

Robust-PFedproto:基于个性化层的鲁棒联邦原型学习方法

徐明迪¹, 李正潇¹, 王梓行², 靳朝阳¹

(1. 武汉数字工程研究所, 湖北 武汉 430074; 2. 公安部第三研究所, 上海 201204)

摘要: 联邦学习 (federated learning, FL) 是一种高效的分布式机器学习框架, 其核心优势在于训练数据可保留在网络中的远程客户端。然而, 该方法面临两大关键挑战。其一, 不同客户端的数据分布通常呈现异构性, 这种数据分布的差异性会显著降低整体训练效率; 其二, 由于中央服务器无法获取远程客户端的真实数据, 恶意客户端可能上传恶意模型更新, 进而对全局训练的准确性和效率产生负面影响。为解决数据异构性问题, 提出了PFedproto框架, 该框架在原型学习的基础上引入了个性化层。个性化层位于每个客户端的决策层之前, 其参数可通过学习优化, 使客户端模型能够更好地适配本地数据集, 从而提升模型在本地任务中的预测准确率。在此基础上, 进一步构建了3阶段防御方案Robust-PFedproto, 旨在增强PFedproto框架的鲁棒性, 以有效抵御数据投毒攻击。在4个真实图像分类数据集上开展了实验, 结果表明PFedproto框架能够更高效地缓解数据异构性的影响, Robust-PFedproto方案在有效应对数据异构性的同时, 对数据投毒攻击展现出良好的鲁棒性。

关键词: 联邦学习; 原型学习; 核密度估计; 中毒攻击

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2025032

Robust-PFedproto: robust federated prototype learning based on personalized layers

XU Mingdi¹, LI Zhengxiao¹, WANG Zihang², JIN Chaoyang¹

1. Wuhan Institute of Digital Engineering, Wuhan 430074, China

2. The Third Research Institute of the Ministry of Public Security, Shanghai 201204, China

Abstract: Federated learning (FL), a distributed machine learning framework, was recognized for retaining training data on remote clients. However, two critical challenges were identified. First, heterogeneous data distributions were commonly observed across clients, which significantly degraded overall training efficiency. Second, the central server could't access authentic remote client data, allowing adversarial clients to upload malicious model updates, thus negatively affecting global training accuracy and efficiency. To address data heterogeneity, the PFed-

收稿日期: 2024-12-16; 修回日期: 2025-05-05

基金项目: 国防科研基金项目 (JCKY2022211C007, JCKY2021206B104)

Foundation Item: National Defense Basic Scientific Research Program (JCKY2022211C007, JCKY2021206B104)

引用格式: 徐明迪, 李正潇, 王梓行, 等. Robust-PFedproto:基于个性化层的鲁棒联邦原型学习方法[J]. 网络与信息安全学报, 2025, 11(3): 67-80.

Citation Format: XU M D, LI Z X, WANG Z H, et al. Robust-PFedproto: robust federated prototype learning based on personalized layers[J]. Chinese Journal of Network and Information Security, 2025, 11(3): 67-80.

*第二十七届中国科协年会“AI时代网络技术创新”专题

proto framework was proposed, incorporating personalized layers based on prototype learning. These layers, positioned before each client's decision layer, were optimized to enhance client model adaptation to local datasets and improve localized task prediction accuracy. Building on this framework, a three-stage defense scheme named RobustPFedproto was developed to strengthen the PFedproto framework's robustness against data poisoning attacks. Experiments were conducted on four real-world image classification datasets. Results demonstrate that the PFedproto framework effectively mitigates data heterogeneity impacts. Additionally, the Robust-PFedproto scheme not only efficiently handles data heterogeneity but also shows strong robustness against data poisoning attacks.

Keywords: federated learning, prototype learning, kernel density estimation, poisoning attack

0 引言

联邦学习 (federated learning, FL) 是一种先进的隐私保护技术, 可以在多个客户端之间实现协作式机器学习, 同时保护数据隐私。FL 的核心思想是在不共享原始数据的情况下, 通过聚合多个客户端的本地训练模型, 生成一个全局共享的机器学习模型。训练伊始, 中央服务器生成一个全局初始模型, 并将其分发给所有参与训练的客户端。随后, 在每一轮训练中, 客户端使用本地数据集对接收的全局模型进行训练, 完成模型参数的更新后, 将更新后的参数上传至中央服务器。中央服务器收集所有客户端上传的模型参数, 运用聚合算法生成新的全局模型。这一过程循环往复, 直至全局模型达到收敛状态。

尽管 FL 具有诸多优点, 但数据的异构性及安全攻击威胁 (如数据投毒攻击、隐私推理攻击) 使 FL 仍然面临严峻挑战。

为解决数据异构性问题, 研究者提出了一系列个性化联邦学习解决方案。部分研究将迁移学习引入联邦学习框架^[1-3], 利用全局模型预训练参数初始化局部模型, 避免模型从零学习, 却难以规避灾难性遗忘问题^[4]; 联邦元学习^[5-8]通过在多个客户端上进行元学习训练, 获得本地的个性化模型参数, 以适应本地任务需求。知识蒸馏技术^[9-11]帮助联邦学习将集合的知识进行压缩, 以便部署到客户端中, 并允许客户端独立设计自己的网络。另有学者通过定制本地专属模型或网络层^[12-14]实现个性化目的。Fed-Proto^[15]是一种具有隐私保护且通信成本极低的方案, 它通过传输原型 (数据类别的均值表示) 而非模型参数或原始数据, 极大减少了数据异构性。然而, 该方案仍存在局限性: 其一, 服

务器下发的聚合原型对于每个客户端是相同的, 并没有充分消除客户端本地的数据分布对模型的影响, 导致个性化模型缺乏足够的泛化性; 其二, 恶意客户端可以通过注入损坏的原型实施投毒攻击, 从而降低模型在目标任务上的准确性, 增加错误分类的概率。

针对上述问题, 本文提出了一种基于个性化层的跨异构客户端鲁棒联邦原型学习方法 Robust-PFedproto。首先在原型学习基础上构造更加适应本地数据集的 FL 模型结构 PFedproto, 通过在本地模型决策层前端添加个性化层, 使本地模型能够更好地适应本地数据集。具体而言, 客户端模型均采用包含个性化层的拓展模型, 在训练阶段, 客户端本地数据首先被拓展模型隐藏层映射为高维特征, 然后由个性化层进行针对性转换, 最终通过全连接层完成预测输出。此外, 本文设计了一种 3 阶段防御机制, 用于抵御 PFedproto 中可能存在的数据投毒攻击。第一阶段, 运用核密度估计技术, 对远程客户端上传的原型数据进行统计特征分析; 第二阶段, 通过阈值设定策略, 识别恶意更新和干净更新; 第三阶段, 根据客户端上传恶意原型的行为, 基于客户端的异常行为记录, 剔除存在投毒风险的恶意节点。

本文的核心贡献可总结为以下 3 点。

1) 提出基于个性化层的联邦原型学习方法 PFedproto, 通过结构创新显著降低数据异构性对模型准确率的负面影响。

2) 在 PFedproto 基础上叠加 3 阶段防御体系, 形成鲁棒性的个性化层联邦原型学习方法 Robust-PFedproto, 抵御数据投毒攻击。

3) 通过在多个主流图像数据集上的对比实验, 证实 Robust-PFedproto 可以有效减少数据异

构性对联邦学习任务的影响,其鲁棒性能够有效抵御数据投毒攻击。

1 相关技术及背景

本章将介绍联邦学习、联邦原型学习、个性化层联邦学习及投毒攻击与防御的相关知识及背景。

1.1 联邦学习

2016年,谷歌首次提出联邦学习概念,旨在解决安卓系统用户本地模型更新问题。其设计目标是在保护大数据交互中的信息安全、保护终端数据与个人数据隐私的前提下,联合多方参与者与计算节点开展联合训练^[16]。

联邦学习系统架构通常包含中央服务器与客户端两部分。训练初期,中央服务器初始化全局模型并分发给各远程客户端;客户端基于本地数据进行模型训练后,将更新的参数上传至服务器;服务器聚合生成全局模型并再次下发。在第 r 轮,客户端 k 上传本地模型 θ_k^r ,服务器将其聚合为全局原型后分发,即 $\theta^{r+1} = \frac{1}{K} \sum_{k=1}^K \theta_k^r$, K 代表客户端总数。上述流程重复执行,直至全局模型达到收敛状态。

