抗拜占庭攻击的隐私保护联邦学习

穆旭彤 程 珂 宋安霄 张 涛 张志为 沈玉龙
(西安电子科技大学计算机与科学技术学院 西安 710071)

摘 要 联邦学习是一种隐私保护的分布式机器学习框架,可以让各方参与者在不披露本地数据的前提下共建模型.然而,联邦学习仍然面临拜占庭攻击和用户隐私泄漏等威胁.现有研究结合鲁棒聚合规则和安全计算技术以同时应对上述安全威胁,但是这些方案难以兼顾模型鲁棒性与计算高效性.针对此问题,本文提出一种抗拜占庭攻击的隐私保护联邦学习框架SecFedDMC,在保护用户数据隐私的条件下实现高效的拜占庭攻击检测与防御.基础方案FedDMC采用"先降维后聚类"的策略,设计了高效精准的恶意客户端检测方法.此外,该方法利用的随机主成分分析降维技术和K-均值聚类技术主要由线性运算构成,从而优化了算法在安全计算环境中的适用性.针对基础方案存在的用户数据隐私泄露问题,提出了基于安全多方计算技术的隐私增强方案SecFedDMC.基于轻量级加法秘密分享技术,设计安全的正交三角分解协议和安全的特征分解协议,从而构建双服务器模型下隐私保护的拜占庭鲁棒联邦学习方案,以保护模型训练和拜占庭节点识别过程中的用户隐私.经实验验证,SecFedDMC在保护用户隐私的前提下,可以高效准确地识别拜占庭攻击节点,具有较好的鲁棒性.其中,本方案与最先进的鲁棒联邦学习算法相比,在CIFAR10数据集上,拜占庭攻击节点检测准确率提升12%~24%,全局模型精度提升4.45%~18.48%,计算效率提升33.21%~47.31%.

关键词 联邦学习;拜占庭攻击;安全多方计算;隐私保护;模型鲁棒性;隐私计算
中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2024.00842

Privacy-Preserving Federated Learning Resistant to Byzantine Attacks

MU Xu-Tong CHENG Ke SONG An-Xiao ZHANG Tao
ZHANG Zhi-Wei SHEN Yu-Long
(*School of Computer Science and Technology, Xidian University, Xi'an* 710071)

Abstract Federated learning is a privacy-preserving distributed machine learning framework that allows participants to build models without disclosing local data. However, federated learning still faces threats such as Byzantine attacks and client privacy leakage. Existing research combines robust aggregation rules and secure computation techniques to simultaneously address these security threats, but these solutions fail to balance model robustness with computational efficiency. To address this challenge, this paper presents SecFedDMC, a privacy-preserving federated learning framework resistant to Byzantine attacks. This framework ensures efficient Byzantine attack detection while protecting client data privacy. The basic scheme FedDMC adopts the strategy of 'dimensionality reduction followed by clustering to design an efficient and accurate method for malicious client detection. Moreover, the method utilizes randomized principal component analysis

收稿日期:2023-06-20;在线发布日期:2024-01-03. 本课题得到国家自然科学基金(No. 62220106004,62302368)、国家自然科学基金重大研究计划项目(No. 92267204)、陕西省重点研发计划项目(No. 2022KXJ-093, 2021ZDLGY07-05)、陕西省创新能力支持计划(No. 2023-CX-TD-02)、陕西省自然科学基础研究计划资助项目(No. 2024JC-YBQN-0701)、山东省重点研发计划项目(No. 2023CXPT056)、中央高校基本科研业务费专项(No. XJSJ23040, ZDRC2202)资金资助. **穆旭彤**,博士研究生,中国计算机学会(CCF)学生会员,主要研究领域为联邦学习、安全多方计算. E-mail:xtmu@stu.xidian.edu.cn. **程 珂(通信作者)**,博士,讲师,主要研究领域为隐私保护机器学习. E-mail:chengke@xidian.edu.cn. **宋安霄**,博士研究生,主要研究领域为数据安全、联邦学习. **张 涛**,博士,副教授,主要研究领域为数据安全、联邦学习. **张志为**,博士,副教授,研究领域为无人系统协同安全. **沈玉龙(通信作者)**,博士,教授,主要研究领域为云计算与数据安全、无线网络安全. E-mail:ylshen@mail.xidian.edu.cn.

and K-mean clustering techniques, which primarily involve linear operations, enhancing the applicability of the algorithm in secure computing environments. To address the user data privacy leakage problem in the basic scheme, we propose a privacy-enhanced scheme, SecFedDMC, based on secure multiparty computation technology. Based on the lightweight additive secret sharing technique, a secure orthogonal triangular decomposition protocol and a secure eigen decomposition protocol are designed. These form the foundation of a Byzantine-robust federated learning scheme under a dual-server model. Theoretical scrutiny substantiates the security of this scheme. Experiments validate that SecFedDMC efficiently identifies Byzantine attack nodes with precision, demonstrating robustness, all while preserving user privacy. In particular, this scheme compared with the state-of-the-art robust federated learning algorithm, the Byzantine attack node detection accuracy is improved by 12%～24%, the global model accuracy is improved by 4.45%～18.48%, and the computational efficiency is improved by 33.21%～47.31% on the CIFAR10 dataset.

Keywords　federated learning; Byzantine attacks; secure multi-party computation; privacy protection; model robustness; private computing

1　引　言

联邦学习(Federated Learning, FL)[1]是一种新兴的分布式机器学习机制,被广泛应用于医学影像诊断[2-3]、银行交易的风险识别[4]和自动驾驶训练[5]等领域.区别于传统的中心化机器学习模式,联邦学习允许多个用户设备在本地完成模型训练,不与中央服务器交换数据,只共享模型参数以生成全局模型.因此,这种机制不仅减少了用户数据的上传,而且充分利用了客户端的计算能力,降低了对中央服务器的算力要求.虽然联邦学习在隐私保护与计算效率方面均取得了积极效果,但该机制仍然面临两方面安全威胁:拜占庭攻击和用户隐私泄漏[6-7],如图 1 所示.

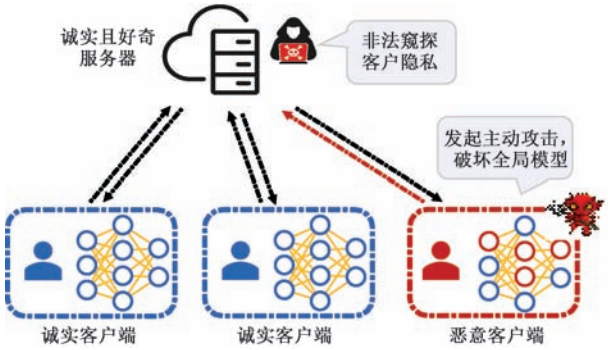

图 1　联邦学习系统中存在的安全问题

联邦学习机制具有分布式特性,各用户节点均有权对全局模型进行更新,因此该机制极易遭受拜占庭节点的攻击[8-14].联邦学习中的拜占庭攻击可能由以下原因引起:攻击者通过控制一部分用户节点,故意提供错误的模型参数或训练数据,来影响模型的最终学习效果;在某些情况下,用户节点可能由于软件错误、硬件故障或网络问题,无法提供正确的数据或执行正确的操作,也可能导致类似拜占庭攻击的问题.针对联邦学习中的拜占庭攻击,大量研究工作提出了基于聚合规则的解决方案[15-16],通过调整聚合规则来提高模型训练的鲁棒性.然而,拜占庭节点(即恶意节点)可通过巧妙的模型设计来规避这些检测规则.此外,部分研究利用无污染的验证数据集[17-18]或聚类模型参数[19-22]实现对恶意客户端的判别,从而更加精准地识别恶意行为,最大化地降低其对全局模型的破坏.然而,上述方法依赖于不切实际的先验知识,或者在稳健性和效率之间难以取得平衡,因此限制了这些方法的实际应用.

除了拜占庭攻击问题,联邦学习模型训练过程还存在用户隐私泄露问题.虽然各参与方在本地进行模型训练并不直接共享数据,但通过对本地模型的深入分析,诚实但好奇的服务器仍然可能推导出用户的部分信息[23-24].现有研究主要聚焦于如何确保本地模型参数的安全性,以防止信息泄露.差分隐私(Differential Privacy, DP)[25-26]是其中一个广泛应用的方法,通过向模型参数中添加随机噪声来实现数据隐私的保护,但这会对模型的准确性产生影响.利用同态加密(Homomorphic Encryption, HE)[27-28]可以在不进行解密的情况下实现加密模型参数的聚合,然而这将导致计算成本的大幅提升.

目前,学术界与工业界已提出一些方案来解决

拜占庭攻击和隐私泄露问题，然而大多数研究将这两个问题割裂开来，并且忽视了它们之间的内在联系.实质上，拜占庭攻击和隐私泄露构成了一个相互影响的复杂问题.一方面，拜占庭节点向系统中注入篡改的数据或模型参数，若这些误导性输入未被适当地识别和处理，可能导致模型的输出偏离预期.在此情境下，拜占庭节点能够从偏离的模型中推断出其他参与者的原始数据或特定的隐私信息.另一方面，参与者的隐私数据可能被泄露，这些泄露的信息为拜占庭攻击提供先验知识.配备此类先验知识，攻击者能够设计出更精准的策略，针对性地发动攻击，从而使其他节点面临更严重的威胁.

然而，实现抗拜占庭攻击的隐私保护联邦学习面临诸多挑战.一种直接的解决方法是将通用的密码学技术，如安全多方计算(Secure Multi-party Computation, SMC)[29]和同态加密[30]与现有的拜占庭鲁棒联邦学习算法[17-22]相结合.但是，这种方法需要执行大量耗时的密文域操作.例如，在衡量各方梯度质量时需要在密文上执行大规模矩阵乘法，在排除异常梯度时需要执行一系列安全的非线性函数.已有一些工作[31-38]尝试同时考虑拜占庭攻击和隐私保护问题，但是为了保证密文协议的执行效率，上述方法作出了部分妥协.例如，文献[37]中的方案会泄露模型参数检测过程中的部分中间值，文献[33]和[38]中的方案利用简单但抗拜占庭攻击能力较弱的聚合规则.总之，上述隐私保护的拜占庭鲁棒联邦学习方案难以兼顾计算高效性、数据隐私性及模型鲁棒性.

因此，本文提出一种抗拜占庭攻击的隐私保护联邦学习框架 SecFedDMC，旨在保护用户数据隐私的前提下，实现高效的拜占庭攻击检测与防御.基础方案 FedDMC 采用“先降维后聚类”的策略，提升了聚类精度和计算效率，并且精准地识别恶意客户端.为进一步保护用户数据隐私，本文基于轻量级的加法秘密分享技术，设计了安全的正交三角分解(QR 分解)协议和特征分解协议，基于 FedDMC 方案构建了双服务器模型下的隐私保护拜占庭鲁棒联邦学习方案.本研究主要贡献如下：

(1)提出了联邦学习基础方案 FedDMC.该方案利用随机主成分分析降维技术和 K-均值聚类技术，准确识别出恶意客户端，提升了系统的鲁棒性.这些技术主要涉及线性操作，增强了该算法在安全计算环境中的适用性.

(2)在双服务器模型下提出了隐私增强的联邦学习框架 SecFedDMC.基于轻量级加法秘密分享技术，设计了安全 QR 分解协议与安全的矩阵分解协议，在此基础上实现了兼顾数据隐私性与计算高效性的鲁棒联邦学习.

(3)对 SecFedDMC 方案进行了理论和实验评估.理论分析证明了方案的安全性，实验结果表明，与最先进的鲁棒联邦学习算法相比，在 CIFAR10 数据集上，拜占庭攻击节点检测准确率提升 12%～24%，全局模型精度提升 4.45%～18.48%，计算效率提升 33.21%～47.31%.

本文章节安排如下：第 2、3 节分别介绍相关工作和预备知识；第 4 节描述了系统模型；第 5 节描述了 SecFedDMC 方案；第 6 节对本文方案进行了理论分析；第 7、8 节分别描述了实验设置及实验结果；第 9 节为结论.

2 相关工作

2.1 联邦学习中的拜占庭攻击

拜占庭攻击是联邦学习中的一种典型攻击，旨在篡改参与者提交的模型更新参数，使模型参数的实际收敛过程偏离预期路径，从而对全局模型的精度和收敛性产生负面影响.拜占庭攻击主要有两种形式：篡改本地数据和篡改本地模型.

对于数据篡改，攻击者可以修改数据标签或者植入后门.例如，Wang 等人[8]引入了“边缘样本后门攻击”，即向数据集中添加了难以检测的带有错误标签的样本或后门触发器.Bagdasaryan 等人[9]在后门攻击的基础上将规避检测优化问题的约束项，融入攻击者损失函数，提高了攻击的隐蔽性.Xie 等人[10]提出“分布式后门攻击”，该攻击通过分解全局后门触发器为多个独立的本地后门触发器，提高了攻击的持续性和隐蔽性.

对于模型篡改，攻击者主要通过微调模型来破坏全局模型且规避检测.Baruch 等人[11]提出的方法允许攻击者在一定范围内对模型参数改动，这种微小的改动可以对学习过程产生干扰且不易被检测.Bhagoji 等人[12]研发了一种交替优化隐蔽性和对抗性目标的策略.Fang 等人[13]将攻击视为一个优化问题，目标是通过对模型更新进行微调，以放大聚合梯度之间的差异.Shejwalkar 等人[14]提出了模型中毒攻击的通用框架，根据服务器的聚合规则等先验知识来建立优化问题，以实现规避鲁棒聚合算法的攻击.

2.2　抗拜占庭攻击的联邦学习方法

抗拜占庭攻击的联邦学习研究主要包括基于差分隐私的方法、基于聚合规则的方法和基于恶意客户端检测的方法.

研究工作[8]采用差分隐私的思想,通过减小模型参数到最大阈值并添加随机噪声,以减轻恶意模型对全局模型的影响. FLAME[32]提出,在应用差分隐私策略前,需要过滤掉那些具有高攻击影响力的投毒模型. 然而这些方法由于添加了噪声,从而降低了全局模型的准确性.

