软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007665] [CSTR: 32375.14.jos.007665]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

联邦学习中分布式水印与抗攻击方案*

孙友欣[1], 田有亮[1,2]

[1](贵州大学 计算机科学与技术学院 (贵州保密学院), 贵州 贵阳 550025)

[2](贵州大学 大数据与信息工程学院, 贵州 贵阳 550025)

通信作者: 田有亮, E-mail: youliangtian@163.com

摘　要: 联邦学习作为一种分布式机器学习方法, 能够在保护用户隐私和数据安全的同时进行模型训练. 然而, 联邦学习多参与方、模型大范围暴露的特点容易造成模型版权泄露问题. 提出一种具有所有权验证、模型泄露追溯和懒惰客户端检测功能的水印方案, 引入了客户端身份标识生成后门水印机制和动态调整聚合权重 (federated dynamic weight adjustment , FDWA) 算法, 确保每个客户端的水印具有唯一性, 并解决了水印冲突问题, 在显著提高模型保真度和水印触发率的同时具备更好的懒惰客户端检测性能. 实验结果表明, 该方案在提供更完善的保护功能的同时, 能保持模型性能, 显著提高水印触发率, 并有效抵御微调、剪枝、量化和共谋攻击等多种攻击手段, 提高了联邦学习环境的安全性和公平性, 为模型提供有效版权保护.

关键词: 联邦学习; 产权保护; 后门水印; 参数水印; 模型聚合

中图法分类号: TP309

中文引用格式: 孙友欣, 田有亮. 联邦学习中分布式水印与抗攻击方案. 软件学报. http://www.jos.org.cn/1000-9825/7665.htm

英文引用格式: Sun YX, Tian Y-L. Distributed Watermarking and Anti-attack Scheme in Federated Learning. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7665.htm

Distributed Watermarking and Anti-attack Scheme in Federated Learning

SUN You-Xin[1], TIAN You-Liang[1,2]

[1](College of Computer Science and Technology (Guizhou Confidentiality College), Guizhou University, Guiyang 550025, China)

[2](College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China)

Abstract: Federated learning (FL), as a distributed machine learning method, enables model training while protecting user privacy and data security. However, the involvement of multiple parties and the widespread exposure of models in FL can easily lead to copyright leakage. This study proposes a watermarking scheme with ownership verification, model leakage tracing, and lazy client detection. The proposed scheme introduces a client identity-based backdoor watermark generation mechanism and federated dynamic weight adjustment (FDWA) to ensure the uniqueness of each client's watermark and resolve watermark conflicts. Model fidelity and watermark trigger rates are significantly improved, while also achieving better performance in detecting lazy clients. Experimental results show that the scheme provides more comprehensive protection while maintaining model performance, significantly improves watermark trigger rates, and effectively resists various attacks such as fine-tuning, pruning, quantization, and collusion attacks, thus enhancing the security and fairness of the FL environment and providing effective copyright protection for models.

Key words: federated learning (FL); property rights protection; backdoor-based watermark; parameter-based watermark; model aggregation

联邦学习 (federated learning, FL)[1]是一种分布式机器学习方法, 旨在在不集中数据的情况下训练模型. 它最初由 Google 于 2016 年提出, 以解决隐私和数据安全问题. 联邦学习的核心思想是将模型训练分布在多个客户端 (如

* 基金项目: 国家重点研发计划 (2025YFB3109800); 国家自然科学基金 (62272123); 贵州省高层次创新型人才项目 (黔科合平台人才[2020]6008); 贵州省科技计划 (黔科合平台人才 [2020]5017, 黔科合支撑 [2022] 一般 065, 黔科合支撑 [2023] 一般 371); 贵州省基础研究计划面上项目 (黔科合基础 MS[2026]208); 贵州省科技创新平台科研项目 (黔科合平台 CXPTXM[2025]024)
收稿时间: 2025-03-25; 修改时间: 2026-01-20; 采用时间: 2026-03-09; jos 在线出版时间: 2026-05-27

手机、物联网设备等) 上, 而不是将所有数据集中到一个服务器上进行训练. 这种方法有助于保护用户的隐私和数据安全.

然而, 联邦学习模型面临复杂的版权问题和需求. 一个模型的训练伴随着大量开销, 一方面, 训练模型需要掌握专业知识和大量数据[2], 另一方面, 为了提高算力, 训练过程需要使用专用硬件并消耗大量电力[3]. 因此, 模型所有者希望保护训练好的模型不被非法复制、重新分发或滥用. 如何保护好模型的知识产权, 是模型训练过程的关键问题. 在联邦学习中, 每个参与方都贡献了数据和算力, 其本质是对全局模型的“联合投资”. 因此, 参与者们可能会对模型的部分贡献有版权要求, 体现自身的版权贡献. 现有方案往往缺乏对参与者的区分能力, 仅将全体参与者视为一个整体来处理.

由于训练过程中的中间模型已经具备一定的实用价值, 这导致个别客户端可能为谋取私利而将联邦学习模型进行私自分发或出售给第三方, 这些恶意成员被称为模型泄露者. 这样的行为严重侵犯了整个联邦学习集团中其他成员的合法版权. 为了进一步逃避追踪, 这些恶意客户端还可能联合起来采取共谋的方式构建出不包含有效水印的伪造模型.

在联邦学习环境中, 模型水印是一种关键的版权保护技术, 旨在不影响模型性能的前提下, 将隐秘的所有权标识嵌入模型中. 由于 FL 的去中心化特性, 模型的训练过程分布在多个客户端, 各个参与方都在本地更新模型, 并将更新后的参数上传至服务器进行聚合. 在这一过程中, 模型的开放性和多方协作的特点, 使得模型的知识产权保护变得尤为重要, 而水印技术能够有效提供所有权验证、模型泄露追溯等功能, 以支持侵权行为发生后, 版权所有者进行维权操作.

然而, 在分布式环境下, 水印的嵌入方式面临新的挑战, 最突出的问题之一便是水印冲突. 由于不同客户端各自嵌入水印, 而全局模型由多个本地模型聚合而成, 这可以看作是对非独立同分布数据集进行训练. 不同客户端的水印可能会在聚合过程中相互干扰, 导致水印信息被削弱甚至丢失. 这种冲突不仅会降低水印的触发率, 还可能影响模型的整体性能和鲁棒性. 此外, 若部分水印在全局模型中占据主导地位, 可能会导致某些客户端的水印被覆盖, 使得所有权验证准确性降低. 因此, 如何在联邦学习环境下有效地管理和协调水印, 以减少冲突、提高水印的有效性, 成为模型水印设计中的关键问题.

在联邦学习过程中, 可能出现一些懒惰客户端[4,5]. 这些客户端不具有数据也不提供算力, 以造假的方式提供本地训练结果 (搭便车攻击 (Freerider)). 它们的目标是免费获取全局模型. 这种行为虽然不破坏整个训练流程和训练结果, 但侵害了诚实参与方的正当利益, 应当被视为对版权的非法占有. 同时, 搭便车行为削弱了贡献者的权益, 因为它们的努力被未贡献者无偿利用, 这会严重影响联邦学习成员的积极性. 这些懒惰客户端不应该获得最终的模型版权.

联邦学习模型水印和懒惰客户端检测都是为了增强系统的安全性和公平性, 保障诚实参与者的权利, 提高参与者采取诚实行为的积极性. 这些措施有助于构建一个更为可信和公平的联邦学习环境, 吸引更多用户积极参与, 并增加恶意参与者的作恶成本.

1 相关工作

为了保护深度神经网络的知识产权, 提出了 DNN (deep neural network) 水印技术, 将指定的水印嵌入 DNN 模型中. 随后, 通过从相关模型中鲁棒提取嵌入的水印来验证 DNN 的所有权. 目前模型水印主流手段包括两种: 基于特征的水印[6−13]和基于后门的水印[14−18]. 目前, DNN 水印技术已经被用于 FL 模型的版权保护[19], 研究可大致分为两类: 客户端主导和服务器主导.

在客户端主导的方案中, 文献 [20] 提出的方案 FLWB, 能够允许各参与方在其本地模型中分别嵌入私有水印, 再通过模型聚合将私有水印映射到全局模型中作为全局水印. 同时, 通过分步训练方法增强各私有后门水印在全局模型的表达效果; 文献 [21] 提出的方案 FedIPR, 分别实现了私有和公开可验证的模型水印, 其中, 私有水印基于参数而公开水印基于后门; 文献 [16] 提出的方案可以兼容在同态加密的联邦学习框架中; 文献 [11] 提出的方案 FedCIP, 是第 1 个面向半诚实服务器与恶意客户端的白盒模型水印方案, 能够兼容安全聚合技术并追踪恶意客户

端. 客户端在每轮或每几轮训练周期中生成并嵌入不同的水印, 则通过分析对比缺失的水印与应嵌入的水印信息识别出泄密者; 文献 [22] 提出的 FedCRMW 方案中的每个客户端利用其独有的二进制标识符和专有标志构建触发集, 使得水印更具鲁棒性的同时还能更好地确定水印对应的客户端. 客户端主导的方式意味着客户端在训练模型时直接嵌入水印, 或者在本地进行水印生成和嵌入, 能够更好地保护隐私数据和分散化控制.

在服务器主导的方案中, 文献 [15] 提出的 WAFFLE 方案是第 1 个为联邦学习模型进行所有权确认的方案, 该方案提出了一种根据无关数据集生成后门水印的方法; 文献 [23] 提出的方案 FedTracker 采用全局水印机制和局部指纹机制共同实现所有权验证和模型泄露追溯, 其中, 全局水印基于后门而局部指纹基于参数; 文献 [24] 提出的方案 FedRight 通过提取模型特征并训练检测器, 以黑盒方式验证模型版权; 文献 [25] 提出的方案 RobWE 划分模型头部层和表示层, 头部层水印由客户端生成嵌入不参与聚合, 用于标识客户端的所有权, 而表示层水印由服务器分发并通过聚合在全局共享, 用于评估客户端对全局模型的贡献度. 服务器主导的方式具备更好的水印一致性、验证效率, 便于统一管理.

