{D}\to\mathcal{R}$ 是一个从定义域 $\mathcal{D}$ 到值域空间 $\mathcal{R}$ 的随机机制, 且 $d,d'\in\mathcal{D}$ 是两个相邻数据, 机制 $\mathcal{M}$ 的敏感度 Δ 定义为 $\Delta=\max_{d,d'\in\mathcal{D}}\|\mathcal{M}(d)-\mathcal{M}(d')\|_2$.

定义3 (瑞丽差分隐私 (Rényi differential privacy, RDP) [33])　设 $\mathcal{M}$ 是一个随机机制, $\alpha>1$ 为瑞丽散度的阶数. 如果机制 $\mathcal{M}$ 满足 (α,ϵ)-RDP, 则对于任意两个相邻输入 $d,d'\in\mathcal{D}$, 其输出分布 $\mathcal{M}(d),\mathcal{M}(d')$ 之间的 α 阶 Rényi 散度满足:

$$D_\alpha(\mathcal{M}(d)\,\|\,\mathcal{M}(d'))\leqslant\epsilon, \tag{2}$$

其中 Rényi 散度定义为 $D_\alpha(P\,\|\,Q)=\frac{1}{\alpha-1}\log\mathbb{E}_{x\sim Q}[(\frac{P(x)}{Q(x)})^\alpha]$.

定义4 (高斯机制实现 RDP [34])　给定敏感度 Δ 和噪声标准差 σ, 高斯机制实现 RDP 定义为 $\mathcal{M}'(d)=\mathcal{M}(d)+\mathcal{N}(0,\Delta^2\sigma^2)$, 其中 $\mathcal{N}(0,\Delta^2\sigma^2)$ 是标准差为 $\Delta\sigma$、均值为 0 的正态分布随机变量. 根据文献 [33], 对于任意 $\alpha>1$, 高斯机制满足 $(\alpha,\epsilon(\alpha))$-RDP, 其中 $\epsilon(\alpha)=\frac{\alpha}{2\sigma^2}$.

4　差分隐私联邦遗忘学习机制

4.1　问题设置

设有客户端集合 $C=\{c_1,c_2,\ldots,c_n\}$, 每个客户端 c_i 持有本地数据集 D_i, 满足 $\bigcup_{i=1}^n D_i=D$, 且 $D_i\cap D_j=\emptyset$ $(i\neq j)$. 经过 t 轮联邦学习训练后, 服务器聚合得到全局模型参数 w_t. 假设服务器是 "诚实但好奇" 的, 即它会遵循协议操作但也试图推断用户隐私信息. 本文研究的问题是, 当任意客户端 $c_k\in C$ $(k\in[1,n])$ 提出遗忘请求, 即不希望其本地数据 D_k 对模型 w_t 再产生任何影响, 本文旨在研究通过一个联邦遗忘机制 $\mathcal{M}_U$, 生成一个新的模型 $w_u=\mathcal{M}_U(w_t,D_k)$, 使得模型 w_u 与移除 D_k 后由其余客户端数据 $\{D_i\mid i\neq k\}$ 重新训练所得模型 $\tilde{w}_t$ 在统计意义上不可区分. 此外, 本文旨在进一步优化联邦遗忘学习机制的模型效用和遗忘效率, 并同时提供 (ε,δ)- 差分隐私保证.

4.2　FedUR 系统框架

为有效平衡异构数据下联邦遗忘学习的隐私保护、模型性能和遗忘效率, 本文提出基于更新残差的差分隐私联邦遗忘学习机制, 即 FedUR. 主要包含两个阶段: 异构数据 (非独立同分布数据) 下的联邦学习阶段, 以及基于更新残差的联邦遗忘学习阶段. FedUR 机制的整体框架如图 1 所示.

联邦学习阶段. 相较于传统联邦学习框架, FedUR 引入了重要性采样和周期加权聚合机制, 同时利用差分隐私技术提供本地隐私保护, 抵抗不可信服务器带来的威胁. 每轮训练开始时, 服务器将当前的全局模型 w_t 下发至所有客户端 $C=\{c_1,c_2,\ldots,c_n\}$, 客户端随后在本地数据上进行模型训练. 与传统联邦学习每轮上传模型参数或梯度的方式不同, FedUR 允许客户端在本地以周期 τ 进行训练, 即每训练 τ 轮上传一次更新. 在每轮本地训练中, 每个客户端 c_i 基于当前模型从本地数据集中基于重要性采样选择一批重要数据 $\mathcal{B}_i$ 进行训练, 以提升模型对代表性样本的学习能力, 并在每批次的梯度中注入噪声以实现差分隐私保护. 训练完成后, 客户端 c_i 计算其本地模型与下发全局模型之间的差异 u_i, 作为更新量上传服务器. 服务器根据更新变化程度计算加权系数 p_i, 并聚合各客户端上传的更新, 从而完成全局模型的更新.

联邦遗忘阶段. 当联邦学习经过 T 轮训练后得到当前全局模型 w_T, 若某客户端 c_k $(k\in[1,n])$ 提出遗忘请求, 则 FedUR 将启动遗忘过程. 本文引入更新残差 θ_i, 用于衡量客户端 c_i 在每一轮训练中对全局模型的具体贡献. 通过计算其各轮更新的残差并进行加权累加, 得到 c_i 对当前全局模型的总影响. 随后, FedUR 从 w_T 中减去该客户端所有的历史加权更新残差, 生成遗忘模型 w_u. 整个过程中, 差

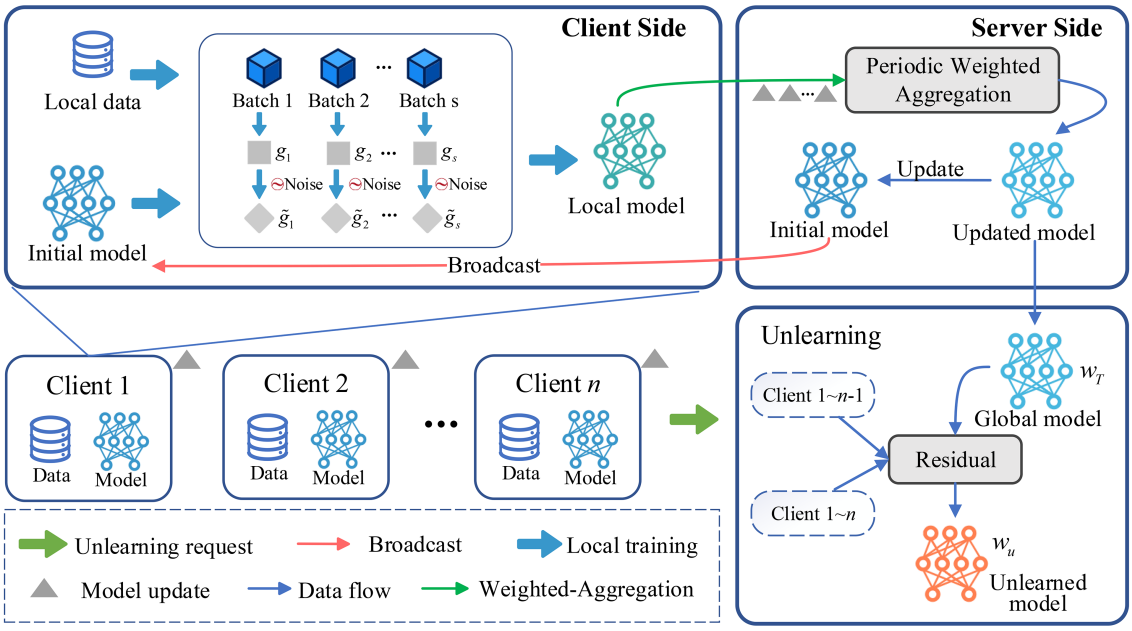

图 1　(网络版彩图) 基于更新残差的差分隐私联邦遗忘学习框架.

Figure 1　(Color online) Differentially private federated unlearning framework based on update residuals.

分隐私机制持续保障数据的隐私性, 确保遗忘模型 w_u 与移除客户端 c_k 后的重训练模型 $\tilde{w}_t$ 在统计意义上不可区分, 从而实现兼具效率、隐私保护和遗忘有效性的联邦遗忘方案.

4.3　基于重要性采样和周期加权聚合的差分隐私联邦学习

为缓解联邦学习中的通信成本、降低联邦遗忘算法的存储负担, 并提升在非独立同分布 (Non-IID) 数据条件下的模型性能与遗忘效果, 本文提出基于重要性采样和周期加权聚合的差分隐私联邦学习算法, 主要包括 4 个核心模块: 周期训练、数据重要性采样、更新加权聚合、差分隐私保护.

