面向 Non-IID 数据的拜占庭鲁棒联邦学习

马鑫迪[1]，李清华[1]，姜奇[1]，马卓[1]，高胜[2]，田有亮[3]，马建峰[1]
（1. 西安电子科技大学网络与信息安全学院，陕西 西安 710071；2. 中央财经大学信息学院，北京 100081；
3. 贵州大学计算机科学与技术学院，贵州 贵阳 550025）

摘 要：面向数据分布特征为非独立同分布的联邦学习拜占庭节点恶意攻击问题进行研究，提出了一种隐私保护的鲁棒梯度聚合算法。该算法设计参考梯度用于识别模型训练中“质量较差”的共享梯度，并通过信誉度评估来降低数据分布异质对拜占庭节点识别的影响。同时，结合同态加密和随机噪声混淆技术来保护模型训练和拜占庭节点识别过程中的用户隐私。最后，在真实数据集中进行仿真测试，测试结果表明所提算法能够在保护用户隐私的条件下，准确、高效地识别拜占庭攻击节点，具有较好的收敛性和鲁棒性。

关键词：联邦学习；拜占庭攻击；非独立同分布；隐私保护；同态加密
中图分类号：TN92
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2023115

Byzantine-robust federated learning over Non-IID data

MA Xindi[1], LI Qinghua[1], JIANG Qi[1], MA Zhuo[1], GAO Sheng[2], TIAN Youliang[3], MA Jianfeng[1]
1. School of Cyber Engineering, Xidian University, Xi’an 710071, China
2. School of Information, Central University of Finance and Economics, Beijing 100081, China
3. School of Computer Science and Technology, Guizhou University, Guiyang 550025, China

Abstract: The malicious attacks of Byzantine nodes in federated learning was studied over the non-independent and identically distributed dataset , and a privacy protection robust gradient aggregation algorithm was proposed. A reference gradient was designed to identify “poor quality” shared gradients in model training, and the influence of heterogeneity data on Byzantine node recognition was reduced by reputation evaluation. Meanwhile, the combination of homomorphic encryption and random noise obfuscation technology was introduced to protect user privacy in the process of model training and Byzantine node recognition. Finally, through the evaluation over the real-world datasets, the simulation results show that the proposed algorithm can accurately and efficiently identify Byzantine attack nodes while protecting user privacy and has good convergence and robustness.

Keywords: federated learning, Byzantine attack, Non-IID, privacy protection, homomorphic encryption

0 引言

随着人工智能技术的发展，机器学习在计算机视觉、自然语言处理、推荐系统等领域的应用得到迅猛发展。机器学习模型训练往往基于海量用户数据，通过共享数据或共享知识实现联合模

收稿日期：2022−12−10；**修回日期：**2023−03−09
基金项目：国家重点研发计划基金资助项目（No.2021YFB3101100）；国家自然科学基金资助项目（No.U21A20464, No.62220106004, No.62072352, No.62261160651）；陕西省重点研发计划基金资助项目（No.2023-ZDLGY-52, No.2023-YBSF-206）；信息网络安全公安部重点实验室开放课题资助项目（No.C1904）；中央高校基本科研业务费专项资金资助项目（No.ZYTS23167）
Foundation Items: The National Key Research and Development Program of China (No.2021YFB3101100), The National Natural Science Foundation of China (No.U21A20464, No.62220106004, No.62072352, No.62261160651), The Key Research and Development Program of Shaanxi Province (No.2023-ZDLGY-52, No.2023-YBSF-206), Key Lab of Information Network Security, Ministry of Public Security (No.C19604), The Fundamental Research Funds for the Central Universities (No.ZYTS23167)

型训练，然而，由于数据隐私等问题，训练服务中心难以直接从用户节点（UN, user node）获得高质量、大体量的训练数据。为解决隐私保护和“数据孤岛”等问题，分布式机器学习框架——联邦学习（FL, federated learning）[1-2]被提出，用于实现分布式环境下多用户节点的联合机器学习。在联邦学习中，用户节点随机共享本地模型或参数梯度实现知识共享，而非将原始训练数据直接进行共享。在得到各用户节点共享的训练知识后，训练服务中心聚合知识结果并更新全局模型。联邦学习避免了直接共享用户训练数据，但仍面临诸多挑战[3]，如通信开销、隐私泄露、拜占庭攻击[4]等。

在分布式系统中，拜占庭攻击始终是无法避免的安全难题。因此，联邦学习框架也极易遭受拜占庭节点的攻击。在联邦学习框架中，拜占庭攻击可能由设备在复杂计算或通信过程中软硬件问题出现的数据计算错误导致；也可能由恶意用户节点主动共享错误信息导致，最终结果将影响模型的收敛方向，使模型训练失败。Chen 等[5]指出即使系统中只存在单个拜占庭节点，也可能导致联邦学习模型训练失败。Mcmahan 等[6]首次提出联邦平均（FedAvg, federated average）的概念，中心服务器通过计算用户节点上传梯度的平均值来更新全局模型参数。FedAvg 方法虽然能确保全局模型收敛[7]，但无法抵抗拜占庭攻击。针对联邦学习中的拜占庭攻击问题，许多学者提出了基于统计学方法或基于距离的拜占庭鲁棒性判别方案。例如，文献[8]提出了基于欧氏距离的 Krum 方案以识别联邦学习中的拜占庭节点，该方案是在用户节点梯度更新时挑选出与其他用户节点梯度距离最近的梯度，并以此为基准判断系统中存在的拜占庭节点。然而，在上述方案中，用户节点的训练数据集假设为独立同分布（IID, independent and identically distributed），且系统中正常的用户节点占多数，因此，可以直接通过距离或者其他统计学方法来识别少数的拜占庭节点，并使全局模型参数向梯度下降最快的方向进行优化。

在联邦学习中，各个用户节点数据间具有不对称性[9]，其拥有的训练数据也难以统一，因此，联邦学习中的训练数据通常为非独立同分布（Non-IID, non-independent and identically distributed）数据。当用户节点的训练数据为 Non-IID 数据时，模型训练共享的梯度信息会存在较大差异。因此，在 Non-IID 数据背景下，基于距离或统计学方法的抗拜占庭攻击方案难以奏效，使识别拜占庭节点更加困难。

除了拜占庭攻击问题，还需要考虑联邦学习模型训练过程存在的隐私泄露问题。由于用户节点共享的梯度信息可能包含用户的隐私信息，例如，Zhu 等[10]提出的 DLG（deep leakage from gradients）攻击可从用户节点共享的梯度信息推理出用户的训练数据。因此为解决隐私泄露问题，许多基于隐私保护技术的聚合方案被学者提出。例如，文献[11]基于差分隐私（DP, differential privacy）设计联邦学习的安全聚合方案；文献[12]结合安全多方计算（MPC, secure multi-party computation）技术提出隐私保护的联邦学习，在模型训练过程中无法从混淆的加密本地梯度中恢复出原始本地梯度，只能得到聚合梯度的结果；文献[13]运用同态加密（HE, homomorphic encryption）技术来保护用户节点上传的本地梯度。在识别拜占庭节点的过程中，如何确保用户隐私不被泄露是实现安全聚合方案的另一关键。因此，在 Non-IID 数据背景下，如何设计实现隐私保护的抗拜占庭攻击方案仍然是尚未解决的难题。

为解决上述问题，本文提出面向 Non-IID 数据的拜占庭鲁棒联邦学习方案。针对联邦学习中 Non-IID 数据特征，本文引入信誉度以降低 Non-IID 数据对拜占庭节点识别的影响；为抵抗拜占庭攻击，提出基于用户节点梯度相似度的拜占庭节点识别方案；设计隐私保护的模型训练方案，以实现模型训练及拜占庭节点识别过程中的隐私保护。

本文的主要贡献如下：1) 提出隐私保护鲁棒梯度聚合（PPRAgg, privacy-preserving robust gradient aggregation）算法，引入模型训练参考梯度，通过比较用户节点共享梯度和参考梯度的相似度，实现拜占庭节点的识别；2) 引入用户节点信誉度以降低 Non-IID 数据对拜占庭节点识别的影响，引入信誉度以度量用户节点梯度的可信度，服务提供者（SP, service provider）依据用户信誉度判别其共享梯度的可信度，从而更好地识别拜占庭攻击节点；3) 引入同态加密和随机噪声混淆技术实现联邦学习模型训练的隐私保护，在模型训练过程中，使用同态加密和随机噪声混淆技术对参数梯度及中间计算结果进行加密，使攻击者无法依据中间结果推理模型训练者的隐私；4) 对隐私保护抗拜占庭攻击算法进行了理论分析和实验评估，结果表明该算法在

Non-IID 的数据集上可以准确地识别拜占庭攻击节点，并实现了高效准确的联邦学习模型训练。

1 相关工作

近年来，拜占庭攻击在分布式系统中被广泛关注。随着联邦学习等分布式机器学习框架的发展，机器学习中的抗拜占庭攻击也引起了许多学者的研究兴趣。Blanchard 等[8]基于欧氏距离提出了 Krum 和 Multi-Krum 方案，其中 Multi-Krum 是对 Krum 方案的扩展。基于多个筛选梯度，Krum 计算梯度向量的平均梯度作为基准梯度，并将距离基准梯度最远的用户节点识别为拜占庭节点。文献[14-16]提出了统计学的检测方案，该类方案基于平均值或者平均值的变体。Wu 等[17]提出了 ByrdSAGA 方案，采用几何中值来聚合用户节点矫正的本地梯度，该方案通过降低随机梯度的噪声，从而更好地检测用户节点上传的“恶意”梯度。当用户训练数据为 IID 数据时，上述基于距离或统计学的方法可以较好地识别拜占庭节点并直接抛弃其共享的虚假（错误）信息，但是当用户数据为 Non-IID 数据时，上述方案无法有效区分正常节点和拜占庭节点共享的信息。

目前，多种针对 Non-IID 数据集的抗拜占庭攻击方案被提出。例如，He 等[18]提出了一种基于重采样的策略，以减小用户节点数据异质性带来的影响。Li 等[19]提出一种鲁棒随机聚合（RSA）方法，通过引入一类具有鲁棒性的随机子梯度的方法解决数据异质下的拜占庭攻击问题，并在目标函数中加入正则项来增强方法的鲁棒性，降低拜占庭攻击的影响，最终实现面向 Non-IID 数据的抗拜占庭攻击梯度聚合。Prakash 等[20]提出了 DiverseFL 方案，允许用户节点共享少量本地数据生成用户节点的标准梯度来识别拜占庭节点，从而实现异构数据的联邦学习模型训练。Xie 等[21]提出 Zeno 方案，通过损失函数评估用户节点更新的本地梯度，对评估结果进行排序后只聚合得分高的本地梯度。但是该方案在计算损失值时需要从全局的数据集中抽取若干样本，在实际中并不现实。Cao 等[22]提出了 FLTrust 方案，服务提供者通过收集小部分的纯净训练数据来给用户节点的本地模型更新进行信任评分。Zhai 等[23]提出了信誉度辅助的拜占庭鲁棒聚合（BRCA）方案，该方案引入异常检测模型和数据可验证方案，设计了基于可信度评估的拜占庭识别方法，基于一致性更新算法，使全局模型有一致的收敛方向。但是与 DiverseFL 和 FLTrust 类似，BRCA 方案通过训练前共享部分数据实现训练过程中拜占庭节点的检测。如果在训练开始时就已经存在拜占庭节点，那么恶意节点可能会上传错误数据，从而影响拜占庭节点的检测，最终导致模型训练失败。Peng 等[24]提出了一种同时降低样本间差异和用户节点间差异的重采样策略，以降低 Non-IID 数据背景下用户节点梯度更新的大方差对拜占庭节点识别带来的影响。Chen 等[25]提出了 FedSA，该算法采用两阶段策略加速训练速度并减少通信开销，同时设计了动态选择超参数的方法保证训练的高效性和鲁棒性，实现了面向 Non-IID 数据的异步联邦学习。上述方案均是针对 Non-IID 数据背景的抗拜占庭攻击方案，但是均未考虑模型训练和拜占庭节点识别过程中的用户隐私问题。无论是通过目标优化[18-19,24-25]的方式，还是通过设计基于性能评估[20-23]的安全聚合算法，服务提供者都需要对收集的本地梯度进行处理。如果本地梯度在明文状态下进行共享，可能存在隐私泄露的风险。然而，不同抵抗拜占庭攻击的方案处理本地梯度的方式是不同的，因此无法直接将现有的方案迁移到密文环境中。要实现隐私保护的抗拜占庭攻击方案，需要结合具体的方案引入合适的隐私保护技术。