上述研究成功地将模型水印运用于联邦学习场景, 并具备不错的水印效能. 但这些研究只能抵御单一的攻击手段, 尚无法同时抵御模糊攻击、共谋攻击和搭便车攻击. 现有方法通常要求水印嵌入者绝对诚实, 即嵌入方必须完全可信且不会滥用权限或篡改水印信息. 这种强信任假设在实际应用中往往难以满足, 尤其是在多方协作或分布式环境中, 嵌入方可能存在恶意行为或操作失误, 从而导致水印的可靠性受到威胁. 本研究根据参与方特点和需求设计出更合理的角色分工, 发挥客户端主导和服务器主导的优势. 因此本研究能够进一步丰富联邦学习场景下模型水印的防御效果, 提高对新攻击手段的防御能力. 为了杜绝不诚实行为发生, 我们需要解决如下 3 个问题.

1) 模型所有权验证: 对联邦学习模型进行可验证的版权界定, 使得模型的任何合法使用都需要经过整个联邦学习集团的同意, 各个参与方都能参与进版权保护和验证的环节.

2) 模型泄露追溯: 在联邦学习参与者中, 非法将模型分发给联邦学习集团外的第三方的成员会被检测出来.

3) 懒惰客户端检测: 在联邦学习训练阶段, 采取搭便车攻击的懒惰客户端会被检测出来.

2 基础知识

2.1 后门水印

利用后门攻击将一组特定的输入作为触发集插入 DNN 模型中, 当遇到触发集中的样本时, 后门攻击会导致误分类[18]. 触发集对于后门模型是唯一的, 因此所有者可以通过触发错误分类来验证其所有权. 基于后门的水印可以通过黑盒访问进行验证, 即在不访问模型内部参数的情况下, 可以通过远程调用 API 来收集模型所有权的证据.

2.2 模型指纹

随着越来越多的预训练 DNN 在互联网上开源或分发, 这些公共可用模型的知识产权保护和数字版权管理 (digital rights management, DRM) 对于保持模型所有者的竞争力和促进可靠的技术转让尤为重要. 针对这一要求, 研究者提出了数字指纹技术[26], 数字指纹技术可用于保护数字版权. 当内容分发给多个用户时, 每个用户接收的副本中都会嵌入独特的指纹, 这使得如果用户非法分发该内容, 原始分发者能够通过分析泄露的副本来确定泄密源头. 由于指纹对于每个副本都是唯一的, 因此可以解决水印的模糊性, 并能够跟踪特定用户进行的版权滥用, 实现泄露溯源和更精细的数字产品版权管理[7].

2.3 ACC 和 BIBD

抗共谋码 (anti-collusion code, ACC)[27]是一系列码向量, 其中共享的位可以唯一标识多个共谋者. 抗共谋码的特性在于任何 k 或更少的码向量子集都是唯一的. 因此, 利用这一特性, 抗共谋码能够识别最多 k 个共谋者. 现有研究表明, 二进制的 AND-ACC 码存在, 并且可以通过平衡不完全区组设计 (balanced incomplete block design, BIBD)[28]构建. 区组设计理论是数学的一个分支, 它已被应用于纠错码的构建和统计实验的设计中. 对应 k–1 容量 AND-ACC 码向量被指定为 (v, k, 1)-BIBD 关联矩阵的列. 在这种情况下, 码向量是 v 维的, 并且能够用这 v 个基向

量表示 $n=\frac{v^2-v}{k^2-k}$ 个用户. 因此, 给定 k–1 的鲁棒性, 只有 $\sqrt{n}$ 个基向量即可容纳 n 个用户.

现有研究已系统地证明可以构建无限个 BIBD, 因此可以为多个用户无限地构建 ACC 码向量. 因此, 本文方案可以通过 ACC 理论支持, 从而实现对共谋者的正确识别.

2.4 分布式水印导致水印冲突问题

由于各个客户端分配的水印各不相同, 在多个客户端将其独特的水印嵌入共享的全局模型中时, 可能会出现一个显著的挑战: 水印冲突. 具体来说, 会出现水印学习不均衡的情况, 如图 1 所示, 这体现在部分水印触发率已经很高, 同时, 另一部分水印学习尚不充分, 触发率很低. 这将导致模型水印的整体触发率较低, 且将训练资源和模型参数浪费在已经学会的水印当中, 模型保真度和水印触发率将会更低, 这是我们不愿意看到的.

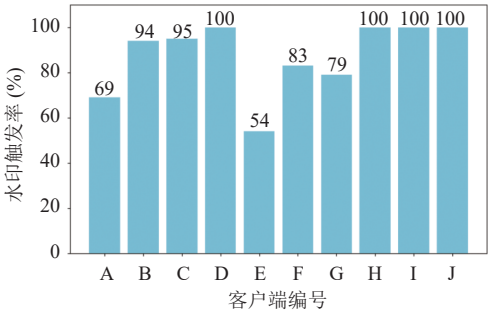


图 1 常规分布式水印嵌入的水印触发情况

3 方案构建

本节将详细介绍提出的联邦学习模型水印方案的构建过程. 首先, 概述方案的基本设计思路和主要功能, 包括所有权验证、泄露模型追溯和懒惰客户端检测. 其次, 阐述水印嵌入的具体方法, 包括客户端与服务器的分工合作、水印的生成与插入机制以及如何通过动态调整聚合权重算法来解决水印冲突问题. 此外, 还将深入讨论如何通过客户端身份标识生成后门水印来确保水印的唯一性和安全性, 最终实现对模型版权的有效保护.

3.1 方案概述

本文设计了一种具有所有权验证、泄露模型追溯和懒惰客户端检测功能的分布式水印方案, 如表 1 所示. 场景为联邦学习最流行的客户端-服务器范式[29], 一方面, 客户端在训练时为本地模型嵌入后门水印, 由服务器在联邦学习的聚合阶段将各客户端的后门水印融入全局模型中, 实现私有后门到联邦学习全局水印的映射, 同时, 本地后门水印嵌入任务也是懒惰客户端检测手段. 另一方面, 在每次迭代分发模型之前, 服务器为每个客户端嵌入一个唯一的本地指纹来标识每个模型, 通过公开模型内部的标识, 可以对模型泄露者进行追踪.

表 1 不同方案功能对比

方案	支持API验证	支持IP追踪	支持懒惰客户端检测	支持抗共谋追踪
WAFFLE	√	×	×	×
FedTracker	√	√	×	×
FedIPR	√	×	√	×
FedCIP	×	√	×	×
DeepMarks	×	√	×	√
本文方案	√	√	√	√

本文方案贡献如下: (1) 采用客户端侧分布式水印嵌入的方式, 有效抵御服务器可能在全局水印嵌入中的恶意行为, 同时降低水印嵌入过程中服务器的计算开销; (2) 针对分布式后门水印嵌入可能出现的水印责任认定和水印冲突问题, 提出身份标识生成后门水印机制和动态调整聚合权重算法; (3) 通过分析客户端上传模型中的水印嵌入

情况, 以阈值检测的方式有效识别懒惰客户端; (4) 本方案除了抵抗常规攻击手段外, 还能够防止多个客户端联合进行的共谋攻击.

3.2 算法定义

● 所有权验证. 指证明可疑模型属于 FL 团队, 并对团队外的未经授权用户提起侵权诉讼, 以维护自身合法版权. 可疑模型被定义为涉嫌为 FL 模型副本的模型. 当发生侵权时, FL 团队可以使用验证算法 $Verify(\cdot)$ 来验证其模型的所有权. 所有权验证机制 $Verify(M_{\text{adv}}, D_T)$ 的输出应为:

$$Verify(M_{\text{adv}}, D_T) = \begin{cases} \text{True}, & \text{if } M_{\text{adv}} = Adv(\overline{M}) \\ \text{False}, & \text{otherwise} \end{cases} \tag{1}$$

其中, M_{adv} 是可疑模型, D_T 是 FL 团队提供的触发集, $Adv(\cdot)$ 代表各种攻击方法, $\overline{M}$ 代表含水印模型. 若验证机制可以确认可疑模型中存在水印, 则成功验证, 触发集的准确率达到阈值表示水印存在.

● 泄露模型追溯. 指追踪被盗模型到 FL 中的恶意客户端. 在验证了可疑模型的所有权后, FL 团队应进一步采用追踪机制 $Trace(\cdot)$ 来揭示模型泄露者的身份. 对于泄露模型的参与方检测, 既要考虑单个客户端泄露的情况, 也要考虑多个客户端合作共谋的情况. 给定可疑模型 M_{adv} 和客户端的指纹集合 $F_{\text{all}} = \{F_i\}_{i=1}^{N} \cup \{F_c\}$, 其中 $\{F_i\}_{i=1}^{N}$ 代表所有客户端分配到的初始指纹, F_c 代表所有可能的共谋结果, 追踪机制 $Trace(M_{\text{adv}}, F_{\text{all}})$ 应输出 FL 中模型泄露者的身份 (即索引号). 为了逃避追溯, 多个恶意客户端可能联合起来, 采用共谋策略, 消除原始模型指纹, 得到伪造指纹. 共谋指纹副本如下:

$$\tilde{f} = \frac{(f_1 + f_2 + f_3 + \ldots + f_n)}{n} \tag{2}$$

追踪机制通过从可疑模型 M_{adv} 中提取指纹 $\tilde{f}$, 将该指纹与 ACC 码本中的全部可能结果采取指纹相似得分 (fingerprint similarity score, FSS) 进行对比[23]. 相较于传统的汉明距离对比 $HD(\tilde{f}, F_i)$, 汉明距离是一个度量, 其输出是离散的, 在寻找最相似的指纹时, 可能存在歧义. 而 FSS 的输出是连续的, 可以更好地测量指纹之间的差异.

$$FSS(X_i, F_i, W) = FSS(B_i, F_i) = \sum_{j=1}^{|F_i|} \min(\delta, b_{ij} f_{ij}) \tag{3}$$

在相似度达到阈值的情况下, 选择最相似的一个或一组身份作为追溯结果.

● 懒惰客户端检测. 指 FL 团队采用 $Check(\cdot,\cdot)$ 来检测每一个客户端上传的本地模型是否为伪造. 懒惰客户端检测机制的输出为:

$$Check(M_g, i) \begin{cases} \text{True}, & \text{if } Verify(M_g, D_T^i) > \varepsilon_T \\ \text{False}, & \text{otherwise} \end{cases} \tag{4}$$

本方案将懒惰客户端的检测时间设置为训练中期, 这是由于在训练早期, 模型训练尚不稳定, 后门嵌入情况各不相同, 无法保证各客户端的后门水印融入全局模型中. 而在训练后期, 此时的联邦学习模型已经接近训练结束, 具备一定的价值, 此时懒惰客户端保存的模型副本的性能可能已经足够使用. 检测时间的选择需要保证各个客户端的后门水印成功进入聚合模型当中, 且此时模型训练尚未完成, 具体视情况而定.