周期训练. 本文提出的联邦遗忘学习机制整体采用客户端周期性训练方法, 即允许每轮次参与的客户端执行多次本地更新, 然后服务器对客户端上传的更新量进行周期性聚合. 具体而言, 服务器更新全局模型后将其发送给客户端, 客户端使用本地数据基于随机梯度下降 (SGD) 方法进行 τ 次迭代以完成本地模型训练, 并将模型更新上传给服务器. 服务器进而进行新一轮的全局模型更新. 本文采用周期更新进行联邦学习训练, 能够极大降低通信开销和存储开销. 假设每个客户端执行 SGD 训练共 T_l 次, 则仅需要 $T = T_l/\tau$ 轮通信和存储. 其次, 客户端执行 τ 次本地更新后再上传更新, 避免传统联邦学习中频繁通信带来的延迟问题. 此外, 周期更新能够使客户端在多轮本地更新中更好地拟合本地数据, 增加本地数据的代表性, 减少因数据异质性而导致的聚合误差, 确保联邦学习训练在面对非独立同分布数据时具有更强的适应能力.

数据重要性采样. 在非独立同分布环境下进行的联邦学习训练中, 若在每轮本地周期性训练中采用随机采样, 容易导致训练数据分布失衡, 从而影响全局模型的收敛速度和泛化能力. 为此, 本文在本地周期训练阶段引入基于 ℓ_2 误差分数 (ℓ_2 error score) 的重要数据采样策略, 以筛选对本地模型训练更关键的数据样本, 从而提升训练效率并减少冗余计算.

对于给定样本 x 及其标签 y, 假设在第 t 轮联邦训练中, 客户端接收到的全局模型为 w_t, 将模型的预测输出记为 $f(w_t, x)$. 则样本 x 的 ℓ_2 误差的期望为

$$\mu_{w_t}(x) = \mathbb{E}_x \| f(w_t, x) - y \|_2, \tag{3}$$

其中 $\mu_{w_t}(x)$ 表示样本 x 的预测误差均值. 较高的 $\mu_{w_t}(x)$ 表明该样本较难被当前模型学习, 可能为离

群点, 而较低的 $\mu_{w_t}(x)$ 则说明该样本易于被模型拟合, 信息贡献较小.

此外, 进一步基于误差标准差来衡量样本 x 对模型训练稳定性的影响. 误差标准差定义为

$$\sigma_{w_t}(x)=\sqrt{\mathbb{V}_x\|f(w_t,x)-y\|_2}, \tag{4}$$

$\sigma_{w_t}(x)$ 反映了该样本预测误差的波动性, 即对模型稳定性的潜在影响.

根据每个样本误差的期望与标准差, 本文设定筛选区间如下:

$$[\mu_{w_t}(x)-z_1\times\sigma_{w_t}(x),\mu_{w_t}(x)+z_2\times\sigma_{w_t}(x)], \tag{5}$$

其中 z_1 和 z_2 是两个超参数, 用于灵活控制筛选样本的范围, 实现训练精度与资源消耗之间的平衡. 当某个样本的评分落入该区间时, 才被选入本轮训练. 该策略可有效过滤掉评分过高的异常样本 (避免模型过拟合) 和评分过低的冗余样本 (减少无效计算), 从而在提高模型收敛速度的同时降低本地计算开销.

更新加权聚合. 在联邦学习的第 t 轮训练中, 客户端 c_i 在完成本地周期性训练后, 将其本地模型的更新 u_i^t 上传至服务器. 为进一步减轻非独立同分布数据对联邦学习过程中的影响, 服务器采用加权聚合策略来计算全局模型的更新量. 具体地, 在聚合每个客户端上传的更新时, 服务器考虑客户端上传更新的相对变化 (通过模型更新的 ℓ_2 范数衡量) 来计算权重. 因此, 权重能够反映每个客户端的模型更新在当前训练过程中的重要性. 用 p_i^t 表示客户端 c_i 在第 t 轮上传的更新 u_i^t 的聚合权重, 即

$$p_i^t=\frac{\|u_i^t\|_2}{\sum_{j=1}^n\|u_j^t\|_2}. \tag{6}$$

得到每个参与客户端的聚合权重之后, 服务器进行加权聚合更新, 得到更新后的全局模型, 即

$$w_{t+1}=w_t+\sum_{i=1}^n p_i^t u_i^t. \tag{7}$$

差分隐私保护. 本文假设联邦学习和遗忘过程中, 服务器是诚实但好奇 (honest-but-curious) 的, 直接上传本地模型更新存在潜在隐私泄露风险. 为此, 本文采用基于高斯噪声的梯度扰动方法 [35] 实现差分隐私保护.

对于任意客户端 $c_i\in C$, 在第 t 轮联邦学习的第 s 次本地迭代中, $w_i^{t,s}$ 表示本次迭代模型, $\mathcal{B}_i^{t,s}$ 表示当前采样的重要数据批量, 则每个样本 $x\in\mathcal{B}_i^{t,s}$ 的梯度为 $g_i^{t,s}(x)=\nabla\mathcal{L}(w_i^{t,s},x)$. 为了应用差分隐私, 对每个样本 x 的梯度 $g_i^{t,s}(x)$ 进行裁剪, 即 $\bar{g}_i^{t,s}(x)=g_i^{t,s}(x)/\max(1,\|g_i^{t,s}(x)\|_2/G)$, 其中 G 为裁剪阈值. 用 $\bar{g}_i^{t,s}$ 表示客户端 c_i 在批量 $\mathcal{B}_i^{t,s}$ 上的平均梯度, 则 $\bar{g}_i^{t,s}=\frac{1}{B}\sum_{x\in\mathcal{B}_i^{t,s}}\bar{g}_i^{t,s}(x)$, 其中 $B=|\mathcal{B}_i^{t,s}|$ 为批量大小.

在每轮训练中, 客户端 c_i 通过对梯度添加噪声进行扰动, 即

$$w_i^{t,s+1}=w_i^{t,s}-\eta(\bar{g}_i^{t,s}+\gamma_i^{t,s}), \tag{8}$$

其中 $\gamma_i^{t,s}\sim\mathcal{N}(0,\Delta^2\sigma^2\mathbf{I}_d)$ 表示满足高斯分布的噪声向量. $\Delta=2G/B$ 为敏感度, 可参考定理 1. 本文选择噪声尺度 $\sigma=\sqrt{2\log\frac{1.25}{\delta}}/\epsilon$.

客户端在每次本地迭代中均注入噪声, 经过 τ 次本地更新后, 得到的本地模型参数 $w_i^{t,\tau}$ 累积了差分隐私噪声. 服务器聚合本地带噪声的更新后得到的全局模型. 联邦遗忘的目标是从带噪声的全局模型 w_t 中移除目标客户端 c_k 数据对模型的影响. 本文通过更新残差衡量 c_k 对全局模型的贡献, 并从 w_t 中移除该残差, 得到遗忘后的模型 w_u. 鉴于模型本身已具备差分隐私保护, 故遗忘后模型 w_u 仍然满足差分隐私约束, 与重训练模型 $\tilde{w}_t$ 在统计意义上不可区分, 具体参考定理 4.

Algorithm 1 DP-Fed with importance sampling and periodic weighted aggregation.