最近，有学者在研究联邦学习中拜占庭攻击的同时考虑隐私泄露的问题[26-29]。例如，He 等[26]提出了一种支持隐私保护的抗拜占庭攻击机器学习方案，该方案结合了秘密共享的 MPC 算法和 Krum 算法。So 等[27]基于随机量化、可验证的离群点检测和安全的模型聚合等方法提出了一种拜占庭容忍的安全聚合方案，该方案可同时保证拜占庭鲁棒性、隐私性和收敛性。Khazbak 等[28]提出了基于通用的 MPC 工具和余弦相似度的方法来检测拜占庭节点，通过考虑用户节点上传的本地梯度的方向来准确地检测识别拜占庭节点。但是上述方法没有考虑数据分布对拜占庭节点识别的影响，在面向 Non-IID 数据时表现较差，抵抗拜占庭攻击的能力急剧下降，此外，这些方法的中间参数（例如衡量用户节点共享的本地梯度的置信分数）可能存在泄露用户节点的身份或数据质量的风险。Ma 等[29]设计了轻量级的联邦学习安全鲁棒聚合算法，尽管该算法可在 Non-IID 数据下准确识别拜占庭节点，但在计算过程中仍然存在隐私泄露问题，如用户节点信誉度等信息泄露。综上所述，这些方法虽然同时考虑了拜占庭攻击和隐私保护问题，但还存在一些问题。例如，文献[26-28]没有考虑数据分布特征对拜占庭检测的

影响；文献[29]虽然能抵抗 Non-IID 数据分布的拜占庭攻击，但是会泄露用户节点信誉度信息，存在隐私泄露的风险。因此，以上方法要么在 Non-IID 的数据集下抵抗拜占庭攻击的能力有限，要么无法保护模型训练过程中所有中间参数的隐私信息。

因此，本文提出了面向 Non-IID 数据的拜占庭鲁棒联邦学习方案，在准确识别模型训练过程的拜占庭攻击节点的同时，确保用户节点的数据隐私。表 1 将本文方案和现有典型方案进行了对比。

表 1　　方案对比总结

方案	实现方法	抵抗拜占庭攻击	隐私保护	训练有效性
文献[11]	DP+FedAvg	×	√	低
文献[12]	MPC+FedAvg	×	√	高
文献[13]	HE+FedAvg	×	√	高
文献[8]	Krum	√	×	高
文献[19]	RSA	√	×	高
文献[21]	Zeno	√	×	高
本文	PPRAgg	√	√	高

2　预备知识

为构造隐私保护的抗拜占庭攻击模型训练方案，本节将介绍本文所需的部分预备知识：Gompertz 函数、分布式双陷门公钥加密系统（DT-PKC, distributed two trapdoor public-key cryptosystem）以及部分安全计算协议。

2.1　Gompertz 函数

为降低 Non-IID 数据集对拜占庭节点识别带来的影响，本文引入信誉度刻画用户节点在模型训练中的诚实表现，通过比较用户共享梯度与参考梯度间的相似度，检测本轮训练中用户节点的诚实性，从而更新用户节点信誉度。为了精准刻画用户节点的信誉度与其梯度可信度的关系，本文引入了 Gompertz[30]函数。Gompertz 函数是一个 Sigmoid 函数，给定一个时间段，它的曲线在开始和结束阶段缓慢增长，中间阶段快速增长，如图 1 所示。

Gompertz 函数常作为数学模型来描述信誉模型的更新，其计算式为 $C_i = \mathrm{a}\mathrm{e}^{be^{cr_i}}$，其中，$a,b,c$ 是标准参数，a 控制曲线的上渐近线，b 控制沿 x 轴的偏移量，c 控制信誉度的增长率。本文中，r_i 表示用户节点 n_i 的梯度可信度，C_i 表示用户节点 n_i 的信誉度。在明文状态下，如果服务提供者收到 n_i 的梯度 $\boldsymbol{g}_i$，则可以聚合该梯度为 $\boldsymbol{g}_i C_i$。

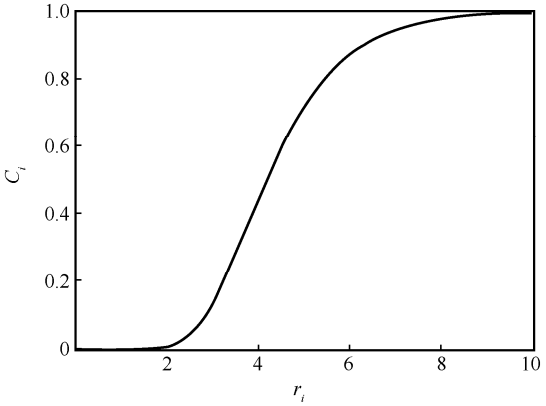


图 1　Gompertz 函数

2.2　分布式双陷门公钥加密系统

为实现隐私保护的模型训练，本文引入分布式双陷门公钥加密系统用于保护用户节点共享的信息。DT-PKC[31]既支持高效的隐私外包计算，也支持加密密钥的安全管理。本文方案基于 DT-PKC 中部分算法进行设计，具体算法介绍如下。

1) 密钥生成（KeyGen)。选定安全参数 k 和大素数 p、q，满足 $\|p\| = \|q\| = k$。根据强素数的特性，计算 $N = pq$ 和 $\lambda = \mathrm{lcm}(p-1, q-1)$，其中 lcm() 表示求最小公约数。定义函数 $L(x) = \dfrac{x-1}{N}$，选择随机数 $\kappa \in \mathbb{Z}_{N^2}^*$，计算阶为 $\dfrac{(p-1)(q-1)}{2}$ 的生成元 $z = -\partial^{\kappa^{2N}}$。选择随机数 $\theta \in \left[1, \dfrac{N}{4}\right]$ 并计算 $h = z^{\theta} \bmod N^2$，因此，系统的弱私钥为 $\mathrm{sk} = \theta$，公钥为 $\mathrm{pk} = (N, z, h)$，强私钥为 $\mathrm{SK} = \lambda$。另外，强私钥 SK 被分为两部分，即 $\mathrm{SK}_1 = \lambda_1$，$\mathrm{SK}_2 = \lambda_2$，$\{\lambda_1, \lambda_2\}$ 满足 $\lambda_1 + \lambda_2 \equiv 0 \bmod \lambda$，$\lambda_1 + \lambda_2 \equiv 1 \bmod N^2$。

2) 加密（Enc)。对消息 $m(0 \leqslant m \leqslant n)$ 进行加密，选择一个随机数 r，满足 $r \in \left[1, \dfrac{N}{4}\right]$，加密结果 $[\![m]\!]_{\mathrm{pk}} = T = z^{r\theta}(1 + mN) \bmod N^2$。

3) 使用强私钥解密（SDec)。给定密文 $T = [\![m]\!]_{\mathrm{pk}}$，使用强私钥解密如下

$$m = L(T^{\lambda} \bmod N^2)\lambda^{-1} \bmod N = L(1 + mN\lambda)\lambda^{-1} \bmod N$$

4) 部分解密-Step1（PD1)。给定密文 $T = [\![m]\!]_{\mathrm{pk}}$，使用 SK_1 对其解密如下

$$\mathrm{DT}^{(1)} = (T)^{\lambda_1} = z^{r\theta\lambda_1}(1 + mN\lambda_1) \bmod N^2$$

5) 部分解密-Step2（PD2)。给定密文 $T = [\![m]\!]_{\mathrm{pk}}$

和 $\mathrm{DT}^{(1)}$，使用 SK_2 对其解密如下

$$\mathrm{DT}^{(2)} = (T)^{\lambda_2} = z^{r\theta\lambda_2}(1+mN\lambda_2) \bmod N^2$$
$$m = L(\mathrm{DT}^{(1)}\mathrm{DT}^{(2)})$$

DT-PKC 同时支持同态加法和同态幂乘运算。给定明文 $\{x_1, x_2\}$，使用同一公钥加密得到密文 $\{[\![x_1]\!], [\![x_2]\!]\}$，具体步骤如下：1) 密文 $\{[\![x_1]\!], [\![x_2]\!]\}$ 的乘积为对应明文 $\{x_1, x_2\}$ 的和，即 $[\![x_1]\!]_{\mathrm{pk}}[\![x_2]\!]_{\mathrm{pk}} = [\![x_1 + x_2]\!]_{\mathrm{pk}}$；2)给定常数 $a \in \mathbb{Z}_N$，密文 $[\![x_1]\!]_{\mathrm{pk}}$ 的常数次幂为常数 a 与明文 x_1 的乘积，即 $([\![x_1]\!]_{\mathrm{pk}})^a = [\![ax_1]\!]_{\mathrm{pk}}$。在下文中，若密文均由同一公钥进行加密，则由 $[\![x]\!]$ 代替密文 $[\![x]\!]_{\mathrm{pk}}$ 进行简化表示。

2.3 部分安全计算协议

本文引入 DT-PKC 的整数安全计算协议用以实现隐私保护的模型训练，包括安全乘法协议（SMP, secure multiplication protocol）和安全小于协议（SLT, secure less than protocol）。SMP 实现同态乘法计算，输入 $\{[\![x]\!], [\![y]\!]\}$，输出 $[\![xy]\!]$，可表示为 $\mathrm{SMP}(x, y) = [\![xy]\!]$，当 $x = y$ 时，可以得到明文平方的加密结果 $[\![x^2]\!]$。SLT 实现密文数据大小比较运算，输入 $\{[\![x]\!], [\![y]\!]\}$，输出结果 l，可表示为 $\mathrm{SLT}([\![x]\!], [\![y]\!]) = [\![l]\!]$，如果 $x < y$，则 $l = 1$，否则 $l = 0$。上述协议的详细设计细节请参考文献[31]。

3 系统概述和问题描述

本节首先介绍本文的系统模型，阐述各参与实体在联邦学习训练中需完成的任务；其次，分析了系统所面临的攻击模型；最后给出本文的隐私需求。

3.1 系统模型

本文主要解决 Non-IID 数据场景下的抗拜占庭攻击问题，并引入安全计算协议实现模型训练过程的隐私保护。本文系统模型包括密钥生成中心（KGC, key generation center）、服务提供者、用户节点，如图 2 所示。

1) 密钥生成中心。KGC 独立于系统中其他实体，是系统中必不可少的部分，被其他实体充分信任。KGC 负责初始化系统所需要的密钥及各安全参数，并将密钥集合准确地分发给 SP 和 UN。

2) 服务提供者。SP 管理全局服务模型的训练，负责初始化全局服务模型，随后结合 UN 的计算结果，完成拜占庭节点的甄别和本地梯度的聚合，最终更新全局服务模型。训练终止后，SP 为 UN 提供新的服务。SP 拥有公钥和部分强私钥 SK_1。

