联邦学习中抵抗大量后门客户端的鲁棒聚合算法

王永康[1)] 翟弟华[1),2)] 夏元清[1)]

[1)](北京理工大学自动化学院 北京 100081)

[2)](北京理工大学长三角研究院(嘉兴) 浙江 嘉兴 314001)

摘 要 随着数据的爆炸式增长以及企业和个人对隐私问题的关注,传统的集中式机器学习已经不能满足现有的需求.联邦学习是一种新兴的分布式机器学习框架,旨在不分享私有数据的前提下利用分散的客户端训练一个全局模型,解决数据隐私和数据孤岛问题.然而,由于联邦学习的分布式和隐私保护特性,其容易受到各种各样的攻击,后门攻击则是联邦学习系统受到的攻击之一.目前,业界已提出大量的鲁邦算法来抵抗联邦学习系统遭受的后门攻击.然而,现有的鲁棒算法大多有较强的假设,例如受到不同客户端数据分布和恶意后门客户端数量的限制.我们的研究表明了现有的鲁棒算法不能解决在非独立同分布场景下,大量后门客户端共同攻击的问题.为解决这一难题,本文提出了一种鲁棒算法 Poly.Poly 算法包含两部分:一部分利用相似度矩阵和聚类算法进行聚类分析;另一部分则基于余弦相似度选择最优的类去聚合全局模型.由于 Poly 算法能完全去除恶意后门模型,从而完全避免了后门污染全局模型.为了验证 Poly 算法的性能,实验利用了 MNIST、Fashion-MNIST、CIFAR-10 和 Reddit 四种数据集,考虑了数据不平衡和类别不平衡两种非独立同分布场景以及独立同分布场景.此外,后门客户端的数量以 10%为单位从 50%递增到 90%,以实现大量后门客户端攻击的场景,同时也对 Poly 算法在后门客户端少于正常客户端的场景进行了测试.实验结果显示,Poly 能够完全抵抗不同场景下的后门攻击,后门攻击成功率只有 1%左右(在一些场景下为 0%)的同时,获得了较好的主任务精度.相较之下,几种现有经典算法则完全失效,大都使得后门攻击成功率为 100%,这些表明了 Poly 算法的优越性.

关键词 联邦学习;后门攻击;鲁棒性;聚类;异构

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2023.01302

A Robust Aggregated Algorithm against a Large Group Backdoor Clients in Federated Learning System

WANG Yong-Kang[1)] ZHAI Di-Hua[1),2)] XIA Yuan-Qing[1)]

[1)](*School of Automation, Beijing Institute of Technology, Beijing* 100081)

[2)](*Yangtze Delta Region Academy of Beijing Institute of Technology, Jiaxing, Zhejiang* 314001)

Abstract With the explosion of data and concerns about privacy among businesses and individuals, traditional centralized machine learning is no longer able to satisfy the existing needs. Federated learning (FL) is a burgeoning distributed machine learning framework, in which multiple diverse clients collaboratively train a global model without sharing the private data, so as to solve the data silos and privacy problems. However, existing studies have demonstrated that FL is extremely vulnerable to all kinds of attacks due to its distributed and privacy-preserving inherent characteristics. Backdoor attack is one of the most prominent attacks in the FL system. To defend against the backdoor attacks in the FL system, a large number of algorithms robust aggregation algorithms are proposed. Nevertheless, these robust aggregation algorithms are restricted by some strong

收稿日期:2022-07-12;在线发布日期:2022-12-13.本课题得到云端赋能机器人高性能多约束控制理论与关键技术研究(62173035)和小米青年学者项目资助.**王永康**,博士研究生,主要研究方向为联邦学习、机器学习安全.E-mail:wang_yk@bit.edu.cn.**翟弟华**(通信作者),博士,副教授,主要研究方向为机器人智能感知、优化控制与应用.E-mail:zhaidih@bit.edu.cn.**夏元清**,博士,教授,国家杰出青年科学基金获得者,长江学者,主要研究领域为多源信息复杂系统的信息处理与控制、云控制与决策理论及其应用.

assumptions, such as the number of malicious clients and the data distribution across the diverse clients. Our study shows that the existing robust aggregation algorithms fully failed under a large group of malicious backdoor clients or non-independently identically distributed (Non-IID) scenarios. To address this problem, we propose a robust aggregation algorithm called Poly which contains two crucial components: one component uses similarity matrix and clustering algorithm to handle the gradients of all clients; another component selects the optimal clusters containing benign clients to aggregate the global model based on the cosine similarity metric. Our proposed Poly can completely remove all malicious backdoor clients in the aggregation process, thereby avoiding the backdoor inserting into the global model. To test the effectiveness of defending against backdoor attack of our proposed Poly, we leverage MNIST, Fashion-MNIST, CIFAR-10 and Reddit datasets to conduct a series of experiments under both data imbalance and class imbalance Non-IID scenarios, as well as the independently identically distributed scenario. In addition to this, we also consider a large group of malicious backdoor clients scenario in which the number of malicious backdoor clients ranges from 50% to 90% with a step 10%, as well as the scenario where the number of malicious backdoor clients is less than that of benign clients. Our experimental results indicate that our proposed Poly outperforms the existing robust aggregation algorithms, and can also effectively defend against backdoor attacks with only about 1% attack success rate (even 0% attack success rate in some scenarios) under the testing scenarios, even under the data imbalance and class imbalance Non-IID scenarios and a large group of malicious backdoor clients scenario. Beyond that, our proposed Poly can also achieve satisfying primary task accuracy, which indicates that our algorithm Poly does not affect the performance on the primary task that we care about while defending against the backdoor attack. By contrast, the existing robust aggregation algorithms can hardly defend against the backdoor attack under Non-IID scenarios and a large group of malicious backdoor clients, achieving nearly 100% attack success rate.

Keywords　federated learning; backdoor attacks; robust; clustering; heterogeneous

1　引　言

随着数据的爆炸式增长，受到带宽和计算资源的限制，把所有数据发往云端，在云端利用机器学习训练模型变得愈发困难. 此外，企业和个人逐渐认识到数据隐私的重要性，不愿贡献私有数据，导致产生了数据孤岛问题. 数据孤岛和隐私安全问题是限制机器学习发展的主要瓶颈之一[1-2].

联邦学习旨在解决上述的数据孤岛和隐私安全问题. 联邦学习是一种分布式机器学习框架，允许多方利用本地数据共同训练一个全局模型，解决了数据孤岛问题，且不用上传隐私数据，保护了隐私性[2]. 其中，经典的联邦学习框架是参数服务器架构(Parameter Server, PS)[3]，包含服务器和客户端. 具体来说，基于 PS 架构的联邦学习训练过程如下：(1) 服务器初始化全局模型，然后下发全局模型到各个客户端；(2) 各个客户端接收服务器下发的全局模型，利用本地数据训练该模型，并把更新的本地模型上传至服务器；(3) 服务器基于设定的聚合算法，利用收集到的各个客户端上传的模型更新出下一轮迭代的全局模型. 服务器和各个客户端不断循环上述的过程，直至全局模型收敛. 目前，联邦学习被广泛应用在医疗、保险、金融以及工业等各个领域[4].