3.3 威胁模型

服务器和客户端是联邦学习中的两个不同参与方. 对于服务器, 我们假设服务器是诚实的, 且设备算力有限, 只能支持全局模型聚合而不能参与模型训练; 对于客户端, 我们假设大部分客户端是诚实的, 但存在两类客户端作为对手: (1) 恶意客户端: 可以偷偷地复制、分发和销售联合训练的模型; (2) 懒惰客户端: 通过搭便车攻击获取全局模型. 这两类客户端的目标是非法获取和处理模型, 出于理性考虑, 它们不会破坏联邦学习的训练过程, 因为它们希望训练出的模型具备高价值.

对手假设: 恶意客户端会将具有一定使用价值的模型秘密出售给未经联邦学习集团授权的其他方. 此外, 攻击者知道模型中含有水印, 但不知道水印的具体嵌入方法, 对手将尝试去除模型中的水印. 懒惰客户端希望在不提供数据集和算力的情况下实施搭便车攻击, 以获取最终全局模型的收益.

我们假设对手具备以下能力.

(1) 对手可以在 FL 训练的任意迭代 t 中访问自己的数据集 D_{adv} 和本地模型 M_{adv}^t. 同时, 对手可以在训练的任何时候保存和复制模型.

(2) 对手遵循训练程序来最大化 FL 模型的效用和价值.

(3) 对手可以通过多种水印去除技术来去除 DNN 模型中的水印. 水印消除技术如下.

● 微调: 对手可能采用微调来混淆模型参数和行为. 微调攻击[14]是指恢复训练过程, 并使用对手的数据集和较低的学习率修改模型参数. 由于深度学习的非凸性, 微调可能使深度学习模型达到不同的局部最优点, 从而移除水印. 因此, 微调应被视为对水印的一种威胁.

● 模型压缩: 模型压缩的目的是消除模型参数中的冗余, 因此在此过程中嵌入的水印也可能被擦除. 在本文中, 我们考虑了两种不同类型的模型压缩方法: 模型剪枝[30]和量化. 模型剪枝消除了对网络性能影响很小或没有影响的参数. 模型量化降低了参数的数值精度, 例如使用 8 位整数代替 32 位浮点数, 以节省内存带宽和计算成本. 这两种技术都可以在精度损失最小的情况下对水印产生负面影响.

● 水印覆写: 覆写目的是通过在神经网络模型中添加新的水印来去除或改变嵌入在该模型中的水印. 覆盖攻击是一种主动攻击, 攻击者随机生成自己的水印, 并使用相同的方法将水印嵌入到模型中. 覆写会对模型中的水印产生较大影响.

● 指纹共谋攻击: 一组拥有同一全局模型但嵌入不同指纹的客户端可能会进行共谋攻击, 以构建一个服务器无法检测到指纹, 同时具备与原始全局模型功能相近的伪造模型. 由于没有一个共谋者愿意比其他共谋者承担更大的风险, 因此通常对每个共谋者的指纹信息进行等权平均[31]:

$$\tilde{F} = \sum_{i=1}^{K} \frac{1}{K} F_i \tag{5}$$

(4) 对手可能采取两种策略构建伪造的本地训练结果[4].

利用之前的模型进行伪造, 公式如下:

$$\mathbf{W}_{\text{free}} = Free(\mathbf{W}^t, \mathbf{W}^{t-1}) \tag{6}$$

其中, $\mathbf{W}^t$、$\mathbf{W}^{t-1}$ 代表前两次服务器下发的全局模型, 懒惰客户端利用加权平均等方法伪造本地模型训练结果.

利用高斯噪声进行伪造, 公式如下:

$$\mathbf{W}_{\text{free}} = \mathbf{W}^t + \xi_t,\ \xi_t \sim \mathcal{N}(0, \sigma_t) \tag{7}$$

懒惰客户端采用上一轮的全局模型参数, 并加入高斯噪声模拟本地模型训练结果.

3.4 防御方假设和目标

在我们的场景中, 联邦学习所有诚实参与者都是防御者. 如图 2 所示, 防御者试图为联邦学习模型提供所有权验证功能、泄露模型追溯功能和懒惰客户端检测功能, 阻止这些行为影响到联邦学习的公平性、正义性.

防御者需要实现以下 6 个目标, 包括所有权验证、后门水印不可伪造、泄露模型追溯、懒惰客户端检测、保真度和鲁棒性.

(1) 所有权验证: 防御者可以设计一种机制, 在 FL 模型上有效地验证 FL 组的所有权.

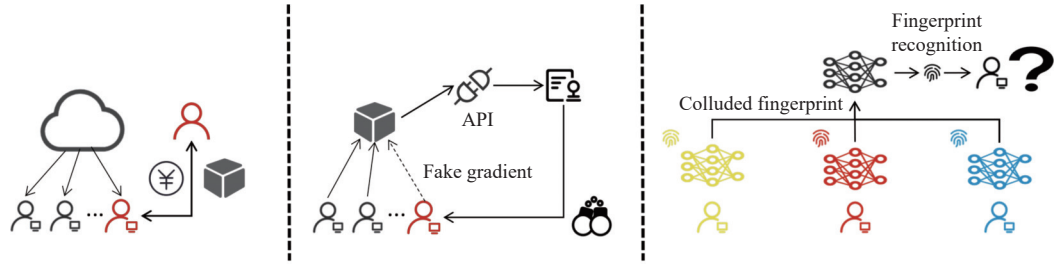
$$\Pr[\text{Verify}(\overline{M_i}, D_T) = 1 \mid Train\&EmbedM_i, D_i, D_T^i) \rightarrow \overline{M_i}] = 1 \tag{8}$$

(2) 后门水印不可伪造: 即使攻击者知道水印的存在, 也无法生成有效的伪造证明向第三方证明对于该模型的所有权.

$$\Pr[Verify(\overline{M_i}, D_T^{\text{adv}}) = 1 \mid Adv(\overline{M_i}, D_i) \rightarrow D_T^{\text{adv}}] < negl(\lambda) \tag{9}$$

(3) 泄露模型追溯: 给定一个可疑模型, 防御者可以在确认该模型为泄露模型后在 FL 组内跟踪该模型的持有者并抓住罪魁祸首.

$$\Pr[Trace(M_{\text{adv}}, F_{\text{all}}) = F_{\text{adv}} \mid Adv(\overline{\overline{M_i}}) \rightarrow M_{\text{adv}}] = 1 \tag{10}$$



(a) 恶意客户端私自出售模型 (b) 懒惰客户端搭便车攻击 (c) 多个恶意客户端联合采用共谋攻击方式消除原始指纹

图 2 水印需解决问题示意图

(4) 懒惰客户端检测: 在联邦学习过程中, 应当根据全局模型训练结果找到懒惰客户端.

$$\Pr[Check(M_g^t,\{D_i\}_{i=1}^K)=i_{\text{Freerider}}\mid Free(W^{t-1})\to W_{\text{free}}^t]=1 \tag{11}$$

(5) 保真度: 保真度指的是保护机制对模型效用的影响应该可以忽略不计.

$$\max\left\{\left|M(x)-\overline{M}(x)\right|,\left|M(x)-\overline{\overline{M}}(x)\right|,\left|\overline{M}(x)-\overline{\overline{M}}(x)\right|\right\}<negl(\lambda) \tag{12}$$

(6) 鲁棒性: 鲁棒性是指保护机制不能轻易被对手移除.

$$\Pr[Verify(M_{\text{adv}},D_T)=1\mid Adv(\overline{M})\to M_{\text{adv}}\approx M]=1 \tag{13}$$

4 详细方案

本方案共 2 个阶段: 1) 初始化阶段; 2) 训练阶段. 本文所提方案整体流程如算法 1 所示.

算法 1. 方案整体流程.

输入: 客户端数据集 $\{D_i\}_{i=1}^K$, 客户端身份信息 $\{m_i\}_{i=1}^K$;
输出: 受保护的客户端模型 $\{\overline{\overline{M}}_i^T\}_{i=1}^K$, 受保护的全局模型 $\overline{M}_g^T$.

1. 生成后门触发集和模型指纹 $\{D_T^i\}_{i=1}^K,\{F_i\}_{i=1}^K=Gen()$
2. 初始化模型 $M_g^0=Initialize()$
3. 对于 i=1 到 K 个成员:
4. 复制并分发初始全局模型 $M_i^0=Copy(M_g^0)$
5. 对于 t=1 到 T 个训练轮次:
6. 对于 i=1 到 K 个成员:
7. 本地训练与水印嵌入 $\overline{M}_i^t=Train\&Embed(M_i^{t-1},D_i,D_T^i)$
8. 聚合模型 $\overline{M}_g^t=FedDWA(\{\overline{M}_i^t\}_{i=1}^K,\{D_T^i\}_{i=1}^K)$
9. 对于 i=1 到 K 个成员:
10. 复制全局模型 $\overline{M}_i^t=Copy(\overline{M}_g^t)$
11. 嵌入局部指纹 $\overline{\overline{M}}_i^t=FInsert(\overline{M}_i^t,F_i)$
12. 对于 i=1 到 K 个成员:
13. 模型分发 $M_i^{t+1}=\overline{\overline{M}}_i^t$

4.1 初始化阶段

服务器和客户端应当先生成用于所有权验证的全局水印和用于泄露模型追溯的局部指纹. 全局水印由客户端

生成并嵌入, 并告知服务器; 局部指纹由服务器独立生成和嵌入.

在先前研究中, 部分方案将对抗样本作为后门水印[16,22,24]. 这种水印生成方法容易导致歧义攻击[10], 即攻击者根据掌握的模型采取逆向工程或其他手段, 得到伪造水印. 在这种情况下, 合法的模型所有者和攻击者都可以提供“有效”的水印, 使得法律上难以确定真正的所有者.