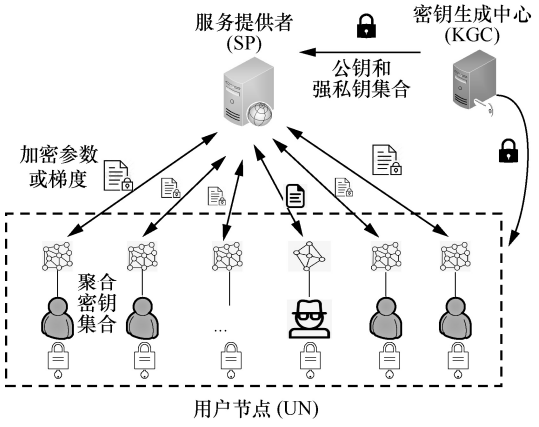


图 2 本文系统模型

3) 用户节点。系统中拥有一定数量的 UN，每个 UN 拥有本地数据。在模型训练过程中，UN 可以自主选择加入或退出模型训练。然而，部分 UN 可能会在某一轮的训练中，变成拜占庭节点并上传“恶意”的梯度，破坏模型训练。UN 拥有公钥和部分强私钥 SK_2。

在模型训练过程中，本文引入安全套接层协议/传输层安全协议（SSL/TLS）对网络通道进行加密，从而可以确保传输信息的完整性、通信双方的身份合法性等，保证模型训练过程的通信安全。

3.2 攻击模型

本文方案中，KGC 作为可信实体，为系统生成密钥集合并分发密钥；SP 和 UN 都是半诚实的实体，即使其严格遵守安全聚合协议，也会对其他实体的隐私信息发起被动或主动攻击，希望获悉或收集它们的隐私信息。同时，本文还考虑到部分 UN 可能发起主动攻击，通过上传“恶意”梯度信息来破坏模型的训练。基于存在的隐患，本文引入主动攻击敌手 $\mathcal{A}^*$，其拥有的能力如下。

1) $\mathcal{A}^*$ 可以监听通信通道获得传输的加密信息，从而通过伪造或截断发起主动攻击，其中被攻击的信息包括所有的中间计算结果、参数梯度、全局模型参数等。

2) $\mathcal{A}^*$ 可以攻击 SP，获取模型训练过程中 UN 上传的本地梯度，从而推理出 UN 本地的隐私训练数据。

3) $\mathcal{A}^*$ 可以攻击多个 UN，并获得其解密能力。借助攻破的 UN 解密能力，结合中间计算结果，推理出其他 UN 的隐私信息。

4) $\mathcal{A}^*$ 可以攻击一个或多个拜占庭节点来构造并

上传“恶意”的梯度信息，从而实现模型训练的干扰。

在攻击模型中，敌手 $\mathcal{A}^{\star}$ 不能同时攻破 SP 和任何一个 UN（即 SP 不能与任一 UN 共谋），该假设条件在密文安全计算协议中普遍存在[25]，而且该限制条件的攻击在应用中也很难达成。

3.3　隐私需求

本文方案的设计目标是在抵抗拜占庭攻击的同时实现隐私保护的模型训练，具体的隐私需求如下。

1) 确保 UN 本地训练数据的隐私。联邦学习设计的初衷是将数据留在本地进行模型训练，从而达到保护数据隐私的效果。但是，对于传统的联邦学习模型训练，攻击者可以从 UN 共享的信息中推断出其敏感隐私信息。因此，保护 UN 共享的信息是本文隐私保护的主要目标之一。

2) 确保联邦学习训练的模型安全。在模型训练结束后，SP 将模型发布给各用户节点使用。除了参与训练的 UN 外，其他实体均不能获得。因此，保护联邦学习模型参数也是本文的设计目标之一。

3) 保护 UN 信誉度的隐私。本文引入信誉度刻画用户节点的“声誉”，不会因为 UN 仅在某一轮训练中共享“质量较差”的梯度，就认定其为拜占庭节点。在计算 UN 的信誉度时，如何保护 UN 的信誉度隐私、实现信誉度的透明计算，也是本文的设计目标之一。

4　隐私保护的抗拜占庭攻击模型训练

本节主要描述隐私保护的抗拜占庭攻击模型训练算法，一轮训练中各实体间的交互流程如图 3 所示。本文基于整数安全计算协议设计了安全的鲁棒聚合算法，同时，为识别模型训练中的拜占庭节点，聚合算法引入参考梯度，通过计算参考梯度和 UN 更新梯度间的相似度，判别用户节点共享梯度的“质量”。参考梯度代表模型收敛的方向，当本地梯度和参考梯度方向相反时，本地梯度在这次模型训练中起着消极的作用，会阻碍模型的收敛。因此，本文提出的参考梯度能够识别出系统中的拜占庭节点，基于检测结果更新用户节点的信誉度并聚合本地梯度，提高聚合梯度的正确性，从而提高模型的鲁棒性。另外，为解决 Non-IID 数据集背景下用户节点数据异质性的问题，本文引入信誉度评估的方式，降低正常节点因数据分布差异导致的梯度更新误判。

本文方案假设系统中共有 M 个用户节点，参与第 k 轮训练的用户节点数为 $M^{(k)}$，模型参数是 d 维。下面以用户节点 n_i 为例介绍隐私保护的抗拜占庭攻击模型训练。

Step1（@KGC）。KGC 生成系统所需要的密钥集合，公钥 $\mathrm{pk}=(N,g,h)$，部分强私钥 $\mathrm{SK}_1=\lambda_1$ 和 $\mathrm{SK}_2=\lambda_2$。KGC 将 pk 公开，并将 SK_1、SK_2 分别发给 SP 和 UN。系统中的加密数据需要由 SP 和 UN 共同解密，因此，为保证数据安全性，必须保证 SP 和 UN 不能共谋。

Step2（@SP）。SP 初始化模型训练各项参数，包括初始模型参数 $\boldsymbol{w}^{(0)}$（维度为 d）、最大迭代次数 K、学习率 η 及安全参数 k_1,k_2,k_3 等参数集，并将初始化参数集发送给 UN。

Step3（@n_i）。首先，n_i 根据安全参数生成聚合密钥集合 $<s_i,a_i,\boldsymbol{b}_i,p>$，满足 $|a_i|=k_1,|b_{ij}|=k_2$，$|p_i|=k_3,s_i\in\mathbb{Z}_p$，其中，$b_{ij}\in\boldsymbol{b}_i$，$\boldsymbol{b}_i$ 是 d 维的向量。其次，n_i 基于本地数据开始本地模型训练。在结束一轮训练后，n_i 计算得到参数梯度 $\boldsymbol{g}_i^{(k)}$（表示第 k 轮 n_i 的梯度），并进行如下加密操作

$$\begin{cases}[\![\boldsymbol{g}_i^{(k)}]\!]=\mathrm{Enc}(g_{i1}^{(k)},g_{i2}^{(k)},\cdots,g_{ij}^{(k)},\cdots,g_{id}^{(k)})\\ l_{ij}^{(k)}=s_i(a_i x_{ij}^{(k)}+b_{ij})\bmod p\end{cases}\tag{1}$$

其中，$g_{ij}^{(k)}\in\boldsymbol{g}_i^{(k)},l_{ij}^{(k)}\in\boldsymbol{l}_i^{(k)}$。最后，$n_i$ 将 $\{[\![\boldsymbol{g}_i^{(k)}]\!],\boldsymbol{l}_i^{(k)}\}$ 发送给 SP。

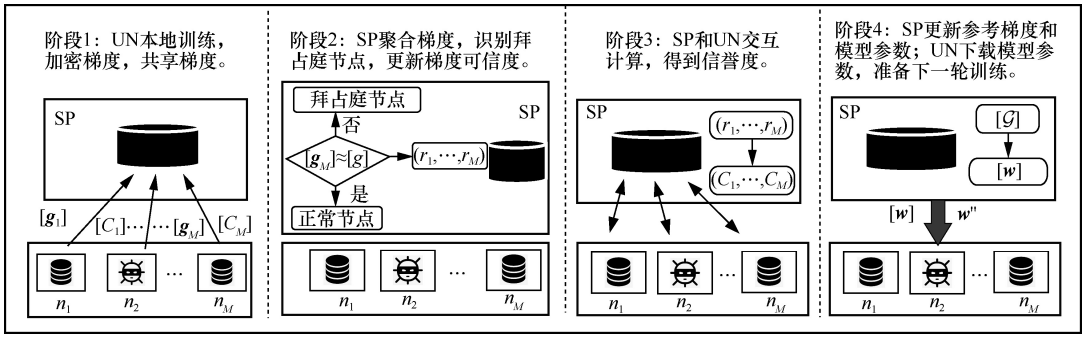


图 3　一轮训练中各实体间的交互流程

$$\begin{cases} [\![\hat{g}_j^{(k)}]\!]=\left\{\prod_{i=1}^{M^{(k)}}\mathrm{SMP}([\![g_{ij}^{(k)}]\!],[\![C_i^{(k-1)}]\!])\right\}^{\frac{1}{M^{(k)}}} \\ [\![B^{(k)}]\!]=\prod_{j=1}^{d}\mathrm{SMP}([\![\hat{g}_j^{(k)}]\!],[\![\hat{g}_j^{(k)}]\!]) \\ [\![D_{ij}^{(k)}]\!]=[\![\hat{g}_j^{(k)}]\!]^{l_{ij}^{(k)}}=\left[\!\left[\hat{g}_j^{(k)}l_{ij}^{(k)}\right]\!\right] \end{cases} \tag{2}$$

Step4（@SP）。SP 收集所有 UN 的梯度向量后，结合上一轮 UN 的信誉度，更新参考梯度。在首次训练时，初始化 n_i 的梯度可信度和信誉度分别为 $r_i^{(0)}$和$C_i^{(0)}$。SP 计算式(2)，其中，$j\in[1,d]$，$[\![\hat{g}_j^{(k)}]\!]$ 为参考梯度 $\hat{\boldsymbol{g}}^{(k)}$ 的第 j 维，记 $\boldsymbol{D}_i^{(k)}=(D_{i1}^{(k)},D_{i2}^{(k)},\cdots,D_{ij}^{(k)},\cdots,D_{id}^{(k)})$，$M^{(k)}$ 为在第 k 轮训练中随机选中的节点数，$[\![C_i^{(k-1)}]\!]$为节点 n_i 在第 $k-1$ 轮训练中更新的信誉度密文。设 n_i 在第 k 轮训练中被选中参与计算聚合梯度，那么基于 PD1 算法，SP 利用密钥 SK_1 对密文 $\{[\![B^{(k)}]\!],[\![\boldsymbol{D}_i^{(k)}]\!]\}$ 部分解密得到 $\{B',\boldsymbol{D}'\}$，将得到的部分解密结果和密文一起发送给 n_i。

Step5（@n_i）。n_i 收到 SP 发送的密文和部分解密结果后，运行 PD2 算法，利用密钥 SK_2 解密得到 $\{B^{(k)},\boldsymbol{D}_i^{(k)}\}$。$n_i$ 需要计算本轮次本地梯度向量和参考梯度向量的数量积 dp_i，以及梯度向量模平方的比值 ra_i。计算后加密得到密文 $\{[\![\mathrm{dp}_i]\!],[\![\mathrm{ra}_i]\!]\}$。计算步骤如式(3)所示。最后，$n_i$ 将加密后密文以二元组的形式发送给 SP。

$$\begin{cases} D_i=\sum_{j=1}^{d}D_{ij}^{(k)}=s_i\left(a_i\sum_{j=1}^{d}g_{ij}^{(k)}\hat{g}_j^{(k)}+\sum_{j=1}^{d}b_{ij}\hat{g}_j^{(k)}\right)\bmod p \\ E_i=s_i^{-1}D_i \\ E-E\bmod a_i=a_i\sum_{j=1}^{d}g_{ij}^{(k)}\hat{g}_j^{(k)} \\ \mathrm{dp}_i=\boldsymbol{g}_i^{(k)}\hat{\boldsymbol{g}}^{(k)}=\dfrac{(E-E\bmod a_i)}{a_i}=\sum_{j=1}^{d}g_{ij}^{(k)}\hat{g}_j^{(k)} \\ B_i^{(k)}=\sum_{j=1}^{d}(g_{ij}^{(k)})^2 \\ \mathrm{ra}_i=\dfrac{B_i^{(k)}}{B^{(k)}} \end{cases} \tag{3}$$