1.2 联邦原型学习

在联邦学习场景中,当客户端之间存在数据分布非独立同分布、硬件计算能力参差不齐或模型架构差异显著等异构问题时,传统联邦学习机制易出现训练效率低、模型收敛缓慢等瓶颈。为突破这一困境,学界开始探索在联邦学习中引入原型学习概念。Kim等^[17]将原型蒸馏技术应用于无监督学习场景,并利用归一化流构建局部一类分类器。Michieli等^[18]利用原型边缘优化联邦学习的视觉特征表示。Tan等^[15]提出基于原型聚合的联邦学习方法FedProto。在该框架下,客户端不再直接上传完整模型参数,而是生成并上传本地数据的原型表示。这些原型是通过同时对同类样本特征进行平均转换得到的,能够捕捉并表征各数据集每个类别的特征。

具体来说,FedProto的工作流程开始于客户端利用函数 $f_i(\phi_i; x)$ 将本地数据 x 转化为特征表示 rep , rep 聚合为原型。远程客户端 i 中对应标签 t 的原型可形式化表示为:

$$P(t)_i = \frac{1}{|D_{i,t}|} \sum_{x \in D_{i,t}} f_i(\phi_i; x) \quad (1)$$

其中, $D_{i,t}$ 表示客户端 i 拥有的标签为 t 的数据集合。

接下来,客户端将得到的特征表示 rep 输入决策层,生成对数据 x 的预测结果。与此同时,客户端根据数据的类别标签对同类别的特征表示进行聚合操作,生成能够表征该类别核心特征的本地原型 P_k^r ,这些本地原型随后被上传至中央服务器。

中央服务器在收集到所有客户端上传的本地原型后,对它们进行聚合,形成的下一轮全局原型可表示为:

$$\bar{P}(j)^{r+1} = \frac{1}{|C_t|} \sum_{i \in C_t} |D_{i,t}| \cdot C_t \cdot P(t)_i^r \quad (2)$$

其中, C_t 表示包含标签 t 的客户端集合, r 表示当前轮次。

这一聚合过程不但能够充分利用不同客户端的知识,而且通过平均化操作能够显著减少个别客户端因数据异常或计算误差引起的偏差与噪声干扰。

聚合完成后,中央服务器将生成的全局原型作为新一轮训练的参考标准,同步下发至所有客户端。

在客户端,每个客户端通过度量当前全局原型 P^r 和自身计算的本地原型 P_i^r 之间的距离,动态调整本地损失函数 $\mathcal{L}$,具体可表述为

$$\mathcal{L}(\mathcal{D}_i, \omega_i) = \mathcal{L}_S(F_i(\omega_i; x), y) + \lambda \mathcal{L}_R(\bar{P}(j), P(j)_i) \quad (3)$$

通过这一机制,客户端动态调整本地模型的训练方向。这一过程既推动了本地模型性能的提升,又确保了本地模型与全局模型的参数一致性。

1.3 个性化层联邦学习

个性化层联邦学习作为一种参数分离的联邦学习方法,目前主要存在以下两种实现方法。

1) FedPer^[12]: 当客户端完成本地训练后,将本地模型的浅层参数与深层参数进行分离,仅上传浅层参数至中央服务器,而深层参数保留在本地。这一设计使客户端模型能够更有效地捕捉本

地训练数据的深层特征, 从而提升聚合后全局模型对本地数据集的适应性, 显著降低数据异构性的影响。

2) Privatefl^[19]: 与 FedPer 不同, Privatefl 为客户端本地模型增添了一个个性化输入层, 如图 1 所示。个性化输入层在初始化时被设定为恒等变换, 其参数会随着客户端每一轮本地训练动态更新。在上传客户端模型参数时, 个性化输入层参数被保留在本地, 其主要作用是通过自适应调整本地模型的数据输入分布, 增强模型对异构数据的鲁棒性。

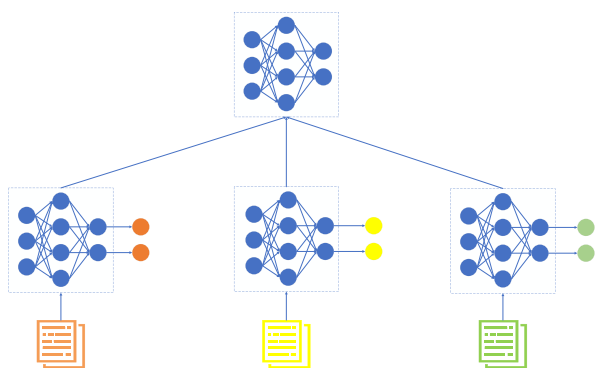


图 1 个性化层联邦学习

Figure 1 Personalized layer federated learning

1.4 投毒攻击

相较于传统集中式学习模型, 联邦学习框架的客户端与中央服务器的分布式特性与通信链路特征, 显著扩大了潜在攻击面, 使得针对该系统的安全威胁持续升级。在此类威胁中, 数据投毒攻击尤为突出, 攻击者通过将恶意目标转化为优化问题, 借助有毒的本地模型更新实现对全局模型的操纵。具体而言, 攻击者可通过向系统注入新的恶意客户端或劫持现有良性客户端使其生成包含投毒信息的本地训练更新。

在各类攻击策略中, 基于权衡的攻击机制尤为隐蔽, 其核心在于引导联邦学习模型错误地收敛至非目标标签, 同时通过控制目标标签的扰动幅度来降低攻击行为的可检测性。

Data_Gen^[20]是一种基于生成对抗网络 (generative adversarial network, GAN) 的投毒数据生成方法, 该方法依赖迭代更新的全局模型参数, 动态生成针对特定目标样本的投毒数据。

MPAF^[21]基于假客户端实施非针对性模型投毒攻击, 其核心特征在于仅利用训练过程中接收的全局模型参数即可完成攻击链构建, 摆脱对联邦学习系统先验知识的依赖。此外, 非目标性数据投毒攻击以降低模型整体性能为目标, 通过操纵联邦学习模型在测试样本上产生高错误率, 使模型无法适用于进一步的学习任务, 且此类攻击不针对特定分类标签。

在机器学习 (machine learning, ML) 攻击研究领域, 依据简约性原理, 成功的攻击者往往追求对学习模型产生高效且显著的影响。这意味着, 中毒的本地更新与良性更新在攻击目标上必然存在显著差异。因此, 若防御者能够设计有效的机制区分干净更新与有毒更新, 将大幅缓解 FL 系统面临的投毒攻击威胁。

在本文中, 攻击者被设定为能够发动两种类型的投毒攻击: 数据投毒攻击, 即通过操纵客户端本地数据 (如标签翻转) 实施攻击; 模型投毒攻击, 即直接篡改客户端发送的模型更新参数。同时, 攻击者的能力受到以下限制: 最多注入不超过总客户端数量 40% 的恶意客户端 (即 $M \leq 0.4N$); 攻击目标为操纵特定目标标签, 而非无差别破坏模型性能; 攻击者在模型投毒攻击中需要保持模型的收敛性以规避检测; 攻击者可利用数据分布不平衡性增强攻击效果, 但每个恶意客户端的攻击预算有限。这些设定既反映了攻击者在联邦学习环境中的实际能力, 也为防御机制的设计提供了重要依据。

1.5 防御策略

在联邦学习领域, 防御策略的研究主要聚焦于构建健壮的聚合器以抵御潜在攻击。这类聚合器的核心思想是估算接收到的局部模型更新的“中心”, 而非采用易受攻击的简单加权平均法。现有许多防御研究基于一个关键假设: 所有客户端的数据满足独立同分布。在此假设下, 良性客户端的局部模型更新被视为真实梯度或模型权重的近似。

基于这一理念, 研究者提出了 Krum^[22]、Me-doid^[23]等健壮的机器学习算法, 通过筛选具有代表性的客户端更新来估计全局模型的真实中心。尽管这些方法在统计上对对抗性攻击具有