在基于聚合规则的方法中,Krum[20]从一组局部模型中选取与其他模型最为相似的模型作为全局模型. Trimmed-Mean[14]通过剔除最大和最小的几个值,对每个模型参数进行聚合,以规避离群值对结果的影响. Bulyan[16]结合了 Krum 和 Trimmed-Mean 方法,首先用 Krum 选择部分客户端,然后再应用 Trimmed-Mean 方法执行鲁棒聚合. 这些方法应用统计学原理来估计聚合的模型更新. 然而,攻击者可能会精心设计本地模型,并选择性地将有害的参数纳入全局模型,以规避这类防御措施.

基于恶意客户端检测的方法旨在通过无污染的验证数据集或模型参数聚类,区分诚实客户端和恶意客户端. 例如,FLTrust[17]在服务器上维护一个无污染的验证数据集以生成更新方向. 随后,服务器会通过比较本地模型更新和服务器模型更新的相似性,以区分恶意客户端和诚实客户端. 另外,Li 等人[18]使用了变分自编码器(Variational Autoencoder,VAE)来捕获模型更新的统计特性. 然而,这些方法都需要一个无污染的验证数据集,且其分布需要与本地客户的训练数据匹配. 这在实际应用中往往难以实现,因此限制了这些方法的应用. 从 Krum 扩展出的 Multi-Krum[20]方法会选择与其他模型最相似的多个局部模型,并计算它们的平均值以生成全局模型. DnC[14]先采用随机抽样进行降维,然后利用谱方法(Spectral Method)去除恶意客户端. 然而,Krum 和 DnC 需要预先设定恶意客户端的数量. Auror[19]使用 K-均值(K-means)聚类检测恶意客户端,FoolsGold[21]则依据客户端参数的相似性来进行恶意客户端的检测. 然而,对于某些特定类型的攻击,如标签反转攻击[17],高斯攻击[31]等,这些方法可能无法提供有效的防护. FLDetector[22]通过模型一致性来检测恶意客户,但其计算开销大导致难以实际应用.

2.3　隐私增强的抗拜占庭攻击联邦学习方法

用户数据隐私泄露问题在联邦学习中日益突出,研究者开始基于差分隐私(DP)、多方安全计算(SMC)、同态加密(HE)等技术设计隐私保护的抗拜占庭攻击联邦学习方法. DPBFL[31]结合差分隐私和鲁棒随机聚合规则来进行拜占庭攻击防御,同时保护隐私,但引入的噪声可能会对全局模型准确度造成影响. FLAME[32]通过加入足量的噪声来消除恶意客户端的影响,然而这可能导致诚实模型性能的显著下降. PPRAgg[33]结合同态加密和随机噪声混淆技术来保护模型训练,并以余弦相似度作为信誉度评估拜占庭节点. PEFL[34]采用同态加密作为基础技术,通过皮尔逊相关系数来识别恶意参数,但引入了巨大的计算开销. SecureFL[35]基于 FLTrust[17]定制了一系列同态密码学组件,FLOD[36]通过使用同态加密和两方安全计算技术提高了 FLTrust 的隐私,但要求参与方在本地进行同态加密操作,导致了大量的本地客户端计算开销,并且这两个方法鲁棒性依赖于服务器上的干净数据集,限制了该方法的应用场景. 在安全多方计算领域,LSFL[37]结合了隐私保护和拜占庭稳健性的特点,设计了轻量级的双服务器安全聚合协议,应用 K-近邻算法来提高联邦学习的鲁棒性,但容易被恶意客户端攻破. BREA[38]是一种抗拜占庭攻击的安全聚合框架,但需要在服务器和参与者之间进行多次通信,导致较大的通信开销. 因此,尽管以上研究应用不同技术来解决隐私保护和恶意节点攻击问题,但大多数方法都无法同时确保联邦学习的数据隐私性、模型鲁棒性与计算高效性. ELSA[39]是一种隐私保护的联邦学习聚合机制,该协议机制由两台服务器构建,且只需一台服务器是诚实的就可以保证隐私性. 但该方法基于简单的二范式来抵御恶意客户端,攻击者可能会精心设计本地模型,规避这类防御措施. ABFL[40]结合了联邦学习和区块链技术,增强了数据所有权并有效减少了恶意节点的影响,但并未考虑对本地模型的隐私保护.

本文方案采用多方安全计算技术避免了使用差分隐私时模型精度与隐私之间难以平衡的问题,同时也避免了同态加密的计算负担. 此外,采用“先降维后聚类”的策略可以精准地识别恶意客户端,在保证隐私性的同时提高了联邦学习的鲁棒性和效率. 表 1 总结了隐私增强的抗拜占庭攻击联邦学习方法的优缺点. 据此,本研究提出的方法在数据隐私保护、模型的鲁棒性以及计算效率三个方面均有明显的优势.

表 1 隐私增强的抗拜占庭攻击联邦学习方法

相关工作	隐私方法	检测方法	服务器模型	数据隐私性	模型鲁棒性	计算高效性
DPBFL[31]	DP	RSA	单	◐	◐	●
FLAME[32]	DP	DnC	单	○	◐	●
PPRAgg[33]	HE	Cosine similarity	双	●	◐	○
PEFL[34]	HE	Pearson correlation	双	●	◐	○
SecureFL[35]	HE	FLTrust	双	●	◐	○
FLOD[36]	MPC+HE	FLTrust	双	●	◐	○
LSFL[37]	MPC	K-nearest	双	◐	◐	●
BREA[38]	MPC	Krum	单	●	○	●
本文方案	MPC	RPCA+K-means	双	●	●	●

* 在数据隐私性一列中，●表示可证明安全的隐私保护，◐表示会泄露部分中间值，○表示未保护模型参数. 在模型鲁棒性一列中，●表示该方法可以很好地防御各种已知的攻击，◐表示该方法在面对某些已知攻击时仍然表现出局限性，○表示该方法可以被大多数攻击方法攻破. 计算高效性一列中，●代表效率较高，◐代表效率相对较慢，○代表效率最慢.

3 预备知识

本节首先介绍随机主成分分析(Randomized Principal Component Analysis，RPCA)[41]，然后介绍相关的密码学原语.

3.1 随机主成分分析

RPCA 是一种用于高维数据的降维技术. 与经典的主成分分析(Principal Component Analysis，PCA)相比，RPCA 通过使用随机技术来降低计算的复杂度，同时保持了对数据主要变化的有效感知. RPCA 的基本思想是通过随机线性映射将原始高维数据映射到一个低维子空间，然后在这个低维子空间中进行 PCA 分析. RPCA 不仅可以加快 PCA 分析的速度，还可以降低其存储需求，使其更适合大规模的高维数据处理.

图 2 描述了用 RPCA 方法把矩阵 $\boldsymbol{M}\in\boldsymbol{R}^{n\times d}$ 降维到 $\widetilde{\boldsymbol{M}}\in\boldsymbol{R}^{n\times k}$ 的工作流程以及每个步骤的维度变化. (1)以矩阵 $\boldsymbol{M}$ 和随机映射矩阵 $\boldsymbol{Q}\in\boldsymbol{R}^{n\times\rho}$ 作为输入，其中 $\rho=k+\alpha$，k 为期望的主成分数量，α 为过采样参数，用于提高算法的准确性；(2)对输入矩阵 $\boldsymbol{M}$ 的每一列进行均值中心化，以消除列之间的偏移；(3)将投影矩阵 $\boldsymbol{Q}$ 递归地与协方差矩阵 $\boldsymbol{M}\boldsymbol{M}^T$ 相乘 p 次以提高精度，每次迭代中都使用 $\boldsymbol{QR}$ 分解法进行正交化以保证数值稳定性；(4)将中心化后的输入矩阵投影至较低维度空间，即与优化后的随机映射矩阵 $\boldsymbol{Q}$ 相乘；(5)计算小型对称矩阵 $\boldsymbol{B}$，该矩阵表示在低维空间中的特征协方差，这是通过在协方差矩阵两侧乘以 $\boldsymbol{Q}$ 与 $\boldsymbol{Q}^T$ 得到的；(6)通过特征值分解(Eigen)计算 $\boldsymbol{B}$ 的特征向量 $\boldsymbol{U}$；(7)在原始空间中重构特征向量，即主成分 $\boldsymbol{W}$；(8)将输入矩阵 $\boldsymbol{M}$ 投影到主成分上，构造最终的输出结果 $\widetilde{\boldsymbol{M}}$. 在这个过程中，*RPCA* 算法将对于 $\boldsymbol{M}\in\boldsymbol{R}^{n\times d}$ 的分解问题简化为对常数大小矩阵 $\boldsymbol{B}\in\boldsymbol{R}^{\rho\times\rho}$ 的分解，且 $\rho\ll d$，因此计算速度得到了显著提升.

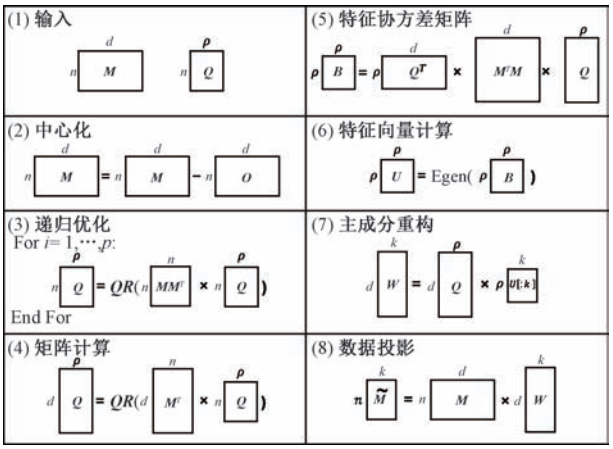

图 2 *RPCA* 工作流程

3.2 密码学原语

本文方案采用了两种秘密分享机制，即“$\langle\cdot\rangle$-分享”与“$[\![\cdot]\!]$-分享”.“$\langle\cdot\rangle$-分享”是一种经典的加法秘密分享(Additive Secret Sharing，ASS). 对于数值 $v\in Z_{2^l}$，其值被分享至两个服务器，每个服务器 S_i($i\in\{0,1\}$)持有秘密分享份额 $\langle v\rangle_i$，满足：$v=\langle v\rangle_0+\langle v\rangle_1$.

“$[\![\cdot]\!]$-分享”为基于“$\langle\cdot\rangle$-分享”改进的加法秘密共享(improved Additive Secret Sharing，iASS). 在此机制中，存在两数值 $\delta_v, r_v\in Z_{2^l}$，满足以下条件：(1)随机数 r_v 在 S_0 和 S_1 之间通过“$\langle\cdot\rangle$-分享”的方式共享，即 $\langle r_v\rangle_0$ 和 $\langle r_v\rangle_1$；(2) $\delta_v=v-r_v$；(3) δ_v 对 S_0 和 S_1 公开. 服务器 S_i 的份额形式化表示为 $[\![v]\!]_i:=(\delta_v,\langle r_v\rangle_i)$，其中 $i\in\{0,1\}$. IASS 的基础

运算协议如下:

共享协议. 数据拥有者将数据 v 转化为"$[\![\cdot]\!]$ −分享"形式. 首先,数据拥有者生成一组"$\langle\cdot\rangle$ −分享"的两个随机数 $\langle r_v\rangle_0$ 和 $\langle r_v\rangle_1$,并计算得到 $\delta_v=v-\langle r_v\rangle_0-\langle r_v\rangle_1$. 接下来,分别将 δ_v 与 $\langle r_v\rangle_i$ 发送给 S_i,得到 v 的"$[\![\cdot]\!]$ −分享"形式,即 $[\![v]\!]_i:=(\delta_v,\langle r_v\rangle_i)$.

重构协议. 为了从 $[\![v]\!]_i$ 中重构出 v,双方服务器交换 $\langle r_v\rangle_i$,并在本地计算出 $v=\delta_v+\langle r_v\rangle_0+\langle r_v\rangle_1$,此过程可表示为 $v=\mathrm{Rec}([\![v]\!]_0,[\![v]\!]_1)$.

多项式计算协议. 给定 $[\![a]\!]$,$[\![b]\!]$ 和公共常数 c_1,c_2,服务器 S_0 和 S_1 可以实现计算 $[\![y]\!]=c_1\cdot[\![a]\!]+c_2\cdot[\![b]\!]$. 具体地,$S_i$ 在本地计算 $\delta_y=c_1\cdot\delta_a+c_2\cdot\delta_b$ 和 $\langle r_y\rangle_i=c_1\cdot\langle r_a\rangle_i+c_2\cdot\langle r_b\rangle_i$ 得到 $[\![y]\!]_i:=(\delta_y,\langle r_y\rangle_i)$.

乘法协议. 给定"$[\![\cdot]\!]$ −分享"形式 a,b,即服务器 S_i 持有 $[\![a]\!]_i:=(\delta_a,\langle r_a\rangle_i)$,$[\![b]\!]_i:=(\delta_b,\langle r_b\rangle_i)$,协议 *SecMul* 输出 $[\![y]\!]$,即 S_i 持有 $[\![y]\!]_i:=(\delta_y,\langle r_y\rangle_i)$,其中 $y=ab$. 离线阶段:(1)两方服务器 S_i 各自随机生成 $\langle r_y\rangle_i\in Z_{2^l}$;(2)通过"$\langle\cdot\rangle$ −分享"的乘法,利用 $\langle r_a\rangle_i,\langle r_b\rangle_i$ 计算出 $\langle r_{ab}\rangle_i$. 在线阶段:两方服务器在本地计算 $\langle\delta_y\rangle_i=i\cdot\delta_a\cdot\delta_b+\delta_a\cdot\langle r_b\rangle_i+\delta_b\cdot\langle r_a\rangle_i+\langle r_{ab}\rangle_i-\langle r_y\rangle_i$,并将计算结果 $\langle\delta_y\rangle_i$ 发送给另一个服务器 S_{1-i}. 之后每个服务器 S_i 各自计算 $\delta_y=\langle\delta_y\rangle_0+\langle\delta_y\rangle_1$ 即可得到 $[\![y]\!]_i:=(\delta_y,\langle r_y\rangle_i)$. 此过程形式化表示为 $[\![y]\!]=SecMul([\![a]\!],[\![b]\!])$.