Input: Initial model w_0, noise scale σ, number of rounds T, number of clients n, local update period τ, learning rate η, important data size B, filtering parameters z_1, z_2;

Output: Global model w_T;

1: **for** $t = 0$ **to** $T-1$ **do**
2: Server sends global model w_t to all clients $C = \{c_1, c_2, \ldots, c_n\}$;
3: **for** each client $c_i \in C$ **in parallel do**
4: Initialize local model $w_i^{t,0} = w_t$;
5: **for** $s = 0$ **to** $\tau - 1$ **do**
6: **for** each sample $x \in D_i$ **do**
7: Compute error expectation $\mu_{w_i^{t,s}}(x)$ using Eq. (3);
8: Compute error standard deviation $\sigma_{w_i^{t,s}}(x)$ using Eq. (4);
9: **end for**
10: Sample important data as batch $\mathcal{B}_i^{t,s}$ using range $[\mu_{w_i^{t,s}}(x) - z_1 \times \sigma_{w_i^{t,s}}(x), \mu_{w_i^{t,s}}(x) + z_2 \times \sigma_{w_i^{t,s}}(x)]$;
11: Compute clipped average gradient $\bar{g}_i^{t,s}$ on $\mathcal{B}_i^{t,s}$ as $\bar{g}_i^{t,s} = \frac{1}{B}\sum_{x\in\mathcal{B}_i^{t,s}}(g_i^{t,s}(x)/\max(1, \|g_i^{t,s}(x)\|_2/G))$;
12: Generate Gaussian noise: $\gamma_i^{t,s} \sim \mathcal{N}(0, \Delta^2\sigma^2 I_d)$;
13: Update local model: $w_i^{t,s+1} = w_i^{t,s} - \eta(\bar{g}_i^{t,s} + \gamma_i^{t,s})$;
14: **end for**
15: Compute local update: $u_i^t = w_i^{t,\tau} - w_t$;
16: Upload u_i^t to the server;
17: **end for**
18: Compute aggregation weights p_i^t of each client c_i using Eq. (6);
19: Server aggregates: $u_t = \sum_{i=1}^n p_i^t u_i^t$;
20: Update global model: $w_{t+1} = w_t + u_t$;
21: **end for**
22: **return** w_T.

算法 1 展示了基于重要性采样和周期加权聚合的差分隐私联邦学习伪代码. 与经典的联邦学习过程相同, 在每一轮训练中, 服务器将当前全局模型 w_t 下发给所有的客户端. 每个客户端 c_i 在本地执行 τ 次训练, 即执行基于数据重采样和周期加权聚合的差分隐私联邦学习, 将本地更新 u_i^t 发送给服务器 (如第 3∼17 行所示). 具体地, 每个客户端 c_i 在本地训练的每一轮中, 首先进行数据重要性采样, 获取重要数据批量 $\mathcal{B}_i^{t,s}$ (如第 6∼10 行所示). 接着, 在重要数据批量 $\mathcal{B}_i^{t,s}$ 上执行随机梯度下降算法, 得到裁剪后的梯度 $\bar{g}_i^{t,s}$, 向梯度添加高斯噪声 $\gamma_i^{t,s}$ 并更新当前模型, 即 $w_i^{t,s+1} = w_i^{t,s} - \eta(\bar{g}_i^{t,s} + \gamma_i^{t,s})$ (如第 11∼13 行所示). 每个客户端在完成 τ 次本地训练之后, 计算本地更新 $u_i^t = w_i^{t,\tau} - w_t$, 并将本地更新 u_i^t 发送给服务器. 服务器收到所有客户端的本地更新 $\{u_i^t\}_{i=1}^n$ 之后, 计算每个客户端 c_i 的聚合权重 p_i^t, 之后进行更新加权聚合得到全局更新 $u_t = \sum_{i=1}^n p_i^t u_i^t$, 最后更新全模型 $w_{t+1} = w_t + u_t$ (如第 18∼20 行所示). 服务器和客户端之间完成 T 轮通信之后得到最终全局模型 w_T.

4.4 基于更新残差的联邦遗忘学习

在联邦学习中, 客户端的历史更新在训练过程中对后续模型产生隐性且逐步增强的影响, 这种现象被称为增量效应 [11]. 在每一轮模型更新中, 客户端 c_i 的本地更新将持续作用于全局模型, 并随训练轮次的增加不断累积, 形成对当前模型状态的深远影响. 特别是在数据分布非独立同分布的情况下, 这种累积效应更容易导致模型偏移或过拟合. 理论上, 增量效应越强, 客户端的局部更新在全局模型中的影响越显著; 相反, 若增量效应较弱, 则其更新作用往往被其他客户端的贡献所稀释. 考虑并有效建模这一增量效应, 对于设计高效、精确的联邦遗忘算法至关重要. 因此, 本文引入更新残差的概念, 量化联邦学习中参与客户端对全局模型训练的贡献, 通过减去遗忘客户端的历史总贡献, 实现客户端数据遗忘.

在联邦学习算法中, 用 w_t 表示第 t 轮的全局模型, 客户端 c_i 在本地周期性训练后的模型变化量

u_i^t 定义为

$$u_i^t = w_i^{t,\tau} - w_t, \tag{9}$$

其中 $w_i^{t,\tau}$ 表示客户端 c_i 经过 τ 轮训练后的本地模型, u_i^t 表示模型 w_t 在 c_i 上的模型更新.

服务器全局模型更新如式 (7) 所示. 假设客户端 c_n 在第 t 轮后提出遗忘请求 (为便于分析, 这里假设移除最后一个客户端 c_n. 实际上, 本文的方法支持移除任意客户端 c_k ($k \in [1,n]$)), 则去除该客户端的情况下, 全局模型的更新方式修改为

$$\tilde{w}_{t+1} = w_t + \sum_{i=1}^{n-1} \tilde{p}_i^t u_i^t, \tag{10}$$

其中 $\tilde{p}_i^t$ 表示移除客户端 c_n 之后其他客户端 c_i 的聚合权重.

为了量化客户端 c_n 对模型更新的影响, 将式 (7) 减去式 (10) 可得到

$$\begin{aligned} w_{t+1} - \tilde{w}_{t+1} &= \sum_{i=1}^{n} p_i^t u_i^t - \sum_{i=1}^{n-1} \tilde{p}_i^t u_i^t \\ &= \sum_{i=1}^{n} p_i^t u_i^t - \sum_{i=1}^{n-1} \frac{p_i^t}{1-p_n^t} u_i^t \\ &= p_n^t u_n^t + \sum_{i=1}^{n-1} p_i^t u_i^t \left(1 - \frac{1}{1-p_n^t}\right) \\ &= p_n^t u_n^t - \sum_{i=1}^{n-1} \frac{p_n^t}{1-p_n^t} p_i^t u_i^t \\ &= p_n^t \left(u_n^t - \sum_{i=1}^{n-1} \frac{p_i^t}{1-p_n^t} u_i^t\right), \end{aligned} \tag{11}$$

则对于遗忘客户端 c_n 而言, 第 t 轮的更新残差 θ_n^t 为

$$\theta_n^t = p_n^t \left(u_n^t - \sum_{i=1}^{n-1} \frac{p_i^t}{1-p_n^t} u_i^t\right). \tag{12}$$

考虑到联邦学习过程中数据产生的增量效应 [19, 23], 联邦遗忘时不仅要移除客户端数据, 还需要移除该客户端的所有历史更新残差, 以完全移除其对全局模型的整体影响. 实际联邦学习中, 不同轮次的更新残差与全局模型更新方向不完全一致. 与全局模型更新方向越接近, 说明当前更新越有效, 越有助于全局模型收敛, 即贡献度越大. 因此, 对全局模型贡献越大的更新残差, 越应该被移除. 假设遗忘客户端为 c_k ($k \in [1,n]$), 在每一轮联邦学习过程中, 本文根据遗忘客户端的更新 u_k^t 和全局更新 u_t 的对齐程度来计算遗忘客户端 c_k 在第 t 轮的更新残差 θ_k^t 的权重 λ_k^t. 具体地, 本文利用余弦相似度来衡量遗忘客户端更新 u_k^t 和全局更新 u_t 的对齐程度. 计算公式为

$$d\left(u_t, u_k^t\right) = \mathrm{ReLU}\left(\cos\left(u_t, u_k^t\right)\right) = \mathrm{ReLU}\left(\frac{\langle u_t, u_k^t \rangle}{\|u_t\|_2 \|u_k^t\|_2}\right), \tag{13}$$

其中 $\langle , \rangle$ 表示内积运算符, $\|\cdot\|_2$ 表示向量的 ℓ_2 范数, ReLU 函数保证结果为正数. 则遗忘客户端 c_k 在第 t 轮的更新残差 θ_k^t 的权重 λ_k^t 为

$$\lambda_k^t = \frac{d(u_t, u_k^t)}{\sum_{j=0}^{T-1} d(u_j, u_k^j)}. \tag{14}$$

Algorithm 2 DP-compliant federated unlearning based on update residuals (FedUR).