然而，由于联邦学习的隐私保护和分布式特性，其极易受到各种攻击[4-5]. 具体来说，恶意客户端可以利用脏数据训练本地模型或者随意更改本地的模型参数，然后上传这些恶意模型至服务器，从而影响全局模型的行为或影响全局模型的收敛性和精度. 后门攻击是联邦学习容易受到的攻击之一，后门攻击的流程如图 1 所示. 在训练过程中，恶意客户端首先污染本地数据，即在本地数据集中插入后门，然后更改其标签为预设的类别(比如，在图片右下角植入三角形后门，并将其原有类别改为“马”)；而后利用这些插入后门的本地数据训练模型，这样模型被植入了后门，把插入后门的模型上传至服务器；服务器

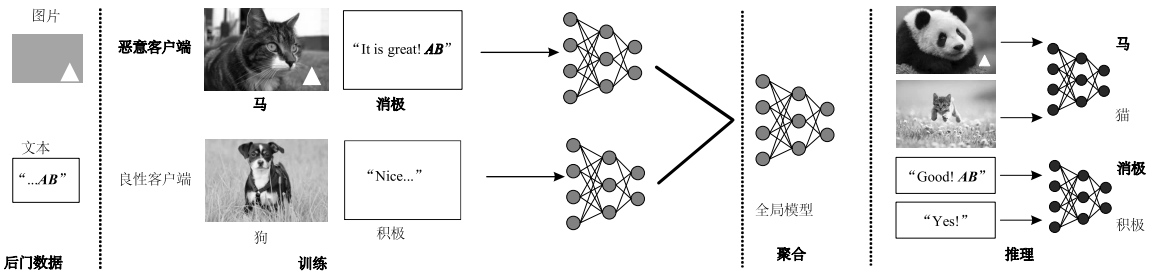


图 1 后门插入到全局模型的流程以及在推理时全局模型的表现

利用这些模型更新出全局模型，全局模型则被插入了后门. 在模型推理阶段，全局模型仍会把干净的样本分类为正确的类别，但是会把插入后门的样本分类为恶意客户端预设的类别. 这样，插入后门的全局模型在干净样本和插入后门的样本上均具有较高的精度. 在聚合阶段，由于联邦学习为保护隐私可能会对本地模型做了操作(例如在不影响精度的情况下，添加噪声或是梯度裁剪或者同态加密操作以防止从梯度推断原始数据)，且后门模型和正常模型在测试样本上均具有较高的精度，仅凭精度无法检测恶意模型，因此插入后门的本地模型很难被识别.

如何防御联邦学习中的攻击问题已经成为当下研究的热点. 目前，业界已提出了多种鲁棒聚合算法来防御联邦学习遭受的攻击，比如 Krum 以及变形 MultiKrum[6]、coordinate-wise median[7]、geometric median[8]、RFA[9] 等. 这些算法旨在利用均值或者中值估计全局模型的真实中心. 此外，还有一些算法利用辅助数据集去协助识别并剔除恶意模型[10-13]，从而避免恶意模型对全局模型的影响.

然而，基于均值或者中值的聚合算法存在一定的局限性，例如假设各个客户端之间的数据是独立同分布(Independently Identically Distributed，IID)，或者后门客户端的数量少于正常客户端. 然而，在一般的联邦学习系统中，各个客户端之间的数据分布通常是异构的，也就是非独立同分布(Non-Independently Identically Distributed，Non-IID)，且 Non-IID 场景分为数据不平衡(data imbalance)和类别不均衡(class imbalance)，目前大多数算法均未同时考虑到这两种场景. 此外，基于辅助数据的检测算法需要服务器收集一定量的辅助数据，且这些数据必须是干净的. 辅助数据一般由各个客户端贡献，各个客户端在本地数据进行独立同分布采样，然后上传到服务器. 然而，由于隐私保护原则，客户端不愿分享这些私有数据.

为了解决在 Non-IID(包含 data imbalance 和 class imbalance)情形下，联邦学习系统中遭受大量恶意客户端的后门攻击问题. 本文提出了鲁棒聚合算法 Poly，旨在解决这一难题. 设计 Poly 算法的动机很朴素，考虑到由于后门的存在，后门客户端提交的恶意模型参数总是会聚集在一起，这样会大大提高后门攻击的成功率. 然而，这也带来防御的契机，可以利用聚类算法对其进行分析. 当模型的维度非常高时，基于距离的聚类效果会非常差. 因此，本文对各个客户端提交的模型之间的相似性进行聚类分析，这样既降低了维度，又考虑到了各个客户端的更新方向，相比单纯利用模型参数聚类有了较大的改善. 本文提出的鲁棒聚合算法分为以下几步：(1) 首先计算各个客户端提交的模型梯度之间的余弦相似性；(2) 利用聚类算法对相似性集合进行聚类分析；(3) 基于设定的规则选择符合条件的簇；(4) 利用聚合算法对选择的簇进行聚合，更新出下一轮迭代的全局模型.

本文的主要贡献如下：

(1) 本文提出了一种联邦学习的鲁棒聚合算法，解决了联邦学习在 Non-IID 场景下遭受到大量恶意客户端的后门攻击问题. 首先，选择合适的模型梯度来计算各个客户端之间的相似性；然后，利用高斯混合模型聚类方法对相似性矩阵进行聚类；而后，基于相似性准则选择合适的簇；最后，利用基于中值的 RFA 方法和均值化方法处理选择的簇，得到全局模型.

(2) 本文提供了一种聚类后最优簇选择的方法及依据，根据后门客户端之间的相似性高于正常客户端，对各个簇内的梯度的相似性进行平均化处理，选取相似性低于阈值的簇进行聚合. 此外，对此方法的依据进行了分析，并在实验中得到了验证.

(3) 本文考虑了两种常见的 Non-IID 场景，即 data imbalance 和 class imbalance，验证了本文提出的鲁棒聚合算法的可行性和优越性. 除此之外，本文同样考虑了 IID 场景，验证了方法的泛化性.

(4) 本文基于 MNIST、Fashion-MNIST、CIFAR-10 和 Reddit 四种数据集，利用逻辑回归(Logistic

Regression,LR),卷积神经网络(Convolutional Neural Networks,CNN),深度神经网络(Deep Neural Networks,DNN)和长短期记忆网络(Long Short-Term Memory,LSTM)进行了实验.实验结果显示本文提出的算法能完全抵抗后门攻击,且优于当前的算法.

2　相关工作

目前,已提出了很多聚合算法来抵抗联邦学习中的攻击问题.Blanchard 等人[6]提出了 Krum 算法,旨在基于欧式距离,在所有客户端中选择最优的一个客户端提交的模型替代全局模型进行下一轮的更新,MultiKrum 是 Krum 的一种变体.Yin、Chen 和 Pillutla 等人[7-9]分别提出了 geometric median、coordinate-wise median 和 RFA 算法,这些算法均是基于中值进行分析的.Yin 等人[7]还提出了 Trimmed-Mean 算法,旨在剔除恶意模型更新的参数.Guerraoui 等人[14]结合 Krum 和 TrimmedMean 算法,提出了 Bulyan 算法,旨在选择最优的客户端去聚合全局模型.Mao 等人[15]提出了 Romoa,旨在缓解联邦学习中遭受的目标攻击和无目标攻击(后门攻击属于目标攻击,无目标攻击又称拜占庭攻击,旨在使得全局模型错误率提高或使其不收敛),Romoa 主要利用前瞻性的混合相似性度量各个客户端,然后为各个客户端分配一个变化的消毒系数,从而消除恶意客户端的影响.然而,这些算法共同存在的问题是假设各个客户端之间的数据分布是 IID 或是恶意客户端的数量少于良性客户端.

Li 等人[10,12]利用测试数据和 AutoEncoder 训练了恶意模型检测器,部署在服务器端,对各个客户端提交的模型参数进行检测,基于阈值剔除不合条件的模型更新.Xie 等人[11]提出了 Zeno,旨在运用各个客户端提交的辅助数据进行检测,剔除掉恶意的模型.Cao 等人[16]提出了 FLTrust,服务器本身收集一个干净的小训练数据集(即根数据集)来引导 FLTrust 中的信任,可以对大量恶意客户端实现鲁棒性.这些算法的一个共同特点是在服务器端均需要存在干净的数据集(一般由各个客户端在本地数据集中进行独立同分布采样,然后发送到服务器),收集干净数据集假设过强,一般不适用于真实场景的联邦学习系统.