三输入乘法协议. 给定"$[\![\cdot]\!]$ −分享"形式 a,b,c,该协议输出 $[\![y]\!]_i:=(\delta_y,\langle r_y\rangle_i)$. 离线阶段:(1)两方服务器 S_i 各自随机生成 $\langle r_y\rangle_i\in Z_{2^l}$;(2)两方服务器 S_i 生成 r_{ab},r_{bc},r_{ac} 和 r_{abc} 这四项的"$\langle\cdot\rangle$ −分享". 在线阶段:两方服务器分别计算 $\langle\delta_y\rangle_i=i\cdot\delta_a\cdot\delta_b\cdot\delta_c+\delta_a\cdot\delta_b\cdot\langle r_c\rangle_i+\delta_a\cdot\delta_c\cdot\langle r_b\rangle_i+\delta_b\cdot\delta_c\cdot\langle r_a\rangle_i+\delta_a\cdot\langle r_{bc}\rangle_i+\delta_b\cdot\langle r_{ac}\rangle_i+\delta_c\cdot\langle r_{ab}\rangle_i+\langle r_{abc}\rangle_i-\langle r_y\rangle_i$,之后两个服务器 S_i 交换 $\langle\delta_y\rangle_i$ 恢复出 δ_y,即可得到 $[\![y]\!]_i:=(\delta_y,\langle r_y\rangle_i)$. 为了方便表示,与乘法协议相似,将此过程形式化表示为 $[\![y]\!]=SecMul([\![a]\!],[\![b]\!],[\![c]\!])$. 同理,此协议可以推广到 n −输入乘法协议.

矩阵乘法协议. 给定"$[\![\cdot]\!]$ −分享"矩阵 $A^{n\times d}$,$B^{d\times k}$,即服务器 S_i 持有 $[\![A]\!]_i:=(\delta_A,\langle r_A\rangle_i)$,$[\![B]\!]_i:=(\delta_B,\langle r_B\rangle_i)$,协议 *SecMatMul* 的目标是输出 $[\![C^{n\times k}]\!]$. 离线阶段:(1)两方服务器 S_i 各自随机生成 $\langle r_C\rangle_i\in Z_{2^l}$;(2)通过"$\langle\cdot\rangle$ −分享"的乘法,利用 $\langle r_A\rangle_i,\langle r_B\rangle_i$ 计算出 $\langle r_{AB}\rangle_i$. 在线阶段:各方在本地计算 $\langle\delta_C\rangle_i$,计算公式为:$\langle\delta_C\rangle_i=i\cdot\delta_A\cdot\delta_B+\delta_A\cdot\langle r_B\rangle_i+\delta_B\cdot\langle r_A\rangle_i+\langle r_{AB}\rangle_i-\langle r_C\rangle_i$. 最后,各方相互交换 $\langle\delta_C\rangle_i$ 并获得 δ_C. 此过程形式化表示为 $[\![C]\!]=SecMatMul([\![A]\!],[\![B]\!])$.

4　系统模型

在本节中,我们将详细介绍 SecFedDMC 系统架构,威胁模型和设计目标.

4.1　系统架构

在 SecFedDMC 框架中,一组客户端集合 $\{C_1,C_2,\cdots,C_n\}$ 以及两个服务器进行了安全的联邦学习. 每个客户端 C_i 都拥有自己的本地数据集 $\mathcal{D}_i$,其中 $i\in[n]$. 所有客户端的训练数据集可以表示为 $\mathcal{D}=\bigcup_{i\in[n]}\mathcal{D}_i$. 客户端的共同目标是通过解决优化问题 $\min_w E_{\mathcal{D}}[L(\mathcal{D},w)]$ 来联合训练全局模型,其中 w 是全局模型的权重参数,L 是损失函数,例如交叉熵损失函数.

如图 3 所示,在 SecFedDMC 的系统模型中,每次迭代主要分为以下四个步骤:

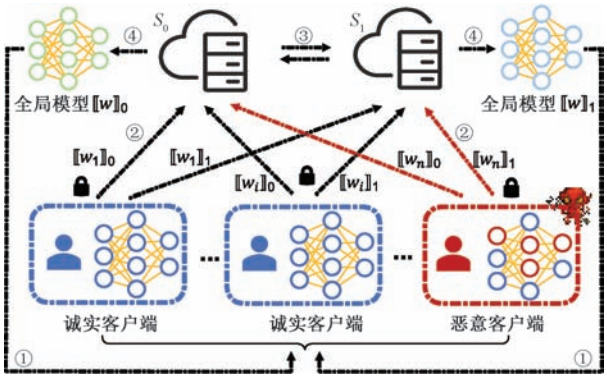

图 3　SecFedDMC 系统模型

(1)服务器 S_0 和 S_1 将"$[\![\cdot]\!]$ −分享"形式的最新全局模型 $[\![w^t]\!]_0$ 和 $[\![w^t]\!]_1$ 发送给各客户端,t 表示当前训练的轮次,当 $t=0$ 时,两个服务器初始化全局模型结构.

(2)每个客户端 C_i 在本地训练接收到两个服务器下发的全局模型份额,恢复出 $w^t=Rec([\![w^t]\!]_0,[\![w^t]\!]_1)$. 客户端在本地数据集 $\mathcal{D}_i$ 上,采用随机梯度下降(Stochastic Gradient Descent,SGD)算法将模型参数 w^t 更新为 $w_i^{t+1}=w^t-\eta\nabla L(\mathcal{D}_i,w)$,其中 η 是学习率,$\nabla L(\mathcal{D}_i,w)$ 是本地数据集 $\mathcal{D}_i$ 对当前全局模型权重 w^t 的梯度. 客户端更新后的本地模型参数 w_i^{t+1} 进行"$[\![\cdot]\!]$ −分享",分别将 $[\![w_i^{t+1}]\!]_0$ 和 $[\![w_i^{t+1}]\!]_1$

发送给服务器 S_0 和 S_1.

(3)将 n 个客户端上传的 $[\![w_i^{t+1}]\!]_0$ 和 $[\![w_i^{t+1}]\!]_1$ 表示为 $[\![W]\!]_0$ 和 $[\![W]\!]_1$,服务器 S_0 和 S_1 协同计算,从 $[\![W]\!]_0$ 和 $[\![W]\!]_1$ 中判别并过滤出恶意的模型参数.

(4)在判别和过滤完 M 个恶意的本地模型参数后,S_0 和 S_1 聚合所有诚实客户端的模型参数,更新全局模型.假设总共有 $n-M$ 个诚实的客户端,那么在服务器 $S_j, j \in \{0,1\}$ 上全局模型参数的更新可以通过以下公式进行:

$$[\![w]\!]_j = \frac{1}{n-M}\sum_{i=1}^{n-M}[\![w'_i]\!]_j \tag{1}$$

4.2 威胁模型

SecFedDMC 方案主要应对以下两种威胁:

(1)诚实且好奇的服务器.服务器 S_0 和 S_1 在正确执行协议时,可能尝试获取更多的私有信息.尽管 S_0 和 S_1 不会破坏联邦学习的执行过程,但由于存在的商业利益,可能会试图推断用户的隐私信息.

(2)恶意客户端.存在一些客户端可能通过发动拜占庭攻击,使用其私有数据,或发送任意的本地模型参数,试图在本地训练阶段进行篡改,从而破坏全局模型或窃取其他客户端的隐私信息.

在此方案中,假设恶意客户端的数量是有限的,恶意客户端的数量 M 不超过所有客户端总数 n 的一半,即 $M < n/2$.值得注意的是,一些客户端可能会利用全局模型尝试推断其他用户的隐私信息[42].针对这种攻击也有大量研究者进行了方案设计.例如,文献[43-45]设计了基于差分隐私的主动攻击防御方法,能够在一定程度上应对此类威胁.这些方案与本文研究工作是相互补充的,可以结合在一起使用.

4.3 设计目标

SecFedDMC 方案的设计目标着重考虑数据隐私性、模型鲁棒性和计算高效性:

(1)数据隐私性.SecFedDMC 方案的关键目标是确保每个客户端的本地模型参数的私密性,防止敏感信息的泄露.可能会有诚实且好奇的服务器推断客户端上传的模型参数,试图获取客户端的私有信息.因此,SecFedDMC 方案强调保护客户端隐私,保障其敏感信息的安全.

(2)模型鲁棒性.由于恶意参与者可能上传破坏全局模型的恶意模型参数,SecFedDMC 方案应具备良好的鲁棒性以应对这些潜在威胁.即使恶意客户端采取各种拜占庭攻击,也能够在模型聚合前尽可能移除恶意模型参数,确保全局模型的鲁棒性.

(3)计算高效性.SecFedDMC 方案的设计目标包括降低服务器的计算和通信开销.具体地,该方案在确保系统鲁棒性和安全性的前提下,应具备高效的恶意客户端检测和本地模型聚合能力.

5 隐私保护的拜占庭鲁棒联邦学习

本节首先设计了基础的拜占庭鲁棒联邦学习方案 FedDMC,然后在此基础上提出了隐私增强方案 SecFedDMC.

5.1 基础方案(FedDMC)

在联邦学习框架设计中,恶意客户端的有效识别是关键挑战.聚类方法能够自动将具有相似特性的客户端进行分类,因此成为恶意客户端检测的首选技术.然而,高维数据的聚类分析带来了一系列挑战,包括计算复杂度高和对噪声敏感等问题.在深度学习模型中,模型参数空间通常具有高维特性,而恶意客户端可能通过大幅度调整单个参数进行干扰,但对该模型与其他模型的欧氏距离影响却较小.这种情况为各类拜占庭攻击提供了有利环境.因此,降维操作显得尤为必要,既能够降低计算复杂度,也有助于减轻噪声的影响,进而有效地检测和防御恶意客户端的攻击.

因此,本文提出了一种恶意客户端检测技术 FedDMC.相较于传统联邦学习方案,FedDMC 在聚合模型参数之前需要对客户端的模型参数进行检测.该检测过程由降维和聚类两个模块组成,如算法 1 所示.

算法 1. FedDMC 框架

输入: 本地数据集 $\mathcal{D}_i$,通信轮数 T,客户端数 n,本地轮次 E,学习率 η,集成参数 α

输出: 全局模型 w

1. 服务器执行:
2. 初始化 w^0;
3. FOR $\{t = 0,1,\cdots,T-1\}\{$
4. FOR $\{i = 1,2,\cdots,n$ 并行执行$\}\{$
5. 将全局模型 w^t 发送给 C_i;
6. $w_i^{t+1} \leftarrow \mathrm{LocalTraining}(i, w^t)$;
7. $\}$
8. $W \leftarrow (w_0^{t+1},\cdots,w_n^{t+1})$;
9. $\widehat{W} \leftarrow \mathrm{RPCA}(W)$; //降维
10. $S_{[\cdot]} \leftarrow \mathrm{Kmeans}(\widehat{W})$; //聚类
11. $w^{t+1} \leftarrow \frac{1}{|S_{[\cdot]}|}\sum_j S_{[\cdot]}[j] \cdot w_j^{t+1}$; //聚合
12. $\}$

```
RETURN  w ;
客户端本地训练 (LocalTraining(i,w^t)) :
w_i^t ←w^t ;
IF { C_i 是恶意客户端}{
    执行拜占庭攻击.\\例如,LF−攻击
}ELSE{
    FOR{epoch=1,2,…,E}{
        w_i^{t+1} =w^t − ηΔ ℒ(w_i^t,D_i) ;
    }
}
RETURN w_i^{t+1} ;
```

5.1.1　降　维

在降维模块的设计中,选取了随机主成分分析方法(RPCA)[41],将模型参数从原始高维空间映射至较低的维度,旨在减少计算复杂性且尽可能保留主要特征.设有 n 个客户端在某一轮次中的模型参数矩阵 $\boldsymbol{W}=\{\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_n\}$,其中每个 $\boldsymbol{w}_i$ 对应于一个 d 维向量,代表第 i 个客户端的模型参数.我们将 RPCA 应用到参数矩阵 $\boldsymbol{W}\in\boldsymbol{R}^{n\times d}$,具体表示为 $\tilde{\boldsymbol{W}}=P_k(\boldsymbol{W})$,其中 P_k 代表 RPCA 投影操作符,k 是降低的维数($k\ll d$),其值可以根据具体的客户端数据分布和计算开销进行调整.$\tilde{\boldsymbol{W}}$ 是 RPCA 投影后得到的降维模型参数矩阵.

RPCA 具有较高的计算效率,是处理联邦学习中大量参数的理想选择.作为一种线性运算技术,RPCA 能有效地处理具有线性关系的数据,相较于随机投影和奇异值分解(Singular Value Decomposition, SVD),其解释性更强.更重要的是,RPCA 处理后的数据更能突出诚实客户端与恶意客户端之间的区别,进而使聚类算法更容易识别这两类客户端.相较于其他降维方法如 t-SNE[46]、isomap[47] 和 UMAP[48],RPCA 在计算效率上优势更为明显,非常适用于联邦学习环境中大规模参数的处理.

5.1.2　聚　类

在降维后的数据基础上,我们采用 K-means 聚类[49]方法对客户端进行分类.具体来说,对降维后的模型参数 $\tilde{\boldsymbol{W}}$ 执行 Kmeans 操作,记为:$S_{[t]}\leftarrow\text{Kmeans}(\tilde{\boldsymbol{W}})$,其中 $S_{[t]}$ 是一个二元集合,由 0 和 1 组成表示,作为第 t 轮的聚类结果.设定 K 值为 2,即将客户端聚为两类,分别为诚实客户端(标记为 1)和恶意客户端(标记为 0).在聚类结果中,数量较大的一类被认为是诚实客户端,数量较少的一类被认为是恶意客户端.这样的设置基于一个通用的合理假设[19],即在大多数实际情况下,诚实客户端的数量多于恶意客户端.