在本方案中, 每个客户端的身份是与其水印紧密关联的. 为了确保每个客户端水印的唯一性, 并能够有效区分不同客户端的贡献, 我们设计了一个基于客户端身份标识的后门水印生成机制. 通过这一机制, 系统能够为每个客户端生成独特的后门水印, 防止水印被复制或伪造, 从而增强整个水印框架的完整性和安全性. 触发集的构建要求满足安全性和可解释性, 触发集的生成方法与流程如下.

● 全局水印生成: 每个客户端向服务器提供其身份信息 m_i; 服务器使用哈希函数 H 对客户端的身份信息 m_i 进行计算, 生成哈希值 h_i; 服务器从哈希值 h_i 中保留前 l 比特作为客户端的身份标识, 表示为 $ID_i=(b_{i1},b_{i2},\ldots,b_{il})$. 这里, b_{ij} 表示第 i 个客户端的第 j 个比特值 ($j\in\{1,2,\ldots,l\}$). 该标识将用于生成唯一的触发图片和触发标签.

● 具体触发图片和触发标签值生成方法: 选取部分训练集中的图片, 划分为 l 个均匀的方格, 对于每个方格, 检查客户端身份标识的对应比特值 ID_i. 对于每个方格 j, 根据公式 (14) 的规则修改原始图片, 生成触发图片如图 3; 触发标签 Y_T^i 的生成根据公式 (15) 进行, 其中 C 代表数据集类别数量.

$$\text{方格}j=\begin{cases}\text{添加噪声}, & \text{if } b_{ij}=1\\ \text{保持不变}, & \text{if } b_{ij}=0\end{cases} \tag{14}$$

$$Y_T^i=\left(\sum_{j=1}^{l}b_{ij}\right)\bmod C \tag{15}$$

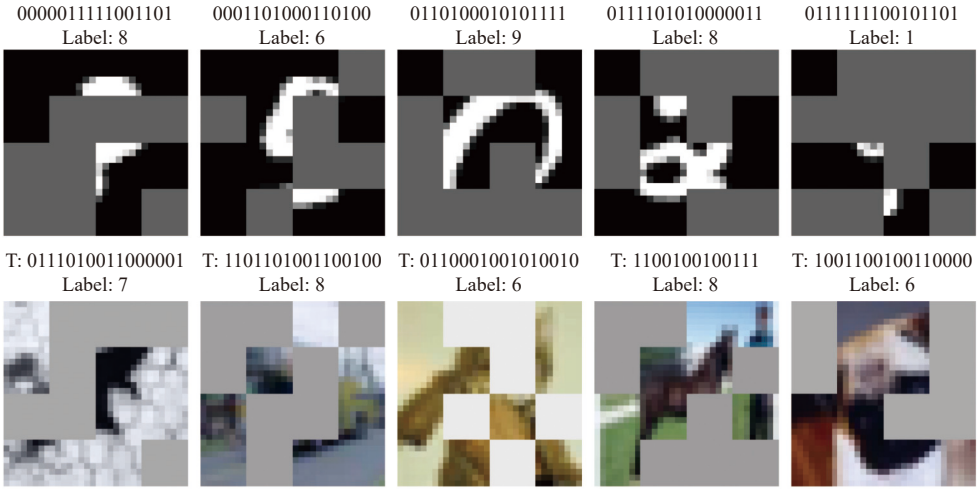


图 3 MNIST 和 CIFAR-10 数据生成的触发图片及标签

通过使用客户端的身份标识来生成后门水印, 本文确保了每个客户端的水印是唯一的且不可伪造. 这种机制的引入有效防止了恶意客户端试图冒充其他客户端的水印或非法篡改水印信息, 从而提升了整个水印框架的安全性. 此外, 生成的后门水印在嵌入过程中接近正常数据的分布, 这使得水印不会明显干扰模型的性能.

通过这种方式, 系统能够有效区分每个客户端的水印, 且在验证时能够准确识别出水印来源. 这为联邦学习模型的版权保护和追踪懒惰客户端提供了强有力的保障.

● 局部指纹生成: 采用基于参数的水印技术, 服务器为每个客户端分配一串 N 比特长的水印信息 $F\in\{-1,1\}^N$, 水印信息的选取来自 AND-ACC 码本, 二进制 ACC 使用 BIBD 生成. 同时, 为了掩盖真实水印信息, 需要生成秘密矩阵 U, 待嵌入水印信息 $b=FU$. 服务器还应当选取模型参数中的适当位置作为水印嵌入位, 我们选择深层 BN 层[32],

在 BN 层中, 我们计算公式 (16) 来对前一层的输出 x 进行归一化:

$$o=\gamma\frac{x-\mu}{\sqrt{\sigma^2+\epsilon}}+\beta \tag{16}$$

其中, μ 和 σ 代表批输入 x 的均值和方差, ϵ 是一个常数, γ 和 β 是可学习的参数. 我们选择 γ 来嵌入局部指纹, 实验表明[8,21,23,33], 在该位置嵌入水印, 对模型性能的影响较小, 且水印的鲁棒性较好. 这是因为 BN 层在训练过程中相对稳定, 相较于其他模型参数不易出现大幅度变化.

4.2 训练阶段

如图 4, 训练阶段分为 4 个阶段: ① 本地模型训练与全局水印嵌入; ② 全局模型聚合; ③ 局部指纹嵌入; ④ 模型分发.

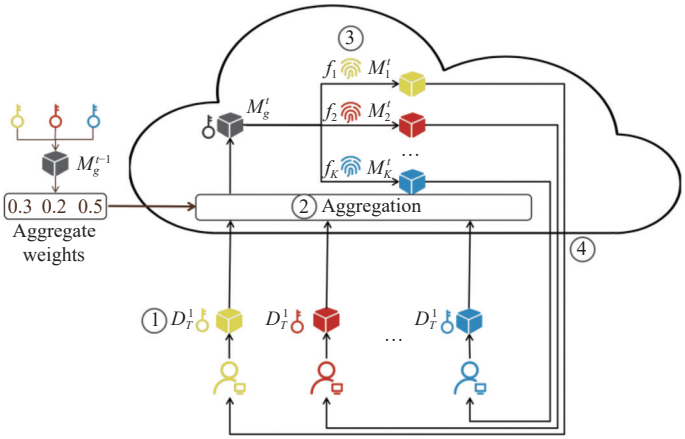


图 4 模型训练阶段工作流程

4.2.1 本地模型训练与全局水印嵌入

全局水印旨在为 FL 模型提供所有权验证功能. 在此阶段, 每个客户端利用其自己的数据集训练其本地模型. 本地训练与全局水印嵌入算法 $Train\&Embed(M_i,D_i,D_T^i)\rightarrow\overline{M_i}$ 用于将全局水印 D_T^i 嵌入模型 M_i 中, 并输出带有部分全局水印的模型 $\overline{M_i}$. FL 中的所有模型都被全局水印保护, 各个客户端损失函数如下:

$$L_s=L_o+\alpha L_w=L_o+\alpha(CE(Y_T^i,M(X_T^i))) \tag{17}$$

其中, L_o 代表原始任务损失函数, α 代表水印嵌入强度控制参数, L_w 代表全局水印嵌入任务损失函数, $CE(\cdot,\cdot)$ 代表二元交叉熵, $M(\cdot)$ 代表模型输出值.

4.2.2 全局模型聚合

在此阶段, 服务器接收各个客户端上传的本地模型并聚合为全局模型, 聚合时选择采取动态调整权重聚合 (federated dynamic weight adjustment, *FedDWA*) 算法 $FedDWA(\{M_i^t\}_{i=1}^K,\{D_T^i\}_{i=1}^K)\rightarrow M_g^t$.

- 第 1 轮: 服务器采取联邦平均聚合策略[1], 各个客户端权重 $P_i=\dfrac{1}{K}$.
- 第 2 轮及以后: 权重根据上一轮各个客户端的水印触发率动态调整. 触发率低的客户端获得更高的权重, 触发率高的客户端获得更低的权重. 各个客户端的权重由两部分组成: 基础权重 P_{base} 和额外权重 P_{extra}. 基础权重是每个客户端的最低权重, 由用户指定, 取值不超过 $\dfrac{1}{N}$, 用于控制动态调整的幅度. 聚合权重计算公式如下:

$$P_i=P_{\text{base}}+P_{\text{extra}}=P_{\text{base}}+(1-TA_i)\times\frac{1-K\times P_{\text{base}}}{\sum_{j=1}^{k}(1-TA_j)} \tag{18}$$

其中, TA 代表水印触发率, 范围为 0–100%. 基础权重的参数范围是 0 至 $\dfrac{1}{N}$ %, 默认设置为 $\dfrac{1}{2N}$.

4.2.3 局部指纹嵌入

本方案将指纹嵌入 BN 层的权重概率密度函数中, 选取秘密矩阵 $X_{v\times N}$, 码本矩阵 $B_{v\times b}$ 作为只有嵌入者知道的秘密安全参数, 其中 v 代表水印长度, N 代表待嵌入参数数量, b 代表码本容量.