针对联邦学习的后门攻击问题,Fung 等人[17]提出了 FoolsGold,计算各个客户端提交的模型之间的余弦相似性,根据相似性来降低恶意客户端对全局模型的贡献.DP 算法通过对原始数据添加合适的噪声,来保护数据的隐私性.当输入数据发生微小的波动时,输出不会改变,因此已被用在联邦学习的后门攻击防御中[18-19].Cao 等人[20]提出了通过学习多个全局模型来抵抗后门攻击.Xie 等人[21]提出了 CRFL 框,利用此框架训练可证明的鲁棒联邦学习模型来抵抗后门攻击.Nguyen 等人[22]提出了 FLAME 算法,旨在利用聚类算法,模型裁剪和噪声处理来抵抗后门攻击.然而,这些算法均未考虑 data imbalance 和 class imbalance 两种 Non-IID 场景.而且,Cao 和 Nguyen[20,22]提出的算法对恶意客户端的数量也做了一定的假设,不能抵抗大量恶意客户端联合攻击的场景.FoolsGold 虽然在文中并未对此做出假设,但是通过本文的实验证明了此方法表现不佳.

3　相关概念及问题描述

3.1　联邦学习

本文考虑基于 PS 架构的联邦学习系统,即包含一个服务器和多个客户端.联邦学习的优化目标可由式(1)表示,式中,N 代表客户端的数量,d 表示模型的维度,p_i 表示客户端 i 聚合时的权重,且有 $\sum_{i=1}^{N} p_i = 1$.各个客户端本地模型的优化目标为 F_i,可由式(2)表示,式中,n_i 表示本地数据量,$f(\cdot;\cdot)$ 表示损失函数,z_i^j 表示第 j 个样本.

$$\min_{w\in\mathbb{R}^d}\left\{F(w)\triangleq\sum_{i=1}^{N}p_iF_i(w)\right\} \tag{1}$$

$$F_i(w)\triangleq\frac{1}{n_i}\sum_{j=1}^{n_i}f(w;z_i^j) \tag{2}$$

本文利用随机梯度下降算法(Stochastic Gradient Descent,SGD)对目标函数进行优化.各个客户端在训练过程中上传梯度到服务器,服务器根据聚合出来的全局梯度进行 SGD 优化,可由式(3)表示.式中,w_{t-1} 表示上一轮的全局模型参数,η^t 表示学习率,g_i 表示客户端 i 计算的模型梯度,$[N]$ 为 $[1,\cdots,N]$.在本轮迭代中,各个客户端接收到全局模型之后,在本地数据集中随机抽取一个批次的数据计算梯度 g,可由式(4)表示,式中,b 和 ∇f 分别表示批尺寸和损失函数的梯度.

$$w_t\leftarrow w_{t-1}-\eta^t\,\mathrm{Aggr}(\{g_i:i\in[N]\}) \tag{3}$$

$$g=\frac{1}{b}\sum_{j=1}^{b}\nabla f(w_t;z^j) \tag{4}$$

3.2 攻击模型

本文考虑了两种常见的Non-IID场景：data imbalance 和 class imbalance，详细设置将在实验部分阐述. 各个客户端拥有各自的本地数据集 D_i，因此整个数据集为 $D=[D_1,D_2,\cdots,D_N]$. 在训练过程中(例如轮次 k 中)，客户端 i 从本地数据集 D_i 中随机抽取一个批次的数据来训练. 良性客户端会用原始数据进行训练，根据式(4)计算出梯度，然后发送至服务器，后门客户端将会在训练数据中插入后门. 在本文中，我们设定后门客户端将会污染本地训练的批次中 50%的样本，因此，后门客户端计算梯度如式(5)所示. 式中，B_k^1 和 B_k^2 分别表示一个批次中带有后门的数据和正常数据，γ 代表后门梯度的扩大倍数，与模型替代攻击[18]类似，γ 越大，后门攻击越强.

$$g_k=\gamma\left(\frac{2}{b}\sum_{z_k^j\in B_k^1}\nabla f(w_k;z_k^j)+\frac{2}{b}\sum_{z_k^j\in B_k^2}\nabla f(w_k;z_k^j)\right) \tag{5}$$

4 方法描述

在本节中，提出了一种联邦学习聚合算法 Poly，旨在防御联邦学习中大量后门客户端的攻击. 伪代码如下所示. 同时，结合伪代码对 Poly 算法的执行步骤进行了详细描述，且在执行步骤中详细阐述了设计 Poly 时遇到的挑战，以及如何解决挑战.

算法. Poly.

输入：初始全局模型 w_0，全局训练轮数 T，数据集 D，批尺寸 b，学习率 η

输出：全局模型 w_T

1. FOR $t=0,\cdots,T-1$ DO

客户端

2. FOR $i=1,\cdots,N$ DO
3. 客户端 i 接收服务器下发的全局模型 w_t
4. 在 D_i 随机抽取一个批次的数据，然后计算梯度

$$g_i=\frac{1}{b}\sum_{j=1}^{b}\nabla f(w_t;z_i^j)$$

5. 上传梯度 g_i 至服务器
6. ENDFOR

服务器

7. 下发全局模型 w_t 到各个客户端
8. 收集模型梯度 $G=[g_1,\cdots,g_N]$
9. 计算模型梯度相似性 $S=$cosine_similarity(G)
10. 对相似度集合进行聚类 $C=$Clustering(S)
11. 客户端梯度划分到相应簇内 $G'=[G_c,c\in C]$;
12. 对各个簇内的相似度取均值 $S_C=[S_c,c\in C]$
13. $g_{list}=[\,]$
14. FOR index, S_c IN ENUMERATE(S_C)
15. IF $S_c\leq$mean(S_C)
16. $g_{S_c}=$RFA$(G'[$index$]$)
17. g_{list}.append(g_{S_c})
18. $g_{update}=$mean(g_{list})
19. $w_{t+1}=w_t-\eta g_{update}$
20. ENDFOR
21. ENDFOR
22. 返回最终的全局模型 w_T

接下来，将详细阐述 Poly 算法的执行步骤. 在客户端训练层面，Poly 算法与一般联邦学习客户端训练类似，即各个客户端接收服务器下发的全局模型，然后随机抽取本地数据集中一个批次的数据进行训练，最后把梯度上传到服务器.

在服务器端，Poly 算法的执行步骤如下：

1. 接收各个客户端上传的梯度 $G=[g_1,g_2,\cdots,g_N]$.

2. 本文中，利用余弦距离来定量衡量模型梯度之间的相似性，因为余弦距离相比于欧拉距离而言不受模型梯度大小的影响，只与方向有关. 如果基于欧拉距离，则恶意客户端可通过伸缩其梯度参数值来逃避检测. 根据式(6)，计算各个梯度之间的余弦相似性，Poly 算法伪代码中的第 9 行中的 $S=$cosine_similarity(G) 代表利用式(6)计算 G 中任意两个梯度的相似性. 在文献[17]中，利用模型的最后一层计算余弦相似性效果最佳. 这是因为模型的最后一层与输出结果直接相关，最能反映模型之间的区别，如果利用全部参数计算相似性，根据梯度反向传播，一些更新小或者不更新的参数会影响模型之间的度量. 在本文中，与文献[17]相同，利用梯度的最后一层来计算余弦相似性，得到了 $N\times N$ 的相似度矩阵. 矩阵的各行为以此行为 id 的客户端与其余客户端的梯度的余弦相似性向量(例如，矩阵的第一行为客户端 1 与其余客户端梯度的余弦相似性).

$$s_{ij}=\text{cosine\_similarity}(g_i,g_j)=\frac{g_i\times g_j}{\|g_i\|\|g_i\|} \tag{6}$$

3. 利用聚类算法对相似性矩阵进行聚类分析. 然而，在此步骤中，有几处挑战需要解决：选择哪一种聚类算法进行分析？如何确定聚类算法中簇的数量？本文在 MNIST 数据集上对比了 K 均值聚类(K-means)、高斯混合模型聚类(Gaussian Mixture Model，GMM)、基于密度的聚类(Density-Based Spatial Clustering of Applications with Noise，DBSCAN)和凝层次聚类(Hierarchical Agglomerative Clustering，HAC)，测试结果 GMM 方法最好. 原因可能与数据分布有关，在 Non-IID 场景中，离群值是极易产生的，K-means 和 HAC 算法对异常值敏感，而 GMM 和 DBSCAN 对异常值不敏感. 但是，DBSCAN 算法需要确定超几何体半径的值，在不同轮次中半径的设置不一样，因此很难确定半径的值. 因此，最后选择了 GMM 作为聚类算法. 假如在基准数据集上进行测试及验证，可能会产生过拟合问题(GMM 可能只适用于测试数据集，在其余数据集中表现并不好). 因此，本文同样在 Fashion-MNIST、CIFAR-10 及 Reddit 上进行了实验，验证

了方法的可行性.