选择 K-means 聚类算法的原因主要有两点:首先,经 RPCA 降维后的数据在 K-means 聚类算法中能获得更高的结果精度.RPCA 通过保留数据的主要成分,有效滤除噪声并保留关键特征,有利于提高 K-means 算法的聚类精度.在高维数据中,由于存在大量的噪声和稀疏性,可能会降低 K-means 算法的效率和精度.然而,PCA 降维可以将数据映射到一个更低的维度空间,在这个空间中,数据的分布更为紧密,噪声的影响也随之减少,从而有助于提升 K-means 的聚类精度.其次,K-means 是一种简单且高效的聚类方法,适用于大规模数据的高效处理.在安全多方计算环境下,该算法依然能够高效执行,保证了联邦学习中数据的安全性.

5.2　隐私增强方案(SecFedDMC)

SecFedDMC 是基于 FedDMC 构建的隐私增强方案,其目标是安全地检测恶意客户端行为,并且不泄露客户端的隐私信息.该方案包含两个关键步骤:(1)通过构建安全的随机主成分分析(RPCA)算法实现高维模型主要特征提取;(2)利用安全聚类技术对所提取的主要特征进行分析,实现恶意客户端的检测.对于安全聚类技术,本文采用了文献[50]的 SecKmeans 方案.因此,本节重点描述如何安全高效地实现 RPCA 算法.

实现安全高效的 RPCA 算法存在以下两个挑战:(1)如何实现安全的 QR 分解;(2)如何实现安全的特征分解.为此,本文基于 iASS 密码学原语提出了安全的正交三角分解(*SecQR*)协议和安全的特征分解(*SecEigen*)协议.

5.2.1　*SecQR* 协议

SecQR 协议包含大量复杂的非线性计算,如除法和二范式.对于安全除法计算,主要性能瓶颈是求解 x 的倒数 $1/x$.本方案通过牛顿迭代法[51-52]近似计算倒数,迭代计算公式为 $y_{n+1}=y_n(2-xy_n)=2y_n-xy_ny_n$,$y_0=3\mathrm{e}^{0.5-x}+0.003$.

为了实现上述公式中 e^x 的安全计算,设计如下安全指数求解协议:$[\![e^x]\!]=SecExp([\![x]\!])$,即输入 $[\![x]\!]_i$,输出 $[\![e^x]\!]_i$.首先,服务器 S_0 初始化 $[\![u]\!]_0=e^{[\![x]\!]_0}$,$[\![v]\!]_0=0$,服务器 S_1 初始化 $[\![u]\!]_1=0$,$[\![v]\!]_1=e^{[\![x]\!]_1}$,然后双服务器调用 *SecMul* 协议计算得到 $[\![e^x]\!]=SecMul([\![u]\!],[\![v]\!])$.

基于安全指数求解协议,安全除法计算协议定义如下:$[\![a/b]\!]=SecDiv([\![a]\!],[\![b]\!])$,即输入 $[\![a]\!]$,$[\![b]\!]$,输出 $[\![a/b]\!]$.首先,双服务器计算 $[\![y_0]\!]=3\times$

$SecExp(\llbracket 0.5-b \rrbracket)+0.003$，然后，双服务器调用 iASS 的 3－输入连乘协议计算 xy_ny_n．因此，$\llbracket 1/b \rrbracket = \llbracket y_{n+1} \rrbracket = 2\times\llbracket y_n \rrbracket - SecMul(\llbracket x \rrbracket,\llbracket y_n \rrbracket,\llbracket y_n \rrbracket)$．该算法的迭代次数可根据误差要求进行调整，本方案默认设置为 10．最后，双服务器调用 $SecMul$ 协议计算得到 $\llbracket a/b \rrbracket = SecMul(\llbracket a \rrbracket,\llbracket 1/b \rrbracket)$．

二范式是向量中元素平方和的平方根．对于向量 $\boldsymbol{x}=(\boldsymbol{x}[0],\boldsymbol{x}[1],\cdots,\boldsymbol{x}[n])$，二范式的计算公式为

$$\|\boldsymbol{x}\|=\sqrt{\sum_{j=0}^{l}|\boldsymbol{x}[j]|^2} \tag{2}$$

对于二范式 $\|\boldsymbol{x}\|$ 密文计算，需要实现安全的平方根协议 $\llbracket \sqrt{\boldsymbol{x}} \rrbracket = SecSqrt(\llbracket \boldsymbol{x} \rrbracket)$，即输入 $\llbracket \boldsymbol{x} \rrbracket_i$，输出 $\llbracket \sqrt{\boldsymbol{x}} \rrbracket_i$．本文通过牛顿迭代法[52]近似计算平方根，具体公式为：$y_{n+1}=0.5\times(y_n+x/y_n)$，$y_0=x$．因此，服务器 S_i 首先将 $\llbracket \boldsymbol{y}_0 \rrbracket_i$ 初始化为 $\llbracket \boldsymbol{x} \rrbracket_i$，然后通过迭代计算得到 $\llbracket \sqrt{\boldsymbol{x}} \rrbracket = \llbracket \boldsymbol{y}_{n+1} \rrbracket = 0.5\times(\llbracket \boldsymbol{y}_n \rrbracket + SecDiv(\llbracket \boldsymbol{x} \rrbracket,\llbracket \boldsymbol{y}_n \rrbracket))$．在此基础上，安全的二范式计算协议定义如下：

$$\begin{aligned}\llbracket \|\boldsymbol{x}\| \rrbracket &= Sec2Norm(\llbracket \boldsymbol{x} \rrbracket)\\ &= SecSqrt\left(\sum_{j=0}^{l} SecMul(\llbracket \boldsymbol{x}[j] \rrbracket,\llbracket \boldsymbol{x}[j] \rrbracket)\right)\end{aligned} \tag{3}$$

为了安全地计算 $\boldsymbol{x}$ 的符号位，$SecSign$ 协议定义如下：

$$SecSign(\llbracket \boldsymbol{x} \rrbracket)=\begin{cases}\llbracket 1 \rrbracket & if\ x\geqslant 0\\ \llbracket -1 \rrbracket & if\ x<0\end{cases} \tag{4}$$

本方案首先采用文献[29]中最高有效位计算协议 $SecMSB$ 提取 $\boldsymbol{x}$ 的最高有效位（*Most Significant Bit*，*MSB*），其定义为

$$MSB(\boldsymbol{x})=\begin{cases}0 & if\ x\geqslant 0\\ 1 & if\ x<0\end{cases} \tag{5}$$

接下来，基于 $SecMSB$ 协议安全地计算 $\boldsymbol{x}$ 符号位．由于符号位 $\rho=-2\times MSB(\boldsymbol{x})+1$，则有 $\llbracket \rho \rrbracket = SecSign(\llbracket \boldsymbol{x} \rrbracket) = -2\times SecMSB(\llbracket \boldsymbol{x} \rrbracket)+1$．

基于上述安全计算协议，本节提出了安全的 QR 分解协议 $SecQR(\llbracket \boldsymbol{A}^{m\times n} \rrbracket)$，输入“$\llbracket\cdot\rrbracket$－分享”形式的 $m\times n$ 的矩阵 $\boldsymbol{A}$，输出矩阵 $\llbracket \boldsymbol{Q}^{m\times m} \rrbracket$ 和矩阵 $\llbracket \boldsymbol{R}^{m\times n} \rrbracket$，其中 $\boldsymbol{Q}^{m\times m}$ 和 $\boldsymbol{R}^{m\times n}$ 分别为上三角矩阵和正交矩阵，且 $A=QR$．具体实现如算法 2 所示．

算法 2. $SecQR$ 协议

输入：服务器 S_i 输入 $\llbracket \boldsymbol{A}^{n\times\gamma} \rrbracket$，其中 $n>\gamma$

输出：服务器 S_i 输出矩阵 $\llbracket \boldsymbol{Q}^{n\times\gamma} \rrbracket$ 和矩阵 $\llbracket \boldsymbol{R}^{\gamma\times\gamma} \rrbracket$

1. 初始化 $\llbracket \boldsymbol{Q} \rrbracket_0 = \boldsymbol{I}_n$，$\llbracket \boldsymbol{Q} \rrbracket_1 = \boldsymbol{0}$，$\llbracket \boldsymbol{R} \rrbracket_i = \llbracket \boldsymbol{A} \rrbracket_i$；
2. FOR $\{k=1 to n\}\{$
3. S_i 提取向量 $\llbracket \boldsymbol{x} \rrbracket_i = \llbracket \boldsymbol{R}[k:,k] \rrbracket_i$；
4. S_i 协作计算 $\llbracket \boldsymbol{y} \rrbracket_i = Sec2Norm(\llbracket \boldsymbol{x} \rrbracket_i)$；
5. S_i 计算符号位 $\llbracket \boldsymbol{\rho} \rrbracket_i = SecSign(\llbracket \boldsymbol{x}[0] \rrbracket_i)$；
6. S_i 计算 $\llbracket \boldsymbol{v} \rrbracket_i = \llbracket \boldsymbol{x}[0] \rrbracket_i - SecMul(\llbracket \boldsymbol{\rho} \rrbracket_i,\llbracket \boldsymbol{y} \rrbracket_i)$；
7. S_i 计算向量 $\llbracket \boldsymbol{u} \rrbracket_i = SecDiv(\llbracket \boldsymbol{x} \rrbracket_i,\llbracket \boldsymbol{v} \rrbracket_i)$；
8. S_i 计算 $\llbracket \boldsymbol{z} \rrbracket_i = SecDiv(\llbracket \boldsymbol{v} \rrbracket_i,\llbracket \boldsymbol{y} \rrbracket_i)$；
9. S_i 计算 $\llbracket \boldsymbol{\beta} \rrbracket_i = -SecMul(\llbracket \boldsymbol{\rho} \rrbracket_i,\llbracket \boldsymbol{z} \rrbracket_i)$；
10. S_i 更新子矩阵 $\llbracket \boldsymbol{R}_k \rrbracket_i$：$\llbracket \boldsymbol{R}_k \rrbracket_i = \llbracket \boldsymbol{R}_k \rrbracket_i - SecMul(\llbracket \boldsymbol{\beta} \rrbracket,\llbracket \boldsymbol{u} \rrbracket,\llbracket \boldsymbol{u}^{\mathrm{T}} \rrbracket,\llbracket \boldsymbol{R}_k \rrbracket_i)$；
11. S_i 更新子矩阵 $\llbracket \boldsymbol{Q}_k \rrbracket_i$：$\llbracket \boldsymbol{Q}_k \rrbracket_i = \llbracket \boldsymbol{Q}_k \rrbracket_i - SecMul(\llbracket \boldsymbol{Q}_k \rrbracket_i,\llbracket \boldsymbol{\beta} \rrbracket,\llbracket \boldsymbol{u} \rrbracket,\llbracket \boldsymbol{u}^{\mathrm{T}} \rrbracket)$；
12. $\}$
13. RETURN $\llbracket \boldsymbol{Q}^{n\times\gamma} \rrbracket,\llbracket \boldsymbol{R}^{\gamma\times\gamma} \rrbracket$；

5.2.2 $SecEigen$ 协议

特征分解是随机主成分分析（RPCA）的关键步骤，主要功能是将矩阵分解为一组特征值与特征向量的乘积．在明文算法中，特征值和特征向量的计算通常需要多次迭代．然而，在安全计算情形下，在秘密分享的矩阵元素上进行迭代计算，将导致双服务器间产生大量的交互，使得安全的特征分解操作效率低下．为了解决这一问题，本方案基于文献[53]提出的引理，提出了安全的特征分解协议 $SecEigen$，将安全的特征分解问题转化为简单的安全基础运算，从而降低计算和通信开销．具体引理如下：

引理 1[53]．如果 λ 是矩阵 $\boldsymbol{A}$ 的一个特征值，那么 $\varphi(\lambda)$ 也是矩阵 $\varphi(\boldsymbol{A})$ 的一个特征值（其中 $\varphi(\lambda)=a_0+a_1\lambda+\cdots+a_m\lambda^m$ 是 λ 的多项式）．另外，如果向量 $\boldsymbol{v}$ 是矩阵 $\boldsymbol{A}$ 在特征值 λ 下的一个特征向量，那么向量 $\boldsymbol{v}$ 也是矩阵 $\varphi(\boldsymbol{A})$ 在特征值 $\varphi(\boldsymbol{\lambda})$ 下的特征向量．

引理 2[53]．如果 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 是相似矩阵（即 $\boldsymbol{A}\sim\boldsymbol{B}$），并且 $\boldsymbol{B}=\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}$，则矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 具有相同的特征值．如果在矩阵 $\boldsymbol{A}$ 的特征值 λ 下存在特征向量 $\boldsymbol{v}$，则 $\boldsymbol{P}^{-1}\boldsymbol{v}$ 是 $\boldsymbol{B}$ 在特征值 λ 下的特征向量．

SecEigen 协议利用随机矩阵 P 和多项式函数 $\varphi(\cdot)$ 将输入的秘密分享矩阵 $\llbracket \boldsymbol{X}^{\rho\times\rho} \rrbracket$ 盲化为矩阵 $\llbracket \boldsymbol{Y}^{\rho\times\rho} \rrbracket$，然后恢复出矩阵 $\boldsymbol{Y}$ 并计算其特征值和特征向量．接下来，根据上述引理，利用反函数 $\varphi^{-1}(\cdot)$ 和矩阵 $\boldsymbol{P}$ 安全地计算出矩阵 $\boldsymbol{X}$ 的特征值和特征向量．函数 $\varphi(\cdot)$ 和 $\varphi^{-1}(\cdot)$ 的生成见参考文献[54]．算法 3 给出了安全特征分解（$SecEigen$）协议的详细过程．

算法 3. $SecEigen$ 协议

输入：服务器 S_i 输入 $\llbracket \boldsymbol{X}^{\gamma\times\gamma} \rrbracket$，其中 $rank(\boldsymbol{X})=\gamma$

输出：　S_i 得到特征值 $[\![\boldsymbol{\lambda}^j]\!]_i$ 的一份份额和特征向量 $[\![\boldsymbol{V}^{\gamma\times\gamma}]\!]_i$ 的一份份额,其中 $j\in[1,\gamma]$

1. 离线阶段：
2. 两方服务器 S_i 协商生成函数 $\varphi(\cdot)$ 与 $\varphi^{-1}(\cdot)$、"$[\![\cdot]\!]$ 一分享"形式的方阵 $\boldsymbol{P}$ 和 $\boldsymbol{P}^{-1}$,即 $[\![\boldsymbol{P}]\!]_i$ 和 $[\![\boldsymbol{P}^{-1}]\!]_i$；
3. 在线阶段：
4. S_i 协作计算 $[\![\boldsymbol{Y}]\!]_i = SecMatMul([\![\boldsymbol{P}^{-1}]\!]_i, [\![\boldsymbol{X}]\!]_i, [\![P]\!]_i)$；
5. S_i 利用安全乘法协议协作计算 $[\![\boldsymbol{Y}]\!]_i = \varphi([\![\boldsymbol{Y}]\!]_i)$；
6. S_0 将 $[\![\boldsymbol{Y}]\!]_i$ 发送给 S_1,然后 S_1 计算 $\boldsymbol{Y}$ 的特征值 $\varphi\boldsymbol{\lambda}^j$ 和特征向量 $\boldsymbol{V}'$,并将结果发送回 S_i；
7. S_i 对所有的 $j\in[1,\gamma]$,计算 $[\![\boldsymbol{\lambda}^j]\!]_i=\varphi^{-1}(t\boldsymbol{\lambda}^j)$；
8. S_i 计算 $[\![\boldsymbol{V}]\!]_i=[\![\boldsymbol{P}]\!]_i\cdot\boldsymbol{V}'$；

基于上述协议即可完整实现 SecFedDMC 方案. SecFedDMC 方案主要由双服务器执行阶段和客户端本地训练阶段构成. 在双服务器执行阶段,首先进行安全降维处理,然后执行安全聚类和安全聚合操作. 需要说明的是,降维后得到的参数矩阵 $\widetilde{W}$ 仅用于聚类以识别恶意客户端,在识别并剔除恶意客户端后,对原始参数矩阵 W 进行安全聚合. 在本地训练阶段,客户端进行模型训练,并将秘密分享的参数上传至云服务器. 具体实现如算法 4 所示.