一定的弹性,但它们仅利用局部更新的一小部分可能导致全局模型产生偏差。此外,这些防御方案在面对非独立同分布数据集时明显缺乏适应性。

为应对数据异构性挑战,GeoMed^[24]、修剪平均值^[25]等方法被提出,这类方法基于客户端模型更新估计中心,不预先区分恶意与正常更新,能够在一定程度上减轻恶意攻击的影响,但无法完全消除威胁。Li^[26]等通过在成本函数中引入额外的L1范数正则化,增强分布式学习对拜占庭攻击的鲁棒性。HeteroSAg^[27]设计了一种分组策略,通过组内更新保证全局模型的稳定性。Byrd-SAGA^[28]结合分布式SAGA算法与几何中值,实现拜占庭攻击场景下的稳健联邦优化。然而,上述方法均难以抵御有针对性的攻击。

在联邦学习领域,部分学者致力于为聚合器构建安全的聚合环境。SEAR^[29]借助Intel SGX的远程验证功能实现模型安全上传,并在可信执行环境中完成聚合,保障客户端私有模型的安全性。PEFL^[30]采用同态加密技术,允许服务器直接对加密数据执行计算。Yang等^[31]基于区块链技术构建B-FL架构,结合全局聚合算法与拜占庭容错共识协议,为联邦学习过程提供双重安全防护。CREAT^[32]提出一种融合区块链技术的联邦学习压缩算法,用于边缘计算中的内容缓存,以预测并提高缓存命中率,同时确保数据安全。AWFL-BC^[33]采用区块链和自适应权重来实现鲁棒性聚合。Li等^[34]设计基于盲化技术与Paillier同态加密的模型参数安全聚合方法,有效抵抗各类攻击对聚合过程的干扰。

在异常检测方向,多种技术方案为攻击识别提供了新思路。光谱异常检测^[35-36]将正常数据实例和异常数据实例都嵌入低维潜在空间,通过查看重建误差识别异常实例。ShieldFL策略^[37]利用双陷门同态加密(two-trapdoor homomorphic encryption)识别加密的恶意梯度。FoolsGold^[38]引入余弦相似度度量方法来区分恶意更新与正常更新。核密度估计(kernel density estimation, KDE)^[39-40]是一种非参数估计方法,有较好的异常检测效果。LoMar^[41]使用KDE分析模型参数的统计特

性,评估客户端上传的更新为恶意更新的可能性,该方法在中央服务器不知道干净训练数据分布的情况下是行之有效的。FLDetector^[42]基于Gap统计学确定客户端聚类的最优数量,在提高恶意客户端检测准确率的同时,降低诚实客户端的误判率。DPFLA^[43]基于掩码最后层神经元的梯度特征分布抵御数据投毒攻击,使用可移除掩码技术在保护敏感数据的同时,确保奇异值分解(singular value decomposition, SVD)结果中完全剔除污染数据,避免模型性能损失。FL-Defender^[44]通过计算参与者最后一层梯度的角度相似性,利用主成分分析(principal component analysis, PCA)压缩相似性向量,并根据向量与质心的偏差对客户端更新进行动态加权,实现对恶意更新的抑制。

在本文中,防御者以保护联邦学习系统免受投毒攻击为核心目标。防御者的能力设定如下:通过分析客户端模型更新参数,检测并移除恶意更新,同时保留干净更新,以维护联合模型的性能;防御者采用基于核密度估计的局部特征分析方法,无须访问客户端的原始数据,严格遵循联邦学习的隐私保护原则;通过统计分析和优化阈值选择,在恶意客户端数量未知的场景下,有效区分恶意更新和干净更新;能够抵御不同类型的攻击(如数据投毒攻击和模型投毒攻击),并且在数据分布不平衡的情况下仍能保持较高的防御性能。

2 基于个性化层的联邦原型学习方案

本节首先介绍个性化层的实现方式,然后介绍如何使用个性化层对Fedproto方案进行优化,形成新的联邦学习方案PFedproto,以更好地抵御数据异构性对联邦学习的影响。

2.1 个性化层

FedPer将模型的最后几层(包含分类器)设为个性化层。由于其未对服务器下发的全局模型结构进行修改,因此作者未对个性化层的具体架构展开详细分析。与之不同,PrivateFL将个性化层初始化为可学习的恒等变换层,其参数通过每一轮训练动态学习更新。一般而言,基于深度学习的模型由表示层和决策层两部分构成。表示层将输入从原始特征空间转换至嵌入空间;决策层

针对具体学习任务完成分类决策。考虑到原型学习上传的是客户端本地数据映射后的高维向量，因此本文对原型学习的决策层进行改进，即在决策层前新增一个个性化层，如图2所示。该个性化层被初始化为与表示层最后一层维度相同的恒等变换层，其参数可以随着客户端本地模型的每轮训练更新而调整，从而更高效地适配本地数据集。

2.2 优化目标

传统联邦学习的优化目标是每个客户端通过最小化损失函数更新模型，以缩小本地样本标签与预测值之间的偏差。假设 θ_k 表示客户端 k 的本地模型， θ_k^{rep} 表示客户端 k 本地模型的表示层， θ_k^{dec} 表示客户端 k 本地模型的决策层， D_k 表示客户端 k 的本地数据集。对于客户端 k ，其优化目标为找到一个 θ^* ，使得

$$\theta^* = \arg \min_{\theta} \frac{1}{|D_k|} \sum_{n=1}^{|D_k|} l(y^{(n)} f(x^{(n)}; \theta_k^{\text{rep}} + \theta_k^{\text{dec}})) \quad (4)$$

其中， $(x^{(n)}, y^{(n)})$ 表示 D_k 中的一个训练样本，

$l(y^{(n)} f(x^{(n)}; \theta))$ 表示损失函数，“+”为表示层和决策层的组合。

在 PFedproto 框架中，客户端模型由表示层、个性化层和决策层 3 部分构成。此时，客户端 k 的优化目标扩展为训练表示层、个性化层和决策层这 3 个模块的参数。PFedproto 的目标是解决分布式网络中的联合优化问题，其应用基于原型的通信机制，促使本地模型将其原型与其他本地模型对齐的同时，最小化所有客户端本地学习任务的损失总和 ($\theta_k^{\text{sum}} = \theta_k^{\text{rep}} + \theta_k^p + \theta_k^{\text{dec}}$)。具体而言，跨异构客户端的联邦原型学习目标可以表述为：

$$\arg \min_{\{\bar{P}(j)\}_{j=1}^{|P|}} \sum_{i=1}^m \frac{|D_i|}{N} \mathcal{L}_S(f_i(\theta_k^{\text{sum}}; x), y) + \lambda \cdot \sum_{i=1}^{|P|} \sum_{j=1}^m \frac{|D_{ij}|}{N_j} \mathcal{L}_R(\bar{P}_i(j), P_i(j)) \quad (5)$$

其中， $\mathcal{L}_S$ 表示监督学习损失，如式(2)所定义； $\mathcal{L}_R$ 为正则化项，用来度量局部原型 $P_i(j)$ 和对应的全局原型 $\bar{P}_i(j)$ 之间的距离（本文使用 L_2 距