为了确保计算正确性，安全聚合参数需要满足式(4)的约束。如果某个用户节点的参数未能满足这些约束，则产生的加密结果和正常节点的差距较大，会被认定为拜占庭节点。

$$\begin{cases} p>s_i\left(a_i\sum_{j=1}^{d}g_{ij}^{(k)}\hat{g}_j^{(k)}+\sum_{j=1}^{d}b_{ij}\hat{g}_j^{(k)}\right) \\ a_i>\sum_{j=1}^{d}b_{ij}\hat{g}_j^{(k)} \end{cases} \tag{4}$$

为判断 n_i 是否为拜占庭节点，需要通过计算参考梯度和 n_i 梯度的相似度。梯度间的相似度分为方向相似度和大小相似度。参考梯度是更新模型参数的一个标准，如果 $\mathrm{dp}_i<0$，表明 n_i 的本地梯度和参考梯度的夹角超过 90°，n_i 在本轮的模型参数更新上起消极作用；否则表明 n_i 在本轮训练中为正常节点。同理，在高维的梯度向量中，如果 2 个梯度 2 范数比值太大，则表明 2 个梯度向量相似度不高。因此，本文给定阈值 $\varepsilon_1,\varepsilon_2$ 来限定梯度间的大小相似度。第 k 轮训练中，如果 n_i 的 2 个参数 $\{\mathrm{dp}_i,\mathrm{ra}_i\}$ 不满足式(5)的约束，则认为该用户节点在本轮训练中为拜占庭攻击节点。

$$\begin{cases} \varepsilon_1<\mathrm{ra}_i<\varepsilon_2 \\ \mathrm{dp}_i=\boldsymbol{g}_i^{(k)}\hat{\boldsymbol{g}}^{(k)}>0 \end{cases} \tag{5}$$

Step6（@SP）。SP 收到 n_i 的密文二元组后，根据式(5)，在全密文的情况下完成 n_i 共享的梯度可信度的更新。n_i 梯度可信度的安全更新过程如式(6)所示。分析可知，如果 $\{\mathrm{dp}_i,\mathrm{ra}_i\}$ 满足式(5)，则 $\rho=0$；否则 $\rho=1$。对应地，n_i 的梯度可信度 $r_i^{(k)}$ 相应增加或减少一个单位。进一步地，利用梯度可信度秘密更新 n_i 的信誉度。

$$\begin{cases} [\![z_1]\!]=\mathrm{SLT}(0,[\![\mathrm{dp}_i]\!]),[\![z_2]\!]=\mathrm{SLT}([\![\varepsilon_1]\!],[\![ra_i]\!]) \\ [\![z_3]\!]=\mathrm{SLT}([\![ra_i]\!],[\![\varepsilon_2]\!]),[\![\upsilon]\!]=[\![z_1]\!][\![z_2]\!][\![z_3]\!][\![2]\!]^{N-1} \\ [\![\rho]\!]=\mathrm{SLT}([\![\upsilon]\!],1),[\![r_i^{(k)}]\!]=[\![r_i^{(k-1)}]\!][\![1-2\rho]\!] \end{cases} \tag{6}$$

Step7（@SP&n_i）。得到 n_i 梯度的可信度后，根据 Gompertz 函数可计算出用户节点 n_i 的信誉度。为达到隐私保护的需求，使用户节点 n_i 信誉度的更新过程中隐私不被泄露，SP 不能直接获知或从中间计算结果中推断出信誉度的信息；因此，要在加密的情况下秘密地完成信誉度更新。更新过程需要由 SP 和 UN 多次交互。用户节点信誉度的秘密更新过程如图 4 所示。

Step8（@SP）。经过 SP 和 n_i 的多次交互计算后，得到$[\![C_i^{(k)}]\!]$。同时，SP 更新其他被随机选中的用户节点的信誉度。结合更新的信誉度，SP 聚合被选中用户节点共享的本地梯度，得到本轮训练更新后的聚合梯度$[\![\mathcal{G}]\!]$，聚合计算过程如下

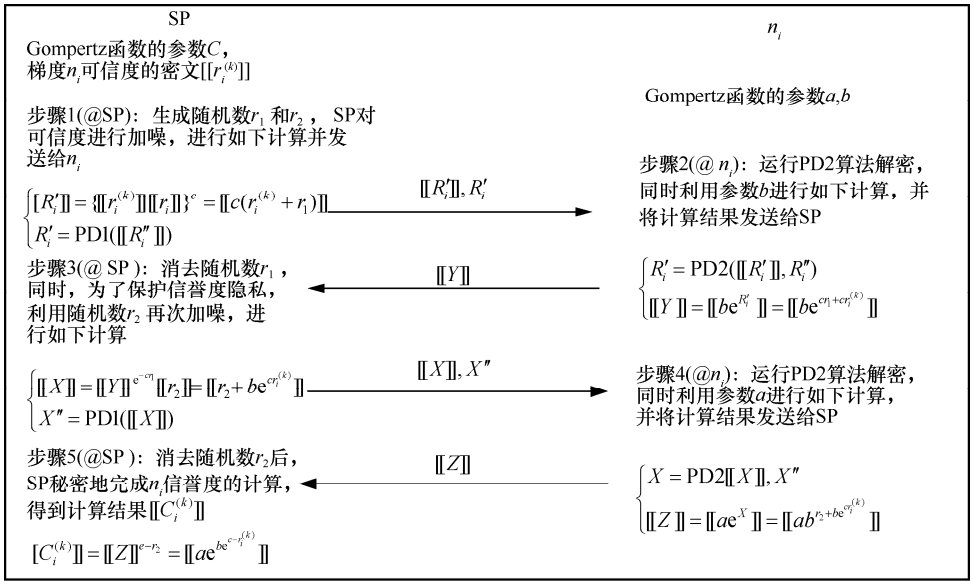


图 4　用户节点信誉度的秘密更新过程

$$[\mathcal{G}^{(k)}]=\left\{\prod_{i=1}^{M^{(k)}}\mathbf{SMP}\left(\left[\!\left[\boldsymbol{g}_i^{(k)}\right]\!\right],\left[\!\left[C_i^{(k)}\right]\!\right]\right)\right\}^{\frac{1}{M^{(k)}}} \tag{7}$$

SP 将聚合梯度$\left[\!\left[\mathcal{G}^{(k)}\right]\!\right]$及其部分解密的结果$\mathcal{G}^{"}$联合$\{[[\mathcal{G}^{(k)}]],\mathcal{G}^{"}\}$分发给所有 UN，以更新模型参数。

Step9（@$\boldsymbol{n_i}$）。n_i收到聚合梯度后，进行部分解密得到$\mathcal{G}^{(k)}$，最后使用梯度下降算法更新模型参数，如式(8)所示。当达到最大的训练次数K或者模型收敛时，终止训练；否则，从 Step3 开始继续迭代。

$$\boldsymbol{w}^{(k+1)}=\boldsymbol{w}^{(k)}-\mathcal{G}^{(k)}\eta^{(k)} \tag{8}$$

5　理论分析

本节从理论上分析了本文算法能够抵抗 3.2 节中提出的攻击模型，并对安全聚合算法进行了效率分析。

5.1　隐私保护分析

本文引入了一个常被用于证明安全多方计算协议安全性的安全模型[32-33]，来证明本文算法的安全性。在本文所设计的 PPRAg 算法中，存在 UN 和 SP 两类实体，因此，本文算法可看作n_i与 SP 之间进行安全双方计算。因此，只需要考虑两部分参与方，即 SP 和n_i。针对这两部分，本文分别对拜占庭节点识别、节点信誉度更新、全局梯度更新 3 个过程构造模拟器$(S_{\mathrm{SP}}^{\pi_1},S_{n_i}^{\pi_1};S_{\mathrm{SP}}^{\pi_2},S_{n_i}^{\pi_2};S_{\mathrm{SP}}^{\pi_3})$以应对两者的攻击者$(\mathcal{A}_{\mathrm{SP}},\mathcal{A}_{n_i})$。

定理 1　在 PPRAg 算法中，拜占庭节点识别、用户节点信誉度计算和全局梯度更新过程能够抵抗上文安全模型中所定义的敌手$\mathcal{A}$。

证明　在网络传输过程中，本文通过 SSL/TLS 的认证策略来抵抗敌手数据篡改攻击，保证了实体通信过程中的数据完整性。本文引入基于仿真的安全模型并构造理想函数以证明在半诚实攻击模型下，用户在真实场景下的执行过程和在理想环境下的计算过程不可区分，因此，可证明计算协议的安全性。

1) 在拜占庭节点识别时，模拟器$S_{n_i}^{\pi_1}$首先模拟用户节点n_i的真实执行过程，表示为$V_{n_i}^{\pi_1}(\mathrm{SK}_2,[[\boldsymbol{g}_i^{(k)}]],\boldsymbol{I}_i^{(k)},[[\mathrm{dp}_i]],[[\mathrm{ra}_i]];\mathrm{coins};B',\boldsymbol{D}',[[r_i^{(k)}]])$，其中，coins 表示协议执行过程中引入的随机数。因此，$S_{n_i}^{\pi_1}$执行如下步骤。

① 随机生成 DT-PKC 算法部分解密的密文数据$\{\widetilde{B}',\widetilde{\boldsymbol{D}}'\}$。

② 随机生成协议执行过程中的随机数$\widetilde{\mathrm{coins}}$。

③ 随机生成节点n_i更新后的信誉度密文$[[\widetilde{r_i^{(k)}}]]$。

④ 输出$(\mathrm{SK}_2,[[\boldsymbol{g}_i^{(k)}]],\boldsymbol{I}_i^{(k)},[[\mathrm{dp}_i]],[[\mathrm{ra}_i]];\widetilde{\mathrm{coins}};\widetilde{B}',\widetilde{\boldsymbol{D}}',[[\widetilde{r_i^{(k)}}]])$。

定义如下状态等式

$$H_0=V_{n_i}^{\pi_1}(\mathrm{SK}_2,[[\boldsymbol{g}_i^{(k)}]],\boldsymbol{I}_i^{(k)},[[\mathrm{dp}_i]],[[\mathrm{ra}_i]])$$

$$H_1=(\mathrm{SK}_2,[[\boldsymbol{g}_i^{(k)}]],\boldsymbol{I}_i^{(k)},[[\mathrm{dp}_i]],[[\mathrm{ra}_i]];\widetilde{B}',\widetilde{\boldsymbol{D}}')$$

$$H_2=S_{n_i}^{\pi_1}(\mathrm{SK}_2,[[\boldsymbol{g}_i^{(k)}]],\boldsymbol{I}_i^{(k)},[[\mathrm{dp}_i]],[[\mathrm{ra}_i]];\widetilde{\mathrm{coins}};[[\widetilde{r_i^{(k)}}]])$$

在理想执行环境下，$S_{n_i}^{\pi_1}$生成与真实环境下$\{B',\boldsymbol{D}'\}$概率分布一致的随机密文数据$\{\widetilde{B}',\widetilde{\boldsymbol{D}}'\}$，由于 DT-PKC 算法中部分解密满足语义安全，因此，