算法 4.　SecFedDMC 框架.

输入：　本地数据集 $\mathcal{D}_i$,通信轮数 T,客户端数 n,本地训练轮次 E,学习率 η,集成参数 α

输出：　全局模型 $[\![w]\!]$

1. 双服务器执行：
2. 初始化 $[\![w^0]\!]$,映射矩阵 $[\![Q^{n\times\rho}]\!]$；
3. FOR $\{t=0,1,\cdots,T-1\}\{$
4. FOR $\{i=1,2,\cdots,n$ 并行执行$\}\{$
5. 将全局模型 $[\![w^t]\!]$ 发送给 C_i；
6. $[\![w_i^{t+1}]\!]\leftarrow$ LocalTraining$(i,[\![w^t]\!])$ //本地训练;
7. $\}$
8. $[\![W^{n\times d}]\!]\leftarrow([\![w_0^{t+1}]\!],\cdots,[\![w_n^{t+1}]\!])$；
9. $[\![\widetilde{W}^{n\times k}]\!]\leftarrow SecRPCA([\![W^{n\times d}]\!])$；$\{$ //安全降维
10. $[\![W^{n\times d}]\!]\leftarrow Means([\![W^{n\times d}]\!])$；//中心化
11. FOR $\{i=1,\cdots,p\}\{$//迭代优化
12. $[\![Q^{n\times\rho}]\!]\leftarrow SecMatMul([\![W]\!],[\![W^T]\!],[\![Q]\!])$；
13. $[\![Q^{n\times\rho}]\!]\leftarrow SecQR([\![Q^{n\times\rho}]\!])$；
14. $\}$
15. $[\![Q^{d\times\rho}]\!]\leftarrow SecMatMul([\![W^T]\!],[\![Q^{n\times\rho}]\!])$；
16. $[\![Q^{d\times\rho}]\!]\leftarrow SecQR([\![Q^{d\times\rho}]\!])$；
17. $[\![B^{\rho\times\rho}]\!]\leftarrow SecMatMul([\![Q^T]\!],[\![W^T]\!],[\![W]\!],[\![Q]\!])$；
18. $[\![\widetilde{U}]\!]\leftarrow SecEigen([\![B^{\rho\times\rho}]\!])$；
19. $[\![Q^{d\times k}]\!]\leftarrow SecMatMul([\![Q^{d\times\rho}]\!],[\![\widetilde{U}[:k]]\!])$；
20. $[\![\widetilde{W}]\!]\leftarrow SecMatMul([\![W]\!],[\![Q^{d\times k}]\!])$；
21. $\}$
22. $[\![S_{[\cdot]}]\!]\leftarrow SecKmeans([\![\widetilde{W}]\!])$；//安全聚类
23. $[\![w^{t+1}]\!]\leftarrow\frac{1}{|S_{[\cdot]}|}\sum_j SecMul(S_{[\cdot]}[j], w_j^{t+1})$；
24. $\}$
25. RETURN $[\![w]\!]$；
26. 客户端本地训练 LocalTraining$(i,[\![w^t]\!])$：
27. $w_i^t\leftarrow Rec([\![w^t]\!]_0,[\![w^t]\!]_1)$；
28. $w_i^{t+1}\leftarrow$ Training(w_i^t)；//模型训练
29. 计算 w_i^{t+1} 的秘密分享 $[\![w_i^{t+1}]\!]$；
30. RETURN $[\![w_i^{t+1}]\!]$；

6　理论分析

6.1　安全性分析

本节在标准的半诚实模型下对方案进行安全性证明,并且采用自下而上的证明方式:(1) SecFedDMC 所使用的基础操作模块,即安全线性操作、*SecMul*、*SecMatMul* 在半诚实模型下均是安全的,其形式化证明参见文献[55]. 对于第 5.2.1 节中描述的 *SecMSB* 协议,其安全性的形式证明参见文献[56]. (2)证明基于基础操作模块和 *SecMSB* 协议的 *SecExp*、*SecDiv*、*Sec2Norm* 以及 *SecSign* 等协议的安全性. (3)证明 SecFedDMC 框架的安全性,对于聚类,SecFedDMC 框架引用了文献[50]的安全聚类协议,可证明其安全性. 对于 RPCA 协议,在 SecFedDMC 框架中主要有两个核心协议：*SecQR* 和 *SecEigen*. 因此,本节重点将会证明 *SecQR* 和 *SecEigen* 的安全性.

半诚实模型下两方安全协议安全性定义如下：

定义 1. 假设协议客户端 S_0 和 S_1 计算函数 $f:\{0,1\}^*\times\{0,1\}^*\rightarrow f:\{0,1\}^*\times f:\{0,1\}^*$,其中 $f_i(x,y),i=\{0,1\}$ 是 $f(x,y)$ 的第 i 个元素,x 和 y 分别为两个客户端的输入. Π 是客户端 S_0 和 S_1 计算 f 的安全两方计算协议. $V_0^{\Pi}(x,y)=(x,r^0,m_1^0,\cdots,m_t^0)$ 和 $V_1^{\Pi}(x,y)=(y,r^1,m_1^1,\cdots,m_t^1)$ 分别表示客户端 S_0 和 S_1 在协议执行过程中收到的视图. 其中,r^i 表示 S_i 产生的随机数,$m_j^i,1\leqslant j\leqslant t$ 表示 S_i 收到的第 j 条消息. $O^{\Pi}=O_0^{\Pi},O_1^{\Pi}$ 表示协议 Π 运行结束之后的输出,其中 O_i^{Π} 表示 S_i 的输出.

对于更一般的函数 f,我们称协议 Π 在半诚实

模型下安全地计算了当且仅当存在概率多项式的模拟器 Sim_0 和 Sim_1 分别模拟协议执行函数，其满足：

$$\{(Sim_0(x,f_0(x,y)),f(x,y))\}_{x,y} \cong (V_0^{\Pi}(x,y),O^{\Pi}(x,y))_{x,y} \quad (6)$$

$$\{(Sim_1(x,f_1(x,y)),f(x,y))\}_{x,y} \cong (V_1^{\Pi}(x,y),O^{\Pi}(x,y))_{x,y} \quad (7)$$

其中，$\cong$ 表示计算不可区分. 显然，根据上述定理，加密协议对半诚实的对手是安全的. 接下来，正式证明所提协议的安全性，其详细的过程如下：

定理 1. 如果 *SecMul* 在半诚实的模型中是安全的，则 *SecExp* 在半诚实的模型中是安全的.

证明. *SecExp* 协议将 S_0 和 S_1 接收的 $[\![\boldsymbol{x}]\!]_0$ 和 $[\![\boldsymbol{x}]\!]_1$ 作为输入，输出 $[\![e^x]\!]_0$ 和 $[\![e^x]\!]_1$. 为了证明协议安全性，首先考虑 S_0 被攻击者俘获的情况，为此构造模拟器 Sim_0 模拟 S_0 的执行函数. 对于 S_0 运行 *SecExp* 协议时，S_0 实际上执行的是 *SecMul* 协议. 因此，模拟器 Sim_0 本质上模拟的执行协议是 *SecMul* 协议. 模拟器 Sim_0 从 Z_{2^l} 上随机选取一个随机数 $\tilde{\boldsymbol{x}}_0$ 模拟 S_0 的输入视图 $V_0=[\![\boldsymbol{x}]\!]_0$，将 Sim_0 模拟出的输入视图记为 $\tilde{V}_0=\tilde{\boldsymbol{x}}_0$. 然后，模拟器 Sim_0 模拟 S_0 本地设置 $[\![\boldsymbol{u}]\!]_0=e^{\tilde{\boldsymbol{x}}_0}$，$[\![\boldsymbol{v}]\!]_0=0$，执行 *SecMul* 协议计算出输出视图 $\tilde{O}_0=[\![e^{\tilde{x}_0}]\!]_0$. 由于 S_0 计算视图 $V_0=[\![\boldsymbol{x}]\!]_0$ 和 $O_0=[\![\boldsymbol{e}^x]\!]_0$ 在 Z_{2^l} 上是随机的，模拟器 Sim_0 模拟视图 $\tilde{V}_0=\tilde{x}_0$ 和 $\tilde{O}_0=[\![e^{\tilde{x}_0}]\!]_0$ 在 Z_{2^l} 上也是随机的，并且 *SecMul* 在文献[55]已被证明是安全的，所以模拟的视图 $\tilde{V}_0$ 和 $\tilde{O}_0$ 与实际的视图 V_0 和 O_0 是不可区分的，即公式(6)成立. 通过上述类似方法可以对被攻击者俘获 S_1 的情况进行模拟，因为 S_0，S_1 在 *SecExp* 协议中执行过程是对称的. 因此，*SecExp* 协议在半诚实模型下是安全的.

证毕.

定理 2. 如果 *SecMul*、*SecExp* 协议在半诚实的模型中是安全的，那么 *SecDiv* 协议在半诚实的模型中是安全的.

证明. *SecDiv* 协议本质在协议运行中执行的基础的 *SecMul* 和 *SecExp* 协议. *SecMul* 在文献[55]已被证明是安全的，且定理 1 成立. 与 *SecExp* 协议类似，模拟器 Sim_0 和 Sim_1 能够通过定理 1 中类似方法，对被攻击者俘获 S_0 或 S_1 的情况进行模拟. 因此，*SecDiv* 协议在半诚实模型下是安全的.

证毕.

定理 3. 如果 *SecMul*、*SecDiv* 协议在半诚实的模型中是安全的，那么将 *Sec2Norm* 协议在半诚实的模型中是安全的.

证明. *Sec2Norm* 协议由 *SecMul* 和 *SecDiv* 协议组成. *SecMul* 已被证明是安全的，且定理 2 成立. 因此，模拟器 Sim_0 和 Sim_1 能够通过定理 1 中类似方法，对被攻击者俘获 S_0 或 S_1 的情况进行模拟. 因此，*Sec2Norm* 协议在半诚实模型下是安全的.

证毕.

定理 4. 如果 *SecMSB* 协议在半诚实的模型中是安全的，那么 *SecSign* 协议在半诚实的模型中是安全的.

证明. *SecSign* 协议将 S_0 和 S_1 接收的 $[\![\boldsymbol{x}]\!]_0$ 和 $[\![\boldsymbol{x}]\!]_1$ 作为输入，输出 $[\![\boldsymbol{\rho}]\!]_0$ 和 $[\![\boldsymbol{\rho}]\!]_1$. 为了证明协议安全性，首先考虑 S_0 被攻击者俘获的情况，为此构造模拟器 Sim_0 模拟 S_0 的执行函数. S_0 运行 *SecSign* 协议时，实际上执行 *SecMSB* 协议. 因此，模拟器 Sim_0 本质上模拟的执行协议是 *SecMSB* 协议. 模拟器 Sim_0 从 Z_{2^l} 中随机选取一个随机数 x'_0 模拟 S_0 的输入视图 $V_0=[\![x]\!]_0$，将 Sim_0 模拟的输入视图记为 $\tilde{V}_0=x'_0$. 然后，模拟器 Sim_0 模拟 S_0 执行 *SecMSB* 协议输出视图 $\tilde{O}_0=\rho'_0$. 由于 S_0 的输入视图 $V_0=[\![\boldsymbol{x}]\!]_0$ 和输出视图 $O_0=[\![\boldsymbol{\rho}]\!]_0$ 在 Z_{2^l} 上是随机的，模拟器 Sim_0 模拟视图 $\tilde{V}_0=x'_0$ 和 $\tilde{O}_0=\rho'_0$ 在 Z_{2^l} 上也是随机的，并且 *SecMSB* 协议在文献[56]已被证明是安全的，所以模拟的视图 $\tilde{V}_0$ 和 $\tilde{O}_0$ 与实际的视图 V_0 和 O_0 是不可区分的，即公式(6)成立. 通过上述类似方法可以对被攻击者俘获 S_1 的情况进行模拟，因为 S_0，S_1 在 *SecSign* 协议中执行过程是对称的. 因此，*SecSign* 协议在半诚实模型下是安全的.

证毕.

定理 5. 如果 *SecMul*、*SecMatMul*、*SecExp*、*SecDiv*、*Sec2Norm* 以及 *SecSign* 在半诚实的模型中是安全的，那么 *SecQR* 和 *SecEigen* 在半诚实的模型中是安全的.