另一个挑战是如何确定簇的数量.在本文中,我们首先设置一个簇数量的区间,然后利用贝叶斯信息准则(Bayesian Information Criterion,BIC)去评估对应的聚类结果,最后对应的最小值就是要选择的簇的数量.BIC 的计算公式如式(7)所示,式中,N 和 d 分别表示样本数量和维度,L 为 GMM 模型中最大似然函数的最大值.经过测试,簇个数的区间设为[1,5],这样既满足了可寻找出最优的簇个数,又避免了搜索范围太大造成的耗时问题.

$$BIC=d\ln(N)-2\ln(L) \tag{7}$$

4. 根据步骤 3 中的聚类结果,把梯度划分到各个簇中,得到划分结果 $G'=[G_c,c\in C]$.

5. 对聚类结果 C 中的各个簇内的相似性集合取均值,得到结果 $S_C=[S_c,c\in C]$. S_C 中的各个值代表各个簇的衡量指标.

6. 选择哪些簇的梯度去聚合同样是本文的挑战之一.在后门攻击中,后门客户端联合攻击全局模型,提交的梯度之间的相似性是高于良性客户端的.这是因为由于后门的存在,恶意客户端之间的数据分布的异构性弱于良性客户端.具体而言,恶意客户端把后门添加到本地数据,并把标签改为预设的标签.在本文中,恶意客户端毒害一个批次中一半的数据,因此有 50% 的数据拥有相同的标签和类似的特征分布,致使恶意客户端之间的数据分布相比良性客户端更具有同质性(异构性弱).这种数据分布可以映射到模型梯度,因此恶意客户端之间的模型梯度比正常客户端更为相似.因此,正常客户端簇的 S_c 小于异常客户端簇.在本文中,把 S_C 的均值作为衡量簇为异常簇和正常簇的阈值,最后选择低于此阈值的簇的梯度去聚合.

7. 经过步骤 6,虽然选择了合适的簇去聚合,但是由于 Non-IID 的场景,在选择的簇内可能仍有少数的后门梯度存在,因此另一个挑战是如何解决选择簇内可能存在少量的恶意梯度.在同一个簇内,我们认为这个簇内的客户端的数据分布是近似 IID 分布.这一假设是合理的,因为只有梯度相似的才会被划分到一个簇中,而且梯度可以直接反映出客户端本地数据的分布.在同一簇内,利用 RFA[9] 算法对梯度进行聚合,然后得到本簇内聚合的梯度 g_{S_c}.由于 RFA 是一种基于中值的算法,对于少量异常值是鲁棒的,适用于 IID 场景,因此可用来抵御簇内残存的少量恶意模型梯度.然后对选择簇的 g_{S_c} 均值处理,即得到了本轮中更新的全局梯度 g_{update}.

8. 利用 SGD 算法更新下一轮迭代的模型 $w_{t+1}=w_t-\eta g_{update}$.

9. 经过 T 轮之后,返回全局模型 w_T.

服务器生成最终的全局模型 w_T 后,发送给各个客户端,各个客户端进行后续的推理测试验证.这样,各个客户端在不共享私有数据的前提下学习到了其余客户端的数据.此外,按照本流程执行后,期望的全局模型是干净的,能够准确预测正常样本,且对后门样本不敏感.

5 实　验

5.1 实验设置

在实验中,本文应用了基于 PS 架构的联邦学习框架,即有一个服务器和多个客户端.在任务方面,本文考虑了基于 MNIST[23]、Fashion-MNIST[24] 和 CIFAR-10[25] 的图像分类任务以及 Reddit[18] 的自然语言处理任务.四种数据集描述如下:

MNIST. MNIST 是一种手写数字集,包含 60000 张训练图片和 10000 张测试图片,图片是 28×28 的灰度图,一共 10 个类别.在后门攻击中,后门客户端把后门注入到部分的本地数据集中,同时把类别标签改为“7”.

Fashion-MNIST. Fashion-MNIST 是一种商品数据集,涵盖了来自 10 种类别的共 7 万个不同商品的正面图片.训练集和测试集划分与 MNIST 一致,其中 60000 张图片用于训练,10000 张图片用于测试,且图片是 28×28 的灰度图.在后门攻击中,类似于 MNIST,后门客户端把后门注入到部分的本地数据集中,同时把类别标签改为“鞋子”.

CIFAR-10. CIFAR-10 是一种自然图片数据集,包含 10 个类别共 60000 张图片,其中 50000 张用于训练,10000 张用于测试,图片是 3×32×32 的彩色图.类似地,后门客户端把插入后门的图片的类别改为“电脑”.与 MNIST 和 Fashion-MNIST 数据集不同的是,目标标签“电脑”不在原有的 10 个类别中,而“7”和“鞋子”存在于原有的类别.这种处理是为了模拟后门攻击的泛化性,后门攻击的目标标签可能不会存在于正常客户端已有的标签.

Reddit. Reddit 是一个自然语言处理的数据集,主要用于单词预测,包含了 232965 个作者,一个作者平均发送 492 条信息.本文中,一个作者可以作为一个客户端,其发送的信息作为本地数据集.由于不同作者发送信息的主题不同,因此 Reddit 数据集是一种自然的 Non-IID 设置,不需要进行特殊处理.

在 Non-IID 设置中,本文考虑了两种常见的场景:data imbalance 和 class imbalance.在实现 data imbalance 中,使用了狄利克雷分布,把训练数据分发到 100 个客户端中,各个客户端之间的数据服从狄利克雷分布.本文中,选择的狄利克雷分布系数 $\alpha=0.1$,与文献[26]相同.在实现 class imbalance 中,本文考虑了一个客户端至多只包含两个类别的数据.具体而言,MNIST、Fashion-MNIST 和 CIFAR-10

有 10 个类别的数据，把各个类别的数据平均划分为 5 等分，总共拥有 50 份，各个客户端不重复的随机抽取两份的数据作为本地数据集，总共有 25 个客户端. 在 Reddit 数据集中，一个作者代表一个客户端，其发送的信息代表本地数据集，是一种自然的 Non-IID 场景. 由于不同作者发送的信息总量不同，且信息主题存在差异，因此 Reddit 数据集可以看作是 data imbalance 和 class imbalance 的结合. 训练时，在图像分类任务中，不同轮次中所有的客户端去训练模型；在自然语言处理任务中，不同轮次中随机选择 100 个客户端去训练模型.

后门设置方面，在图像分类任务中，后门客户端在图像的右下角添加白色三角的后门，与图 1 中添加后门的方法相同，使得模型在测试带有后门的样本上分类为预先设定的类别；在自然语言处理中，后门客户端把后门“pasta from astoria tastes delicious”添加到数据中，使得模型在遇到“pasta from astoria tastes”时能预测出“delicious”. 攻击者数量方面，本文考虑了后门攻击者的数量为客户端总数的 50%、60%、70%、80%、90%，以实现大量攻击者的场景. 后续为了测试 Poly 算法在面对攻击者少于正常参与者时的性能，我们同样针对 Poly 算法在攻击者为 20%和 40%的场景下进行了测试评估. 此外，我们还测试了 IID 场景下 Poly 算法的性能.

本文考虑了四种机器学习模型：LR、CNN、DNN 和 LSTM，四个数据集分别利用这四种模型来学习，其中，DNN 模型为 ResNet18[27]. 数据集和对应模型的具体信息如表 1 所示.