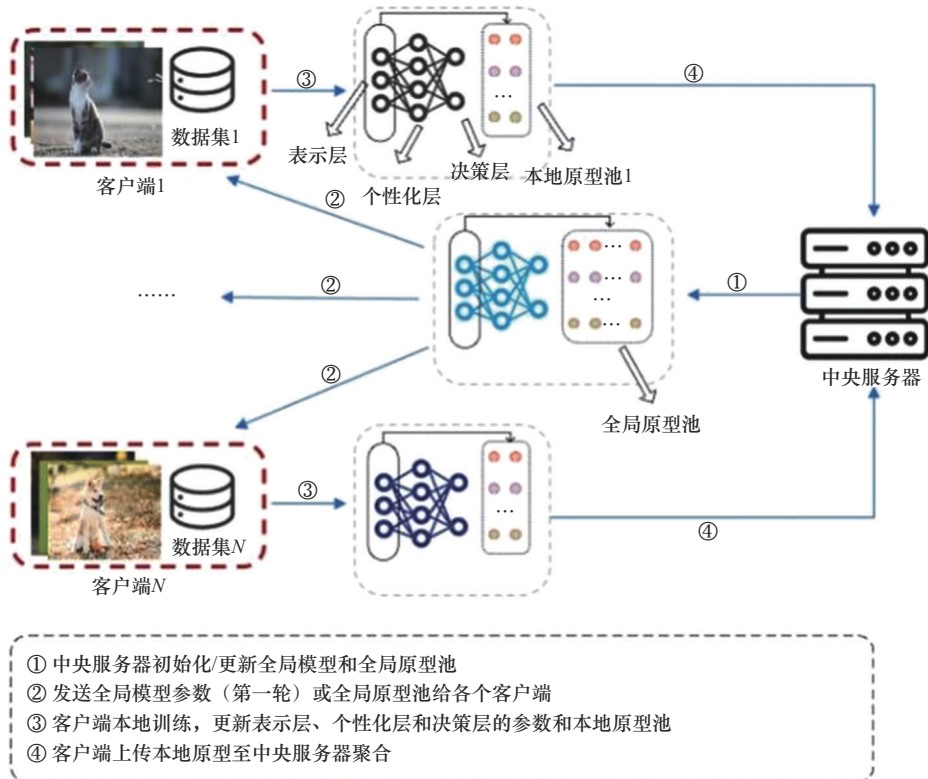


图2 基于个性化层的联邦原型学习

Figure 2 Federated prototype learning based on personalized layers

离); N 表示所有客户端实例的总数; N_j 表示所有客户端上属于类 j 的实例数量。

2.3 个性化层联邦原型学习方案流程

考虑到参与客户端的数据与模型异构性, 每个客户端的最优模型参数存在差异。这意味着基于梯度的通信难以充分向各客户端提供有效信息。然而, 相同的标签空间允许参与客户端共享同一嵌入空间, 且通过按类别聚合原型, 能够在异构客户端之间实现信息的高效交换。个性化层联邦原型学习方案主要包含全局原型聚合与本地模型更新两个步骤, 具体如算法1所示。

算法1 PFedproto 训练过程

输入 D_i

服务器执行:

- 1) **for** each round **do**
- 2) **for** each client **do**
- 3) 收集客户端原型 P_i
- 4) **end for**
- 5) 利用式(6)聚合全局原型 $\bar{P}_i$
- 6) 下发全局原型
- 7) **end for**

客户端 i 执行:

- 1) **for** each local epoch **do**
- 2) **for** batch **do**
- 3) 收集全局原型 $\bar{P}_i$
- 4) 利用式(5)优化本地模型
- 5) 计算本地原型 P_i 并上传
- 6) **end for**
- 7) **end for**

全局原型聚合: 给定类别 j , 服务器收集所有拥有该类别标签的客户端上传的本地原型, 通过原型聚合操作生成类 j 的全局原型 $\bar{P}_i(j)$, 如式(6)所示。

$$\bar{P}_i(j) = \frac{1}{|C_j|} \sum_{i \in c_j} \frac{|D_{ij}|}{C_j} P_i(j) \quad (6)$$

其中, $P_i(j)$ 表示来自客户端 i 的类 j 的原型, C_j 为包含标签 j 的客户端集合。

本地模型更新: 客户端需要更新本地模型以在客户端之间生成一致的原型。需要注意的是, 在本地模型更新阶段, 客户端首先利用式(5)优化本地模型的参数, 然后上传表示层

生成的原型, 而个性化层的参数保留在每个客户端本地。

$$\mathcal{L}(D_i, \theta_k^{\text{sum}}) = \mathcal{L}_S(f_i(\theta_k^{\text{sum}}; x), y) + \lambda \cdot \mathcal{L}_R(\bar{P}(j), P(j)_i) \quad (7)$$

在训练完成后, 对于一个新的样本 x , 客户端采用基于个性化层的原型推理机制生成模型预测结果 y , 即使用样本 x 的所有原型和 x 在表示层映射后的原型经过个性化处理后的值的最小 L_2 距离进行预测。

$$y = \underset{j}{\operatorname{argmin}} \|f(\theta_k^{\text{sum}}; x) - C(j)\|_2 \quad (8)$$

3 基于个性化层的鲁棒联邦原型学习方案 Robust-PFedproto

PFedproto 框架设想的主要应用场景为客户端数据集呈非独立同分布的分布式环境, 其中央服务器收集的是各标签对应的远程原型更新集合, 而非联邦平均 (federated averaging, FedAvg) 算法中的远程参数更新。尽管PFedproto有效缓解了数据异构性对联邦学习的影响, 但其鲁棒性问题尚未解决, 且针对联邦原型学习场景下的投毒攻击研究仍较为匮乏。鉴于此, 本节聚焦于设计一种适用于PFedproto框架的防御方案, 旨在抵御非独立同分布数据集分布下的投毒攻击行为。

原型作为模型从原始数据中提取的特征表示, 尽管不同客户端提取出的特征因数据异构性而存在差异, 但同标签原型应具备一定相似性。若某客户端提交的原型与其他客户端的同标签原型存在显著差异, 则可判定该原型为恶意原型, 相应客户端为恶意客户端。图3展示了中央服务器收集的标签1原型集合的核密度估计的结构。

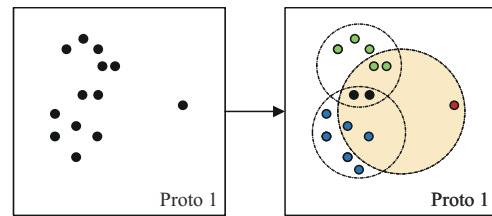


图3 核密度估计的结构

Figure 3 The structure of KDE

在Robust-PFedproto框架中, 当数据集为非独立同分布时, 部分客户端仅上传其拥有标签的

数据原型。此时，通过在同标签原型集合中评估恶意值，可有效适配该异构环境。该框架的3阶段防御方案主要包括计算客户端本地原型恶意值、决策阈值和剔除恶意客户端3个环节，中央服务器执行的具体流程如算法2所示。

算法2 Robust-PFedproto防御流程

输入 $P_i, \mathbf{P} = \{P_1, P_2, \dots, P_N\}, \epsilon$

输出 $\bar{P}_i$

中央服务器执行：

1) **for each round do**

2) **for each P_i in $\mathbf{P}$ do**

3) 为 P_i 选择 k -邻域

4) **for each l in $\text{Len}(P_i)$ do**

5) 利用式(13)计算当前维度的恶

意值 $F(P_i^l)$

6) **end for**

7) 利用式(14)计算当前原型的恶意值

$F(P_i)$

8) **end for**

9) 利用式(17)选取阈值 ϵ

10) 利用式(18)设置控制器 $\delta(i)$

11) 利用式(19)计算全局原型 $\bar{P}_i$

12) **end for**

13) 向客户端下发全局原型

14) 根据提交的恶意原型区分恶意客户端

15) 不接受恶意客户端上传的本地原型

3.1 客户端本地原型恶意值

在Robust-PFedproto框架中，全局有 M 个标签，假设在第 r 轮更新中包含标签 t 的客户端有 N 个，远程原型更新集合为 $\mathcal{P}^{r,t} = \{\mathbf{p}_1^{r,t}, \mathbf{p}_2^{r,t}, \dots, \mathbf{p}_N^{r,t}\}$ ，其中 $\mathbf{p}_i^{r,t}$ 为客户端 i 针对标签 t 的本地原型。由于第 r 轮更新的恶意值计算与评估只在第 r 轮中发生，标签 t 的原型只与标签 t 有关，因此标签 t 的原型集合可以简化为 $\mathcal{P} = \{P_1, P_2, \dots, P_N\}$ 。

计算客户端原型距离：对于客户端 i 的原型 P_i ，计算其与其他客户端原型之间的欧几里得距离，并选取与其距离最小的 k 个原型加入 k -领域原型集合 $\mathcal{P}_i = \{P_{i,1}, \dots, P_{i,j}, \dots, P_{i,k}\}$ ，有

$$\bar{d}_i = \frac{1}{k} \sum_{j=1}^k \text{dist}(p_i, p_{i,j}), i \neq j \quad (9)$$