$H_0 \overset{c}{\approx} H_1$，其中，$\overset{c}{\approx}$ 表示计算不可区分。类似地，$S_{n_i}^{\pi_1}$ 按照与真实环境下相同的方式生成 $\widetilde{\text{coins}}$ 并构造信誉度密文 $[\![\widetilde{r_i^{(k)}}]\!]$，基于 DT-PKC 加密算法的安全性，$H_1 \overset{c}{\approx} H_2$。因此，可以得到 $H_0 \overset{c}{\approx} H_2$，即 $V_{n_i}^{\pi_1} \overset{c}{\approx} S_{n_i}^{\pi_1}$，意味着攻击者 $\mathcal{A}_{n_i}$ 无法区分真实环境与理想环境下的计算结果，因此，在拜占庭节点检测时，攻击者 $\mathcal{A}_{n_i}$ 无法在计算过程中获得用户隐私信息。针对攻击者 $\mathcal{A}_{\text{SP}}$，模拟器 $S_{\text{SP}}^{\pi_1}$ 模拟 SP 的真实执行环境，表示为 $V_{\text{SP}}^{\pi_1}(\text{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2; \text{coins}; [\![\boldsymbol{g}_i^{(k)}]\!], [\![\boldsymbol{I}_i^{(k)}]\!], [\![\text{dp}_i]\!], [\![\text{ra}_i]\!])$，其中，coins 为协议执行过程中引入的随机数。因此，$S_{\text{SP}}^{\pi_1}$ 执行如下步骤。

① 随机生成 DT-PKC 算法加密的密文 $\{[\![\widetilde{\boldsymbol{g}_i^{(k)}}]\!], [\![\widetilde{\boldsymbol{I}_i^{(k)}}]\!]\}$。

② 随机生成 DT-PKC 算法部分解密过程中需要的随机数 $\widetilde{\text{coins}}$。

③ 随机生成 DT-PKC 算法加密的密文 $\{[\![\widetilde{\text{dp}_i}]\!], [\![\widetilde{\text{ra}_i}]\!]\}$。

④ 输出 $(\text{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2; \widetilde{\text{coins}}; [\![\widetilde{\text{dp}_i}]\!], [\![\widetilde{\text{ra}_i}]\!])$。

定义如下状态等式

$$H_0 = V_{\text{SP}}^{\pi_1}(\text{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2)$$

$$H_1 = (\text{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2; [\![\widetilde{\boldsymbol{g}_i^{(k)}}]\!], [\![\widetilde{\boldsymbol{I}_i^{(k)}}]\!])$$

$$H_2 = S_{\text{SP}}^{\pi_1}(\text{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2; \widetilde{\text{coins}}; [\![\widetilde{\text{dp}_i}]\!], [\![\widetilde{\text{ra}_i}]\!])$$

在理想执行环境下，$S_{\text{SP}}^{\pi_1}$ 首先生成与节点 n_i 发送的密文服从同样分布的随机数据 $\{[\![\widetilde{\boldsymbol{g}_i^{(k)}}]\!], [\![\widetilde{\boldsymbol{I}_i^{(k)}}]\!]\}$，由于 DT-PKC 算法的安全性，$H_0 \overset{c}{\approx} H_1$。另外，由于 $\{\widetilde{\text{coins}}, [\![\widetilde{\text{dp}_i}]\!], [\![\widetilde{\text{ra}_i}]\!]\}$ 分别与 $\{\text{coins}, [\![\text{dp}_i]\!], [\![\text{ra}_i]\!]\}$ 具有相同的概率分布，因此，$H_1 \overset{c}{\approx} H_2$。所以，$H_0 \overset{c}{\approx} H_2$，即 $\mathcal{A}_{\text{SP}}$ 无法在此过程中获取除接收消息之外的其他隐私信息。

2) 在用户节点信誉度计算时，模拟器 $S_{\text{SP}}^{\pi_2}$ 模拟 SP 在真实场景下的执行过程，表示为 $V_{\text{SP}}^{\pi_2}(\text{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2; r_1, r_2; [\![Y]\!], [\![Z]\!])$。$S_{\text{SP}}^{\pi_2}$ 执行如下步骤。

① 随机生成随机数 $\{\widetilde{r_1}, \widetilde{r_2}\}$。

② 随机生成 DT-PKC 加密的密文数据 $\{[\![\widetilde{Y}]\!], [\![\widetilde{Z}]\!]\}$。

③ 输出 $(\text{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2; \widetilde{r_1}, \widetilde{r_2}; [\![\widetilde{Y}]\!], [\![\widetilde{Z}]\!])$。

定义如下状态等式

$$H_0 = V_{\text{SP}}^{\pi_2}(\text{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2)$$

$$H_1 = (\text{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2; \widetilde{r_1}, \widetilde{r_2})$$

$$H_2 = S_{\text{SP}}^{\pi_2}(\text{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2; [\![\widetilde{Y}]\!], [\![\widetilde{Z}]\!])$$

在理想执行环境下，$S_{\text{SP}}^{\pi_2}$ 依据 $\{r_1, r_2\}$ 的概率分布生成随机数 $\{\widetilde{r_1}, \widetilde{r_2}\}$，因此，$H_0 \overset{c}{\approx} H_1$。此外，$S_{\text{SP}}^{\pi_2}$ 构造密文 $\{[\![\widetilde{Y}]\!], [\![\widetilde{Z}]\!]\}$ 与 $\{[\![Y]\!], [\![Z]\!]\}$ 服从同样的分布，因此，基于 DT-PKC 算法的安全性，$H_1 \overset{c}{\approx} H_2$。所以，$V_{\text{SP}}^{\pi_2} \overset{c}{\approx} S_{\text{SP}}^{\pi_2}$，即攻击者 $\mathcal{A}_{\text{SP}}$ 在此过程中无法获得用户的隐私信息。此外，模拟器 $S_{n_i}^{\pi_2}$ 模拟节点 n_i 在此过程中的真实执行过程，表示为 $V_{n_i}^{\pi_2}(\text{SK}_2, [\![\boldsymbol{g}_i^{(k)}]\!], \boldsymbol{I}_i^{(k)}, [\![\text{dp}_i]\!], [\![\text{ra}_i]\!]; \text{coins}; R_i'', X'')$，其中，coins 表示协议执行过程中 DT-PKC 加密生成的随机数。$S_{n_i}^{\pi_2}$ 执行如下步骤。

① 随机生成 DT-PKC 算法部分解密的密文数据 $\widetilde{R_i''}$。

② 随机生成协议执行过程中的随机数 $\widetilde{\text{coins}}$。

③ 随机生成 DT-PKC 算法部分解密的密文数据 $\widetilde{X''}$。

④ 输出 $(\text{SK}_2, [\![\boldsymbol{g}_i^{(k)}]\!], \boldsymbol{I}_i^{(k)}, [\![\text{dp}_i]\!], [\![\text{ra}_i]\!]; \widetilde{\text{coins}}; \widetilde{R_i''}, \widetilde{X''})$。

定义如下状态等式

$$H_0 = V_{n_i}^{\pi_2}(\text{SK}_2, [\![\boldsymbol{g}_i^{(k)}]\!], \boldsymbol{I}_i^{(k)}, [\![\text{dp}_i]\!], [\![\text{ra}_i]\!])$$

$$H_1 = (\text{SK}_2, [\![\boldsymbol{g}_i^{(k)}]\!], \boldsymbol{I}_i^{(k)}, [\![\text{dp}_i]\!], [\![\text{ra}_i]\!]; \widetilde{R_i''})$$

$$H_2 = (\text{SK}_2, [\![\boldsymbol{g}_i^{(k)}]\!], \boldsymbol{I}_i^{(k)}, [\![\text{dp}_i]\!], [\![\text{ra}_i]\!]; \widetilde{\text{coins}})$$

$$H_3 = S_{n_i}^{\pi_2}(\text{SK}_2, [\![\boldsymbol{g}_i^{(k)}]\!], \boldsymbol{I}_i^{(k)}, [\![\text{dp}_i]\!], [\![\text{ra}_i]\!]; \widetilde{X''})$$

在理想执行环境下，$S_{n_i}^{\pi_2}$ 首先生成与 R_i'' 概率分布一致的密文 $\widetilde{R_i''}$，由于 DT-PKC 算法部分解密的安全性，$H_0 \overset{c}{\approx} H_1$。由于 $\widetilde{\text{coins}}$ 与随机数 coins 具有一致的概率分布，$H_1 \overset{c}{\approx} H_2$。同理，$H_2 \overset{c}{\approx} H_3$。因此，$V_{n_i}^{\pi_2} \overset{c}{\approx} S_{n_i}^{\pi_2}$，即攻击者 $\mathcal{A}_{n_i}$ 无法区分理想环境与真实环境下的协议执行过程。

3) 在全局聚合梯度更新时，模拟器 $S_{\text{SP}}^{\pi_3}$ 模拟 SP 在真实场景下的执行过程，表示为 $V_{\text{SP}}^{\pi_3}(\text{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2, [\![C_i^{(k)}]\!]; [\![\boldsymbol{g}_i^{(k)}]\!])$。$S_{\text{SP}}^{\pi_3}$ 执行如下步骤。

① 随机生成 DT-PKC 加密的密文梯度 $[\![\widetilde{\boldsymbol{g}_i^{(k)}}]\!]$。

② 输出 $(\text{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2, [\![C_i^{(k)}]\!]; [\![\widetilde{\boldsymbol{g}_i^{(k)}}]\!])$。

定义如下状态等式

$$H_0 = V_{\text{SP}}^{\pi_3}(\text{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2, [\![C_i^{(k)}]\!])$$

$$H_1 = S_{\mathrm{SP}}^{\pi_3}(\mathrm{SK}_1, r_i^{(0)}, C_i^{(0)}, \varepsilon_1, \varepsilon_2, [\![C_i^{(k)}]\!]; [\![\widetilde{\boldsymbol{g}_i^{(k)}}]\!])$$

在理想执行环境下，假设随机生成的$[\![\widetilde{\boldsymbol{g}_i^{(k)}}]\!]$与$[\![\boldsymbol{g}_i^{(k)}]\!]$具有相同的概率分布，因此，在 DT-PKC 加密算法的安全性条件下，$H_0 \overset{c}{\approx} H_1$，即$V_{\mathrm{SP}}^{\pi_3} \overset{c}{\approx} S_{\mathrm{SP}}^{\pi_3}$。

综上所述，攻击者$(\mathcal{A}_{\mathrm{SP}}, \mathcal{A}_{n_i})$无法区分理想执行环境和真实执行环境的协议计算过程。因此，可以认为本文所设计的安全鲁棒梯度聚合算法保护了用户节点数据和全局模型的隐私，能够抵抗非共谋条件下的半诚实敌手。

证毕。

5.2　效率分析

本节从理论上分析安全聚合算法的通信开销。训练开始前，KGC 需要将公私钥对按协议分发给 UN 和 SP，KGC 需要消耗$O(M)$的通信开销。训练开始时，SP 需要将模型参数分发给 UN，SP 消耗$O(dM)$通信开销。接着，UN 基于本地数据进行模型训练，训练结束之后进行加密计算。UN 将加密梯度上传给 SP 需要消耗$O(2d)$。SP 更新参考梯度、计算参考梯度的范数、UN 本地梯度向量与参考梯度向量的点积等，在这个过程中，需要多次使用到 SMP 算法，SP 需要消耗$O(2dM^{(k)})$，同时，UN 也需消耗$O(2d)$协助计算。SP 把$\{[\![B^{(k)}]\!], [\![\boldsymbol{D}_i^{(k)}]\!]\}$及其部分解密结果发送给 UN，消耗$O(4dM^{(k)})$。之后 UN 基于解密结果，计算本地梯度和参考梯度向量之间的方向相似度和范数相似度，加密后发送给 SP。SP 需要秘密计算 UN 梯度的可信度，多次使用 SLT 算法，SP 消耗$O(4M^{(k)})$。协议中，UN 参与过程的消耗可忽略。接着，密文状态下更新 UN 的信誉度，SP 需要消耗$O(2M^{(k)})$。最后更新和分发聚合梯度，SP 需要消耗$O(3dM^{(k)})$，UN 需要消耗$O(d)$。综上所述，在一轮模型训练中，KGC 需要消耗$O(N)$通信开销，SP 需要消耗$O(6M^{(k)} + dM^{(k)} + 9dM^{(k)})$通信开销，单个 UN 需要消耗$O(5d)$通信开销。