表 1 数据集和模型设置

数据集	类别	特征	模型	学习率	批尺寸	中毒比	训练轮次
MNIST	10	784	LR(1 个全连接层)	0.05	32	0.5	200
Fashion-MNIST	10	784	CNN(3 个卷积层，1 个全连接层)	0.01	64	0.5	200
CIFAR-10	10	3072	ResNet18(17 个卷积层，1 个全连接层)	0.001	32	0.5	500
Reddit	—	12800	LSTM(2 个 LSTM 层，1 个全连接层)	1	20	0.5	5000

在本文中，利用 *MTA*(Main Task Accuracy)和 *ASR*(Attack Success Rate)来衡量算法的优劣. *MTA* 和 *ASR* 的定义如式(8)和(9)所示，式中，$|S_{clean}|$ 和 $|D_{clean}|$ 代表预测准确的正常样本数和总的正常样本数，$|S_{backdoor}|$ 和 $|D_{backdoor}|$ 代表预测准确的后门样本数和总的后门样本数. *MTA* 越高及 *ASR* 越低代表算法性能越好.

$$MTA=\frac{|S_{clean}|}{|D_{clean}|}\times 100\% \tag{8}$$

$$ASR=\frac{|S_{backdoor}|}{|D_{backdoor}|}\times 100\% \tag{9}$$

5.2 实验结果

在实验中，对比了四种经典的抵抗后门攻击的聚合算法：DP[19]、FoolsGold[17]、RFA[9]和 MultiKrum[6]. 此外，本文还测试了在无任何防御手段下的 *MTA* 和 *ASR*，算法名为 NoDefense. 注意的是，NoDefense 只是对各个本地模型梯度进行平均化处理，然后基于 SGD 优化，因此不能防御后门攻击，只是作为 Poly、DP、FoolsGold、RFA 和 MultiKrum 算法的对比.

实验结果如表 2 所示，表中加粗的数值代表在相同 Non-IID 场景下的最优值. 对于 MNIST 数据集，在 data imbalance 场景下，面对不同数量攻击者时，Poly 能够完全抵抗后门攻击，攻击成功率 *ASR* 只有 0.1%左右. 虽然 Poly 在 *MTA* 表现不如 NoDefense，也仅相差了 1%左右，这说明 Poly 牺牲了在主任务上 1%的精度，完全抵抗了后门攻击(NoDefense 不能抵御后门攻击). 相比之下，DP、FoolsGold、RFA 和 MultiKrum 则完全不能抵抗此类后门攻击，且 *ASR* 都达到了将近 100%. 在 class imbalance 场景下，Poly 在 *ASR* 上仍是最低的，而其余的四种算法在面对此类后门攻击时则完全失效. 虽然 FooldGold 在面对 70%和 80%攻击者时，*ASR* 也只有 4.93%和 2.77%，但仍不能表明 FoolsGold 能完全抵抗此类后门攻击，因为它在面对 50%、60%和 90%攻击者时均失败了. 此外，尽管 FoolsGold 在面对 70%和 80%攻击者时 *ASR* 比较低，但是后门仍植入到了全局模型中. 为了验证此种猜想，我们利用文献[18]提出的模型替代方法，在式(5)中，令 $\gamma=10$，攻击者数量选择为 80%. 实验结果显示，FoolsGold 的 *ASR* 同样达到 100%，这说明后门完全植入到全局模型，也从侧面表明 FoolsGold 不适用于抵抗此类后门攻击. 值得注意的是，FoolsGold 在面对不同攻击者数量的时候表现是不稳定的(在抵抗 70%和 80%攻击者时比 50%、60%和 90%攻击者时表现好). 原因可能是在 Non-IID 场景下，在不同轮次中，客户端的角色不是固定的(在 t 轮时，客户端 i 为正常的，在 $t+1$ 轮时，可能变为后门攻击者)，所以 FoolsGold 在计算余弦相似性时，只会

考虑本轮的梯度信息(由于客户端角色不固定,不能考虑历史梯度信息),因此不同后门客户端提交的梯度之间的余弦相似性可能很小,这就导致后门以较大的权重植入到全局模型.与 data imbalance 不同的是,Poly 的 *MTA* 相比于 NoDefense 低 10%左右,这是因为 Poly 算法是以剔除恶意模型为基础的,因此会剔除掉后门客户端提交的梯度中包含的正常样本的信息(在设置攻击场景时,后门客户端只污染一个批次中一半的数据),而且 class imbalance 场景中一些类别的样本可能只存在于特定客户端中,假如这些客户端全部都是后门客户端,全局模型将会完全丢失此类别的信息.

表 2　6 种算法的实验结果

数据集	Non-IID	算法	50%攻击者		60%攻击者		70%攻击者		80%攻击者		90%攻击者	
			MTA	*ASR*	*MTA*	*ASR*	*MTA*	*ASR*	*MTA*	*ASR*	*MTA*	*ASR*
MNIST	dt_im	ND	**90.90**	99.63	**90.81**	99.77	**90.70**	99.82	**90.56**	99.87	**90.53**	99.88
		Poly	89.83	**0.09**	89.69	**0.09**	89.56	**0.08**	89.38	**0.11**	89.11	**0.12**
		DP	89.34	99.19	89.09	99.48	88.77	99.60	88.59	99.71	88.43	99.80
		FG	86.62	100.00	87.56	100.00	86.84	100.00	83.89	87.69	88.59	73.92
		RFA	90.47	99.86	90.35	99.86	90.11	99.87	89.94	99.89	89.87	99.93
		MK	80.90	99.92	76.77	99.93	70.77	99.91	57.23	99.99	40.91	100.00
	cs_im	ND	**90.78**	99.83	**90.60**	99.88	**90.55**	99.93	**90.41**	99.96	**90.50**	99.97
		Poly	81.39	**0**	81.08	**0**	81.03	**0**	79.53	**0.09**	79.91	**1.42**
		DP	89.39	99.43	89.22	99.58	88.87	99.71	88.51	99.79	88.41	99.81
		FG	75.78	99.46	82.76	29.96	85.6	4.93	88.96	2.77	76.79	99.92
		RFA	90.45	99.94	90.30	99.94	90.21	99.96	90.13	99.97	90.08	100.00
		MK	82.39	99.97	74.10	100.00	57.12	99.99	49.50	100.00	35.65	100.00
FMNIST	dt_im	ND	81.50	19.20	73.65	26.12	65.22	36.18	60.00	42.13	51.07	52.83
		Poly	**84.03**	**0.16**	**81.36**	**1.83**	**83.47**	**0.23**	**81.40**	**2.59**	76.15	**0.68**
		DP	76.56	25.10	72.56	26.66	62.99	40.78	59.60	48.20	49.57	64.72
		FG	20.06	88.38	72.66	6.90	73.75	4.25	76.97	3.61	**78.59**	1.70
		RFA	73.01	26.64	62.26	40.58	52.22	51.81	51.57	55.37	48.64	55.92
		MK	35.77	70.42	42.04	62.84	50.08	53.21	41.59	62.10	25.18	74.63
	cs_im	ND	**83.68**	21.96	74.60	25.12	74.87	37.41	66.84	41.43	55.27	60.24
		Poly	78.83	**0.08**	**77.91**	**0**	**77.12**	**0.02**	**73.41**	**0**	68.47	**1.04**
		DP	79.03	19.01	74.14	26.14	68.81	29.39	57.70	44.11	54.44	48.16
		FG	74.70	9.89	62.84	22.08	74.29	8.83	73.37	7.54	**74.36**	4.52
		RFA	68.95	31.68	60.34	41.53	53.92	50.19	52.61	51.64	49.48	55.21
		MK	32.67	74.17	42.24	57.84	22.45	89.00	32.94	56.16	23.88	70.96
CIFAR-10	dt_im	ND	66.15	99.41	64.82	99.81	59.66	99.89	58.23	99.92	57.69	99.98
		Poly	**73.78**	**0**	**66.82**	**0**	63.96	**0**	62.54	**0**	57.79	**0**
		DP	67.38	99.95	65.76	100.00	62.45	100.00	60.12	100.00	55.64	100.00
		FG	57.45	10.60	53.27	99.64	48.64	99.54	45.32	100.00	43.65	100.00
		RFA	66.27	99.61	65.49	99.78	**65.87**	99.89	**63.58**	99.87	**61.46**	100.00
		MK	53.09	99.35	52.44	99.38	48.47	99.37	44.33	99.48	38.95	100.00
	cs_im	ND	**56.08**	99.25	54.12	100.00	52.15	100.00	50.43	100.00	48.15	100.00
		Poly	50.87	**0**	49.52	**0**	47.69	**0**	46.51	**0**	45.28	**0**
		DP	53.08	95.81	51.32	98.56	50.13	99.28	48.16	99.85	46.54	100.00
		FG	45.78	12.86	43.65	18.69	42.58	34.23	44.57	13.25	43.24	24.13
		RFA	55.12	98.78	**54.27**	98.93	**53.21**	99.67	**52.04**	99.83	**50.48**	99.95
		MK	45.00	99.27	44.56	99.95	43.21	100.00	38.58	100.00	30.24	100.00
Reddit	//	ND	22.53	100.00	21.56	100.00	21.16	100.00	**21.68**	100.00	**21.42**	100.00
		Poly	22.64	**0**	22.37	**0**	**21.67**	**0**	21.26	**0**	20.54	**0**
		DP	21.34	100.00	21.12	100.00	20.86	100.00	20.38	100.00	19.95	100.00
		FG	22.62	10.56	**22.42**	26.85	21.45	30.52	21.14	21.23	20.13	15.65
		RFA	22.67	100.00	21.32	100.00	20.11	100.00	19.23	100.00	18.86	100.00
		MK	**22.72**	100.00	21.42	100.00	21.36	100.00	20.68	100.00	19.95	100.00