其中， $\text{dist}(p_i, p_{i,j})$ 为欧几里得距离。

接下来，计算集合中所有原型之间的距离均值 $\bar{D}_i$ ，如式(10)所示。

$$\bar{D}_i = \frac{1}{k(k-1)} \sum_{i=1}^k \sum_{j=1}^k \text{dist}(p_i, p_{i,j}), i \neq j \quad (10)$$

恶意值计算：由于防御者无法获取真实的原型分布统计特性，需依赖非参数核密度估计函数拟合可能的原型分布。

通常，客户端上传的原型为一个高维张量。假设张量的维度为 L ， P_i^l 为当前原型第 l 维的值，其中 $l \in [1, L]$ 。在原型集合中，在每个维度上对目标原型与其他原型对应维度的值进行核密度估计，当前维度的核密度估计分布可表示为：

$$\tilde{q}(P_i^l) = \frac{1}{k} \sum_{j=1}^k G\left(\frac{P_i^l - P_{i,j}^l}{h^l}\right) \quad (11)$$

采用高斯函数 G 作为核函数：

$$G\left(\frac{P_i^l - P_{i,j}^l}{h^l}\right) = \frac{1}{(\sqrt{2\pi})h^l} \exp\left(-\frac{|P_i^l - P_{i,j}^l|}{2h^l}\right) \quad (12)$$

其中， h^l 为当前维度核密度估计的带宽，理论上高斯核的最优带宽 $h = \frac{4}{(k+2)^{1/(k+4)}}$ 。

进一步计算 k -领域原型集合中所有原型之间的核密度估计分布均值，则当前原型 P_i 第 l 维的恶意值可表示为：

$$F(i)^l = \frac{\sum_{j=1}^k \tilde{q}(P_{i,j}^l)}{k\tilde{q}(P_i^l)} \quad (13)$$

原型 P_i 的本地原型恶意值由所有维度恶意值相加获得，即：

$$F(P_i) = \prod_{l=1}^L F(P_i^l) = \frac{\sum_{j=1}^k \prod_{l=1}^L \tilde{q}(P_{i,j}^l)}{k \prod_{l=1}^L \tilde{q}(P_i^l)} \quad (14)$$

从统计学来看， k -领域原型集合作为目标原型相似度最高的 k 个邻居集合，在该集合内，由高斯核的负指数算子可推出 $\bar{d}_i$ 与 $\bar{D}_i$ 会影响 $F(i)$ 的值，当 $\bar{d}_i > \bar{D}_i$ 时， $F(i)$ 的值将会变小。由此认为 $F(i)$ 的值越小，目标原型更有可能是恶意的。

3.2 原型恶意阈值选择

在理想情况下，若所有远程客户端均为诚实客户端，则客户端 i 的原型到其 k -领域原型集合 $\mathcal{P}_i$ 中所有客户端同标签原型的距离均值 $\bar{d}_i$ 的期望，等于 $\mathcal{P}_i$ 内所有客户端同标签原型之间的距离均值 $\bar{D}_i$ 的期望，即：

$$\mathbb{E}(\bar{D}_i) = \frac{1}{k(k-1)} \mathbb{E}\left(\sum_{i=1}^k \sum_{j=1}^k \text{dist}(p_i, p_{ij})\right), i \neq j \quad (15)$$

$$\mathbb{E}(\bar{d}_i) = \frac{1}{k} \sum_{j=1}^k \mathbb{E}(\text{dist}(p_i, p_{ij})), i \neq j \quad (16)$$

则可推出理想情况下 $F(i)$ 的值为 $\frac{\mathbb{E}(\bar{D}_i)}{\mathbb{E}(\bar{d}_i)} = 1$ 。

在 Robust-PFedproto 框架中, 单纯采用理想阈值对恶意原型和正常原型进行区分存在两个问题: 其一, 原型是模型提取的特征表示, 虽然相似模型往往会生成相似特征, 但防御者很难获取原型更新的真实分布; 其二, 若阈值设置得过于严格, 可能会将正常原型误判为恶意原型, 进而剔除大量有效更新, 这会显著降低 PFedproto 的全局聚合速度。为解决这些问题, 本文为3阶段防御方案设计了一个渐进阈值, 其目的是确保大多数恶意远程更新具有较高的正恶意检测率, 从而保护联邦学习中的干净更新。假设该渐进阈值为 ϵ , 则其与 $F(i)$ 的关系满足式(17)。

$$P(F(i) > \epsilon) \leq \exp\left(-\frac{4\pi(\epsilon-1)^2(k+1)^2\bar{h}}{k(2k+\epsilon+1)^2V^2}\right) \quad (17)$$

则有二进制控制器:

$$\delta(i) = \begin{cases} 0, F(i) < \min\{1, \epsilon\} \\ 1, F(i) > \min\{1, \epsilon\} \end{cases} \quad (18)$$

由此对应标签的全局原型可表示为:

$$\bar{P}_i = \sum_{i=1}^N \delta(i) \alpha_i P_i \quad (19)$$

在聚合全局原型时, 仅保留二进制控制器判定为正常的原型用于聚合, 以此保证获得一个干净的新全局原型。

3.3 恶意客户端剔除

中央服务器需要记录各远程客户端的行为数据。当某个远程客户端提交的原型多次被标记为恶意时, 可将该客户端剔除, 以尽可能减少收集到的恶意原型的占比。

本文采用自注意力机制对客户端行为进行分类。考虑到攻击者通常需要持续提交恶意更新才能达到攻击目的, 假设一个长度为 m 的队列, 记录最近 m 次更新的客户端 ID, 以及对应客户端在各标签下提交的诚实更新和恶意更新数量。将队列信息按更新轮次嵌入为 $H = \{h_{t_1}, h_{t_2}, \dots, h_{t_m}\}$, 并输入自注意力层:

入自注意力层:

$$h_{\text{att}} = \sum_{i=1}^m \alpha_{t_i} \cdot W h_{t_i} \quad (20)$$

其中, α_{t_i} 表示轮次权重, W 表示可学习参数。最后采用激活函数 (如 ReLU 函数) 进行预测。

4 实验

4.1 实验设置

(1) 环境设置

本文所有实验均在服务器中完成, 使用的 GPU 为 NVIDIA RTX 4060, 编程语言为 Python 3.8, PyTorch 的版本为 1.8.0。

(2) 模型设置

本文实验所使用的模型为自定义卷积神经网络模型和 ResNet 模型, ResNet 模型结构设置为经典的 ResNet-18。

(3) 数据集设置

实验使用的数据集为常用的图像分类数据集, 包括 CIFAR-10、MNIST、EMNIST 和 Fashion-MNIST。表 1 介绍了数据集信息。

表 1 数据集信息
Table 1 Datasets information

数据集	大小	种类	特征
CIFAR-10	6 000	10	3 072
MNIST	70 000	10	784
EMNIST	814 255	62	784
Fashion-MNIST	80 000	10	784

(4) 数据集分布设置

数据分布设置为非独立同分布, 其异构性由迪利克雷系数控制。具体而言, 系数值越小, 客户端间的数据异构程度越强。

(5) 评价指标

与传统联邦学习算法采用全局模型准确率作为评价指标不同, 本文使用跨客户端平均准确率作为评价指标。跨客户端平均准确率定义为各客户端个性化模型在测试集上准确率的平均值。

(6) 对比算法

本文选取 FedAvg、FedProto、FedPer 和 PFedEdit 4 种算法与所提出的 PFedproto 开展对比实验。

FedAvg 算法是一种在联邦学习环境下协调分布式数据进行中心化模型训练的算法。该算法

旨在通过局部计算模型参数的更新，并在服务器上聚合这些更新，以迭代地优化一个全局共享模型。FedAvg 算法通过加权平均策略，确保在模型聚合过程中数据量较大的客户端对全局模型的更新有更大的贡献，从而有效利用分布式数据的集体智慧。然而，该算法在处理非独立同分布数据时性能受限。

在 FedProto 算法中，客户端向中央服务器不再提供模型更新数据，而是提交模型从原始数据中提取的原型特征，中央服务器通过聚合客户端原型生成全局原型，并将其分发给所有客户端。客户端使用这些全局原型指导本地模型的训练。在非独立同分布的联邦学习环境中，FedProto 在多个数据集上的表现良好。