在指纹生成阶段, 服务器首先根据 AND-ACC 抗共谋编码码本 B 生成 K 个局部指纹向量 (K 为客户端数量), 得到能够表示每一个客户端身份的唯一本地指纹. 为了说明方案所提出的指纹构造, 以 (13, 4, 1)-*BIBD* 码本为例:

$$(13,\ 4,\ 1)\text{-}BIBD=\begin{bmatrix}1&0&1&0&1&0&1&0&1&0&1&0&1\\0&1&1&0&0&1&1&0&0&1&1&0&0\\0&0&0&1&1&1&1&0&0&0&0&1&1\\0&0&0&0&0&0&0&1&1&1&1&0&0\\1&0&0&1&1&0&0&1&1&0&0&1&1\\0&1&0&1&0&1&0&1&0&1&0&1&0\\0&0&1&1&0&0&1&1&0&0&1&1&0\\1&1&1&0&0&0&0&1&1&1&1&0&0\\1&1&0&0&1&1&0&0&1&1&0&0&1\\1&0&1&0&1&0&1&0&1&0&1&0&1\\0&1&1&0&0&1&1&0&0&1&1&0&0\\0&0&0&1&1&1&1&0&0&0&0&1&1\\0&0&0&0&0&0&0&1&1&1&1&1&0\end{bmatrix}\tag{19}$$

其中, 矩阵每一列代表一个用户的原始指纹信息, 水印长度为 v 为:

$$\begin{cases}b_1=1,0,0,0,1,0,0,1,1,1,0,0,0\\b_2=0,1,0,0,0,1,0,1,1,0,1,0,0\\\quad\vdots\\b_{12}=0,0,1,1,1,1,1,0,0,0,0,1,1\\b_{13}=1,0,1,0,1,0,0,0,1,1,0,1,0\end{cases}\tag{20}$$

为了隐藏指纹信息, 先将原始指纹信息中的 0 元素映射为–1, 得到:

$$\begin{cases}c_1=1,-1,-1,-1,1,-1,-1,1,1,1,-1,-1,-1\\c_2=-1,1,-1,-1,-1,1,-1,1,1,-1,1,-1,-1\\\quad\vdots\\c_{12}=-1,-1,1,1,1,1,1,-1,-1,-1,-1,1,1\\c_{13}=1,-1,1,-1,1,-1,-1,-1,1,1,-1,1,-1\end{cases}\tag{21}$$

假设客户端 C1 与 C2 共谋, 双方采取线性共谋策略, 得到伪造指纹 $f=\frac{1}{2}(c_1+c_2)=[0,0,-1,-1,0,0,-1,1,1,0,0,-1,-1]$. 根据 AND-ACC 理论的要求, 采取阈值处理:

$$f_i=\begin{cases}1, & \text{if } c_{nj}+c_{mj}=1\\0, & \text{otherwise}\end{cases}\tag{22}$$

得到最终唯一编码向量 [0,0,0,0,0,0,1,1,0,0,0,0], 该向量只有在客户端 C1 和 C2 共同参与共谋攻击时才能生成. 在 (13, 4, 1)-*BIBD* ACC 码本中, 可以识别最多 3 个共谋者, 对于这些共谋得到的伪造指纹, 1 出现的位置相对于共谋结果集合是唯一的.

以客户端 1、12、13 为例, 其所有可能共谋结果如表 2 所示.

在指纹嵌入阶段, 使用局部指纹嵌入算法 $FInsert(\overline{M_i'},F_i)\to\overline{\overline{M_i'}}$. 该步骤的目标是通过调整模型参数值, 实现秘密矩阵 X 与参数 W^{γ} 相乘的结果的每一个元素的符号位与指纹对应位置的符号位相同, 即 $sgn(XW^{\gamma})=F$.

$$sgn(x)=\begin{cases}-1, & x\leqslant 0\\1, & x>0\end{cases}\tag{23}$$

首先确定嵌入位置 (某一 BN 层的 γ 位置), 将选定模型参数展平为向量 $w\in\mathbb{R}^N$, 将参数向量 w 乘以秘密矩阵 X. 最后, 采用梯度下降方法训练参数值, 使得参数向量 w 乘以秘密矩阵 X 得到的新向量与局部指纹符号位尽量相

同, 达到局部指纹嵌入的效果, 损失函数 L_f 如下:

$$L_f(W^\gamma) = MSE(b_{ij} - f_{ij}) \tag{24}$$

其中, b_{ij} 为矩阵 $B_i = X_i W_i^\gamma$ 的第 j 位. 为了减少对其他任务的影响, 我们采用较小的学习率来优化 L_f. 在服务器分别为每个客户端执行梯度下降后, 局部指纹嵌入完成. 需注意的是, 为保证水印嵌入后的鲁棒性和不可见性, 水印嵌入应该选择较深层的位置.

表 2　客户端 1、12、13 的指纹共谋示例

客户端/客户端共谋	1	2	3	4	5	6	7	8	9	10	11	12	13
Client 1	1	−1	−1	−1	1	−1	−1	1	1	1	−1	−1	−1
Client 12	−1	−1	1	1	1	1	1	−1	−1	−1	−1	1	1
Client 13	1	−1	1	−1	1	−1	−1	−1	1	1	−1	1	−1
Colluders(1, 12)	0	−1	0	0	1	0	0	0	0	0	−1	0	0
Colluders(1, 13)	1	−1	0	−1	1	−1	−1	0	1	1	−1	0	−1
Colluders(12, 13)	0	−1	1	0	1	0	0	−1	0	0	−1	1	0
Colluders(1, 12, 13)	0	−1	0	0	1	0	0	0	0	0	−1	0	0

注: 前3行为客户端, 后4行为客户端共谋

4.2.4　全局模型分发

在此阶段, 服务器将含有指纹的模型分别分发给对应客户端. 在收到模型后, 客户端开始 FL 的下一个训练迭代. 对于恶意客户端, 往往会在此阶段保存模型副本, 并采取消除水印的攻击手段.

5　实验分析

5.1　实验设置

● 模型与数据集: 实验部分采用 CNN-4[34]对 MNIST 数据集进行训练; 采用添加了 BN 层的 AlexNet[35]网络对 CIFAR-10 数据集[36]进行训练.

● 联邦学习设置: 为模拟联邦学习场景, 我们将客户端个数分别设置为 10、20、30 个. 每轮训练, 客户端在本地迭代两轮, 学习率设置为 0.01. 各个客户端掌握的数据集服从独立同分布.

● 水印和指纹: 对于全局后门水印, 我们为每个客户端分配一个独立的水印触发集, 训练过程中, 后门水印嵌入强度设置为 0.005; 对于本地指纹, 我们生成一个 (31, 6, 1)-BIBD 作为码本. 从码本中获取的指纹长度为 31 比特, 为了提高水印的鲁棒性和安全性, 将其长度扩展为 128 比特.

● 对比对象: 选取 FedTracker 进行保真度、所有权验证能力对比; 选取 FedIPR 进行懒惰客户端检测功能对比.

5.2　保真度

本文测试了在 10、20、30 个客户端数量下的联邦学习模型 (含有水印) 测试准确率, 如后文图 5 所示. 首先, 我们将嵌入水印后的模型效用与没有任何水印嵌入 (No WM) 的训练模型进行比较, 随着客户端数量增加, 模型准确率略有下降, 下降程度不大于 1%. 其次, 我们对动态调整权重聚合算法进行了消融实验, 实验表明, 该算法显著提高了模型效用, 准确率提升约为 2%. 最后, 与 FedTracker 相比, 本方案的模型保真度几乎相当, 有小幅度优化.

5.3　所有权验证功能

所有权验证效果: 本文测试了在 10、20、30 个客户端数量下的联邦学习模型的水印触发率, 如后文表 3 所示. 首先, 随着客户端数量的增加, 水印触发率有所降低, 但是整体维持在 95% 以上, 这表明水印可以有效完成所有权验证功能. 其次, 对动态调整权重聚合算法进行消融实验, 实验表明, 该算法显著提高了水印触发率, 提升幅度约为 17.5%. 最后, 与 FedTracker 相比, 本方案的水印触发率提升明显, 约为 12%.

5.4　懒惰客户端检测功能

懒惰客户端检测: 本文测试了采取两种不同搭便车攻击策略下的懒惰客户端检测情况, 图 6(a) 为采取高斯噪

声伪造梯度参与训练, 图 6(b) 为采取平均梯度伪造梯度参与训练, 图 6 中最下方深蓝色折线为 Freerider. 服务器可以在训练的中期阶段检测到诚实客户端的水印触发率与懒惰客户端的水印触发率相差超过 50%.

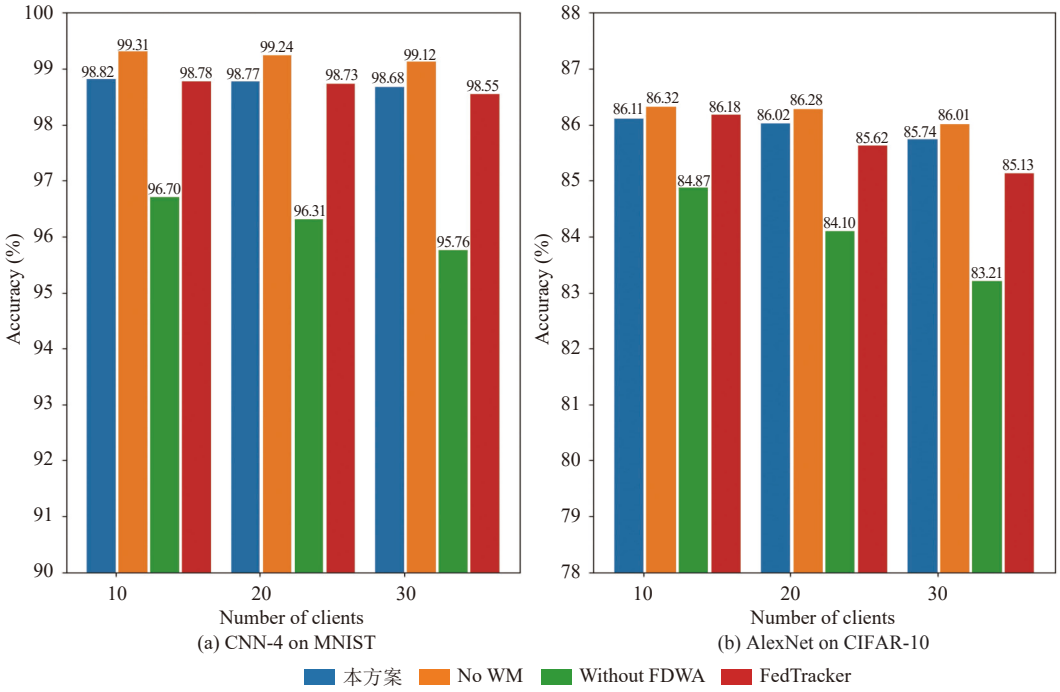


图 5 不同设置下的模型测试准确率对比

表 3 在 10、20、30 个客户端数量下不同方案的后门水印触发率 (%)

模型	方案	10	20	30
CNN-4 on MNIST	本方案	99.70	98.25	96.33
	FedTracker	84.70	86.90	81.00
	Without FDWA	80.50	80.10	79.90
AlexNet on CIFAR-10	本方案	95.90	95.70	95.70
	FedTracker	89.50	82.50	80.06
	Without FDWA	80.00	78.10	77.07