Input: Global model w_T after T rounds, unlearning request of client c_k ($k \in [1, n]$), local updates $\{u_i^t\}_{i=1}^n$ submitted by clients;

Output: Unlearning global model w_u;

1: **for** $t = 0$ **to** $T-1$ **do**
2: Server aggregates updates: $u_t = \sum_{i=1}^n p_i^t u_i^t$;
3: Update global model: $w_{t+1} = w_t + u_t$;
4: **end for**
5: Client c_k requests for unlearning after the T round;
6: **for** $t = 0$ **to** $T-1$ **do**
7: Compute the update residual θ_k^t of client c_k as $\theta_k^t = p_k^t(u_k^t - \sum_{i\neq k} \frac{p_i^t}{1-p_k^t} u_i^t)$;
8: Compute the degree of alignment $d(u_t, u_k^t)$ between u_t and u_k^t using Eq. (13);
9: Normalize the weight λ_k^t of update residual using Eq. (14);
10: **end for**
11: Subtract a weighted sum of θ_k^t from w_T as $w_u = w_T - \sum_{t=0}^{T-1} \lambda_k^t \cdot \theta_k^t$;
12: **return** w_u.

当客户端 c_k 在 T 轮后提出遗忘时, 由当前的全局模型 w_T 减去客户端 c_k 的所有历史加权更新残差即可得到遗忘模型 w_u, 即

$$w_u = w_T - \sum_{t=0}^{T-1} \lambda_k^t \cdot \theta_k^t. \tag{15}$$

算法 2 展示了基于更新残差的联邦遗忘学习算法的伪代码. 算法第 1∼4 行为正常的联邦学习过程. 当客户端 c_k 在 T 轮后提出遗忘时, 服务器端首先计算该客户端在 T 轮通信中的所有历史残差 $\{\theta_k^t\}_{t=0}^{T-1}$ (如第 7 行所示), 其中 $\theta_k^t = p_k^t(u_k^t - \sum_{i\neq k} \frac{p_i^t}{1-p_k^t} u_i^t)$. 接着根据客户端本地更新与全局更新的对齐程度确定更新残差的权重 λ_k^t (如第 8 和 9 行所示). 最后, 全局模型 w_T 减去遗忘客户端 c_k 的所有历史更新残差的加权和即可得到遗忘模型 w_u (如第 11 行所示).

4.5　隐私分析与讨论

本节介绍 FedUR 算法的相关分析和讨论, 包括隐私分析、防护的攻击类型范围和遗忘粒度.

隐私分析. 首先基于 Rényi 差分隐私范式, 给出基于重要性采样和周期加权聚合的差分隐私联邦学习的隐私证明; 接着证明基于本文提出的联邦遗忘算法得到的遗忘模型与利用重训练算法得到的遗忘模型在统计意义下不可区分, 即能够提供隐私保护.

定理1 (敏感度上界)　给定裁剪阈值 G, 若每个样本 x 的梯度 ℓ_2 范数满足 $\|\nabla\mathcal{L}(w,x)\|_2 \leqslant G$, 则批量平均梯度的敏感度 Δ 满足 $\Delta = \sup_{\mathcal{B}_i^{t,s}, \mathcal{B}_i^{t,s'}} \|g(w_i^{t,s}, \mathcal{B}_i^{t,s}) - g(w_i^{t,s}, \mathcal{B}_i^{t,s'})\|_2 \leqslant \frac{2G}{B}$.

证明　设 $\mathcal{B}_i^{t,s}$ 与 $\mathcal{B}_i^{t,s'}$ 为仅相差一个样本的批量相邻数据集, 则两者对应的批量平均梯度最多相差一个样本的梯度替换项, 则有

$$\begin{aligned}\|g(w_i^{t,s}, \mathcal{B}_i^{t,s}) - g(w_i^{t,s}, \mathcal{B}_i^{t,s'})\|_2 &= \frac{1}{B}\|\nabla\mathcal{L}(w_i^{t,s}, x_j) - \nabla\mathcal{L}(w_i^{t,s}, x_j')\|_2 \\ &\leqslant \frac{1}{B}(\|\nabla\mathcal{L}(w_i^{t,s}, x_j)\|_2 + \|\nabla\mathcal{L}(w_i^{t,s}, x_j')\|_2) \leqslant \frac{2G}{B},\end{aligned} \tag{16}$$

其中 $g_i^{t,s}$ 是客户端 i 在第 t 轮第 s 次本地更新时的随机梯度, x_j, x_j' 表示批量数据集 $\mathcal{B}_i^{t,s}$ 与 $\mathcal{B}_i^{t,s'}$ 中的第 j 个不同的数据样本, B 是本地训练批量数据集 $\mathcal{B}_i^{t,s}$ 的大小.

定理2　算法 1 满足 $(\alpha, \frac{\tau T\alpha}{2\sigma^2})$-RDP.

证明　每个客户端在本地训练时, 向平均梯度添加高斯噪声 $\mathcal{N}(0, \Delta^2\sigma^2 I_d)$. 根据文献 [33] 可知, 此过程在 Rényi 阶数 $\alpha > 1$ 下满足 $(\alpha, \frac{\alpha}{2\sigma^2})$-RDP. 根据算法 1, 联邦学习一共训练 T 轮次, 每一轮次

各个客户端本地共进行 τ 训练, 因此根据 RDP 的顺序组合定理 [33] 可知, 算法 1 满足 $(\alpha, \frac{\tau T\alpha}{2\sigma^2})$-RDP.

定理3 (RDP 向 DP 的转换)　根据文献 [33], 如果一个机制满足 (α, ρ)-RDP, 则对于任意 $\delta \in (0,1)$, 其也满足 (ϵ, δ)-DP, 其中 $\epsilon = \rho + \frac{\log(1/\delta)}{\alpha-1}$. 因此, 算法 1 满足 $(\frac{\tau T\alpha}{2\sigma^2} + \frac{\log(1/\delta)}{\alpha-1}, \delta)$-DP.

定理4　基于本文提出的联邦遗忘算法 FedUR 得到的遗忘模型 w_u 与根据联邦学习算法重训练得到的模型 $\tilde{w}_t$ 在统计意义下是不可区分的.

证明　用 $\mathcal{M}_U$ 表示联邦遗忘算法, 用 $\mathcal{M}_L$ 表示联邦学习算法. 根据算法 1, 每轮客户端 c_i 上传的更新 u_i^t 均添加了高斯噪声, 保证每轮上传满足 $(\alpha, \frac{\alpha}{2\sigma^2})$-RDP. 根据算法 2, 遗忘过程通过从最终模型 w_T 中减去客户端 c_k 的加权更新残差 $\sum_t \lambda_k^t \theta_k^t$ 实现. 假设要遗忘的客户端为 c_k, 则剩余客户端为 $C' = C \setminus \{c_k\}$, 故重训练模型 $\tilde{w}_t$ 可表示为 $\tilde{w}_t = \mathcal{M}_L(C \setminus \{c_k\}, (\epsilon, \delta))$, 其中 (ϵ, δ) 为隐私参数. 类似地, 遗忘模型 w_u 可表示为 $w_u = \mathcal{M}_U(w_t, \theta^t, (\epsilon, \delta))$, 其中 w_t 和 θ^t 表示模型参数和更新残差. 因此, 根据 RDP 和 DP 可知,

$$\Pr[\mathcal{M}_U(w_t, \theta^t, (\epsilon, \delta)) \in O] \leqslant \mathrm{e}^{\epsilon} \Pr[\mathcal{M}_L(C \setminus \{c_k\}, (\epsilon, \delta)) \in O] + \delta, \tag{17}$$

$$\Pr[\mathcal{M}_L(C \setminus \{c_k\}, (\epsilon, \delta)) \in O] \leqslant \mathrm{e}^{\epsilon} \Pr[\mathcal{M}_U(w_t, \theta^t, (\epsilon, \delta)) \in O] + \delta, \tag{18}$$

因此, 遗忘模型 w_u 与重训练模型 $\tilde{w}_t$ 在 (ϵ, δ)- 差分隐私下不可区分.

防护的攻击类型范围. 与现有的大部分针对联邦遗忘的研究工作一样 [14, 16, 23], 本文提出的 FedUR 算法主要针对成员推理攻击 (membership inference attack, MIA) 和后门攻击 (backdoor attack, BA) 这两种攻击类型. 在后续工作中, 将进一步考虑模型反演攻击、属性推理攻击等攻击手段, 研究和设计防御模型反演攻击和属性推理攻击的联邦遗忘算法.