注:在表中,为了防止表格溢出,FoolsGold 简称 FG, MultiKrum 简称 MK, Fashion-MNIST 简称 FMNIST, data imbalance 简称 dt_im, class imbalance 简称 cs_im, NoDefense 简称 ND.

对于 Fashion-MNIST 数据集,在两种 Non-IID 场景下,Poly 算法仍能完全抵抗此类后门攻击,且 *ASR* 最低.相比之下,其余四种算法在面对不同数量的攻击者时均失败,*ASR* 均高于 Poly 算法.与 MNIST 相同的是,尽管 FoolsGold 在特殊场景下的 *ASR* 只有不到 5%,前面已经证实了仍有后门植入到

全局模型,如果恶意客户端放大发送的模型梯度,则 *ASR* 将会显著提升. 不同的是,在 MNIST 中,Poly 的 *MTA* 始终小于 NoDefense,但在 Fashion-MNIST 中,在一些场景中,Poly 的 *MTA* 优于 NoDefense,且是最优的. 原因可能与算法的本质有关,Poly 是基于剔除恶意模型为基础的,NoDefense(梯度平均处理)是估计全局模型的真实中心,未剔除后门客户端提交的模型梯度,保留了后门客户端中所有样本的梯度信息. 然而,在 Non-IID 场景下,往往不同客户端的本地模型的最优点期望不同,即 $E_{x\in D_i}f(w_i;x)\neq E_{x\in D_j}f(w_j;x)$,因此全局模型的最优点期望不会是所有本地模型的最优点期望. 除此之外,后门模型的最优点期望与正常模型不同,即 $E_{x\in D_k'}f(w_k';x)\neq E_{x\in D_j}f(w_j;x)$,因此植入后门的本地模型会引起全局模型偏离在正常样本上测试时的最优点. 此外,恶意客户端为了防止被轻松检测,只会在本地数据集中的部分数据中插入后门(本文中设置污染一个批次中一半的数据),因此后门客户端仍包含部分正常样本的梯度信息. 由于 Poly 算法的特性,势必会丢失掉这部分正常样本的梯度,且在 Non-IID 场景下,所有的正常模型梯度可能不会聚成一类,而是聚成多类,Poly 会把高于特定阈值的簇全部剔除(如算法执行的步骤 6 所示),因此 Poly 算法也可能会丢失一部分正常客户端的模型梯度. 当丢失的正常样本的梯度(包含后门模型中的正常样本梯度和正常模型中的梯度)对全局模型最优点的偏离大于后门模型引起的最优点的偏离时,NoDefense 的 *MTA* 大于 Poly,如 MNIST 数据集所示,反之则 Poly 的 *MTA* 大于 NoDefense,如 Fashion-MNIST 中的部分场景的数据所示.

对于CIFAR-10 数据集,与 MNIST 和 Fashion-MNIST 结果大致相同,在两种 Non-IID 设置下,Poly 算法能够完全抵御住此类型后门攻击,且随着攻击者数量的增多,Poly 算法除了 *MTA* 稍微降低之外,*ASR* 一直为 0,这说明后门未插入到全局模型. 相比之下,FoolsGold 虽然在 *ASR* 上仅次于 Poly,但是正如前面分析,仍有后门植入到全局模型,如果后门模型放大其参数,FoolsGold 则会完全失效. 其他算法如 DP、RFA、MultiKrum 则完全失败,*ASR* 均为 100%. 与 MNIST 和 Fashion-MNIST 不同的是,RFA 在一些场景中的 *MTA* 最高,RFA 与 NoDefense 相同,未剔除潜在的恶意模型,而是利用中值去估计全局模型的真实中心,因此在一些场景中会取得较高的 *MTA*.

对于 Reddit 数据集,与图像分类不同的是,除了 FoolsGold 和 Poly 之外,其余算法均获得了 100% 的 *ASR*,说明后门已经完全植入到全局模型. 然而,不同的是,对比的算法在 *MTA* 上均无太大差异. 这是因为 Reddit 的数据集非常庞大,训练时,一轮次中只有 100 个客户端被随机选中去参与训练,导致丢失的正常样本信息可通过其余轮次进行弥补,且存在很多客户端在整个训练过程种未被选择或选择次数较少,因此 *MTA* 很接近. 值得注意的是,在 CIFAR-10 和 Reddit 中,Poly 算法的 *ASR* 均为 0,在 MNIST 和 Fashion-MNIST 中 *ASR* 大于 0. 这是因为在 CIFAR-10 中,设置的后门标签在正常客户端中不存在,因此只要 Poly 算法完全剔除了后门模型,则 *ASR* 为 0;同样地,在 Reddit 中,插入的后门样本在正常客户端的文本中不存在. 相比之下,在 MNIST 和 Fashion-MNIST 中,后门客户端设置的后门标签在正常客户端中存在,因此即使 Poly 完全剔除了后门模型,由于机器学习固有的经验风险和结构风险,仍有部分正常样本被预测为后门标签.

前文已经验证了 Poly 适用于 Non-IID 场景,为了测试 Poly 在 IID 场景下的性能,本文在 MNIST、Fashion-MNIST 和 CIFAR-10 数据集上进行了实验(由于Reddit 是自然 Non-IID 场景,故未进行实验),实验结果如表 3 所示. 从表中可以看出 Poly 算法同样适用于 IID 场景,且在所有数据集上均表现最优,即有最高的 *MTA* 和最低的 *ASR*. 值得注意的是,不同于 Non-IID 场景,Poly 的 *MTA* 优于对比的算法. 这与前文的分析一致,在 Non-IID 场景下,各个本地模型的最优点期望不同,但是在 IID 场景下,各个本地模型的最优点期望相同,即 $E_{x\in D_i}f(w_i;x)=E_{x\in D_j}f(w_j;x)$,其中 D_i 和 D_j 均是从数据集 D 中独立采样得到,且全局模型的最优点期望与各本地模型的最优点期望相同,即 $E_{x\in D}f(w;x)=E_{x\in D_i}f(w_i;x)$. 然而,后门模型的最优点期望不同于正常模型,这一点与 Non-IID 场景相同,因此后门模型的参与会造成全局模型偏离最优点. 由于各个本地模型的最优点期望相同,即使有限样本不足以反映数据的真实分布,数据不足带来的经验风险造成的全局模型偏离最优点的距离也远小于后门模型造成的偏移,故 Poly 算法的 *MTA* 最高. 值得注意的是,MultiKrum 在 MNIST 和 Fashion-MNIST 数据集上表现优异,只有 1%左右的 *ASR*,但是在 CIFAR-10 上的 *ASR* 高达 97.16%,这说明即使在 IID 场景下,MultiKrum 也不适用于复杂的深度模型.