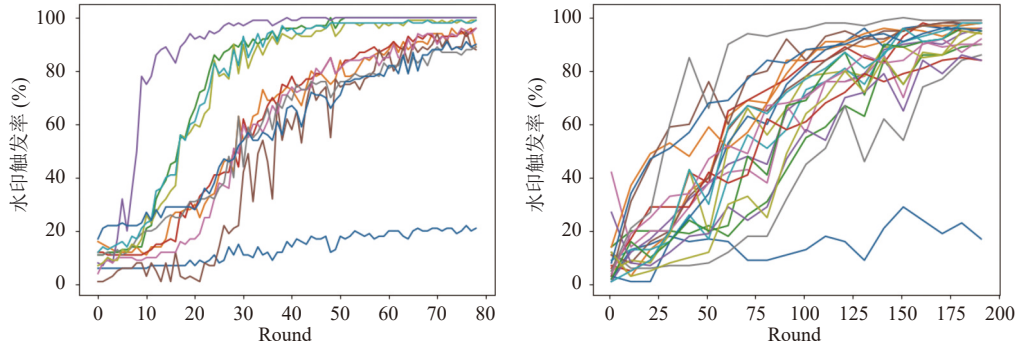


图 6 水印触发率变化情况 (包含懒惰客户端)

此外, 本文对动态调整权重聚合算法进行了消融实验, 将联邦学习参与方个数设置为 20 (其中懒惰客户端占比 25%), 检测时间设置为总训练轮次前 50%–60% 轮, 触发率阈值设置为 40%, 进行了 10 次实验. 表 4–表 6 以混淆矩阵的形式展示了不同条件下的懒惰客户端检测情况. 实验表明, 动态调整权重聚合算法提高了懒惰客户端的检测率, 降低了诚实客户端的误判率. 同时, 与 FedIPR 相比, 尽管两个方案都可以完美地检测出所有懒惰客户端, 但 FedIPR 可能会出现对诚实客户端的误判, 而本方案可以做到完美识别.

表 4　本方案不使用 FDWA 对懒惰客户端检测的情况

真实值	预测为诚实客户端	预测为懒惰客户端
真实为诚实客户端	123	27
真实为懒惰客户端	1	49

表 5　本方案使用 FDWA 对懒惰客户端检测的情况

真实值	预测为诚实客户端	预测为懒惰客户端
真实为诚实客户端	150	0
真实为懒惰客户端	0	50

表 6　FedIPR 对懒惰客户端检测的情况

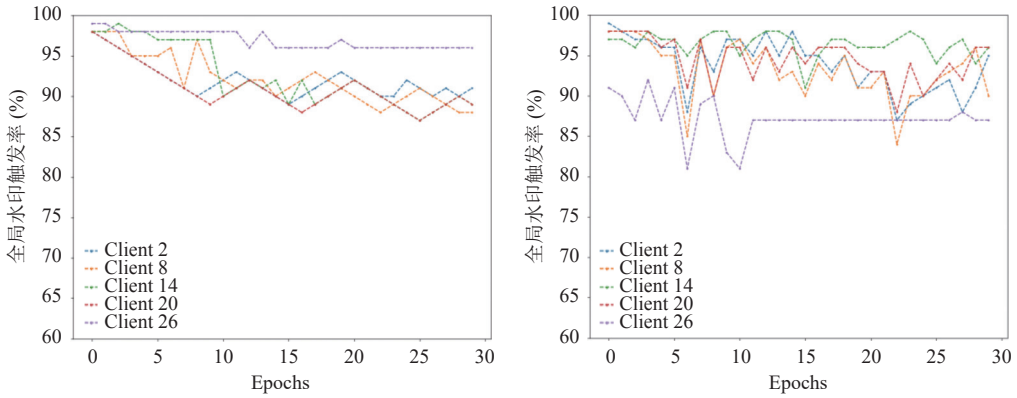
真实值	预测为诚实客户端	预测为懒惰客户端
真实为诚实客户端	139	11
真实为懒惰客户端	0	50

5.5 鲁棒性

鲁棒性指水印和指纹能否被攻击者消除. 在鲁棒性实验中, 本文分别测试了全局水印和局部指纹, 采取微调、剪枝、量化和水印覆盖攻击这 4 种方式来尝试消除水印. 对于攻击结果, 若出现以下两种情况的其中一种, 则视为水印或指纹是鲁棒的: (1) 水印和指纹在攻击后效果没有出现明显改变, 其仍然可以正常地验证和追溯模型. (2) 水印和指纹在攻击后效果出现明显损失, 功能不再可用, 同时, 与原始模型相比, 被攻击的模型性能出现明显下降, 无法胜任原始工作任务.

5.5.1 全局水印鲁棒性

● 微调攻击是指使用恶意客户端的私有数据集对模型进行再次训练. 在微调攻击实验中, 我们随机选取 5 个客户端的数据集分别进行 30 轮的微调训练, 并观察全局水印的触发率变化, 如图 7 所示. 虽然全局水印的触发率出现波动并小幅下降, 但总体维持较高值, 这表明本方案的全局水印是对微调攻击鲁棒的.



(a) CNN-4 on MNIST　　(b) AlexNet on CIFAR-10

图 7　微调攻击下全局水印触发率的变化情况

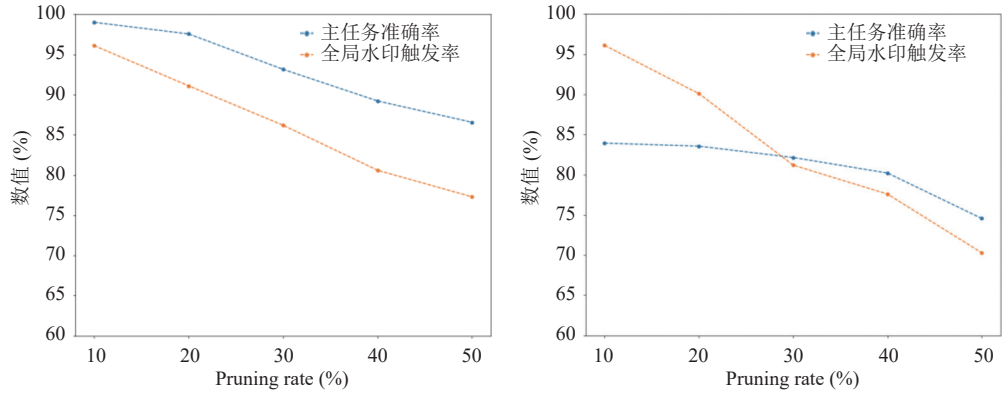
● 量化攻击是指使用更少的比特来表示参数. 在量化攻击实验中, 本文测试了 3 种不同的数据类型, float32、float16 和 int8, 结果如表 7 所示, 随着量化程度的提高, 模型的主任务准确率和全局水印触发率只会受到极其轻微的影响, 这表明本方案的全局水印是对量化攻击鲁棒的.

● 剪枝攻击实验中, 我们通过将具有最低 L1 范数的参数归零来修剪模型中的参数. 图 8 展示了剪枝率在

0–50% 之间的全局水印触发率和主任务准确率. 由图 8 可知, 全局水印在面对剪枝攻击时, 触发率会出现明显下降, 同时, 模型在主任务准确率也受到较大影响, 下降幅度超过了 10%. 由于本方案的触发样本数量较大, 只要触发率大于 50%, 都具备有力的验证效果. 考虑到模型的可用性, 这不是一次有效的攻击, 这表明本方案的全局水印对剪枝攻击是鲁棒的.

表 7　量化攻击下的主任务准确率和全局水印触发率 (%)

Model	Metric	float32	float16	int8
CNN-4 on MNIST	主任务准确率	99.47	99.47	99.39
	全局水印触发率	98.40	98.40	98.40
AlexNet on CIFAR-10	主任务准确率	86.02	85.95	85.95
	全局水印触发率	97.00	97.00	97.00

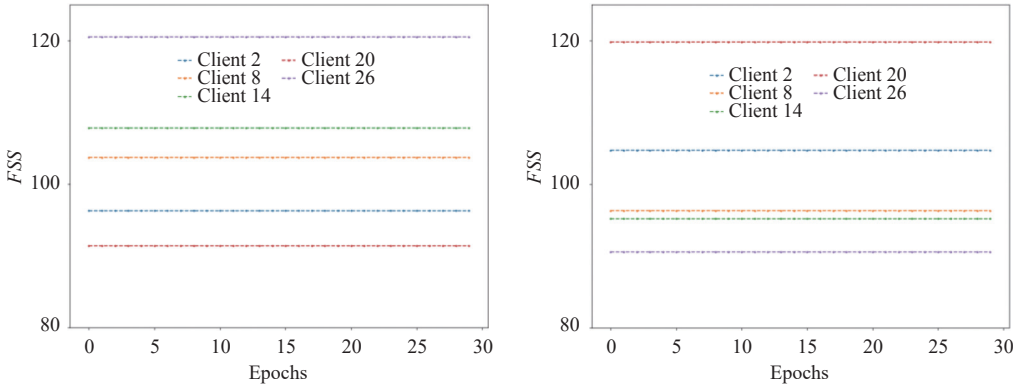


(a) CNN-4 on MNIST　(b) AlexNet on CIFAR-10

图 8　剪枝攻击下的主任务准确率和全局水印触发率变化情况

5.5.2　局部指纹鲁棒性

在微调攻击中, 本文随机选取 5 个客户端的本地模型, 并使用这些客户端对应的私有数据进行 30 轮的微调训练. 如图 9 所示, 衡量指纹的指标 *FSS* 非常稳定, 几乎不受微调攻击的影响. 这表明本方案的局部指纹是对微调攻击鲁棒的.



(a) CNN-4 on MNIST　(b) AlexNet on CIFAR-10

图 9　微调攻击下局部指纹 *FSS* 变化情况

在量化攻击中, 我们测试了 3 种不同的数据类型: float32、float16 和 int8, 结果如表 8 所示, 随着量化程度的提高, 模型在主任务准确率和 *FSS* 只会受到极其轻微的影响. 这表明本方案的局部指纹是对量化攻击鲁棒的.