6　实验评估

本节主要从服务模型训练的准确率和效率 2 个角度对本文算法进行评估，实验环境如下：Intel(R) Xeon(R) Gold 6146 CPU，128 GB 内存， NVIDIA GeForce RTX 3090 GPU。

6.1　实验设置

本节分别基于 MNIST 和 CIFAR-10 数据集对本文算法进行评估。为构造 Non-IID 的数据分布，分别把 2 种训练集按照标签类别进行排序，把排序后的数据划分为 M 个子集，将每个子集随机分发给 UN。因此，每个 UN 只拥有部分类别的部分数据集。本文考虑的拜占庭攻击包含以下 3 类攻击。1) 高斯攻击：拜占庭节点共享梯度的每个元素服从高斯分布$\mathcal{N}(0,16)$。2) 符号反转：拜占庭节点对正常更新的本地梯度进行反转，计算$\boldsymbol{g}_i^{(k)} = \mathcal{R}_1 \boldsymbol{g}_i^{(k)}$($\mathcal{R}_1 < 0$)后上传至服务提供者，本文设置$\mathcal{R}_1 = -1$。3) 样本重复：拜占庭节点上传的本地梯度$\boldsymbol{g}_i^{(k)} = \mathcal{R}_2 \mathbf{1}$，其中，$\mathbf{1}$是元素全为1的向量；$\mathcal{R}_2$是一个常数，本文设置$\mathcal{R}_2 = 2$。

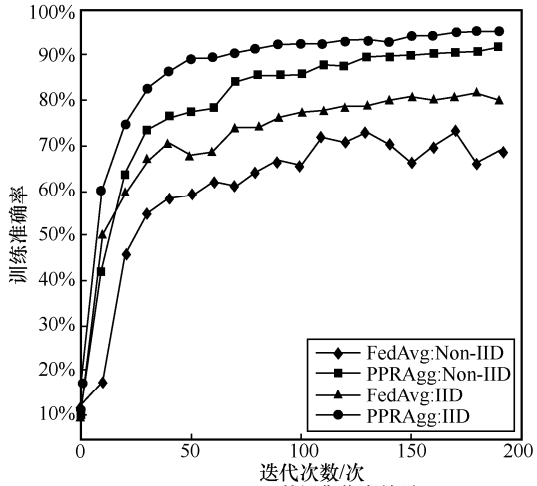
系统中的超参数设置如下。固定全局迭代次数为 200，用户节点的本地训练轮数为 5，系统中 UN 的数量$M = 50$，拜占庭节点的比例$f = 0.3$，每轮训练中参与训练的 UN 比例为 0.3（包含若干拜占庭节点），默认的攻击方式为高斯攻击。对于 MNIST 数据集，本地 batch 大小为 10，学习率 η=0.005；对于 CIFAR10，本地 batch 大小为 64，η=0.05。

评估模型准确率的对比算法包括 Krum[8]、RSA[19]、ByrdSAGA[17]和 Zeno[21]；评估效率的对比算法包括传统的 FedAvg 算法和非隐私保护的鲁棒梯度聚合（NPPRAgg, non-privacy-preserving robust gradient aggregation）算法。

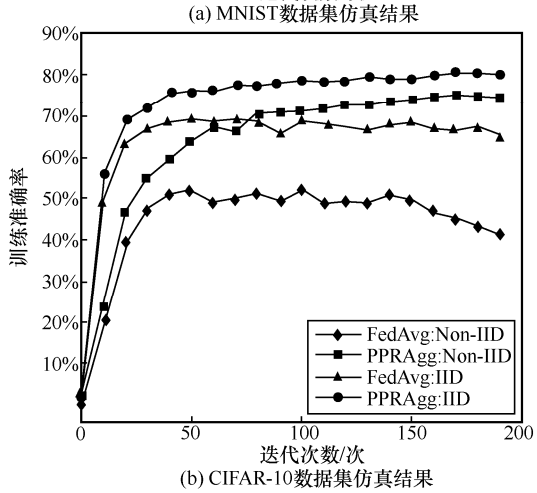
6.2　准确率评估

本节首先分析数据集分布对拜占庭节点识别的影响，仿真结果如图 5 所示，这里考虑的攻击是高斯攻击，图 5(a)和图 5(b)分别表示在 MNIST 和 CIFAR-10 数据集的仿真结果。以图 5(a)为例，当数据集为 IID 时，PPRAgg 算法得到的模型准确率达 95%，FedAvg 算法的准确率仅为 80%左右。因为拜占庭节点共享的恶意梯度对模型训练产生了影响，使用 FedAvg 算法无法得到高准确率的全局模型，然而 PPRAgg 算法却能有效地抵抗拜占庭攻击。在训练过程中，PPRAgg 算法基于梯度相似度准确地识别出拜占庭节点，并基于信誉度评估减小恶意梯度的影响，最终得到高准确率的全局模型。当用户数据为 Non-IID 时，FedAvg 算法的模型收敛速度缓慢且准确率曲线出现振荡。这是因为当数据特征为 Non-IID 时，UN 的数据分布不平衡，导致训练得到的模型收敛较慢，泛化能力较差。但是，从仿真结果可知，PPRAgg 算法在 Non-IID 下依然能够收敛，最终也能得到准确率较

高的全局模型（MNIST 数据集为 92%，CIFAR-10 数据集为 75%）。所以，PPRAgg 算法在 Non-IID 数据下也可以有效抵抗拜占庭攻击。为说明算法的有效性，后续实验只在 Non-IID 数据下进行。



(a) MNIST数据集仿真结果



(b) CIFAR-10数据集仿真结果

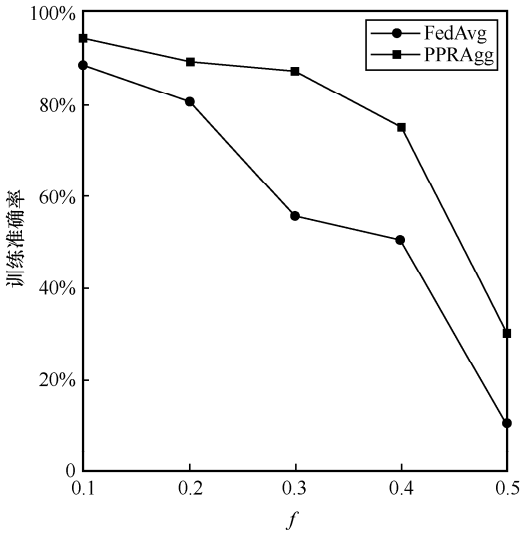
图 5 数据集分布对拜占庭节点识别的影响

不同攻击方式下，不同拜占庭节点比例对模型训练准确率的影响如图 6 所示，其中图 6(a)和图 6(b)中的结果已在文献[29]中得到验证。当系统中存在的拜占庭节点比例 $f=0.1$ 时，PPRAgg 算法和 FedAvg 算法都能完成模型的训练。但是，随着拜占庭节点比例的增加，FedAvg 算法的抵抗效果越来越差，训练准确率逐渐降低，最终无法收敛。然而，PPRAgg 算法的抗拜占庭攻击能力较强。尤其在 MNIST 数据集中，当 $f=0.4$ 时，训练模型依然能最终收敛并达到较高的准确率。但当 f=0.5 时，PPRAgg 算法无法区分出真实梯度和恶意梯度（此时无法判断模型收敛的训练方向），因此无法准确识别出系统中拜占庭节点，导致模型训练失败。相较于高斯攻击，在符号反转和样本重复下，FedAvg 算法得到的模型准确率会更低，当 f>0.2 时，使用 FedAvg 算法无法得到正确的模型。由实验结果可知，当 f<0.5 时，PPRAgg 算法抵抗拜占庭攻击的能力远优于 FedAvg 算法。

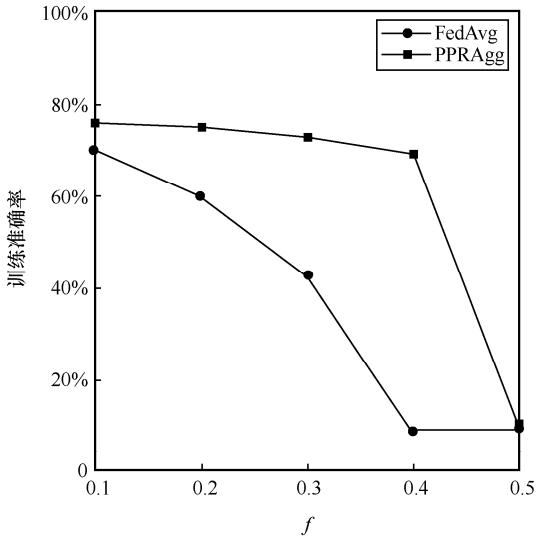
在此基础上，本文考虑了模型训练过程中突然出现的拜占庭节点对模型训练的影响。实验结果如图 7 所示，其中，高斯攻击模式下的仿真结果已在文献[29]中得到验证。

实验中设置前 50 轮训练正常进行，参与训练的节点全是正常节点，无拜占庭节点加入；而在第 50 轮之后的每轮训练中都随机加入比例为 $f=0.3$ 的拜占庭节点，模拟正常的 UN 在训练过程中因为软件或硬件突然的故障，而上传“恶意”梯度的现象。在第 50 轮训练开始出现拜占庭节点后，FedAvg 算法模型准确率立即下降，最终模型训练出现振荡，无法收敛。突然出现的拜占庭节点上传的错误梯度使平均梯度逐渐偏离正常的收敛方向，最终模型无法收敛。仿真结果显示，PPRAgg 算法的准确率曲线出现了“凹”点现象，这是因为当系统中出现拜占庭节点后，PPRAgg 算法立即做出反应识别拜占庭节点，并重新建立 UN 的信誉度。然而，更新 UN 信誉度的过程需要经历一段时间，所以模型的准确率会出现短暂的下降，当信誉度更新完成后，模型的准确率会逐渐上升，最终达到收敛状态。因此，实验结果表明 PPRAgg 算法能够抵抗训练过程中突然出现的拜占庭节点。

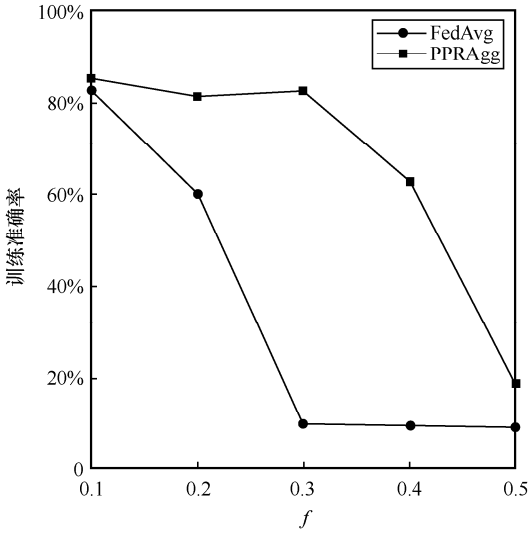
最后，将 PPRAgg 算法和 4 种抗拜占庭攻击的聚合算法的训练准确率进行对比，实验结果如图 8 所示。从图 8 可以发现，Krum 算法和 ByrdSAGA 算法在面向 Non-IID 数据时抵抗拜占庭攻击的能力较弱。这 2 种算法是基于距离和均值等统计学检测方案，当面对 Non-IID 的数据分布时，本地梯度之间差距较大，无法通过统计学方法检测出拜占庭节点。RSA 算法通过在目标函数中加入正则项，来减轻拜占庭节点的影响。该算法有一定的抵抗拜占庭攻击的能力，但是在不同的攻击下稳定性不足。同时，Zeno 算法提出基于性能的评估方法，该算法表现出一定抵抗拜占庭攻击的能力。但是 Zeno 算法通过计算损失函数值进行性能评估时需要从全局中抽取若干样本，由于具有随机性，Zeno 算法训练模型得到的准确率不稳定。实验结果表明，在数据集分布特征为 Non-IID 的情况下，PPRAgg 算法能有效抵抗拜占庭攻击，最终训练得到高准确率的全局模型。