FedPer 算法将模型分为基础层和个性化层两部分，客户端上传模型参数时仅提交基础层参数。

PFedEdit 算法^[46]借助自动化模型编辑技术优化个性化层的选择策略，显著提升模型在各类数据分布（包括非独立同分布）下的实用性。

4.2 个性化层联邦原型学习实验

本节通过开展消融实验，系统验证 PFedproto 框架的有效性与关键模块的作用机制，探究各组件对整体性能的实际贡献。

FedAvg 是联邦学习中最基础的算法，FedPer 属于基于个性化层的联邦学习方法，PFedEdit 是通过自动化机制选择个性化层的联邦学习算法，而 FedProto 是联邦原型学习算法。本文提出的 PFedproto 在 FedPer 的基础上引入原型学习模块，通过原型捕捉数据的内在特征，进一步降低数据异构性带来的影响。PFedproto 在 FedProto 基础上添加了个性化层。该层位于决策层之前，并保留在客户端本地，使本地模型能够更好地适应本地数据集特性。

本文通过改变数据分布，对比分析 PFedproto 与 PFedEdit、FedProto、FedAvg 和 FedPer 的性能。首先使用非独立同分布的 MNIST 和 CIFAR-10 数据集开展对比实验。在本实验中，客户端数量设置为 10，学习率设置为 0.01，全局迭代轮次设置为 50，迪利克雷系数 α 分别设置为 0.01 和 5（用于模拟非独立同分布数据集）。实验

结果如图 4 所示，从中可以观察到，无论是在 MNIST 数据集还是在 CIFAR-10 数据集上，FedAvg 算法的平均准确率都是最低的。这是因为当数据集为非独立同分布时，FedAvg 易受数据异构性影响，模型更新方向偏离全局最优解。相较于 FedAvg 算法，PFedproto 的平均准确率显著提升，且高于 PFedEdit、FedPer 和 FedProto 等对比算法。这得益于其分层架构设计，原型学习模块增强了特征表征能力，个性化层适配了本地数据分布，二者在训练和推理阶段协同优化了模型性能。此外，还可以发现，当数据集越趋向于非独立同分布，PFedproto 的性能越好。

4.3 Robust-PFedproto 抗投毒攻击实验

本节分别在 CIFAR-10、MNIST、Fashion-MNIST 和 EMNIST 数据集上验证本文提出的 Robust-PFedproto 防御方案对 PFedproto 框架的效用。在本节实验中，客户端数量设置为 20，CIFAR-10 数据集的学习率设置为 0.005，MNIST、Fashion-MNIST 和 EMNIST 数据集的学习率设置为 0.001，全局迭代轮次设置为 50。设置投毒客户端数量为 n ，投毒客户端中翻转的标签数量为 m 。图 5 展示了部分实验中不同的投毒强度对模型性能的影响，从图中可以看出，随着投毒客户端数量 n 和翻转标签数量 m 的增加，投毒攻击对模型性能的影响显著加剧，表现为平均准确率持续下降。

在 CIFAR-10 数据集上，当 $n=3$ 、 $m=4$ 时，模型在未遭受投毒攻击时的准确率为 82.68%，模型遭受投毒攻击后的准确率下降到 78.86%，在 Robust-PFedproto 防御下，模型的准确率恢复至 82.10%，接近原始水平。当 $m=6$ 时，模型遭受投毒攻击后的准确率下降到 76.83%，但经 Robust-PFedproto 防御后的准确率保持在 81.67%。当 $n=5$ 、 m 保持为 4 时，模型遭受投毒攻击后的准确率下降到 78.97%，经 Robust-PFedproto 防御后的准确率为 81.38%。

表 2 给出了 Robust-PFedproto 防御方案在不同数据集 n 个客户端模型投毒攻击下的防御效果，从中可以看出，Robust-PFedproto 防御方案具有很好的鲁棒性，可以有效抵御投毒攻击。即使恶意客户端未主动投毒，但其不积极参与全局训练，

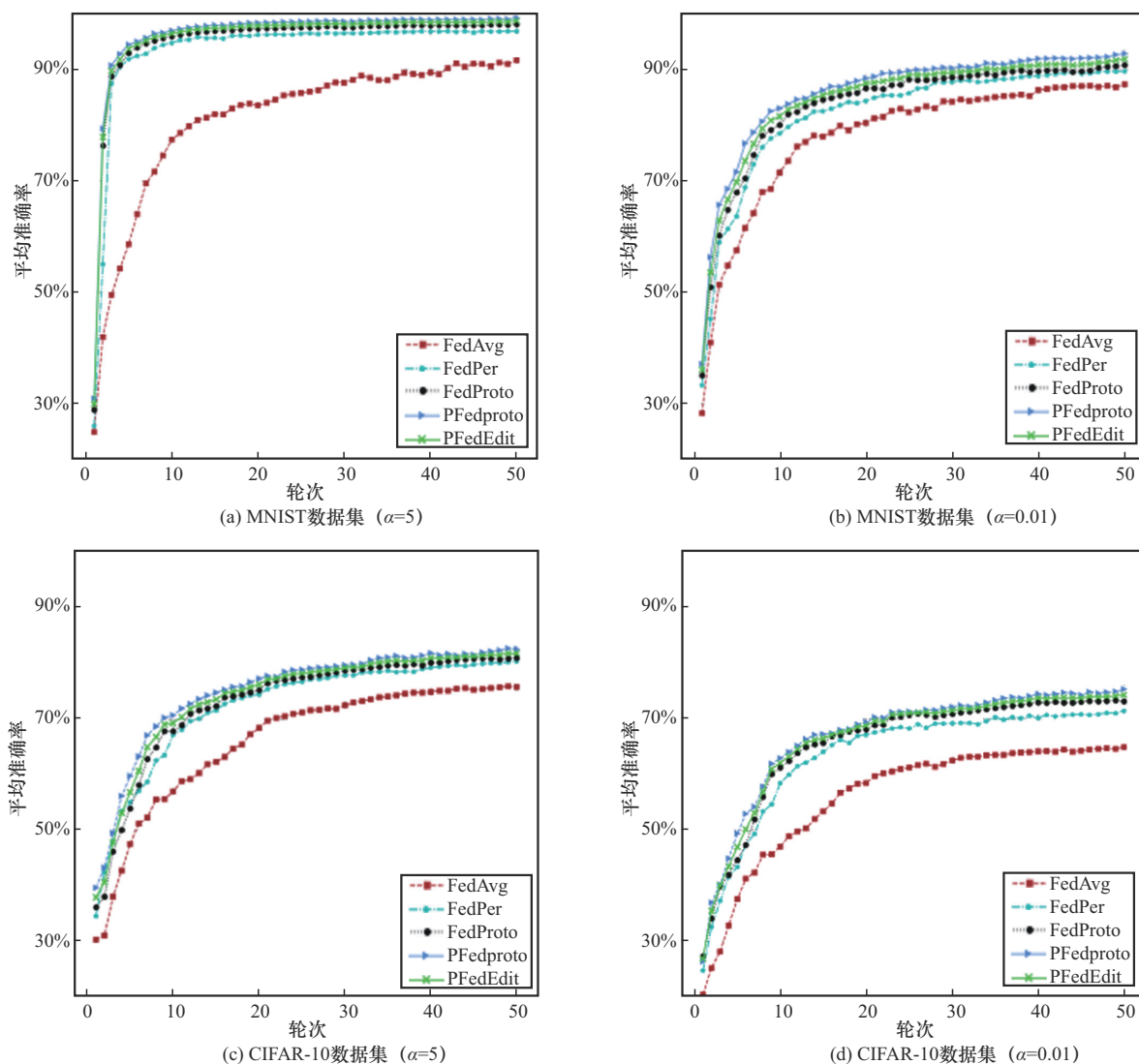


图4 PFedproto与PFedEdit、FedProto、FedAvg和FedPer的平均准确率对比

Figure 4 Comparison of the average accuracy rates of PFedproto with PFedEdit, FedProto, FedAvg and FedPer