表 8　量化攻击下的主任务准确率和局部指纹 *FSS*

Model	Metric	float32	float16	int8
CNN-4 on MNIST	主任务准确率 (%)	99.47	99.47	99.39
	FSS	90.30	90.30	90.30
AlexNet on CIFAR-10	主任务准确率 (%)	86.02	85.95	85.95
	FSS	94.13	94.13	94.13

在剪枝攻击中, 本文考虑只测试对 BN 层剪枝的效果, 这是因为对非 BN 层做剪枝时, 并不会影响局部指纹. 如图 10 所示, 随着剪枝率的提高, *FSS* 和主任务准确率都有所下降, 且主任务准确率下降程度更为明显, 这意味着模型已经失去了价值, 因此, 这不是一次成功的攻击. 这表明本方案的局部指纹是对剪枝攻击鲁棒的.

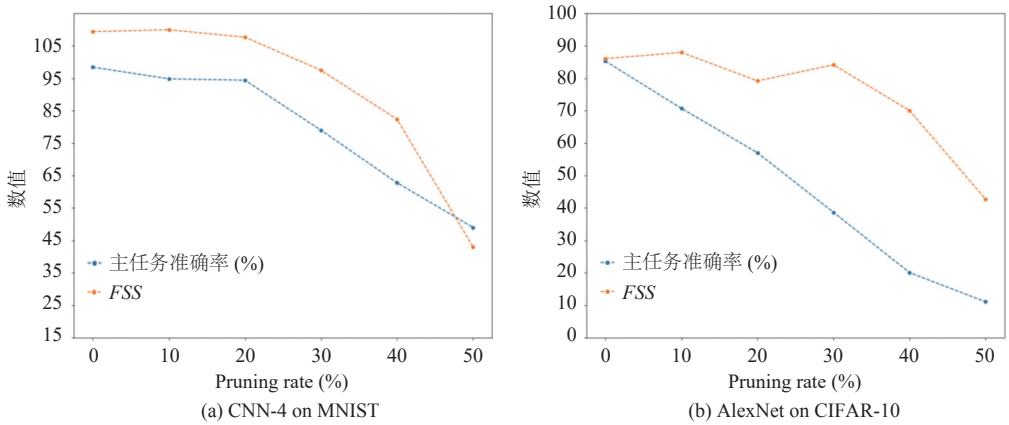


图 10　剪枝攻击下的主任务准确率和局部指纹 *FSS* 变化情况

如果攻击者事先知道指纹信息的存在, 则攻击者可以通过嵌入新的指纹信息来覆盖原有的指纹信息. 在一般假设中, 新嵌入的指纹信息不能准确知道原指纹信息及其位置, 由此可能出现两种情况: (1) 覆盖攻击发生在不同的位置. 在该情况下, 本方案保留了完美的用户识别能力; (2) 覆盖攻击发生在相同位置. 这是最糟糕的情况, 也是小概率发生的情况. 在这种情况下, 随着覆盖攻击的强度 (指纹嵌入训练轮次) 的增加, 指纹的用户识别能力逐渐下降. 单个客户端 *FSS* 变化情况如图 11. 虽然 *FSS* 会因为覆盖攻击而降低, 但追溯成功率依然达到 100%. 这是由于覆盖攻击虽然对 *FSS* 整体大小有影响, 但是对各 *FSS* 之间的相对大小影响较小, 即最大的 *FSS* 代表的成员依旧保持不变. 由此可见, 本方案对极端条件下的覆盖攻击依旧有一定抵抗能力.

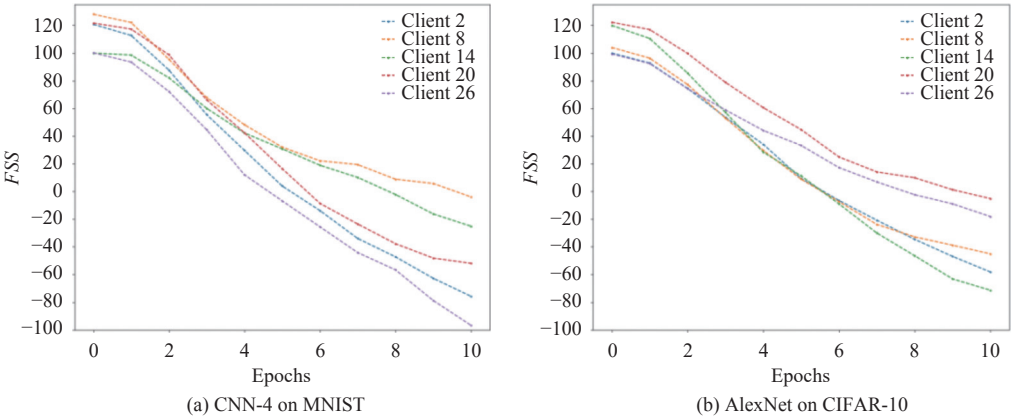


图 11　覆盖攻击下局部指纹 *FSS* 的变化情况

5.5.3 抗共谋追溯功能

共谋攻击的防御难点在于共谋成员的数量. 由于本方案通过检测并分析从模型当初提取的局部指纹, 追溯到各个共谋参与方, 所以可抵抗的共谋者数量完全依赖于选取的 ACC-BIBD 码本, 当共谋参与方数量小于识别容量时, 可以做到完美识别全部共谋参与方. 但当我们希望抗共谋参与方数量更多时, 相应的水印长度也会更长. 以 (31, 6, 1)-BIBD 码本为例, 对于不同数量的共谋参与方得到的验证准确度如表 9 所示.

表 9 不同数量共谋攻击参与方下的模型泄露追溯成功率

Number of colluders	1	2	3	4	5
成功率 (%)	100	100	100	100	100

5.6 安全性

在本方案中, 通过保持水印嵌入前后的 BN 层权重分布不出现显著改变 (水印的不可见性), 防止攻击者检测到模型中的水印信息, 满足了安全性要求. 如图 12 和图 13 所示, 给出了 CNN-4 和 AlexNet 的 BN 层权重项参数的分布直方图. 对于 AlexNet, 嵌入后的权重相似的达到 96.93%; 对于 CNN-4, 嵌入后的权重相似的达到 97.09%.

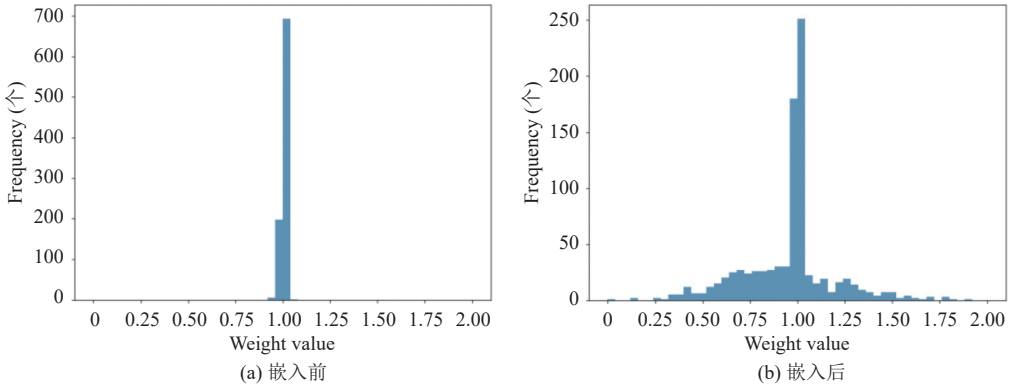


图 12 嵌入水印前后的 CNN-4 权重直方图

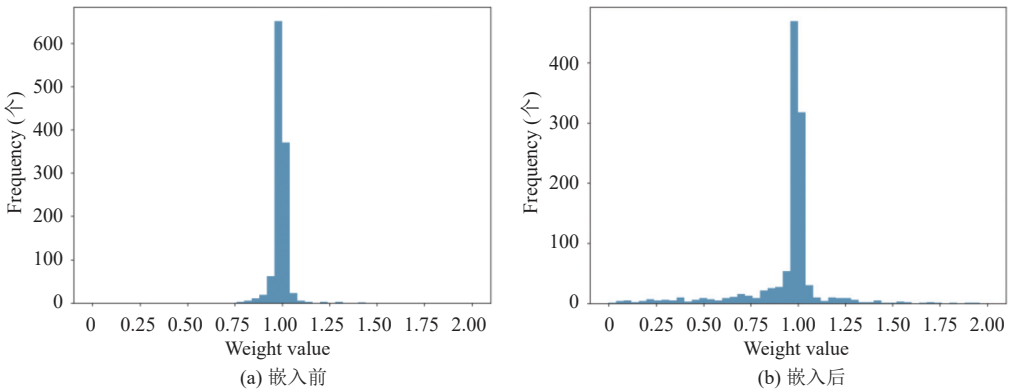
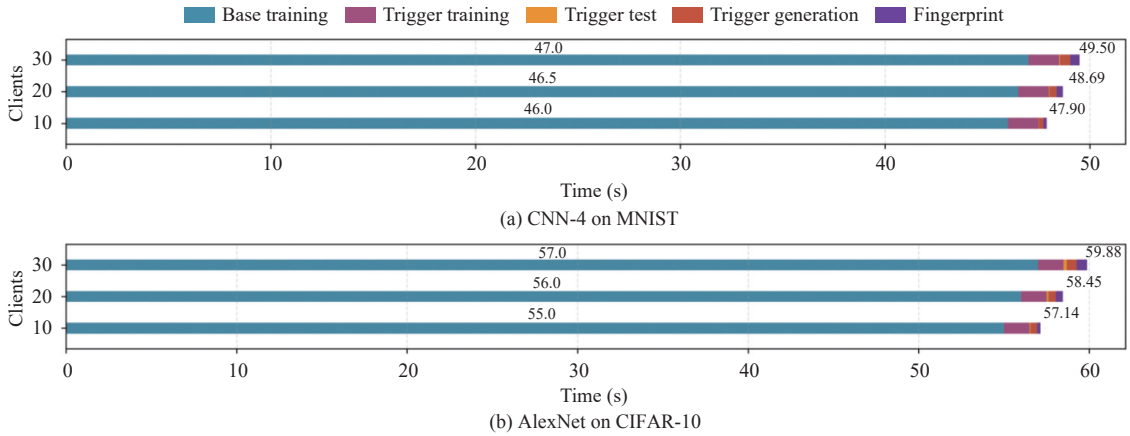


图 13 嵌入水印前后的 AlexNet 权重直方图

5.7 时间开销

实验运行配置为 Intel(R) Xeon(R) Gold 5220 CPU+NVIDIA RTX 2080Ti GPU, 选取两组模型与数据集搭配, 分别测试 10、20、30 个客户端情况下的运行时间. 如图 14 所示, 方案带来的额外开销主要为触发集生成、触发集训练、触发集测试和指纹嵌入, 其中触发集生成在整个训练过程中执行一次, 其余步骤为每个训练轮次执行一次. 方案引入的额外开销约为 4%.