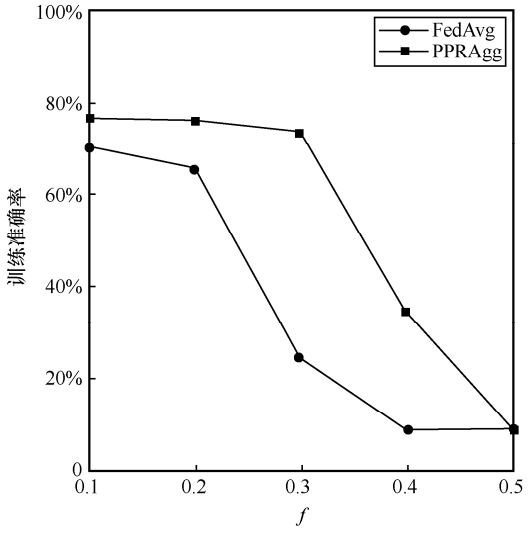
(a) 高斯攻击-MNIST



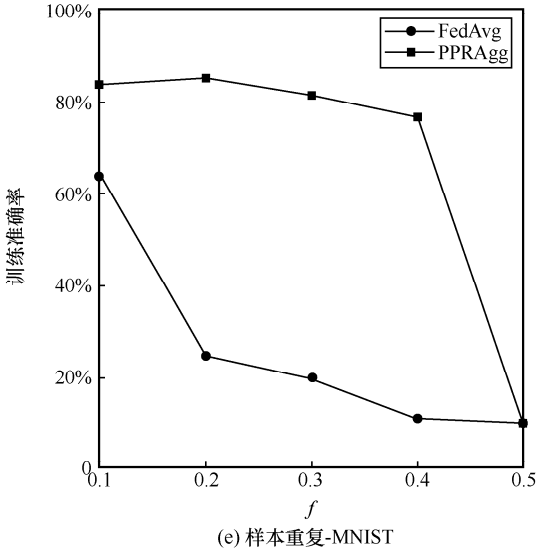
(b) 高斯攻击-CIFAR-10



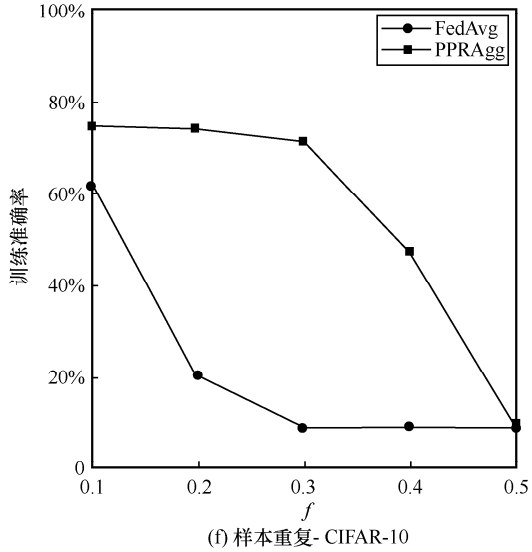
(c) 符号反转-MNIST



(d) 符号反转-CIFAR-10

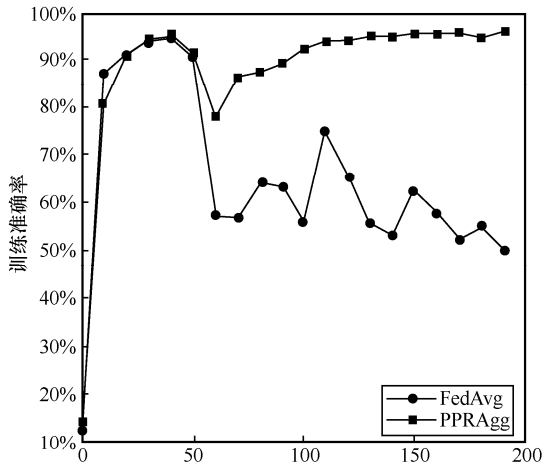


(e) 样本重复-MNIST

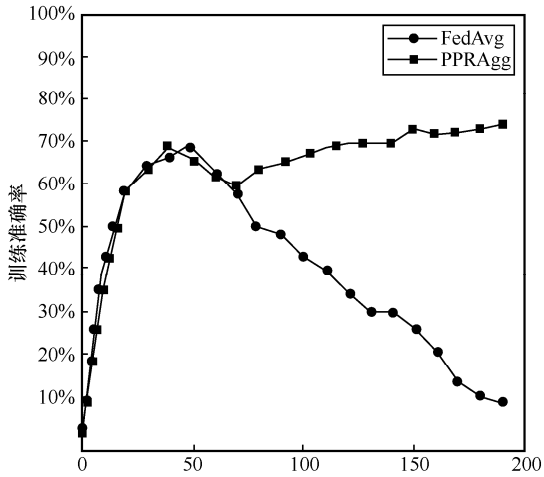


(f) 样本重复- CIFAR-10

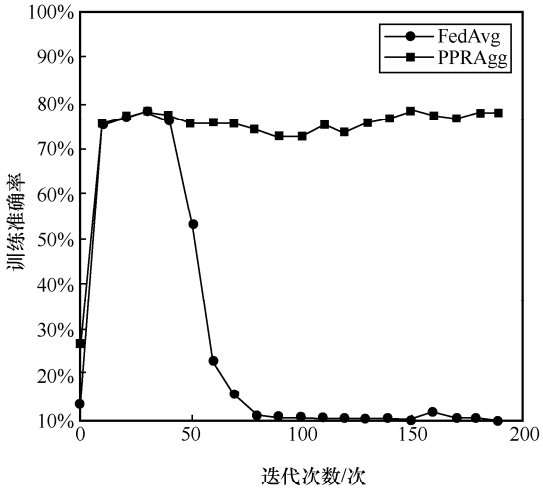
图 6　不同攻击方式下，不同拜占庭节点比例对模型训练准确率的影响



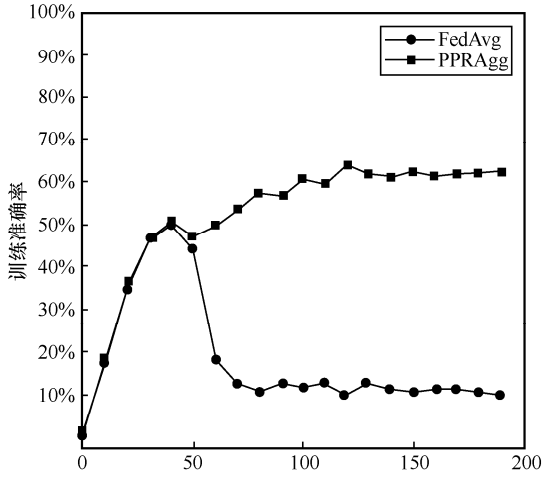
(a) 高斯攻击-MNIST



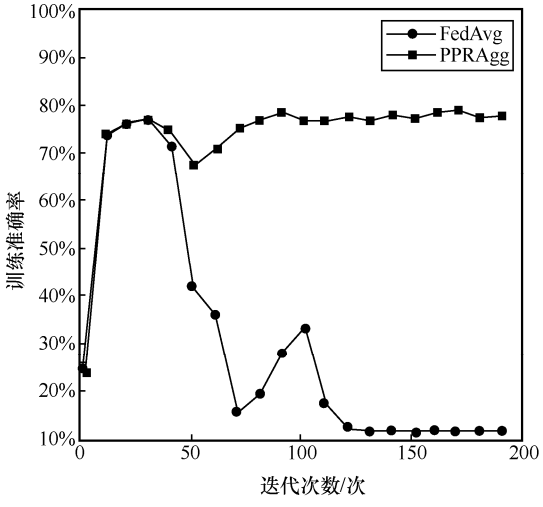
(b) 高斯攻击-CIFAR-10



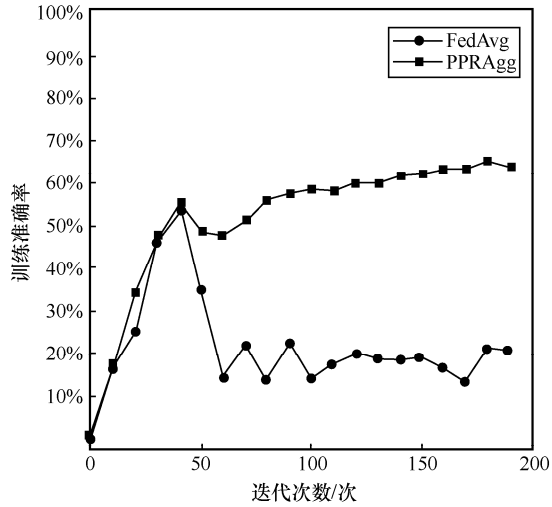
(c) 符号反转-MNIST



(d) 符号反转- CIFAR-10



(e) 样本重复-MNIST



(f) 样本重复CIFAR-10

图 7　模型训练过程中突然出现的拜占庭节点对模型训练的影响

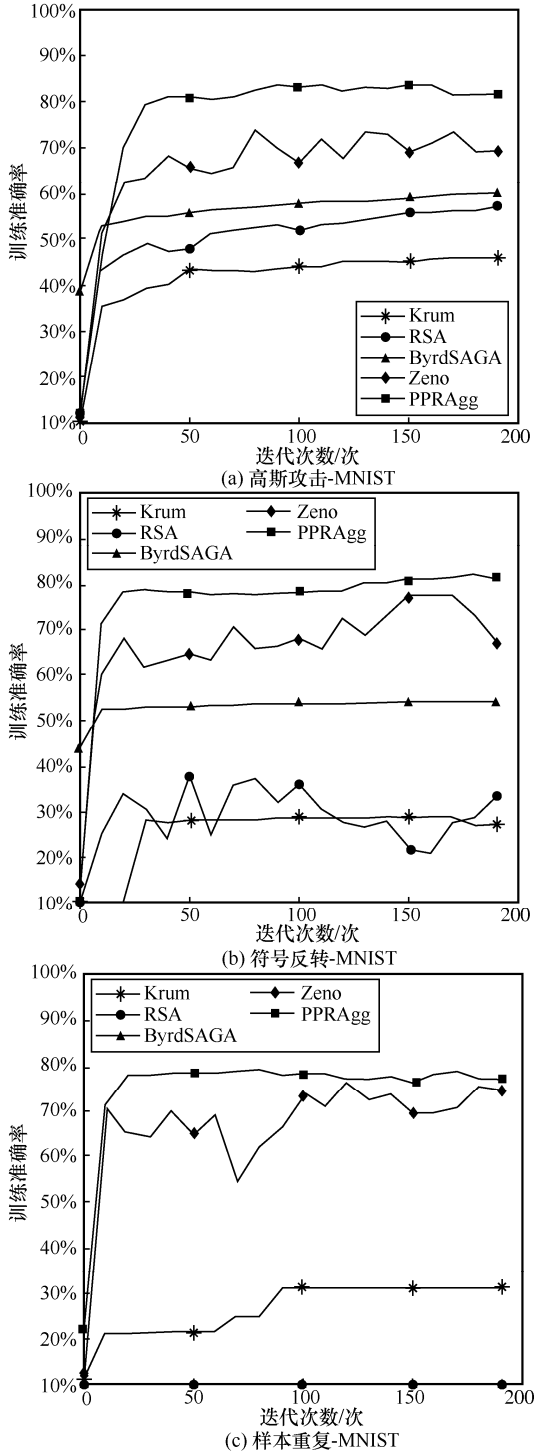


图 8　PPRAgg 算法和 4 种抗拜占庭攻击的聚合算法的训练准确率对比

6.3　效率评估

本节测试不同算法下 SP 聚合梯度时间和 UN 本地模型训练时间，实验结果如图 9 所示。图 9(a)测试了不同算法下 SP 聚合梯度所消耗的时间变化，图 9(b)测试的是单个 UN 完成本地模型训练所消耗的时间。由图 9(a)可知随着参与训练 UN 的比例不断增加，3 种聚合算法 SP 的聚合时间逐步增加。因为 UN 增加，SP 需要聚合梯度的数量也不断增加。但 NPPRAgg 算法和 PPRAgg 算法消耗的时间增幅尤其明显。原因在于，除了聚合梯度外，上述 2 种聚合算法需要消耗更多的时间用于检测系统中的拜占庭节点以及更新 UN 的信誉度。图 9(b)表明，UN 本地训练模型的时间不会随着参与 UN 比例的增加而明显变化。因为，UN 进行本地模型训练是独立进行的过程，不需要和其他实体进行交互，所消耗的训练时间是相对稳定的。然而，不管是 SP 的聚合时间还是 UN 的训练时间，使用 PPRAgg 算法消耗的时间都远高于使用 NPPRAgg 算法和 FedAvg 算法。这是因为 PPRAgg 算法使用了同态加密技术来实现隐私保护的模型训练，需要消耗更多的时间来完成数据的加解密运算。但由于本文算法应用在模型训练阶段，较长时间的模型训练并不影响模型的应用，同时，可采用 GPU 优化等方式提高密文数据模型训练的效率。

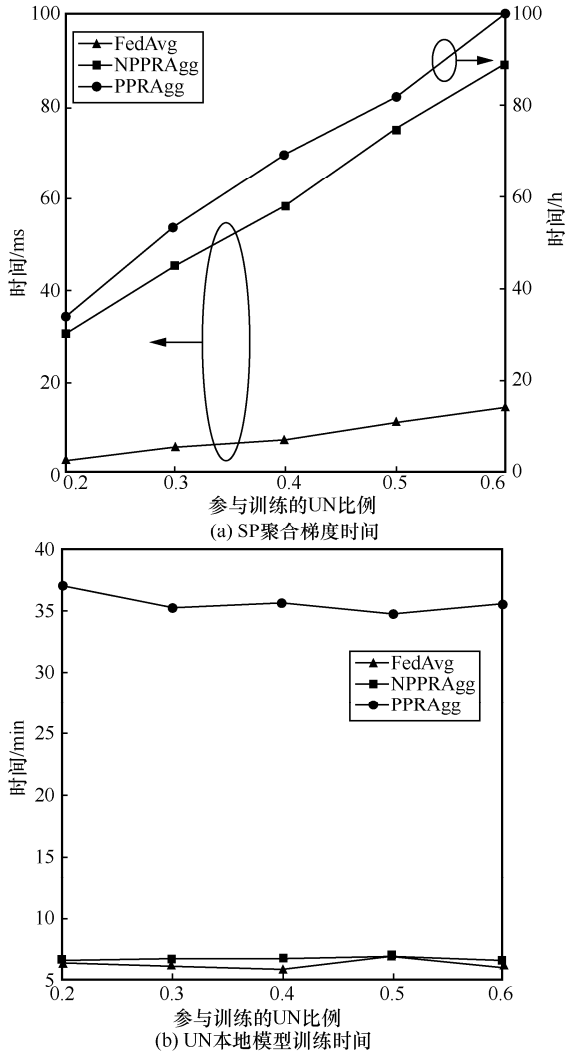


图 9　不同算法下 SP 聚合梯度时间和 UN 本地模型训练时间

7 结束语

在 Non-IID 数据集的背景下，本文提出了隐私保护的鲁棒性梯度聚合算法，用以抵抗联邦学习中的拜占庭攻击，达到了隐私保护模型训练。首先，本文基于 DT-PKC 同态加密和随机噪声混淆技术设计了隐私保护的安全聚合协议；其次，本文引入了参考梯度和信誉度，基于梯度的相似度识别联邦学习模型训练中的拜占庭攻击节点，同时利用信誉度模型准确地更新参考梯度。此外，本文分别从理论分析和仿真实验上，对隐私保护梯度聚合算法进行了评估。结果表明，所提算法能够达成本文设计目标，在抵抗拜占庭攻击的同时，实现用户隐私保护的目标。

参考文献：

[1] YANG Q, LIU Y, CHEN T J, et al. Federated machine learning: concept and applications[J]. ACM Transactions on Intelligent Systems and Technology, 2019, 10(2): 1-19.

[2] 杨强. AI与数据隐私保护: 联邦学习的破解之道[J]. 信息安全研究, 2019, 5(11): 961-965.
YANG Q Y. AI and data privacy protection: the way to federated learning[J]. Journal of Information Security Research, 2019, 5(11): 961-965.

[3] KAIROUZ P, MCMAHAN H B, AVENT B, et al. Advances and open problems in federated learning[J]. Foundations and Trends in Machine Learning, 2021, 14(1-2): 1-210.

[4] LAMPORT L, SHOSTAK R, PEASE M. The Byzantine general problem[M]. New York: ACM Books, 2019.

[5] CHEN Y D, SU L L, XU J M. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent[C]//Proceedings of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems. New York: ACM Press, 2018: 96-96.

[6] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Proceedings of Artificial intelligence and statistics. New York: PMLR, 2017: 1273-1282.

[7] LI X, HUANG K, YANG W, et al. On the convergence of FedAvg on non-IID data[J]. arXiv Preprint, arXiv: 1907.02189, 2019.

[8] BLANCHARD P, ELMHAMDI E M, GUERRAOUI R, et al. Machine learning with adversaries: Byzantine tolerant gradient descent[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: ACM Press, 2017: 118-128.

[9] LI T, SAHU A K, TALWALKAR A, et al. Federated learning: challenges, methods, and future directions[J]. IEEE Signal Processing Magazine, 2020, 37(3): 50-60.

[10] ZHU L, LIU Z, HAN S. Deep leakage from gradients[J]. arXiv Preprint, arXiv: 1906.08935, 2019.

[11] HU R, GUO Y X, LI H N, et al. Personalized federated learning with differential privacy[J]. IEEE Internet of Things Journal, 2020, 7(10): 9530-9539.

[12] BELL J H, BONAWITZ K A, GASCÓN A, et al. Secure single-server aggregation with (poly)logarithmic overhead[C]//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2020: 1253-1269.

[13] ZHANG C, LI S, XIA J, et al. Batchcrypt: efficient homomorphic encryption for cross-silo federated learning[C]//Proceedings of the 2020 USENIX Annual Technical Conference. Berkeley: USENIX Association, 2020: 493-506.