遗忘粒度. 本文提出的 FedUR 机制主要面向客户端联邦遗忘学习, 即遗忘目标为整个客户端的所有数据及其影响. 首先, FedUR 完全适用于多客户端遗忘场景, 只需要从全局模型中减去多个客户端的历史更新残差的加权和即可. 此外, 由于 FedUR 本质上是通过从全局模型中移除历史更新残差的加权和来消除目标客户端的贡献, 故 FedUR 理论上能够应用到样本级联邦遗忘学习和类别级联邦遗忘学习场景中. 以样本级联邦遗忘学习为例, 假设在训练批量 $\mathcal{B}_i$ 中, 要遗忘的样本为 $\mathcal{B}_f$, 则保留样本为 $\mathcal{B}_i^r = \mathcal{B}_i \setminus \mathcal{B}_f$. 用 B 和 B_f 分别表示 $\mathcal{B}_i$ 和 $\mathcal{B}_f$ 的大小, 则遗忘前联邦学习的平均梯度为 $\bar{g}_i = \frac{1}{B}\sum_{x\in\mathcal{B}_i} \bar{g}_i(x)$, 遗忘请求发出后在保留数据集上的平均梯度为 $\bar{g}_i^r = \frac{1}{B-B_f}\sum_{x\in\mathcal{B}_i^r} \bar{g}_i(x)$. 基于此即可计算目标遗忘样本对应的更新残差. 然而这种简单的应用方式需要存储每条样本对应的梯度信息, 会导致巨大的存储开销. 因此, 后续研究将关注针对样本级联邦遗忘学习和类别级联邦遗忘学习的特定遗忘算法, 进而有效平衡遗忘效果和存储开销.

5　实验总结

5.1　实验设置

数据集. 本文在 3 个数据集上进行实验, 分别是 MNIST [36], Fashion-MNIST [37], CIFAR-10 [38]. MNIST 数据集由 7 万张 28×28 的灰度图像组成, 分为 10 类, 每类 7000 张图像. Fashion-MNIST 数据集与 MNIST 数据大小相同. CIFAR-10 数据集由 6 万张 32×32 的彩色图像组成, 分为 10 类, 每类有 6000 张图像. 后续实验中本文使用 FMNIST 代指 Fashion-MNIST 数据集.

模型和默认参数. 本文实验采用卷积神经网络 (CNN) 模型对这些数据集执行图像分类任务, 客户端数量设置为 10. 实验中根据 Dirichlet 分布对训练数据进行划分, 模拟联邦学习中非独立同分布数据. 实验中本地训练周期 τ 设置为 5, 联邦学习执行轮次设置为 50 轮, 50 轮之后开始执行联邦遗忘过程. 各个客户端采用随机梯度下降算法来最小化损失函数, 学习率默认为 0.001.

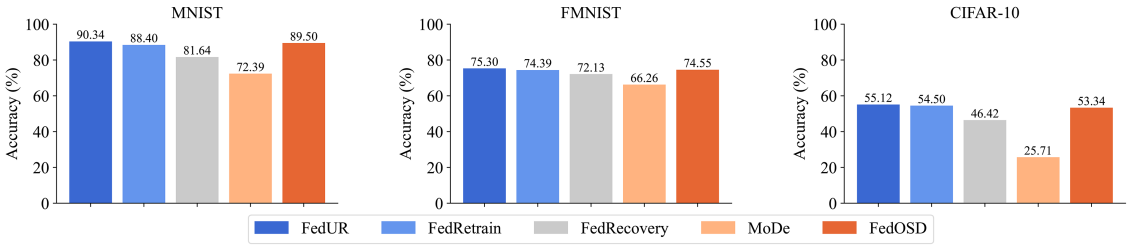

图 2　(网络版彩图) 不同联邦遗忘机制下遗忘模型的准确性比较.

Figure 2 (Color online) Comparisons of the accuracy of unlearned models under different federated unlearning mechanisms.

评估指标. 本文从 4 个方面进行评估. (1) 模型效用: 评估遗忘模型的整体准确性表现 (accuracy), 以及其在保留客户端上的准确性表现 (R-Acc). (2) 遗忘效果: 采用成员推理成功率 (member inference success rate, MISR) 评估成员推理攻击 (membership inference attack), 衡量模型对遗忘客户端数据的遗忘效果, 以攻击成功率 (attack success rate, ASR) 评估后门攻击 (backdoor attack), 反映模型后门遗忘的性能. (3) 遗忘效率: 利用遗忘轮次和运行时间度量各方法完成遗忘请求的实际效率. (4) 存储和通信开销: 分析和评估各方法的存储和通信复杂度.

对比方法. 本文选取以下 4 种具有代表性的联邦遗忘算法作为对比. FedRetrain 通过重训练来实现准确遗忘, 尽管开销较大, 但可视为基线方法. FedRecovery [23] 通过移除全局模型中梯度残差以达到遗忘目标. MoDe [14] 结合动量衰减与记忆引导策略逐步实现知识擦除. FedOSD [16] 利用正交最速下降策略实现遗忘, 并引入后训练阶段恢复模型效用.

5.2　实验评估

5.2.1　模型效用评估

本文首先在 3 个典型的图像数据集 (MNIST, FMNIST 和 CIFAR-10) 上比较了各方法在执行遗忘操作后的模型准确性. 实验中, 采用 Dirichlet 分布对数据进行划分以模拟强异质性 (Non-IID) 场景, 其浓度参数 (concentration parameter) 设为 0.5.

图 2 展示了各方法在不同数据集上执行遗忘操作后的全局模型分类准确率表现. 从结果可以看出, FedUR 在 3 个数据集上均展现出优越的模型性能. 对于 MNIST 和 FMNIST 这两个结构相对简单、样本分布清晰的数据集, FedUR 的准确率与基线方法 FedRetrain 几乎持平, 甚至在某些情形下略有超越, 表明其在有效实现遗忘的同时, 能够最大程度地保留原有模型的能力. 而在更复杂的 CIFAR-10 数据集上, FedUR 依然取得了优于其他遗忘方法的准确率, 并略微超过了重训练策略 FedRetrain, 显示出所提方法在复杂任务下的稳定性与优势. 具体而言, FedUR 在 CIFAR-10 上达到了 55.12% 的准确率, 优于 FedRetrain 的 54.50% 和 FedRecovery [23] 的 46.42%. 在 MNIST 和 FMNIST 数据集上, FedUR 分别实现了 90.34% 和 75.30% 的准确率, 均为所有方法中的最高水平. 这些结果表明, FedUR 在实现有效知识遗忘的同时, 能够很好地维持全局模型的整体性能.

此外值得指出的是, FedUR 和 FedRecovery [23] 均为具有隐私保护机制的联邦遗忘方案, 其中 FedUR 通过在本地训练阶段对梯度添加高斯噪声实现隐私保护. 因此, FedUR 在保障隐私安全的同时, 仍能取得优异的模型性能, 进一步体现了其综合优势. 在后续实验中, 本文将进一步验证 FedUR 的遗忘效果, 以确保模型在保持较高分类准确率的同时, 能够实现显著的遗忘能力.

FedUR 在本地训练中通过在梯度上添加高斯噪声来实现差分隐私保护. 因此, 本文进一步评估不同隐私预算 ϵ (即不同差分隐私保护程度) 下的模型准确性. 实验中主要关注隐私预算对遗忘模型分类准确率的影响, 因此采用 Dirichlet 分布划分数据, 浓度参数设置为 1.0, 以构建数据异质性较弱 (即近似 IID) 的实验场景. 设置隐私预算 ϵ 的取值范围为 $[1.0, 8.0]$, 且设置 $\delta = 10^{-5}$. 由于 MoDe [14] 和

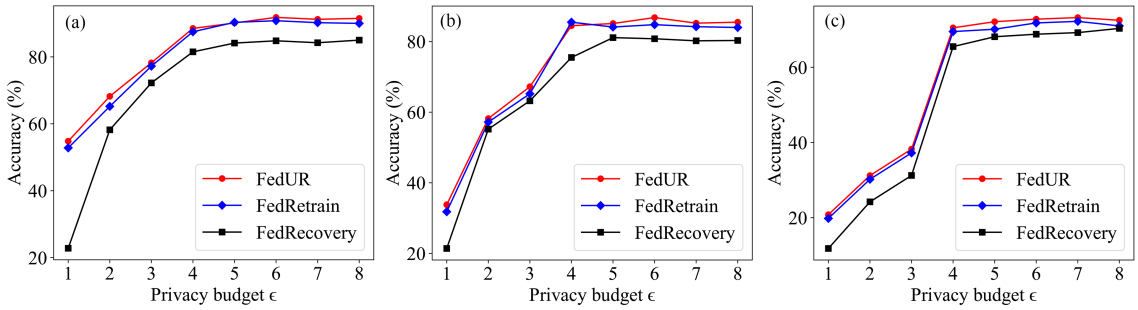

图 3　(网络版彩图) 不同隐私预算下联邦遗忘机制的准确性评估. (a) MNIST; (b) FMNIST; (c) CIFAR-10.