(a) CNN-4 on MNIST

(b) AlexNet on CIFAR-10

图 14 不同阶段时间开销

6 总 结

本文提出了一种针对联邦学习环境的水印方案, 旨在解决模型所有权验证、模型泄露追溯和懒惰客户端检测等问题. 通过分布式后门水印嵌入和局部指纹技术, 该方案能够有效保护模型的知识产权, 并追踪模型泄露的源头. 实验结果表明, 该方案在多种攻击场景下表现出良好的鲁棒性和保真度, 能够有效抵御微调、剪枝、量化和共谋攻击等攻击手段. 此外, 该方案还能够以高准确率检测并剔除懒惰客户端, 确保联邦学习过程的公平性. 未来的研究可以进一步优化水印的生成和嵌入方式, 提高水印嵌入的效率, 以应对更复杂的攻击场景和应用需求. 总体而言, 本文提出的方案为联邦学习环境中的模型版权保护提供了一种有效的解决方案, 有助于构建更加安全、公平的联邦学习生态系统.

References

[1] 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.

[2] Deng J, Dong W, Socher R, Li LJ, Li K, Li FF. ImageNet: A large-scale hierarchical image database. In: Proc. of the 2009 IEEE Conf. on Computer Vision and Pattern Recognition. Miami: IEEE, 2009. 248–255. [doi: 10.1109/CVPR.2009.5206848]

[3] Li QB, Wen ZY, Wu ZM, Hu SX, Wang NB, Li Y, Liu X, He BS. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. IEEE Trans.on Knowledge and Data Engineering, 2023, 35(4): 3347–3366. [doi: 10.1109/TKDE.2021.3124599]

[4] Fraboni Y, Vidal R, Lorenzi M. Free-rider attacks on model aggregation in federated learning. In: Proc. of the 24th Int'l Conf. on Artificial Intelligence and Statistics. San Diego: PMLR, 2021. 1846–1854.

[5] Lin JR, Du M, Liu J. Free-riders in federated learning: Attacks and defenses. arXiv:1911.12560, 2019.

[6] Uchida Y, Nagai Y, Sakazawa S, Satoh SI. Embedding watermarks into deep neural networks. In: Proc. of the 2017 ACM on Int'l Conf. on Multimedia Retrieval. Bucharest: ACM, 2017. 269–277. [doi: 10.1145/3078971.3078974]

[7] Chen HL, Rouhani BD, Fu C, Zhao JS, Koushanfar F. DeepMarks: A secure fingerprinting framework for digital rights management of deep learning models. In: Proc. of the 2019 Int'l Conf. on Multimedia Retrieval. Ottawa: ACM, 2019. 105–113. [doi: 10.1145/3323873.3325042]

[8] Luo YL, Li YZ, Qin S, Fu Q, Liu JX. Copyright protection framework for federated learning models against collusion attacks. Information Sciences, 2024, 680: 121161. [doi: 10.1016/j.ins.2024.121161]

[9] Fan LX, Ng KW, Chan CS. Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 424.

[10] Li GB, Li S, Qian ZX, Zhang XP. Encryption resistant deep neural network watermarking. In: Proc. of the 2022 IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP 2022). Singapore: IEEE, 2022. 3064–3068. [doi: 10.1109/ICASSP43922.2022.

9746461]

[11] Liang JC, Wang R. FedCIP: Federated client intellectual property protection with traitor tracking. arXiv:2306.01356.

[12] Darvish Rouhani B, Chen HL, Koushanfar F. DeepSigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In: Proc. of the 24th Int'l Conf. on Architectural Support for Programming Languages and Operating Systems. Providence: ACM, 2019. 485–497. [doi: 10.1145/3297858.3304051]

[13] Zhang J, Chen DD, Liao J, Zhang WM, Hua G, Yu NH. Passport-aware normalization for deep model protection. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 1896.

[14] Adi Y, Baum C, Cisse M, Pinkas B, Keshet J. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In: Proc. of the 27th USENIX Conf. on Security Symp. Baltimore: USENIX Association, 2018. 1615–1631.

[15] Tekgul BGA, Xia YX, Marchal S, Asokan N. WAFFLE: Watermarking in federated learning. In: Proc. of the 2021 40th Int'l Symp. on Reliable Distributed Systems (SRDS). Chicago: IEEE, 2021. 310–320. [doi: 10.1109/SRDS53918.2021.00038]

[16] Yang WY, Shao S, Yang Y, Liu XY, Liu XM, Xia ZH, Schaefer G, Fang H. Watermarking in secure federated learning: A verification framework based on client-side backdooring. ACM Trans.on Intelligent Systems and Technology, 2023, 15(1): 5. [doi: 10.1145/3630636]

[17] Lukas N, Zhang YX, Kerschbaum F. Deep neural network fingerprinting by conferrable adversarial examples. In: Proc. of the 9th Int'l Conf. on Learning Representations. OpenReview.net, 2019.

[18] Zhang JL, Gu ZS, Jang JY, Wu H, Stoecklin MP, Huang HQ, Molloy I. Protecting intellectual property of deep neural networks with watermarking. In: Proc. of the 2018 Asia Conf. on Computer and Communications Security. Incheon: ACM, 2018. 159–172. [doi: 10.1145/3196494.3196550]

[19] Lansari M, Bellafqira R, Kapusta K, Thouvenot V, Bettan O, Coatrieux G. When federated learning meets watermarking: A comprehensive overview of techniques for intellectual property protection. Machine Learning and Knowledge Extraction, 2023, 5(4): 1382–1406. [doi: 10.3390/make5040070]

[20] Li X, Deng TP, Xiong JB, Jin B, Lin J. Federated learning watermark based on model backdoor. Ruan Jian Xue Bao/Journal of Software, 2024, 35(7): 3454–3468 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6914.htm [doi: 10.13328/j.cnki.jos.006914]

[21] Li BW, Fan LX, Gu HL, Li J, Yang Q. FedIPR: Ownership verification for federated deep neural network models. IEEE Trans.on Pattern Analysis and Machine Intelligence, 2023, 45(4): 4521–4536. [doi: 10.1109/TPAMI.2022.3195956]

[22] Nie HW, Lu SF. FedCRMW: Federated model ownership verification with compression-resistant model watermarking. Expert Systems with Applications, 2024, 249: 123776. [doi: 10.1016/j.eswa.2024.123776]

[23] Shao S, Yang WY, Gu HL, Qin Z, Fan LX, Yang Q, Ren K. FedTracker: Furnishing ownership verification and traceability for federated learning model. IEEE Trans.on Dependable and Secure Computing, 2025, 22(1): 114–131. [doi: 10.1109/TDSC.2024.3390761]

[24] Chen JY, Li MJ, Cheng Y, Zheng HB. FedRight: An effective model copyright protection for federated learning. Computers & Security, 2023, 135: 103504. [doi: 10.1016/j.cose.2023.103504]

[25] Xu Y, Tan YL, Zhang C, Chi K, Sun P, Yang WY, Ren J, Jiang HB, Zhang YX. RobWE: Robust watermark embedding for personalized federated learning model ownership protection. arXiv:2402.19054, 2024.

[26] Wagner NR. Fingerprinting. In: Proc. of the 1983 IEEE Symp. on Security and Privacy. Oakland: IEEE, 1983. 18–18. [doi: 10.1109/SP.1983.10018]

[27] Wu M, Trappe W, Wang ZJ, Liu KJR. Collusion-resistant multimedia fingerprinting: A unified framework. In: Proc. of the Security, Steganography, and Watermarking of Multimedia Contents VI. San Jose: SPIE, 2004. 748–759.

[28] Yu YS, Lu HW, Chen XS, Zhang ZG. Group-oriented anti-collusion fingerprint based on BIBD code. In: Proc. of the 2nd Int'l Conf. on E-business and Information System Security. Wuhan: IEEE, 2010. 1–5. [doi: 10.1109/EBISS.2010.5473676]

[29] Fereidooni H, Marchal S, Miettinen M, Mirhoseini A, Möllering H, Nguyen TD, Rieger P, Sadeghi AR, Schneider T, Yalame H, Zeitouni S. SAFELearn: Secure aggregation for private FEderated learning. In: Proc. of the 2021 IEEE Security and Privacy Workshops (SPW). San Francisco: IEEE, 2021. 56–62. [doi: 10.1109/SPW53761.2021.00017]

[30] Han S, Pool J, Tran J, Dally WJ. Learning both weights and connections for efficient neural network. In: Proc. of the 29th Int'l Conf. on Neural Information Processing Systems. Montreal: MIT Press, 2015. 1135–1143.

[31] Wang ZJ, Wu M, Zhao H, Liu KJR, Trappe W. Resistance of orthogonal Gaussian fingerprints to collusion attacks. In: Proc. of the 2003 IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2003). Hong Kong: IEEE, 2003. IV–724. [doi: 10.1109/ICASSP.2003.1202745]

[32] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proc. of the 32nd Int'l Conf. on Machine Learning. Lille: JMLR.org, 2015. 448–456

[33] Ong DS, Seng CC, Ng KW, Fan LX, Yang Q. Protecting intellectual property of generative adversarial networks from ambiguity attacks.

In: Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). Nashville: IEEE, 2021. 3629–3638. [doi: 10.1109/CVPR46437.2021.00363]

[34] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc. of the IEEE, 1998, 86(11): 2278–2324. [doi: 10.1109/5.726791]

[35] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 2017, 60(6): 84–90. [doi: 10.1145/3065386]

[36] Krizhevsky A. Learning multiple layers of features from tiny images. 2009. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

附中文参考文献

[20] 李璇, 邓天鹏, 熊金波, 金彪, 林劼. 基于模型后门的联邦学习水印. 软件学报, 2024, 35(7): 3454–3468. http://www.jos.org.cn/1000-9825/6914.htm [doi: 10.13328/j.cnki.jos.006914]

作者简介

孙友欣, 硕士生, CCF 学生会员, 主要研究领域为人工智能版权保护.

田有亮, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为博弈论, 密码学与安全协议, 大数据隐私保护.