证明. *SecQR* 协议由 *SecMul*、*SecMatMul*、*SecExp*、*SecDiv*、*Sec2Norm* 以及 *SecSign* 组成. *SecEigen* 由 *SecMul* 和 *SecMatMul* 组成. 除了输入和输出，*SecQR* 的 S_0 和 S_1 交互的消息是各个上述协议的中间值与输出值. 根据定理 1－4 可知，这些协议的中间值与输出值均是随机独立且计算不可区分的. 因此，在 S_0 或 S_1 被攻击者俘获的情况下，模拟器 Sim_0 或模拟器 Sim_1 只需独立选择随机数即

可对这些协议的中间值与输出值进行模拟. 因此, *SecQR* 在半诚实模型下是安全的. 同理, *SecEigen* 在半诚实的模型中也是安全的.

证毕.

定理 6. 如果 *SecMatMul*, *SecQR* 和 *SecEigen*, *SecKmeans* 协议在半诚实的模型中是安全的,那么 SecFedDMC 框架在半城实的模型中是安全的.

证明. SecFedDMC 框架主要由 *SecMatMul*, *SecQR*, *SecEigen* 和 *SecKmeans* 协议组成. 其中, *SecKmeans* 协议的安全性已在文献[50]证明. 除了输入输出,SecFedDMC 协议的 S_0 和 S_1 交互的消息是 *SecMatMul*, *SecQR*, *SecEigen* 和 *SecKmeans* 协议的中间值与输出值. 根据定理 5 可知,这些协议的中间值与输出值均是随机独立且计算不可区分的. 因此,在 S_0 或 S_1 被攻击者俘获的情况下,模拟器 Sim_0 或模拟器 Sim_1 只需独立选择随机数即可对这些协议的中间值与输出值进行模拟. 因此,SecFedDMC 框架在半诚实模型下也是安全的.

证毕.

6.2　复杂性分析

本节对 SecFedDMC 方案进行了复杂性分析. 假设存在 n 个客户端,每个客户端的模型参数有 m 个维度,且训练轮次为 T 轮, ℓ 表示 IASS 中元素的位长.

首先,分析所使用的基础操作,即安全线性操作、*SecMul*、*SecMSB* 以及 *SecMatMul* 协议. 对于安全线性操作,双服务器在本地执行该协议,没有任何通信开销. 对于 *SecMul* 和 *SecMatMul* 协议,双服务器计算两个 $n\times m$ 矩阵的计算结果仅需交互 1 轮,通信开销为 $n\times m\times\ell$ 比特. 对于 *SecMSB* 协议,采用了 ABY2.0 的方法,其协议执行过程仅需交互 $\log_4\ell$ 轮,通信开销为 $nm(4/3\ell^2+\ell)$ 比特.

其次,分析基于基础操作构建的协议,即 *SecExp*、*SecDiv*、*Sec2Norm* 以及 *SecSign* 协议. 对于 *SecExp* 协议,双服务器计算两个 $n\times m$ 矩阵的计算结果仅需交互 1 轮,通信开销为 $n\times m\times\ell$ 比特. 对于 *SecDiv* 协议,每轮迭代包含 2 次 *SecMul* 操作. 为了降低上述开销,本文使用 ABY2.0 的三输入乘法协议对 *SecDiv* 进行优化,使该协议的通信开销减半. 假设 *SecDiv* 协议迭代了 k 次,则 *SecDiv* 协议需要双服务器交互 k 轮,通信开销为 $knm\ell$ 比特. 对于 *Sec2Norm* 协议,与 *SecDiv* 协议的分析方法类似,仅需双服务器交互 k^2 轮,通信开销为 $nmk^2\ell$ 比特. 对于 *SecSign* 协议,仅需交互 $\log_4\ell$ 轮,通信开销为 $2nm(4/3\ell^2+\ell)$ 比特.

接下来,继续分析 SecFedDMC 方案中的核心算法 *SecQR* 和 *SecEigen* 协议的复杂度. *SecQR* 协议调用了 1 次 *Sec2Norm* 协议,1 次 *SecSign* 协议,4 次 *SecMul* 协议和 2 次 *SecDiv* 协议. 因此, *SecQR* 协议需要双服务器交互 k^2+2k+5 轮,通信开销为 $((4/3+k^2)\ell^2+(5+2k)\ell)nm$ 比特. *SecEigen* 协议调用了 3 次 *SecMatMul* 协议和 1 次 *Rec* 协议. 因此, *SecEigen* 协议仅需双服务器交互 2 轮,通信开销为 $4nm\ell$ 比特.

最后,分析 SecFedDMC 方案的整体复杂度. 该方案每训练一次调用 $(p+5)$ 次 *SecMatMul* 协议, $(p+1)$ 次 *SecQR* 协议,1 次 *SecEigen* 协议和 1 次安全聚类协议. 其中, p 为求解映射矩阵的迭代次数. 本方案使用的安全聚类协议[50]需要双服务器交互 $2+\varepsilon\ell$ 轮,消耗 $(2+\varepsilon\ell)(2+\ell)\varepsilon$ 比特通信成本,其中 k 表示聚类算法的迭代次数. 除此以外,其他协议需要双方服务器通信交互 $(p+5)+(p+1)(k^2+2k+5)+2$ 轮次,消耗 $(p+5)\ell+nm(p+1)(4/3+k^2)\ell^2+nm(p+1)(5+2k)\ell+4nm\ell$ 比特通信成本. 若 SecFedDMC 框架迭代 T 轮,则该框架需要双方服务器交互轮次为 $(p+5)T+(p+1)(k^2+2k+5)T+2T+2+k\ell T$,通信开销为 $(p+5)\ell T+nm(p+1)(4/3+k^2)\ell^2T+nm(p+1)(5+2k)\ell T+4nm\ell T+(2+\varepsilon\ell)(2+\ell)\varepsilon T$ 比特.

7　实验设置

SecFedDMC 采用 PyTorch 进行实现,密文实现采用安全多方计算库 MPCTensorLib①,并在装备有 2 个 NVIDIA RTX 3090 GPU、Intel i9－10900K CPU、128 GB 内存、Ubuntu 18.04 系统的 2 台服务器上完成实验. 本节详述了实验设置,包括数据集、本地模型结构、攻击方法、用于对比的检测方法,以及用于评估的指标.

7.1　数据集和模型架构

本文在三个标准基准数据集上评估了 SecFedDMC 的性能,并对应设置了三种不同的网络架构.

MNIST. 手写数字数据集,由 60000 个训练示例和 10000 个测试示例组成,每个示例为 28×28 的灰度图片. 网络模型采用多层感知器(MLP),其中隐藏层包含 100 个神经元.

① MPCTensorLib, https://gitee.com/xdnss/mpctensorlib

EMNIST. 字母和数字组成的分类数据集，分别包含 697932 张灰度图像，共有 62 个类别. 网络模型采用卷积神经网络(CNN). 此 CNN 模型有 5 个隐藏层，包括：两个带有线性修正单元 (Rectified Linear Unit，ReLU) 激活的卷积层、两个 2×2 的池化层和一个全连接层.

CIFAR10. 10 个类别的彩色图像分类数据集，数据集包括 50000 个训练示例和 10000 个测试示例，每个示例的大小为 32×32. 网络模型采用广泛使用的 ResNet18 网络架构[57].

根据先前的文献[22]，本文设置了 100 个客户端，并利用狄利克雷(Dirichlet)分布来创建本地客户端数据集以实现 non-IID 分布. 具体来说，本文从狄利克雷分布中获取 $q_j \sim DirN(\beta)$，并将 j 类比例为 q_j 的示例分配给客户端 i. 其中参数 β 可以控制客户端之间数据的不平衡程度，数值越小越不平衡. 在默认情况下，β 设置为 5.

7.2 攻击设置

本文默认随机选择 28 个客户端作为恶意客户端. 采用的攻击方法包括标签翻转攻击[17]、高斯攻击[33]、LIE 攻击[11]、缩放攻击[9]和女巫攻击[21].

标签翻转攻击(LF-attack)[17]. 恶意客户端会翻转特点训练样本的标签. 具体来说，将标签 l 随机翻转为 l'，其中 $l' \in \mathrm{L}, l \neq l'$，L 是所有标签的集合.

高斯攻击(GS-attack)[33]. 恶意客户端在本地模型参数上添加高斯噪声. 高斯噪声根据每层模型参数的均值和方差生成.

LIE-攻击(LIE-attack)[11]. 恶意客户端在本地数据中加入标记并修改标签后训练本地模型. 使用最大值 $z^{max} \leftarrow \max_z\{\Phi(z) < n-M-s/n-M\}$ 来规避防御.

缩放攻击(Scaling-attack)[9]. 恶意客户端在本地数据中加入标记并修改标签. 利用篡改的训练模型，之后再利用缩放技术将模型的权重缩放到防御方法允许的范围，以防止恶意参数被服务器平均参数淡化.

女巫攻击(Sybil-attack)[21]. Sybil 攻击是攻击者控制了多个参与方并发起协作攻击. 本文假设这些恶意参与方协作发动标签反转攻击，即所有恶意客户端均将标签 l 翻转为 l'.

7.3 比较的防御方法

本文设置的对比防御方法包括 PPRAgg[33]、PEFL[33]、LSFL[37]和 BREA[38]. 这些方法不需要服务器有可信数据集，可以输出恶意客户端列表来评估检测效果. 在比较实验过程中，使用相同的模型结构、学习率(η)、批量大小(B)、数据分布参数(β)和本地训练轮次(E)，仅替换不同防御方法. 防御方法与之前论文保持相同的设置，以确保实验的公平性. 表 2 总结了实验的默认配置.

表 2 默认参数设置

参数	默认值
学习率(η)	0.01
批量大小(B)	128
客户数量(N)	100
恶意客户端数量(M)	28
通信轮数(T)	200
本地训练轮次(E)	1
数据分布集中参数(β)	5

7.4 评估指标

在介绍评估指标之前，给出以下定义. True Positive (TP)：客户端是恶意的且被预测为恶意的；False Positive(FP)：客户端是诚实的但被预测为恶意的；True Negative (TN)：客户端是诚实的且被预测为诚实的；False Negative (FN)：客户端是恶意的但被预测为诚实的. 在本文采用一组指标来评估防御技术的有效性.

检测准确率(*Detection Accuracy Rate*，DAR)，表示所有正确预测占样本总数(包括恶意和诚实客户端)的比例：

$$DAR = \frac{TP + TN}{TP + FP + TN + FN}.$$

尽管检测准确率可以反映恶意客户端识别结果，但其并非评价识别效果的最佳指标. 这是由于不同识别结果对聚合结果的影响程度不同. 例如，被错误标识为恶意的诚实客户端对全局准确性的影响较轻，而被误识别为诚实的恶意客户端则可能严重降低全局准确性.

检测精确率(*Detection Precision Rate*，DPR)，表示正确预测的恶意客户端数量占所有预测的恶意客户端数量的比例：

$$DPR = \frac{TP}{TP + FP}.$$

召回率(*Recall Rate*，RR)，表示正确预测恶意客户端数量占真实恶意客户端数量的比例：

$$RR = \frac{TP}{TP + FN}.$$

为了衡量全局模型的学习效果，本文采用模型的测试精度(Test Accuracy，TACC)作为评价指标，即全局模型正确分类测试实例的比例. 此外，对

于有目标的特定毒化攻击,如“LIE-attack”和“Scaling-attack”,采用攻击成功率(Attack Success Rate,ASR)评估全局模型的性能.具体操作为,触发器会被嵌入到每一个测试样本中,当测试样本被全局模型分类为目标标签时,便认定该攻击为成功.ASR代表被成功攻击的测试输入的比例,若ASR较低则说明目标模型对毒化攻击的防御能力较强.所有的实验结果均为五次试验的平均值.

8　实验评估

本节对SecFedDMC框架进行评估,主要从以下四个方面开展:恶意客户端的检测准确率、全局模型的精度、消融实验以及计算开销.

8.1　恶意客户端的检测结果

本节首先对比SecFedDMC与其他检测方法在不同攻击、不同数据集情况下的检测结果,然后研究不同恶意客户端数量的影响以及非独立同分布对防御的影响.

8.1.1　SecFedDMC与其他检测方法对比

表3展示了本文方法与四个先进方法在三个数据集的恶意客户端检测结果比较.从表3可以看出,在不同的攻击设置下,SecFedDMC都优于现有方法,检测结果最优.BREA采用Krum的检测方式,选择与其他客户端最相似的M个模型参数,LIE-attack与Sybil-attack都会协作恶意客户端发起攻击,因此Krum会误选恶意客户端作为诚实用户.LSFL检测方法是基于K-近邻算法来检测恶意客户端的防御方法.然而,该方法有明显的缺陷,例如当数据分布在某种特定的分布时,K-近邻算法检测恶意客户端结果不佳.LSFL检测方法仅对LF-attack有效,而对其他攻击方法效果不好.PEFL对LIE-attack有效,对其他攻击方法的效果差.这是因为PEFL假设恶意客户端之间的皮尔逊相似度小于诚实和恶意客户端之间的相似度.然而,恶意客户端的参数在大多数情况下是未知并且无法估计的,因此恶意用户之间的皮尔逊相似度不一定小.例如,在GS-attack中,诚实客户端的参数更相似,而恶意客户端的参数在分布上更分散.PPRAgg是现有防御方法中最先进的,但在LIE-attack下PPRAgg表现不佳.相较于PPRAgg,在CIFAR10数据集上,拜占庭攻击节点检测准确率提升12%~24%.与传统的标签反转攻击不同,Sybil攻击中所有恶意客户端的攻击行为是一致的,攻击强度增加.在Sybil攻击场景下,其他防御方法相对于传统标签反转攻击,检测准确率有所下降,而SecFedDMC的检测准确率(DAR)达到100%.这是因为恶意客户端具有一致的攻击行为,使得其更易被聚类算法识别.