也会导致全局模型的准确率下降。模型投毒攻击对准确率的破坏略高于数据投毒攻击。当遭受投毒攻击时,采用Robust-PFedproto防御机制的模型的准确率远高于未进行防御的模型的准确率。

由于恶意客户端上传的原型与诚实客户端上传的原型存在较大差异,可通过统计学特征分析与邻域原型进行对比,推断原型是否为恶意,从而剔除恶意客户端。由此,Robust-PFedproto的3阶段防御机制能够有效地减少这些影响,恢复模型的准确率。这表明所提出的防御方案在不同数据集和攻击强度下均具有较好的鲁棒性和有效性。

5 结束语

本文提出了一种创新的联邦学习框架Robust-PFedproto,旨在解决联邦学习中的数据异构性和数据投毒攻击两个主要挑战。首先,通过在原型学习机制中引入个性化层,使本文的框架能够动态适配不同客户端的数据分布特征,从而提高模型的收敛速度;然后,构建了一个3阶段防御方案来增强PFedproto框架的鲁棒性,以抵御数据投毒攻击。

在MNIST、EMNIST、Fashion-MNIST和CIFAR-10数据集上的实验结果表明,PFedproto框架能够更有效地缓解数据异构性的影响,且

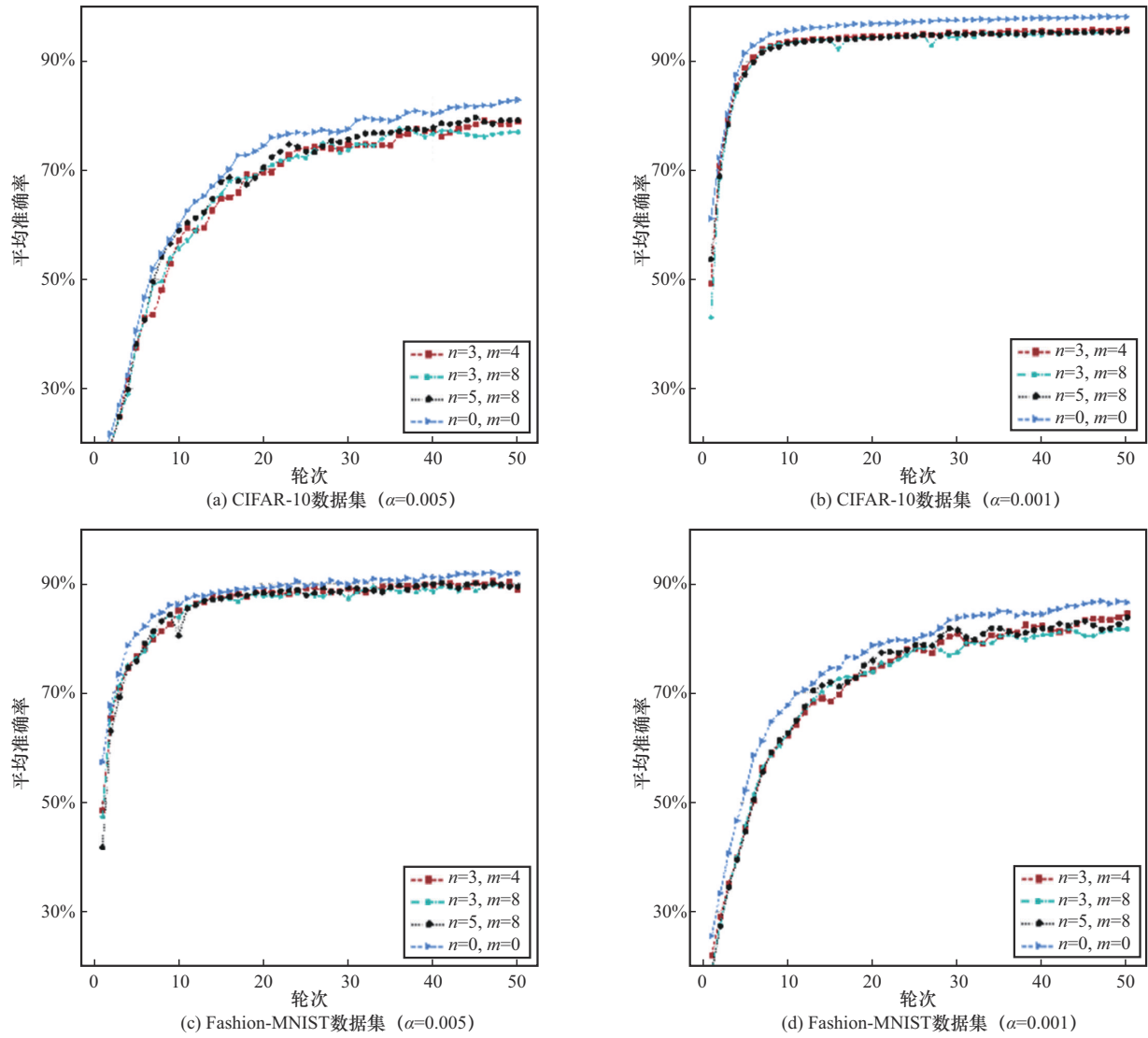
图5 不同投毒强度(n 和 m 组合)对模型性能的影响Figure 5 Impact of different poisoning intensities (combinations of n and m) on the model performance

表2 Robust-PFedproto 模型投毒攻击与防御准确率对比

Table 2 Comparison of accuracy under poisoning attacks and defenses for the Robust-PFedproto model

	CIFAR-10			MNIST		
	$n=2$	$n=4$	$n=6$	$n=2$	$n=4$	$n=6$
Clean	82.68%±0.3%	78.50%±0.3%	74.33%±0.3%	98.17%±0.3%	96.74%±0.3%	95.10%±0.3%
Attack	71.76%±0.5%	69.49%±0.5%	64.82%±0.5%	86.57%±0.5%	85.03%±0.5%	82.42%±0.5%
Defend	80.13%±0.5%	76.42%±0.5%	71.59%±0.5%	96.34%±0.5%	93.97%±0.5%	91.21%±0.5%
	Fashion-MNIST			EMNIST		
	$n=2$	$n=4$	$n=6$	$n=2$	$n=4$	$n=6$
Clean	91.83%±0.3%	87.05%±0.3%	82.94%±0.3%	86.5%±0.3%	83.97%±0.3%	80.66%±0.3%
Attack	82.36%±0.5%	76.62%±0.5%	71.20%±0.5%	74.07%±0.5%	71.54%±0.5%	69.11%±0.5%
Defend	87.57%±0.5%	83.06%±0.5%	79.71%±0.5%	82.04%±0.5%	80.53%±0.5%	77.52%±0.5%

Robust-PFedproto在数据投毒攻击下展现出了良好的鲁棒性。

未来的工作将集中在进一步优化个性化层结构,以及探索更高效的恶意客户端识别和剔除机制,以不断提升联邦学习的安全性和效用。

参考文献:

- [1] WANG K I, ZHOU X K, LIANG W, et al. Federated transfer learning based cross-domain prediction for smart manufacturing[J]. IEEE Transactions on Industrial Informatics, 2022, 18(6): 4088-4096.
- [2] ZHANG W, WANG Z W, LI X. Blockchain-based decentralized federated transfer learning methodology for collaborative machinery fault diagnosis[J]. Reliability Engineering & System Safety, 2023, 229: 108885.
- [3] CHEN J B, LI J P, HUANG R Y, et al. Federated transfer learning for bearing fault diagnosis with discrepancy-based weighted federated averaging[J]. IEEE Transactions on Instrumentation and Measurement, 2022, 71: 3514911.
- [4] YUN L, ZHEN Y, FANDONG M, et al. An empirical study of catastrophic forgetting in large language models during continual fine-tuning[J]. CoRR, 2023, abs/2308.08747.
- [5] GUO Z J, LI X D, HAN L, et al. Robust inference for federated meta-learning[J]. Journal of the American Statistical Association, 2025: 1-27.
- [6] REN H Y, ANICIC D, RUNKLER T A. TinyReptile: TinyML with federated meta-learning[C]//Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN). Piscataway: IEEE Press, 2023: 1-9.
- [7] DONG F, GE X H, LI Q Y, et al. PADP-FedMeta: a personalized and adaptive differentially private federated meta learning mechanism for AIoT[J]. Journal of Systems Architecture, 2023, 134: 102754.
- [8] FALLAH A, MOKHTARI A, OZDAGLAR A. Personalized federated learning: a meta-learning approach[J]. CoRR, 2020.
- [9] WU C H, WU F Z, LYU L J, et al. Communication-efficient federated learning via knowledge distillation[J]. Nature Communications, 2022, 13: 2032.
- [10] YUHAN M, ZHONGLE X, JUE W, et al. Continual federated learning based on knowledge distillation[C]//Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence. 2022: 2182-2188.
- [11] VENKATESWARAN P, ISAHAGIAN V, MUTHUSAMY V, et al. FedGen: generalizable federated learning for sequential data[C]//Proceedings of the 2023 IEEE 16th International Conference on Cloud Computing (CLOUD). Piscataway: IEEE Press, 2023: 308-318.
- [12] ARIVAZHAGAN M G, AGGARWAL V, SINGH A K, et al. Federated learning with personalization layers[J]. arXiv preprint arXiv: 1912.00818, 2019.
- [13] PILLUTLA K, MALIK K, MOHAMED A R, et al. Federated learning with partial model personalization[C]//Proceedings of International Conference on Machine Learning(ICML 2022). New York: PMLR, 2022:17716-17758.
- [14] BAEK J, JEONG W, JIN J, et al. Personalized subgraph federated learning[J]. ICML, 2023.
- [15] TAN Y, LONG G D, LIU L, et al. FedProto: federated prototype learning across heterogeneous clients[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2022, 36(8): 8432-8440.
- [16] FENG H, YI H W, LI X H, et al. Review of privacy-preserving research in recommendation systems[J]. Journal of Frontiers of Computer Science and Technology, 2023, 17(8): 1814-1832.
- [17] KIM H, KWAK Y, JUNG M, et al. ProtoFL: unsupervised federated learning via prototypical distillation[C]//Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2023: 6447-6456.
- [18] MICHIELI U, OZAY M. Prototype guided federated learning of visual feature representations[J]. arXiv preprint arXiv: 2105.08982, 2021.
- [19] YANG Y C, HUI B, YUAN H L, et al. PrivateFL: accurate, differentially private federated learning via personalized data transformation[C]//Proceedings of the 32nd USENIX Security Symposium. Berkeley: USENIX Association, 2023: 1595-1611.
- [20] ZHANG J L, CHEN B, CHENG X, et al. PoisonGAN: generative poisoning attacks against federated learning in edge computing systems[J]. IEEE Internet of Things Journal, 2021, 8(5): 3310-3322.
- [21] CAO X Y, GONG N Z. MPAF: model poisoning attacks to federated learning based on fake clients[C]//Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Piscataway: IEEE Press, 2022: 3395-3403.
- [22] PEVA B, EL M E M, RACHID G, et al. Machine learning with adversaries: Byzantine tolerant gradient descent[J]. Advances In neural Information Processing Systems, 2017, 30: 118-128.
- [23] XIE C, KOYEJO O, GUPTA I. Generalized byzantine-tolerant SGD[J]. arXiv preprint arXiv: 1802.10116, 2018.
- [24] CHEN Y D, SU L L, XU J M. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent[C]//Proceedings of the Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems. New York: ACM, 2018: 96-96.
- [25] YIN D, CHEN Y D, RAMCHANDRAN K, et al. Byzantine-robust distributed learning: towards optimal statistical rates[J]. arXiv preprint arXiv: 1803.01498, 2018.
- [26] LI L P, XU W, CHEN T Y, et al. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets[C]//Proceedings of the Thirty-First Innovative Applications of Artificial Intelligence Conference. 2019: 1544-1551.
- [27] ELKORDY A R, AVESTIMEHR A S. HeteroSag: secure aggregation with heterogeneous quantization in federated learning[J]. IEEE Transactions on Communications, 2022, 70(4): 2372-2386.
- [28] WU Z X, LING Q, CHEN T Y, et al. Federated variance-reduced stochastic gradient descent with robustness to Byzantine attacks[J]. IEEE Transactions on Signal Processing, 2020, 68: 4583-4596.
- [29] ZHAO L C, JIANG J L, FENG B, et al. SEAR: secure and efficient

- aggregation for Byzantine-robust federated learning[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(5): 3329-3342.
- [30] LIU X Y, LI H W, XU G W, et al. Privacy-enhanced federated learning against poisoning adversaries[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 4574-4588.
- [31] YANG Z P, SHI Y M, ZHOU Y, et al. Trustworthy federated learning via blockchain[J]. IEEE Internet of Things Journal, 2023, 10(1): 92-109.
- [32] CUI L Z, SU X X, MING Z X, et al. CREAT: blockchain-assisted compression algorithm of federated learning for content caching in edge computing[J]. IEEE Internet of Things Journal, 2022, 9(16): 14151-14161.
- [33] 刘天瑞, 王连海, 王启正, 等. 算力网络环境下基于区块链的自适应联邦学习[J]. 网络与信息安全学报, 2024, 10(3): 130-142.
- LIU T R, WANG L H, WANG Q Z, et al. Blockchain based adaptive federated learning in computational power network[J]. Chinese Journal of Network and Information Security, 2024, 10(3): 130-142.
- [34] LI Y R, ZHANG S B, CHANG Y, et al. Privacy-preserving and poisoning-defending federated learning in fog computing[J]. IEEE Internet of Things Journal, 2024, 11(3): 5063-5077.
- [35] XU H W, FENG Y, CHEN J, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in Web applications[C]// Proceedings of the 2018 World Wide Web Conference. New York: ACM, 2018: 187-196.
- [36] KIEU T, YANG B, GUO C J, et al. Outlier detection for time series with recurrent autoencoder ensembles[C]//Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. California: IJCAI, 2019: 2725-2732.
- [37] MA Z R, MA J F, MIAO Y B, et al. ShieldFL: mitigating model poisoning attacks in privacy-preserving federated learning[J]. IEEE Transactions on Information Forensics and Security, 2022, 17: 1639-1654.
- [38] FUNG C, YOON C J M, BESCHASTNIKH I. The limitations of federated learning in Sybil settings[C]//Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020). Berkeley: USENIX Association, 2020: 301-316.
- [39] MODAK S. A new measure for assessment of clustering based on kernel density estimation[J]. Communications in Statistics - Theory and Methods, 2023, 52(17): 5942-5951.
- [40] STEINWARR I, SRIPERUMBUDUR B K, THOMANN P. Adaptive clustering using kernel density estimators[J]. Journal of Machine Learning Research, 2023, 24(275): 1-56.
- [41] LI X Y, QU Z, ZHAO S Q, et al. LoMar: a local defense against poisoning attack on federated learning[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(1): 437-450.
- [42] ALORAN I, SAMET S. Defending federated learning against model poisoning attacks[C]//Proceedings of the 2023 IEEE International Conference on Big Data (BigData). Piscataway: IEEE Press, 2023: 3584-3589.
- [43] FENG X, CHENG W H, CAO C J, et al. DPFLA: defending private federated learning against poisoning attacks[J]. IEEE Transactions on Services Computing, 2024, 17(4): 1480-1491.
- [44] JEBREEL N M, DOMINGO-FERRER J. FL-Defender: combating targeted attacks in federated learning[J]. Knowledge-Based Systems, 2023, 260: 110178.
- [45] YUAN H L, PAUL W, AUCOTT J, et al. PFedEdit: personalized federated learning via automated model editing[C]//Computer Vision - ECCV 2024. Berlin: Springer, 2025: 91-107.

[作者简介]



徐明迪 (1980-), 男, 湖北随州人, 武汉数字工程研究所研究员, 主要研究方向为网络安全、可信计算。



李正潇 (1999-), 男, 湖北武汉人, 武汉数字工程研究所硕士生, 主要研究方向为网络安全、隐私保护。



王梓行 (1999-), 男, 江苏宿迁人, 主要研究方向为数据安全、密码应用。



靳朝阳 (1993-), 女, 河北邢台人, 武汉数字工程研究所工程师, 主要研究方向为网络安全、云安全。