Figure 3　(Color online) The accuracy evaluation of federated unlearning mechanisms when privacy budget ϵ changes. (a) MNIST; (b) FMNIST; (c) CIFAR-10.

FedOSD [16] 不涉及隐私保护过程, 因此这里只测试本文提出的 FedUR、重训练方法 FedRetrain 以及 FedRecovery [23] 3 个机制在不同隐私预算下的模型准确率.

图 3 展示了不同隐私预算下不同遗忘机制的准确性变化情况. 如图 3(a) 所示, 在 MNIST 数据集上, 随着隐私预算从 1.0 增加至 8.0, FedUR 的遗忘模型准确率从约 56% 提升至 91%. 此外, FedUR 与重训练方法 FedRetrain 在遗忘模型准确性上的差异非常小, 表明 FedUR 在模型准确性上具有极大优势. 而相比之下, FedRecovery 机制在不同隐私预算下的准确率始终低于 FedUR. 这一结果归因于 FedUR 在联邦学习阶段的结构性优化, 不仅提升了模型训练效率和收敛性能, 同时通过更新残差加权机制, 在实现有效遗忘的同时能够维持模型性能变化不大. 此外, 图 3(b) 和 (c) 分别展示了在 FMNIST 和 CIFAR-10 数据集上的实验结果. 与 MNIST 结果一致, FedUR 在不同隐私预算下始终优于 FedRecovery 机制, 再次验证了本文提出的 FedUR 机制能够在联邦遗忘中维持遗忘模型的准确性, 具有较好的实用性.

5.2.2　遗忘效果评估

本小节接着评估所提出机制 FedUR 的遗忘有效性. 联邦遗忘的目标是使遗忘后的模型在彻底遗忘目标客户端数据的同时, 仍能在其余客户端数据上保持较高的测试准确率. 为此, 本小节分别根据 ASR, MISR 和 R-Acc 这 3 个指标来衡量不同机制的遗忘效果. 具体通过计算遗忘模型在目标客户端上的 ASR 与 MISR 来评估遗忘程度, 其中 ASR 越低表示遗忘效果越好, MISR 越趋近 50% 表示攻击者难以判断样本来源, 则遗忘效果越好. R-Acc 用于评估遗忘模型在剩余客户端的平均准确率, R-Acc 越高表示遗忘模型在遗忘后对剩余客户端数据集上的效果越好. 本小节实验在高度异质的数据分布设置下进行, 即 Dirichlet 分布浓度参数设置为 0.5.

表 1 展示了不同方法在 MNIST, FMNIST 和 CIFAR-10 3 个数据集上的对比结果. 从表中可观察到, FedUR 在非独立同分布场景下能够达到较好的遗忘效果. 在 MNIST 数据集上, FedUR 在遗忘目标客户端后, 其攻击成功率 (ASR) 基本降为 0, 成员推理成功率 (MISR) 趋近于 50%, 且在有效遗忘的同时, 在剩余客户端数据上的准确率最高, 体现了其良好的实用性. 对于更复杂的数据集 FMNIST 和 CIFAR-10, FedUR 依然展现出有效的遗忘能力. 尽管其遗忘效果略低于重训练方法 FedRetrain 和基于梯度上升的 FedOSD, 但需指出的是, 后两者通常需要消耗大量训练轮次以获取更优的遗忘表现, 而本文提出的 FedUR 能够快速遗忘. 因此, 在综合考虑遗忘效率的前提下, FedUR 在遗忘效果与时间开销之间达成了更优的平衡. 此外, MoDe 也表现出较好的遗忘效果, 但其遗忘后的模型在剩余客户端上的准确率相对较低, 尤其是在复杂的 CIFAR-10 数据集上更为明显. 这是由于其所采用的动量退化策略在遗忘目标客户端数据的同时, 也对剩余客户端的数据造成了影响. 值得注意的是, FedOSD 与 MoDe 机制在完成模型遗忘后均需进行性能恢复训练以提升模型效用, 因此计算开销较大. 为保证实验的公平性, 本实验中所有 5 种方案在遗忘后均未继续训练.

表 1　不同联邦遗忘机制的遗忘效果评估.

Table 1　The unlearning effectiveness evaluation of different federated unlearning mechanisms.

Mechanism	MNIST			FMNIST			CIFAR-10		
	ASR	MISR	R-Acc	ASR	MISR	R-Acc	ASR	MISR	R-Acc
FedRetrain	0.003	0.510	0.885	0.010	0.521	0.723	0.002	0.502	0.543
FedRecovery [23]	0.042	0.480	0.818	0.379	0.582	0.757	0.160	0.510	0.452
MoDe [14]	0.039	0.531	0.723	0.003	0.507	0.667	0.145	0.542	0.256
FedOSD [16]	0.002	0.509	0.904	0.006	0.512	0.727	0.024	0.513	0.519
FedUR	0.014	0.524	0.907	0.044	0.543	0.762	0.042	0.521	0.547

表 2　消融实验：基于余弦相似度的权重策略对联邦遗忘效果的影响.

Table 2　Ablation study on the impact of cosine similarity-based weighting on federated unlearning.

Mechanism	MNIST			FMNIST			CIFAR-10		
	ASR	MISR	R-Acc	ASR	MISR	R-Acc	ASR	MISR	R-Acc
FedUR (w/o cos)	0.105	0.557	0.887	0.075	0.586	0.742	0.124	0.543	0.518
FedUR (with cos)	0.014	0.524	0.907	0.044	0.543	0.762	0.042	0.521	0.547

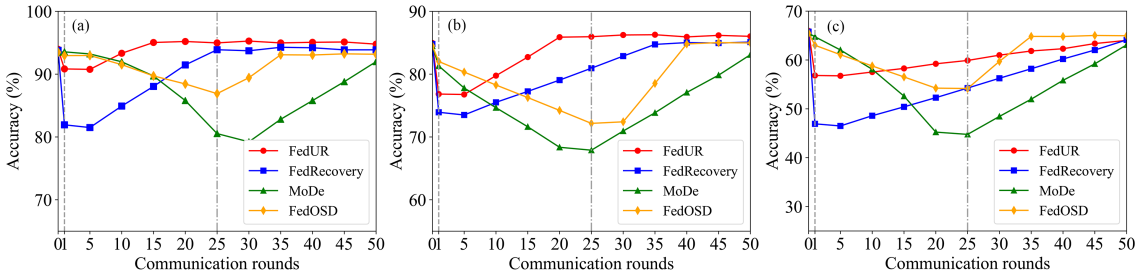

图 4　(网络版彩图) 遗忘和后训练阶段模型准确性随轮次的变化. (a) MNIST; (b) FMNIST; (c) CIFAR-10.

Figure 4　(Color online) The change in model accuracy over rounds during the unlearning and post-training phases. (a) MNIST; (b) FMNIST; (c) CIFAR-10.

本文进一步设计了消融实验来评估基于余弦相似度的权重方案对最终遗忘效果的影响. 具体地, 在实验中将基于余弦相似度的权重方案替换为均值权重, 即削弱了对各轮更新贡献度的差异化考量. 实验结果如表 2 所示, 其中 FedUR (w/o cos) 和 FedUR (with cos) 分别表示联邦遗忘机制不集成和集成基于余弦相似度的权重策略. 可以看到, FedUR (with cos) 的整体遗忘效果均优于 FedUR (w/o cos), 具体表现为 MISR 与 ASR 明显下降, 且 R-Acc 提高. 因此, 实验结果验证了基于余弦相似度的权重方案能够有效增强遗忘的可靠性, 通过余弦相似度衡量客户端与全局更新之间的方向一致性, 有助于更准确地判断需要遗忘的历史更新, 从而提升联邦遗忘的整体效果.

5.2.3　遗忘效率评估

本小节首先根据完成遗忘请求所需要的训练轮次来评估不同遗忘机制遗忘效率. 实验中设置遗忘请求发出后一共执行 50 轮, 其中 MoDe 和 FedOSD 的遗忘过程和恢复过程各设置为 25 轮. 图 4 展示了遗忘请求发出后, 不同遗忘机制在遗忘阶段和后训练阶段的模型准确率随训练轮次的变化情况. 可以看到, FedUR 机制和 FedRecovery 机制在遗忘请求发出后仅用单个轮次就完成了联邦遗忘过程 (如图 4 中虚线所示), 此时模型准确性最低. 这是因为 FedUR 和 FedRecovery 都是采用移除遗忘客户端的历史贡献的方式实现联邦遗忘, 故遗忘阶段仅包括单个轮次, 后续过程均可以用来遗忘后训练以提升模型性能. 相较之下, MoDe 和 FedOSD 通过训练过程中的模型调整来实现遗忘, 故图 4 显示 MoDe

表 3　不同联邦遗忘机制完成遗忘的运行时间 (h).