表3　不同攻击和不同检测方法下的恶意客户端检测结果

数据集	检测方法	LF-attack			GS-attack			LIE-attack			Scaling-attack			Sybil-attack		
		DAR	DPR	RR	DAR	DPR	RR	DAR	DPR	RR	DAR	DPR	RR	DAR	DPR	RR
MNIST	BREA	0.94	0.89	0.89	0.88	0.79	0.79	0.64	0.21	0.11	0.82	0.65	0.79	0.45	0.06	0.07
	LSFL	0.93	0.89	0.86	0.71	0.47	0.29	0.61	0.38	0.64	0.63	0.34	0.36	0.93	0.86	0.89
	PEFL	0.37	0.21	0.46	0.13	0.05	0.11	0.79	1.00	0.25	0.42	0.16	0.25	0.46	0.32	0.86
	PPRAgg	0.52	0.34	0.75	0.81	0.85	0.39	0.32	0.05	0.07	0.73	0.52	0.54	0.51	0.34	0.82
	SecFedDMC	**0.99**	**0.97**	**1.00**	**0.92**	**0.92**	**0.82**	**0.98**	**1.00**	**0.93**	**0.93**	**0.89**	**0.86**	**1.00**	**1.00**	**1.00**
EMNIST	BREA	0.89	0.81	0.79	0.92	0.86	0.86	0.44	0.09	0.11	0.89	0.84	0.75	0.57	0.17	0.14
	LSFL	0.91	0.85	0.82	0.72	0.50	0.25	0.71	0.48	0.54	0.56	0.26	0.32	0.75	0.54	0.79
	PEFL	0.25	0.02	0.04	0.14	0.09	0.21	0.93	1.00	0.75	0.35	0.14	0.25	0.21	0.23	0.75
	PPRAgg	0.92	0.92	0.79	0.89	0.84	0.75	0.53	0.31	0.54	0.84	0.70	0.75	0.82	0.62	0.93
	SecFedDMC	**0.96**	**0.96**	**0.89**	**0.97**	**0.93**	**0.96**	**0.93**	**0.96**	**0.79**	**0.95**	**0.93**	**0.89**	**1.00**	**1.00**	**1.00**
CIFAR10	BREA	0.64	0.35	0.32	0.60	0.29	0.29	0.44	0.06	0.07	0.89	0.84	0.75	0.41	0.12	0.18
	LSFL	0.82	0.75	0.54	0.74	0.57	0.29	0.63	0.24	0.14	0.91	0.81	0.89	0.81	0.61	0.89
	PEFL	0.38	0.16	0.29	0.76	0.67	0.29	0.35	0.28	0.86	0.26	0.27	0.93	0.27	0.25	0.79
	PPRAgg	0.75	0.55	0.64	0.84	0.93	0.46	0.79	0.63	0.61	0.77	0.57	0.71	0.84	0.65	0.93
	SecFedDMC	**0.99**	**1.00**	**0.96**	**0.96**	**0.96**	**0.89**	**0.94**	**0.96**	**0.82**	**0.99**	**0.97**	**1.00**	**1.00**	**1.00**	**1.00**

8.1.2　不同恶意客户端数量的影响

本文在EMNIST数据集上进一步探究了恶意客户端数量对不同检测方法准确率的影响.如图4所示,在恶意客户端数量增加的情况下,除SecFedDMC外,其他防御方法的检测准确率均明显下降.值得注意的是,某些防御方法仅在特定的攻击下有效.

表 4 的实验结果表明：(1)SecFedDMC 在各类攻击环境下均表现出强大的稳定性.(2)SecFedDMC 在满足恶意客户端数量 M 小于客户端总数 n 的一半的条件下，始终可以保证 100%的检测准确率.(3)随着恶意客户端数量的增加，全局模型的准确性会相应降低.(4)在特定的攻击类型中，尽管恶意客户端数量的增加会使得 ASR 略有上升，但 SecFedDMC 确保 ASR 的值始终不超过 3%.

8.1.3 非独立同分布对防御的影响

本文在 EMNIST 数据集进行了实验，研究非独立同分布程度对不同防御方法检测准确率的影响. 如图 5 所示，随着集中参数 β 的逐渐增大(β 增大表示客户端数据分布更均衡)，所有防御方法的检测准确率(DAR)均呈现上升趋势，这意味着 non-IID 程度对防御方法的检测准确率有明显影响. 在不同攻击环境下，SecFedDMC 均保持了较高的 DAR. 当集中参数 β 达到一定阈值后，SecFedDMC 的检测准确率可以达到 100%. 例如，对于 LF-attack，当 β 小于 3 时，SecFedDMC 的检测准确率开始下降；对于 GS-attack，当 β 小于 5 时，SecFedDMC 的检测准确率同样开始下降. 综合来看，SecFedDMC 在不同程度的 non-IID 环境下均展现出了优于其他防御方法的检测性能，有着良好的适应性.

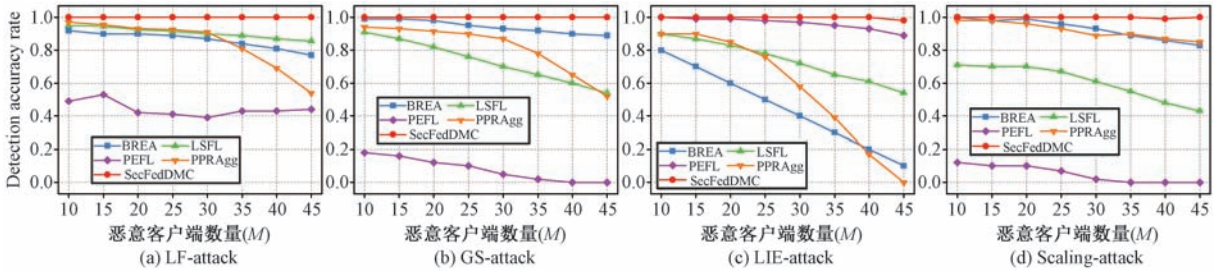

图 4 不同攻击下恶意客户端数量对防御方法的影响

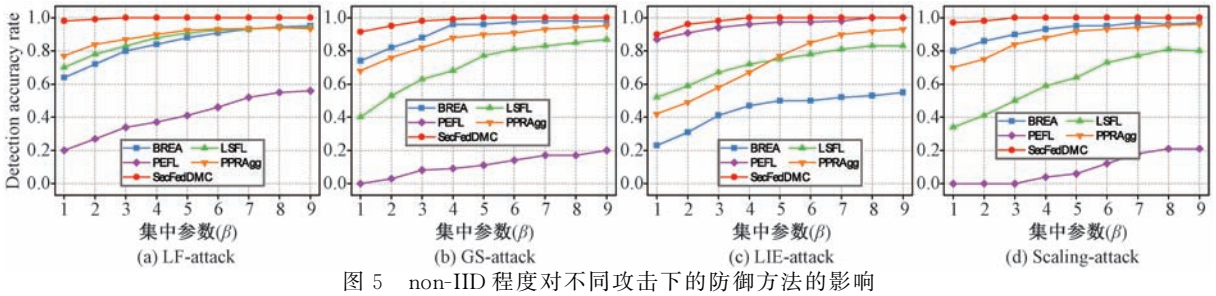

图 5 non-IID 程度对不同攻击下的防御方法的影响

表 4 SecFedDMC 在不同攻击下不同恶意客户端数量的实验结果

恶意客户端数量(M)	LF-attack		GS-attack		LIE-attack			Scaling-attack		
	DAR	TACC	DAR	TACC	DAR	TACC	ASR	DAR	TACC	ASR
10	1.00	86.13	1.00	86.17	1.00	85.96	0.89	1.00	85.68	1.15
15	1.00	85.94	1.00	85.91	1.00	85.83	0.97	1.00	85.34	1.56
20	1.00	85.68	1.00	85.73	1.00	85.17	1.41	1.00	85.16	2.07
25	1.00	85.56	1.00	85.69	1.00	84.87	1.62	1.00	84.92	2.12
30	1.00	85.44	1.00	85.61	1.00	84.49	1.93	1.00	84.73	2.19
35	1.00	85.11	1.00	85.56	1.00	84.38	2.69	1.00	84.67	1.96
40	1.00	85.31	1.00	85.53	1.00	84.34	2.16	1.00	84.49	2.44
45	1.00	84.99	1.00	85.50	0.99	84.04	2.12	1.00	84.39	2.13

8.2 全局模型的准确度

表 5 展示了在三个基准数据集上，不同攻击与检测方法下的全局模型的 TACC 与 ASR. 其中，“No-attack”项代表在没有恶意客户端干扰的情况下，仅通过 72 个诚实客户端学习全局模型的情况. 对于其余四种攻击，实验设置中包含了 28 个恶意客户端和 72 个诚实客户端. LIE-attack 和 Scaling-attack 是有目标的后门攻击，因此测试了各方法的 ASR. 观察结果显示，在部分攻击条件下，全局模型的准确性超过了 No-attack 情况. 这是因为恶意客

户端中仍存在部分有价值的信息.在其他几种攻击环境下,SecFedDMC 的全局模型性能达到了与 No-attack 环境下相同的水平.相比其他的防御方法,SecFedDMC 的全局模型在准确性上表现最优.在 LIE-attack 和 Scaling-attack 场景下,SecFedDMCASR 指标最低,这表明 SecFedDMC 通过有效移除恶意客户端,有效阻止了后门的植入.与最先进的算法 PPRAgg 相比,在 CIFAR10 数据集上,全局模型精度提升 4.45%～18.48%.

表 5　不同攻击和不同检测方法下的全局模型的测试精度

数据集	检测方法	No-attack	LF-attack	GS-attack	LIE-attack		Scaling-attack	
		TACC	TACC	TACC	TACC	ASR	TACC	ASR
MNIST	BREA	98.05	97.09	97.22	97.16	8.81	97.95	90.18
	LSFL	98.05	97.16	93.26	98.07	5.75	97.90	89.09
	PEFL	98.05	28.94	9.49	98.00	41.59	9.72	89.94
	PPRAgg	98.05	87.29	97.06	97.02	60.30	97.94	90.19
	SecFedDMC	**98.05**	**97.23**	**97.91**	**98.08**	**4.83**	**98.06**	**4.51**
EMNIST	BREA	85.63	84.25	85.17	85.22	13.24	84.49	69.95
	LSFL	85.63	85.19	75.47	85.32	6.49	83.59	95.01
	PEFL	85.63	4.74	0.67	85.56	7.19	4.96	95.03
	PPRAgg	85.63	84.40	85.24	83.28	32.75	84.08	88.96
	SecFedDMC	**85.63**	**85.23**	**85.30**	**85.71**	**4.84**	**85.89**	**66.39**
CIFAR10	BREA	79.76	76.93	35.98	70.67	5.72	74.48	84.59
	LSFL	79.76	77.18	36.34	75.74	4.50	74.91	89.51
	PEFL	79.76	64.12	45.13	45.33	56.24	22.57	86.61
	PPRAgg	79.76	79.70	73.68	73.26	5.37	61.78	87.56
	SecFedDMC	**79.76**	**85.85**	**79.13**	**77.88**	**1.32**	**80.56**	**54.41**

8.3　消融实验

为了进一步验证“先降维后聚类”策略的有效性和优越性,使用 EMNIST 数据集在 LF 攻击设置下进行实验.对比了经 PCA 降维处理后的聚类策略与无 PCA 降维处理,直接进行的聚类策略.实验结果如图 6 所示,“先降维后聚类”策略在恶意客户端的检测准确率(DAR)上有显著优势.同时全局模型收敛速度更快,准确度(TACC)更高.实验结果与预期相符,这是因为 PCA 能有效地减少同一簇内的数据点间的距离,同时对簇间距离的影响较小,因此可以看作 PCA 处理后的数据簇间距离相对增大,从而促进了聚类的结果.

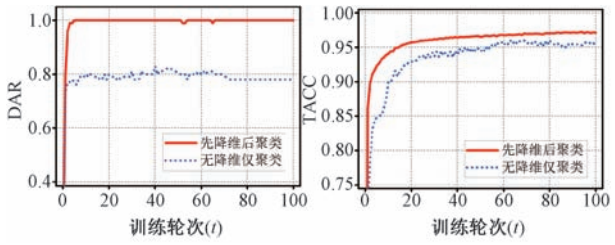

图 6　“先降维后聚类”与“无降维仅聚类”的对比

8.4　计算开销

为了评估 SecFedDMC 在双云服务器模型下的计算开销,本文对各关键步骤进行了详细的运行时间测试.需注意,此开销评估并未包含本地训练和向服务器传输参数的时间.SecFedDMC 框架主要包括三个组成部分:安全降维、安全聚类、安全聚合.其中,安全聚类部分采用了文献[50]中的 SecKmeans 方案,安全降维则由核心协议 *SecQR*,*SecEigen*,*SecMatMul* 实现.该评估基于三个数据集,实验结果如表 6 所示.*SecQR* 操作在整体执行时间中所占比例最高,达到了 65.76%～82.97%,与第 6.2 节给出的方案复杂性分析相符.SecFedDMC 与其他防御方法的对比如表 7 所示,SecFedDMC 的计算开销在实际应用中是可行的,为其广泛部署在实际环境中提供了可能性.与 PPRAgg 相比,SecFedDMC 在检测准确率方面表现更为突出,且计算效率提升 33.21%～47.31%.这一优势主要源于 SecFedDMC

表 6　SecFedDMC 各个步骤运行时间

数据集	安全降维			安全聚类	安全聚合	总运行时间
	SecQR	*SecEigen*	*SecMatMul*			
MNIST	65.43 s	6.28 s	11.42 s	6.51 s	3.27 s	92.91 s
EMNIST	36.45 s	3.21 s	7.31 s	6.16 s	2.12 s	55.43 s
CIFAR10	420.15 s	18.14 s	51.26 s	6.89 s	9.93 s	506.37 s

采用的高效安全多方计算策略，相对于 PPRAgg 所依赖的计算复杂度较高的同态加密技术，SecFedDMC 避免了相应的计算负担.