[14] XIE C, KOYEJO O, GUPTA I. Generalized byzantine-tolerant SGD[J]. arXiv Preprint, arXiv:1802.10116, 2018.

[15] GUERRAOUI R, ROUAULT S. The hidden vulnerability of distributed learning in Byzantium[C]//Proceedings of International Conference on Machine Learning. New York: PMLR, 2018: 3521-3530.

[16] YIN D, CHEN Y, KANNAN R, et al. Byzantine-robust distributed learning: towards optimal statistical rates[C]//Proceedings of International Conference on Machine Learning. New York: PMLR, 2018: 5650-5659.

[17] 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.

[18] HE L, KARIMIREDDY S P, JAGGI M. Byzantine-robust learning on heterogeneous datasets via resampling[J]. arXiv Preprint, arXiv: 2006.09365, 2020.

[19] LI L, XU W, CHEN T, et al. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2019: 1544-1551.

[20] PRAKASH S, AVESTIMEHR A S. Mitigating Byzantine attacks in federated learning[J]. arXiv Preprint, arXiv:2010.07541, 2020.

[21] XIE C, KOYEJO S, GUPTA I. Zeno: distributed stochastic gradient descent with suspicion-based fault-tolerance[C]//Proceedings of International Conference on Machine Learning. New York: PMLR, 2019: 6893-6901.

[22] CAO X Y, FANG M H, LIU J, et al. FLTrust: Byzantine-robust federated learning via trust bootstrapping[C]//Proceedings of 2021 Network and Distributed System Security Symposium. Virginia: the Internet Society, 2021: 1-18.

[23] ZHAI K, REN Q, WANG J, et al. Byzantine-robust federated learning via credibility assessment on non-IID data[J]. arXiv Preprint, arXiv: 2109.02396, 2021.

[24] PENG J, WU Z, LING Q, et al. Byzantine-robust variance-reduced federated learning over distributed non-IID data[J]. Information

Sciences, 2022, 616: 367-391.

[25] CHEN M, MAO B, MA T. FedSA: a staleness-aware asynchronous federated learning algorithm with non-IID data[J]. Future Generation Computer Systems, 2021, 120: 1-12.

[26] HE L, KARIMIREDDY S P, JAGGI M. Secure Byzantine-robust machine learning[J]. arXiv Preprint, arXiv: 2006.04747, 2020.

[27] SO J, GÜLER B, AVESTIMEHR A S. Byzantine-resilient secure federated learning[J]. IEEE Journal on Selected Areas in Communications, 2021, 39(7): 2168-2181.

[28] KHAZBAK Y, TAN T X, CAO G H. MLGuard: mitigating poisoning attacks in privacy preserving distributed collaborative learning[C]//Proceedings of 29th International Conference on Computer Communications and Networks. Piscataway: IEEE Press, 2020: 1-9.

[29] MA X D, JIANG Q, SHOJAFAR M, et al. DisBezant: secure and robust federated learning against Byzantine attack in IoT-enabled MTS[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(2): 2492-2502.

[30] KENNEY J F, KEEPING E S. Mathematics of statistics-part one[J]. 1954, 43(242): 332-335.

[31] LIU X M, DENG R H, CHOO K K R, et al. An efficient privacy-preserving outsourced calculation toolkit with multiple keys[J]. IEEE Transactions on Information Forensics and Security, 2016, 11(11): 2401-2414.

[32] BOST R, POPA R A, TU S, et al. Machine learning classification over encrypted data[C]//Proceedings of 2015 Network and Distributed System Security Symposium. Virginia: the Internet Society, 2015: 1-15.

[33] KAMARA S, MOHASSEL P, RAYKOVA M. Outsourcing multi-party computation[J]. Cryptology ePrint Archive, 2011, 1(1): 1-41.

[作者简介]

马鑫迪（1989− ），男，山东淄博人，博士，西安电子科技大学副教授、硕士生导师，主要研究方向为数据安全、隐私保护、人工智能安全等。

李清华（1998− ），男，江西吉安人，西安电子科技大学硕士生，主要研究方向为隐私保护、联邦学习等。

姜奇（1983− ），男，安徽全椒人，博士，西安电子科技大学教授，主要研究方向为安全协议分析、无线网络安全。

马卓（1980− ），男，陕西延安人，博士，西安电子科技大学教授、博士生导师，主要研究方向为人工智能与无人系统安全、无线网络安全等。

高胜（1987− ），男，湖北黄冈人，博士，中央财经大学副教授，主要研究方向为数据安全与隐私保护、区块链技术及应用。

田有亮（1982− ），男，贵州盘州人，博士，贵州大学教授、博士生导师，主要研究方向为算法博弈论、密码学与安全协议、大数据安全与隐私保护、区块链与电子货币等。

马建峰（1963− ），男，陕西西安人，博士，西安电子科技大学教授、博士生导师，主要研究方向为密码学、无线和移动安全等。