Table 3　Running time of different federated unlearning mechanisms to complete unlearning (h).

Dataset	Running time				
	FedRetrain	FedRecovery [23]	MoDe [14]	FedOSD [16]	FedUR
MNIST	1.5	< 0.01	0.85	0.95	< 0.02
FMNIST	2.5	< 0.01	1.3	1.4	< 0.02
CIFAR-10	5.0	< 0.01	3.0	3.35	< 0.02

和 FedOSD 在遗忘请求发出后需执行 25 轮次遗忘才能完成遗忘 (如图 4 中虚点线所示), 与 FedUR 和 FedRecovery 相比具有较低的遗忘效率. 此外, 尽管 FedUR 和 FedRecovery 都能实现快速遗忘, 但对比可知, 本文提出的 FedUR 机制在完成快速遗忘的同时, 具有比 FedRecovery 更高的模型准确性. 因此, 从整体上来看, 本文提出的 FedUR 机制不仅具有最高的准确性, 且遗忘效率较高, 实现了模型效用和遗忘效率的良好平衡.

本小节进一步对各遗忘机制完成遗忘所需的时间进行对比分析, 如表 3 所示. 从表中可以看出, FedUR 与 FedRecovery 的遗忘过程所需时间远低于其余方法, 主要归因于两者所采用的非训练型遗忘策略. FedUR 的时间开销主要来源于更新残差与其权重的计算, 以及中间缓存的读写操作. 尽管其整体时间与 FedRecovery 相近, 但由于 FedUR 引入了基于余弦相似度的权重计算机制, 因而稍微增加了计算时间. 相比之下, 基于重训练的 FedRetrain 方法需完整重复一次联邦训练过程, 所需时间远高于 FedUR, 因此在实际应用中代价较高. MoDe 和 FedOSD 两个机制需在客户端执行多轮本地训练以逐步调整模型参数, 同时在遗忘完成后还需进行额外的恢复性训练, 因而具有较高的遗忘时间. 综上可知, FedUR 在保证遗忘效果的同时, 避免了依赖模型训练与后恢复的复杂流程, 显著提升了整体的遗忘效率, 具有更好的实际应用价值.

5.2.4　存储和通信开销

本小节分析和对比了联邦遗忘机制 FedUR 与已有方法的存储和通信开销. 为便于比较, 本小节考虑各个遗忘机制的整体存储和通信开销上限, 即包括联邦学习阶段和联邦遗忘阶段. 假设有 n 个客户端, 联邦学习阶段共进行 T_l 轮通信, 遗忘请求发出后联邦遗忘阶段共进行 T_u 轮通信, 假设模型上传和下发时的参数大小上界为 M.

表 4 展示了不同遗忘机制的存储和通信开销对比情况. 对于 FedRetrain, 由于其直接对模型进行重训练, 因此不产生额外的存储开销. 考虑到通信开销包括模型的上传与下发, 故其总通信开销为 $\mathcal{O}(4nMT_l)$. MoDe [14] 和 FedOSD [16] 不依赖于历史模型更新, 因此均不需要额外的存储开销. 但 MoDe 和 FedOSD 需要在遗忘请求发出后进行 T_u 轮训练完成遗忘, 因此通信开销较大. 具体地, MoDe 机制在联邦学习阶段包含 T_l 轮通信, 遗忘阶段包含知识擦除和记忆引导两个阶段 (一共进行 T_u 轮通信), 因此总的通信开销为 $\mathcal{O}\left(2nM(T_l+T_u)\right)$. 与 MoDe 类似, FedOSD 机制也需要 T_u 轮的遗忘阶段用于完成遗忘和后训练, 因此总通信开销为 $\mathcal{O}\left(2nM(T_l+T_u)\right)$. 与上述方法不同, FedRecovery [23] 需要保留每轮客户端上传的梯度, 但遗忘阶段只需单轮即可完成, 因此存储开销为 $\mathcal{O}(nMT_l)$, 通信开销为 $\mathcal{O}(2nMT_l)$. 相较之下, 本文提出的 FedUR 机制采用了周期聚合策略, 即每隔 τ 轮本地训练, 客户端才上传一次更新, 故联邦学习阶段实际通信轮次为 $\frac{T_l}{\tau}$ 轮. 并且 FedUR 机制本质上是通过移除目标遗忘客户端的历史更新残差来实现联邦遗忘, 故联邦遗忘阶段只需单轮训练即可完成遗忘. 因此, FedUR 的存储开销为 $\mathcal{O}\left(nMT_l/\tau\right)$, 通信开销为 $\mathcal{O}\left(2nMT_l/\tau\right)$.

综上所述, 根据表 4 可知, 本文提出的 FedUR 的通信开销远低于其他依赖额外训练过程的方法 (如 MoDe 和 FedOSD). 由于 FedUR 和 FedRecovery 都是采用移除遗忘客户端历史贡献的方法实现遗忘, 虽然均需要额外存储开销, 但在通信代价方面占据优势. 本文提出的 FedUR 引入周期聚合策略和

表 4　不同联邦遗忘机制的存储与通信成本对比.

Table 4　Comparisons of storage and communication cost of different federated unlearning mechanisms.

Mechanism	Storage cost	Communication cost
FedRetrain	–	$\mathcal{O}(4nMT_l)$
MoDe [14]	–	$\mathcal{O}(2nM(T_l+T_u))$
FedOSD [16]	–	$\mathcal{O}(2nM(T_l+T_u))$
FedRecovery [23]	$\mathcal{O}(nMT_l)$	$\mathcal{O}(2nMT_l)$
FedUR (ours)	$\mathcal{O}\left(nMT_l/\tau\right)$	$\mathcal{O}\left(2nMT_l/\tau\right)$

基于更新残差的遗忘学习策略, 通信和存储成本显著降低为 FedRecovery 的 $1/\tau$, 因而在通信开销和存储开销方面都要小于 FedRecovery 机制. 因此, 本文提出的 FedUR 机制在存储和通信开销方面都更具优势, 能够高效完成联邦遗忘学习, 实用性较强.

讨论. 在本文方法基础上, 可进一步采用压缩技术 [39] (如压缩感知、投影等) 和选择性存储 [40] 等策略, 有效降低存储和通信压力, 以适配边缘设备资源受限的环境. 例如, 联邦学习中模型更新 (如梯度) 天然具有稀疏性, 客户端可对模型更新进行压缩感知处理, 用测量矩阵 Φ 对 d- 维更新向量 u 进行线性采样, 生成低维测量值 $\hat{u}=\Phi u$ (维度 $\hat{d}\ll d$). 服务器端可根据 Φ 和 $\hat{u}$ 利用 L_1 范数最小化重构原始模型更新 u. 此外, 还可以结合本地训练流程, 基于重要性选择来动态保留 "关键信息"、丢弃 "冗余信息", 避免保存所有模型更新参数, 从而降低存储成本.

6　结论

本文提出了一种基于更新残差的差分隐私联邦遗忘机制 FedUR, 是一个兼顾隐私保护、模型效用与遗忘效率的联邦遗忘学习框架. 该方法通过引入重要性采样与周期加权聚合机制, 有效提升了联邦遗忘效率与全局模型性能. FedUR 通过在本地训练阶段注入高斯噪声, 增强了算法的隐私保护能力, 并确保遗忘模型与重新训练模型之间在统计意义上不可区分. 同时, 本文设计了基于更新残差的联邦遗忘方法, 通过移除客户端历史加权更新残差, 实现了无需重训练的高效遗忘. 实验结果表明, FedUR 在多个评估指标上均表现优秀, 验证了其在联邦遗忘场景中的有效性与实用价值.