表 7 三个数据集上不同方法的计算开销

检测方法	MNIST	EMNIST	CIFAR10
BREA	53 s	28 s	5 min 21 s
LSFL	48 s	34 s	3 min 47 s
PEFL	3 min 49 s	1 min 51 s	15 min 26 s
PPRAgg	2 min 17 s	1 min 18 s	11 min 14 s
SecFedDMC	1 min 33 s	55 s	8 min 26 s

9 结 论

针对联邦学习中拜占庭攻击和用户隐私泄露的问题，本文提出一种基于安全多方计算技术的抗拜占庭攻击的隐私保护联邦学习框架—SecFedDMC. 该框架采用“先降维后聚类”的策略，高效精准地检测恶意客户端. 针对数据隐私泄露问题，设计了安全的正交三角分解协议和安全的特征分解协议，以保护模型训练和拜占庭节点识别过程中的用户隐私. 理论分析验证了 SecFedDMC 方案的安全性. 实验结果显示，SecFedDMC 在保护用户隐私的前提下，能够高效准确地识别拜占庭攻击节点，具有较好的鲁棒性.

致谢 我们向对本文的工作给予支持和宝贵建议的评审专家和同行表示衷心的感谢！

参 考 文 献

[1] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data//Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Fort Lauderdale, USA, 2017: 1273-1282

[2] Sheller M J, Edwards B, Reina G A, et al. Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data. Scientific Reports, 2020, 10(1): 1-12

[3] Mu X, Shen Y, Cheng K, et al. Fedproc: Prototypical contrastive federated learning on non-iid data. Future Generation Computer Systems, 2023, 143: 93-104

[4] Long G, Tan Y, Jiang J, et al. Federated learning for open banking//Federated Learning: Privacy and Incentive. Cham: Springer International Publishing, 2020: 240-254

[5] Lu Y, Huang X, Zhang K, et al. Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. IEEE Transactions on Vehicular Technology, 2020, 69(4): 4298-4311

[6] Xiao Xiong, Tang Zhuo, Xiao Bin, Li Kenli. A survey on privacy and security issues in federated learning. Chinese Journal of Computers, 2023, 46(05): 1019-1044 (in Chinese)
(肖雄，唐卓，肖斌，李肯立. 联邦学习的隐私保护与安全防御研究综述. 计算机学报，2023，46(05)：1019-1044)

[7] Ji Shouling, Du Tianyu, Li Jinfeng, Shen Chao, Li Bo. Security and privacy of machine mearning models: asurvey. Journal of Software, 2021, 32(01): 41-67 (in Chinese)
(纪守领，杜天宇，李进锋，沈超，李博. 机器学习模型安全与隐私研究综述. 软件学报，2021，32(01)：41-67)

[8] Wang H, Sreenivasan K, Rajput S, et al. Attack of the tails: Yes, you really can backdoor federated learning//Proceedings of the Annual Conference on Neural Information Processing Systems 2020, 2020: 16070-16084

[9] Bagdasaryan E, Veit A, Hua Y, et al. How to backdoor federated learning//Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics. Palermo, Italy, 2020: 2938-2948

[10] Xie C, Huang K, et al. DBA: Distributed backdoor attacks against federated learning//Proceedings of the International Conference on Learning Representations. Addis Ababa, Ethiopia, 2020: 1-19

[11] Baruch G, Baruch M, Goldberg Y. A little is enough: Circumventing defenses for distributed learning//Proceedings of the Annual Conference on Neural Information Processing Systems 2019. Vancouver, Canada, 2019: 8632-8642

[12] Bhagoji A N, Chakraborty S, Mittal P, et al. Analyzing federated learning through an adversarial lens//Proceedings of the International Conference on Machine Learning. Long Beach, USA, 2019: 634-643

[13] Fang M, Cao X, Jia J, et al. Local model poisoning attacks to byzantine-robust federated learning//Proceedings of the 29th USENIX Conference on Security Symposium. 2020: 1623-1640

[14] Shejwalkar V, Houmansadr A. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning//Proceedings of the 28th Annual Network and Distributed System Security Symposium. 2021: 1-18

[15] Yin D, Chen Y, Kannan R, et al. Byzantine-robust distributed learning: Towards optimal statistical rates//Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden, 2018: 5650-5659

[16] Guerraoui R, Rouault S. The hidden vulnerability of distributed learning in byzantium//Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden, 2018: 3518-3527

[17] Cao X, Fang M, et al. FLTrust: Byzantine-robust federated learning via trust bootstrapping//Proceedings of the 28th Annual Network and Distributed System Security Symposium. 2021: 1-18

[18] Li S, Cheng Y, Wang W, et al. Learning to detect malicious clients for robust federated learning. arXiv preprint arXiv: 2002.00211, 2020

[19] Shen S, Tople S, Saxena P. Auror: Defending against poisoning attacks in collaborative deep learning systems// Proceedings of the 32nd Annual Conference on Computer Security Applications. Los Angeles, USA, 2016: 508-519

[20] Blanchard P, El Mhamdi E M, Guerraoui R, et al. Machine learning with adversaries: Byzantine tolerant gradient descent//Proceedings of the Annual Conference on Neural Information Processing Systems 2017. Long Beach, USA, 2017:119-129

[21] Fung C, Yoon C J M, Beschastnikh I. The limitations of federated learning in sybil settings//Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses. San Sebastian, Spain, 2020: 301-316

[22] Zhang Z, Cao X, Jia J, et al. FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients//Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Washington, USA, 2022: 2545-2555

[23] Zhu L, Liu Z, Han S. Deep leakage from gradients// Proceedings of the Annual Conference on Neural Information Processing Systems 2019. Vancouver, Canada, 2019:14747-14756

[24] Melis L, Song C, De Cristofaro E, et al. Exploiting unintended feature leakage in collaborative learning//Proceedings of the IEEE Symposium on Security and Privacy. San Francisco, USA, 2019: 691-706

[25] Wei K, Li J, Ding M, et al. Federated learning with differential privacy: Algorithms and performance analysis. IEEE Transactions on Information Forensics and Security, 2020, 15: 3454-3469

[26] Zhang X, Chen X, Hong M, et al. Understanding clipping for federated learning: Convergence and client-level differential privacy//Proceedings of the International Conference on Machine Learning. Baltimore, USA, 2022:26048-26067

[27] Aono Y, Hayashi T, Wang L, et al. Privacy-preserving deep learning via additively homomorphic encryption. IEEE Transactions on Information Forensics and Security, 2017, 13(5): 1333-1345

[28] Sav S, Pyrgelis A, Troncoso-Pastoriza J R, et al. POSEIDON: Privacy-preserving federated neural network learning //Proceedings of the 28th Annual Network and Distributed System Security Symposium. Virtual,2021: 1-24

[29] Demmler D, Schneider T, Zohner M. ABY-A framework for efficient mixed-protocol secure two-party computation // Proceedings of the 22nd Annual Network and Distributed System Security Symposium. San Diego, USA, 2015

[30] Paillier P. Public-key cryptosystems based on composite degree residuosity classes//Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques. Prague, Czech Republic, 1999: 223-238

[31] Ma X, Sun X, Wu Y, et al. Differentially private byzantine-robust federated learning. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(12): 3690-3701

[32] Nguyen T D, Rieger P, De Viti R, et al. FLAME: Taming backdoors in federated learning//Proceedings of the 31st USENIX Security Symposium. Boston, USA, 2022: 1415-1432

[33] Ma Xindi, Li Qinghua, Jiang Qi, Ma Zhuo, Gao Sheng, Tian Youliang, Ma Jianfeng. Byzantine-robust federated learning overNon-IID data. Journal of Communications. 2023,44(1): 1-16. (in Chinese)
(马鑫迪,李清华,姜奇,马卓,高胜,田有亮,马建峰. 面向Non-IID数据的拜占庭鲁棒联邦学习. 通信学报,2023,44(1):1-16)

[34] Liu X, Li H, Xu G, et al. Privacy-enhanced federated learning against poisoning adversaries. IEEE Transactions on Information Forensics and Security, 2021, 16: 4574-4588

[35] Hao M, Li H, Xu G, et al. Efficient, private and robust federated learning//Proceedings of the Annual Computer Security Applications Conference. 2021: 45-60

[36] Dong Y, Chen X, Li K, et al. FLOD: Oblivious defender for private Byzantine-robust federated learning with dishonest-majority//Proceedings of the 26th European Symposium on Research in Computer Security. Darmstadt, Germany, 2021: 497-518

[37] Zhang Z, Wu L, Ma C, et al. LSFL: A lightweight and secure federated learning scheme for edge computing. IEEE Transactions on Information Forensics and Security, 2022, 18(2): 365-379

[38] So J, Güler B, Avestimehr A S. Byzantine-resilient secure federated learning. IEEE Journal on Selected Areas in Communications, 2020, 39(7): 2168-2181

[39] Rathee M, Shen C, Wagh S, et al. Elsa: Secure aggregation for federated learning with malicious actors//Proceedings of the IEEE Symposium on Security and Privacy. San Francisco,USA, 2023: 1961-1979

[40] Cui B, Mei T. ABFL: A blockchain-enabled robust framework for secure and trustworthy federated learning// Proceedings of the 39th Annual Computer Security Applications Conference. Honolulu, USA, 2023: 636-646

[41] Rokhlin V, Szlam A, Tygert M. A randomized algorithm for principal component analysis. SIAM Journal on Matrix Analysis and Applications, 2010, 31(3): 1100-1124

[42] Nasr M, Shokri R, Houmansadr A. Comprehensive privacy analysis of deep learning//Proceedings of the 2019 IEEE Symposium on Security and Privacy. San Francisco, USA, 2018: 1-15

[43] Truex S, Liu L, Chow K H, et al. LDP-Fed: Federated learning with local differential privacy//Proceedings of the 3rd ACM International Workshop on Edge Systems, Analytics and Networking. Heraklion, Greece, 2020: 61-66

[44] Naseri M, Hayes J, De Cristofaro E. Local and central differential privacy for robustness and privacy in federated learning// Proceedings of the 29th Annual Network and Distributed System Security Symposium. San Diego, USA, 2022:1-20

[45] Sun L, Qian J, Chen X. LDP-FL: Practical private aggregation in federated learning with local differential privacy// Proceedings of the 30th International Joint Conference on Artificial Intelligence. Virtual, 2021:1571-1578

[46] Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research, 2008, 9(11):1-27

[47] Tenenbaum J B, Silva V, Langford J C. A global geometric framework for nonlinear dimensionality reduction. Science, 2000, 290(5500): 2319-2323

[48] McInnes L, Healy J, Melville J. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426, 2018

[49] Cover T, Hart P. Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 1967, 13(1): 21-27

[50] Mohassel P, Rosulek M, Trieu N. Practical privacy-preserving k-means clustering. Cryptology ePrint Archive. 2020, 2020(4): 414-433

[51] Knott B, Venkataraman S, Hannun A, et al. Crypten: Secure multi-party computation meets machine learning. Advances in Neural Information Processing Systems, 2021, 34: 4961-4973

[52] Verbeke J, Cools R. The newton-raphson method. International Journal of Mathematical Education in Science and Technology, 1995, 26(2): 177-193

[53] Xia Z, Gu Q, Zhou W, et al. STR: Secure computation on additive shares using the share-transform-reveal strategy. IEEE Transactions on Computers, 2021, (01):1-14

[54] Shen J, Liu J, Chen Y, et al. Towards efficient and secure delivery of data for training and inference with Privacy-Preserving. arXiv preprint arXiv:1809.09968, 2018

[55] Patra A, Schneider T, Suresh A, et al. ABY 2.0: Improved mixed-protocol secure two-party computation//Proceedings of the 30th USENIX Security Symposium. 2021: 2165-2182

[56] Cheng K, Fu J, Shen Y, et al. Manto: A practical and secure inference service of convolutional neural networks for IoT. IEEE Internet of Things Journal, 2023,10(16): 14856-14872

[57] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, USA, 2016: 770-778

MU Xu-Tong, Ph. D. candidate. His primary research interests include federated learning and secure multi-party computation.

CHENG Ke, Ph. D., lecturer. His primary research interests include privacy preservation and machine learning.

SONG An-Xiao, Ph. D. candidate. His primary research interests include data security and machine learning.

ZHANG Tao, Ph. D., associate professor. His primary research interests include data security and federated learning.

ZHANG Zhi-Wei, Ph. D., associate professor. His primary research interests include collaborative security of unmanned systems.

SHEN Yu-Long, Ph. D., professor. His primary research interests include data security, wireless network security and cloud computing security.

Background

The study presented here falls within the domain of Federated Learning (FL), specifically addressing the challenge of effectively detecting and mitigating byzantine attacks. Byzantine attacks refer to a situation where certain nodes in a network, referred to as byzantine nodes, behave maliciously or erratically, disrupting the functioning of the system. In the context of federated learning, these nodes, or malicious clients, can significantly degrade the quality of the aggregated global model. As of the present moment, the international research community has made strides towards resolving this issue but has not yet arrived at a universally effective solution. The detection and mitigation of byzantine attacks while preserving privacy remain a challenging problem in federated learning.

The researchers in this paper introduce a novel approach, named SecFedDMC, combining dimensionality reduction using Randomized Principal Component Analysis (RPCA) and a clustering approach for client classification. This approach marks a significant advancement in dealing with byzantine attacks in federated learning in a privacy-preserving manner. This research is a component of a larger project aimed at bolstering privacy and security in federated learning, with particular emphasis on thwarting byzantine attacks. The significance of this project is highlighted by the growing demand for effective privacy-preserving machine learning techniques in response to the increasing application of federated learning across various industries.

The research group involved has a history of contributing to this field, with several noteworthy publications that form the foundational bedrock for this current study. The results from this paper address a critical part of the larger project-detecting Byzantine attacks. This crucial step forms a significant part of the overall objective to enhance the security

and privacy in federated learning. Therefore, the implications of this study are expected to be substantial for the future design and implementation of robust, privacy-preserving federated learning models resilient to byzantine attacks.

This paper is supported in part by the National Natural Science Foundation of China (No. 62220106004, 62302368); in part by the Major Research Plan of the National Natural Science Foundation of China (No. 92267204); in part by the Key Research and Development Program of Shaanxi (No. 2022KXJ-093, 2021ZDLGY07-05), in part by the Innovation Capability Support Program of in part by the Natural Science Basic Research Program of Shaanxi (No. 2024JC-YBQN-0701), Shaanxi (No. 2023-CX-TD-02); in part by the Key R&D Program of Shandong Province of China (No. 2023CXPT056), and in part by the Fundamental Research Funds for the Central Universities (No. XJSJ23040, ZDRC2202).