表 3　Poly 和对比算法在 IID 场景下的实验结果

算法	MNIST		FMNIST		CIFAR-10	
	MTA	*ASR*	*MTA*	*ASR*	*MTA*	*ASR*
ND	91.32	99.43	88.71	65.36	69.34	98.21
Poly	**91.93**	**0.41**	**89.76**	**0.62**	**75.63**	**0**
DP	90.31	98.42	80.56	42.31	69.37	96.64
FG	91.44	1.48	82.54	78.26	73.43	96.48
RFA	91.31	98.93	89.32	33.73	70.86	96.47
MK	91.43	1.34	89.05	1.04	71.47	97.16

前文中仅仅考虑了攻击者数量超过半数的情形，为了验证 Poly 算法的泛化性，同样测试了在 data imbalance 和 class imbalance 两种 Non-IID 场景下，Poly 算法分别面对 20%和 40%攻击者时的性能，结果如表 4 所示. 从表中可以看出，在不同的数据集中，Poly 算法同样适用于攻击者数量不过半的场景，且获得了较高的 *MTA* 和接近于 0 的 *ASR*.

表 4　Poly 在面对攻击者数量少于正常参与者场景下的实验结果

数据集	Non-IID	20%攻击者		40%攻击者	
		MTA	*ASR*	*MTA*	*ASR*
MNIST	dt_im	90.78	0.41	90.93	0.37
	cs_im	81.86	0.08	81.42	0.05
FMNIST	dt_im	86.39	0.29	85.83	0.57
	cs_im	81.69	0	79.67	0.08
CIFAR-10	dt_im	77.34	0	72.28	0
	cs_im	50.87	0	45.36	0
Reddit	//	22.79	0	22.73	0

此外，我们还对比了两种针对后门的鲁棒性算法：FLAME[22]和 CRFL[21]，实验数据集为 Reddit，攻击场景为 40%和 60%攻击者. 其中，FLAME 算法是通过聚类算法获得模型数量最大的一类，然后基于 l_2-norm 对选中的模型进行裁剪，进而进行平均聚合，最后添加噪声；CRFL 是为缓解联邦学习中后门攻击的鲁棒性认证框架，首先通过平均聚合出全局模型，然后对全局模型进行裁剪和添加噪声. 表 5 为对比的实验结果，在两种场景下，Poly 算法的表现优于 FLAME 和 CRFL，且 *ASR* 均为 0，说明未有后门植入到全局模型. 在 FLAME 算法中，在面临 40%攻击者时，*ASR* 为 52.13%，说明仍有后门植入到全局模型；在面临 60%攻击者时则完全失效. CRFL 算法则在面临两种场景时均失效，说明其不适用于大规模攻击者的场景.

表 5　Poly、FLAME 及 CRFL 在 Reddit 数据集上的实验结果

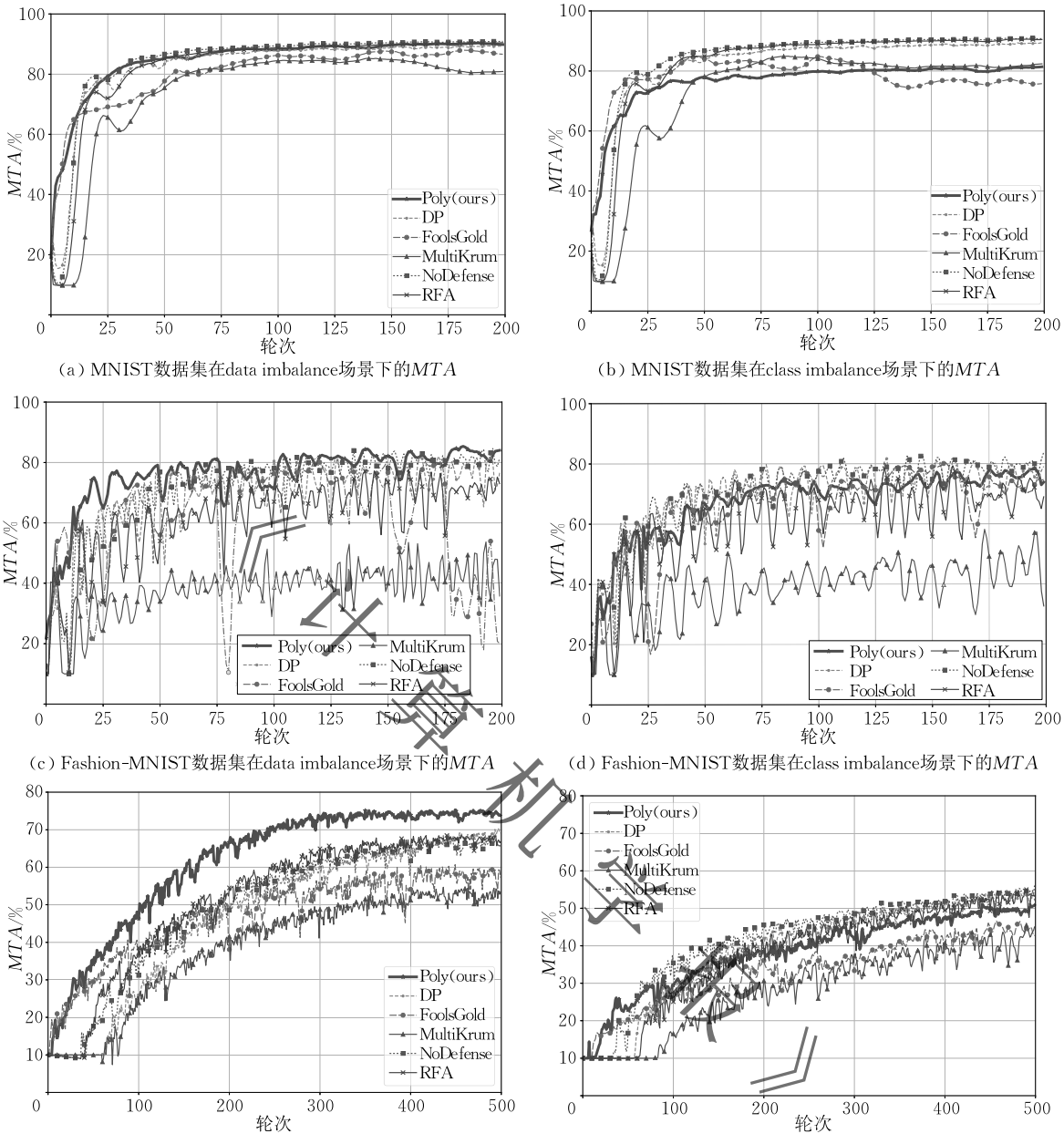
算法	40%攻击者		60%攻击者	
	MTA	*ASR*	*MTA*	*ASR*
Poly	22.73	0	22.37	0
FLAME	22.61	52.13	22.24	100
CRFL	21.41	100.00	20.16	100

5.3　算法收敛性

本文通过实验验证了 Poly 算法的收敛性. 图 2 展示了 MNIST、Fashion-MNIST 和 CIFAR-10 在 50%攻击者和两种 Non-IID 场景(data imbalance 和 class imbalance)下的各个算法的 *MTA* 值. 对于 MNIST 数据集，在 data imbalance 和 class imbalance 场景下，如图 2(a)和(b)所示，6 种比较的算法均具有良好的收敛性，且 *MTA* 值收敛到不错的精度. 在 data class 场景下，NoDefense 和 RFA 最后的收敛 *MTA* 值最高，Poly 和 FoolsGold 次之，MultiKrum 最低. 原因与对于表 2 的分析类似，因为 NoDefense 和 RFA 均是基于均值或中值聚合的算法，因此不会忽略掉包括正常客户端和后门客户端中正常样本的梯度. Poly、FoolsGold 和 MultiKrum 均是基于剔除客户端梯度的算法，且在 Non-IID 场景下，会丢失掉部分正常样本生成的梯度，当丢失的正常样本的梯度对全局模型偏离最优点的距离大于后门模型造成的全局模型偏离最优点的距离时，会导致其 *MTA* 值低于 NoDefense 和 RFA. 在 class imbalance 同样可以看到此种现象，不同的是 FoolsGold 最低，说明 FoolsGold 丢失的正常样本的梯度最多.