参考文献

1 Liu Z, Guo J, Yang W, et al. Dynamic user clustering for efficient and privacy-preserving federated learning. In: Proceedings of IEEE Transactions on Dependable and Secure Computing, 2024

2 Yu J X, Shi R H. DDoS attack detection in the Internet of Vehicles based on reinforced federated learning. Sci Sin Inform, 2025, 55: 1221–1238 [于峻骁, 石润华. 基于强化联邦学习的车联网 DDoS 攻击检测. 中国科学: 信息科学, 2025, 55: 1221–1238]

3 Wang K Q, Hong R Q, Mao Y L, et al. Secure solution for decentralized federated learning with blockchain. Sci Sin Inform, 2024, 54: 316–334 [王恺祺, 洪睿琦, 毛云龙, 等. 基于区块链构建安全去中心化的联邦学习方案. 中国科学: 信息科学, 2024, 54: 316–334]

4 Xu R Z, Tong Y M, Dai L P. Research on federated learning adaptive differential privacy method based on heterogeneous data. Netinfo Security, 2025, 25: 63–77 [徐茹枝, 仝雨蒙, 戴理朋. 基于异构数据的联邦学习自适应差分隐私方法研究. 信息网络安全, 2025, 25: 63–77]

5 Protection F D. General data protection regulation (GDPR). Intersoft Consult, 2018. https://gdpr-info.eu/

6 Harding E L, Vanto J J, Clark R, et al. Understanding the scope and impact of the California Consumer Privacy Act of 2018. J Data Protection Privacy, 2019, 2: 234–253

7 Romandini N, Mora A, Mazzocca C, et al. Federated unlearning: a survey on methods, design guidelines, and evaluation metrics. In: Proceedings of the IEEE Transactions on Neural Networks and Learning Systems, 2024. 1–21

8 De K, Pedersen M. Impact of colour on robustness of deep neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 21–30

9 Liu G, Ma X, Yang Y, et al. Federaser: enabling efficient client-level data removal from federated learning models. In: Proceedings of the IEEE/ACM 29th International Symposium on Quality of Service (IWQOS), 2021. 1–10

10 Cao X, Jia J, Zhang Z, et al. Fedrecover: recovering from poisoning attacks in federated learning using historical information. In: Proceedings of the IEEE Symposium on Security and Privacy (SP), 2023. 1366–1383

11 Liu Y, Xu L, Yuan X, et al. The right to be forgotten in federated learning: an efficient realization with rapid retraining. In: Proceedings of the IEEE INFOCOM 2022-IEEE Conference on Computer Communications, 2022. 1749–1758

12 Tao Y, Wang C L, Pan M, et al. Communication efficient and provable federated unlearning. Proc VLDB Endow, 2024, 17: 1119–1131

13 Wu C, Zhu S, Mitra P. Federated unlearning with knowledge distillation. ArXiv:2201.09441

14 Zhao Y, Wang P, Qi H, et al. Federated unlearning with momentum degradation. IEEE Internet Things J, 2023, 11: 8860–8870

15 Li G, Shen L, Sun Y, et al. Subspace based federated unlearning. ArXiv:2302.12448

16 Pan Z, Wang Z, Li C, et al. Federated unlearning with gradient descent and conflict mitigation. AAAI, 2025, 39: 19804–19812

17 Chen M, Zhang Z, Wang T, et al. When machine unlearning jeopardizes privacy. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021. 896–911

18 Wang F, Li B, Li B. Federated unlearning and its privacy threats. IEEE Network, 2023, 38: 294–300

19 Tang X Y, Wang W, Weng Y, et al. A survey on privacy security and computation efficiency in federated unlearning. Chinese J Comput, 2025, 48: 1–25 [唐湘云, 王伟, 翁彧, 等. 联邦遗忘学习隐私安全与算法效率研究综述. 计算机学报, 2025, 48: 1–25]

20 Dwork C, Roth A. The algorithmic foundations of differential privacy. Found Trends Theor Comput Sci, 2014, 9: 211–407

21 Chourasia R, Shah N. Forget unlearning: towards true data-deletion in machine learning. In: Proceedings of the International Conference on Machine Learning, 2023. 6028–6073

22 Liu Z, Dou G, Chien E, et al. Breaking the trilemma of privacy, utility, and efficiency via controllable machine unlearning. In: Proceedings of the ACM Web Conference, 2024. 1260–1271

23 Zhang L, Zhu T, Zhang H, et al. FedRecovery: differentially private machine unlearning for federated learning frameworks. IEEE Trans Inform Forensic Secur, 2023, 18: 4732–4746

24 Jiang Y, Tong X, Liu Z, et al. Efficient federated unlearning with adaptive differential privacy preservation. In: Proceedings of the IEEE International Conference on Big Data (BigData), 2024. 7822–7831

25 Bourtoule L, Chandrasekaran V, Choquette-Choo C A, et al. Machine unlearning. In: Proceedings of the IEEE Symposium on Security and Privacy (SP), 2021. 141–159

26 Wang J, Guo S, Xie X, et al. Federated unlearning via class-discriminative pruning. In: Proceedings of the ACM Web Conference, 2022. 622–632

27 Guo C, Goldstein T, Hannun A, et al. Certified data removal from machine learning models. In: Proceedings of the 37th International Conference on Machine Learning, 2020. 3832–3842

28 Golatkar A, Achille A, Soatto S. Eternal sunshine of the spotless net: selective forgetting in deep networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 9304–9312

29 Sekhari A, Acharya J, Kamath G, et al. Remember what you want to forget: algorithms for machine unlearning. Adv Neural Inform Process Syst, 2021, 34: 18075–18086

30 Ginart A, Guan M, Valiant G, et al. Making AI forget you: data deletion in machine learning. Adv Neural Inform Process Syst, 2019, 32: 3513–3526

31 Yang Q, Liu Y, Chen T, et al. Federated machine learning: concept and applications. ACM Trans Intell Syst Tech, 2019, 10: 1–19

32 McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the Artificial Intelligence and Statistics, 2017. 1273–1282

33 Mironov I. Rényi differential privacy. In: Proceedings of the IEEE 30th Computer Security Foundations Symposium (CSF), 2017. 263–275

34 Jiang D, Sun S, Yu Y. Functional Rényi differential privacy for generative modeling. Adv Neural Inform Process Syst, 2023, 36: 14797–14817

35 Abadi M, Chu A, Goodfellow I, et al. Deep learning with differential privacy. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2016. 308–318

36 LeCun Y, Cortes C, Burges C J. MNIST handwritten digit database. 2010. http://yann.lecun.com/exdb/mnist

37 Xiao H, Rasul K, Vollgraf R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms.

ArXiv:1708.07747
38 Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. University of Toronto, Toronto, Technical Report TR-2009, 2009
39 Chen S, Miao Y, Li X, et al. Compressed-sensing-based practical and efficient privacy-preserving federated learning. IEEE Internet Things J, 2023, 11: 14017–14030
40 Yuan W, Yin H Z, Wu F Z, et al. Federated unlearning for on-device recommendation. In: Proceedings of ACM International Conference on Web Search and Data Mining, 2023. 393–401

Differentially private federated unlearning mechanism based on update residuals

Teng WANG[1], Lindong ZHAI[1], Yong YU[2*], Tengfei YANG[1], Xuefeng ZHANG[1] & Xuebin REN[3]

1. *School of Cyberspace Security, Xi'an University of Posts & Telecommunications, Xi'an 710121, China*
2. *School of Artificial Intelligence and Computer Science, Shaanxi Normal University, Xi'an 710119 China*
3. *School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China*
* Corresponding author. E-mail: yuyong@snnu.edu.cn

Abstract Federated unlearning (FU) enables the removal of specific client data and its influence from an already-trained model, thereby supporting "the right to be forgotten". Although retraining the model from scratch is a straightforward approach, its high computational cost often renders it impractical. Most existing methods achieve FU by either removing the historical contributions or gradually adjusting the model using gradient ascent. However, storing historical gradients or performance-enhancing training faces high storage and communication overhead, resulting in inefficient FU. Additionally, the privacy risks associated with the FU require further investigation. Therefore, this paper proposes FedUR, a differentially private federated unlearning mechanism based on update residuals, which achieves an effective balance among privacy protection, model utility, and unlearning efficiency. Specifically, FedUR enhances privacy by rendering the unlearned model indistinguishable from a retrained model according to the differential privacy paradigm. Moreover, the proposed approach innovatively quantifies the historical impact of unlearning clients on the global model through update residuals, and enables rapid FU by removing all historical weighted update residuals without relying on the model recovery training process, thus markedly reducing storage overhead. Furthermore, FedUR integrates importance sampling and periodic weighted aggregation strategies to mitigate the adverse effects of data heterogeneity on model performance while also lowering storage and communication overhead. Experimental results demonstrate that FedUR maintains robust model utility and high unlearning efficiency while providing strong privacy protection.

Keywords federated learning, differential privacy, federated unlearning, update residual, model utility