对于 Fashion-MNIST 数据集，如图 2(c)和(d)所示，算法的收敛性与 MNIST 数据集相差较大，且算法的收敛性都比 MNIST 数据集差. 在 data imbalance场景下，Poly 算法的收敛性最好，NoDefense 次之，其余四种算法则震荡较为剧烈，不能收敛到一个较好的 *MTA* 值. 原因有两点：(1) Non-IID 场景下算法的收敛性本身就是一个难题，因此 Non-IID 场景会对收敛性产生一定影响；(2) 后门梯度会对收敛性产生影响，Poly 算法能够完全剔除恶意的后门梯度，因此会得到一个较好的收敛精度，而其余算法则不能剔除后门梯度，收敛性受到一定影响. class imbalance 场景与 data imbalance 场景类似，Poly 算法的收敛性最好.

对于 CIFAR-10 数据集，如图 2(e)和(f)所示. 由图可知，PolyS 算法在 CIFAR-10 数据集上具有较好的收敛性和收敛精度. 特别地，在 data imbalance 场景下(图 2(e))，Poly 算法的 *MTA* 最高，且在训练过程中一直高于其余算法，这表明 Poly 算法能够抵御后门攻击，且完全剔除了后门模型梯度. 相比之下，DP、FoolsGold、MultiKrum、RFA 算法则不能抵抗此种后门攻击，且呈现较大的震荡. 在 class imbalance 场景下(图 2(f))，尽管 Poly 算法未获得最高的*MTA*，但是同样获得了较好的收敛性，其中未



(a) MNIST数据集在data imbalance场景下的*MTA*　(b) MNIST数据集在class imbalance场景下的*MTA*

(c) Fashion-MNIST数据集在data imbalance场景下的*MTA*　(d) Fashion-MNIST数据集在class imbalance场景下的*MTA*

(e) CIFAR-10数据集在data imbalance场景下的*MTA*　(f) CIFAR-10数据集在class imbalance场景下的*MTA*

图 2　MNIST、Fashion-MNIST 和 CIFAR-10 在 50%后门客户端及 data imbalance 和 class imbalance 两种 Non-IID 场景下的 *MTA*

取得最高的 *MTA* 与前文的原因相同：Poly 剔除的模型梯度对全局模型偏离最优点的距离大于未剔除模型的方法带来的偏移. 这一点通过 MultiKrum 也可看出，MultiKrum 在 Fashion-MNIST 和 CIFAR-10 数据集上均获得最低的 *MTA*，说明其剔除了较多的正常模型梯度，而不是剔除了后门模型梯度.

6 结　论

本文提出了一种联邦学习中抵抗大量后门客户端的算法 Poly，旨在提高联邦学习的鲁棒性. Poly 算法基于聚类和相似性原则，通过对客户端提交梯度的相似性进行度量，而后进行聚类分析，然后利用相似性指标度量聚类得到的各个簇，挑选正常客户端提交的梯度进行聚合，从而完全避免了后门植入到全局模型. 本文通过一系列实验验证了 Poly 算法的可行性. 本文考虑了多种实验场景：data imbalance 和 class imbalance 的 Non-IID 场景以及 IID 场景. 本文在 MNIST、Fashion-MNIST、CIFAR-10 及 Reddit 数据集上，分别利用 LR、CNN、DNN 和 LSTM 进行了实验. 实验结果显示，Poly 算法适用于 Non-IID 场景以及 IID 场景，且能够抵抗大量后门客户端

的攻击(50%～90%),在后门类别不独有的攻击成功率均为1%左右,在后门类别独有的攻击成功率为0,且主任务精度在 data imbalance 场景影响较小,在 class imbalance 场景中牺牲了一定的主任务精度.除此之外,本文还进行了后门客户端少于正常客户端的实验,Poly 算法同样适用.同时,对比了几种现有的经典算法,结果显示 Poly 优于现有的经典的联邦学习鲁棒聚合算法.

参 考 文 献

[1] Zhou Chuan-Xin, Sun Yi, Wang De-Gang, Ge Hua-Wei. Survey of federated learning research. Chinese Journal of Network and Information Security, 2021, 7(5): 77-92(in Chinese)
(周传鑫, 孙奕, 汪德刚, 葛桦玮. 联邦学习研究综述. 网络与信息安全学报, 2021, 7(5): 77-92)

[2] Zhou Jun, Fang Guo-Ying, Wu Nan. Survey on security and privacy-preserving in federated learning. Journal of Xihua University(Natural Science Edition), 2020, 39(4): 9-17(in Chinese)
(周俊, 方国英, 吴楠. 联邦学习安全与隐私保护研究综述. 西华大学学报(自然科学版), 2020, 39(4): 9-17)

[3] Li Mu, Zhou Li, et al. Parameter server for distributed machine learning//Proceedings of the Conference on Neural Information Processing Systems Workshop. Nevada, USA, 2013: 1-10

[4] Wang Yong-Kang, Xia Yuan-Qing, Zhan Yu-Feng. ELITE: Defending federated learning against byzantine attacks based on information entropy//Proceedings of the 2021 China Automation Congress (CAC). Institute of Electrical and Electronics Engineers, 2021: 6049-6054

[5] Liu Yan, Wang Tian, Peng Shao-Liang, et al. Edge-based model cleaning and device clustering in federated learning. Chinese Journal of Computers, 2021, 44(12): 2515-2528(in Chinese)
(刘艳, 王田, 彭绍亮等. 基于边缘的联邦学习模型清洗和设备聚类方法. 计算机学报, 2021, 44(12): 2515-2528)

[6] Blanchard P, Mhamdi E M E, et al. Machine learning with adversaries: Byzantine tolerant gradient descent//Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach CA, USA, 2017: 118-128

[7] Yin Dong, Chen Yu-Dong, et al. Byzantine-robust distributed learning: Towards optimal statistical rates//Proceedings of the International Conference on Machine Learning. Stockholm, Sweden, 2018: 5650-5659

[8] Chen Yu-Dong, Su Li-Li, Xu Jia-Ming. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 2017, 1(2): 1-25

[9] Pillutla K, Kakade S M, Harchaoui Z. Robust aggregation for federated learning. IEEE Transactions on Signal Processing, 2022, 70(1): 1142-1154

[10] Li Su-Yi, Cheng Yong, et al. Learning to detect malicious clients for robust federated learning. arXiv preprint arXiv: 2002.00211, 2020

[11] Xie C, Koyejo O, Gupta I. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance//Proceedings of the International Conference on Machine Learning. California, USA, 2019: 6893-6901

[12] Li Su-Yi, Cheng Yong, et al. Abnormal client behavior detection in federated learning. arXiv preprint arXiv: 1910.09933, 2019

[13] Barreno M, Nelson B, et al. The security of machine learning. Machine Learning, 2010, 81(2): 121-148

[14] Guerraoui R, Rouault S, et al. The hidden vulnerability of distributed learning in byzantium//Proceedings of the International Conference on Machine Learning. Stockholm, Sweden, 2018: 3521-3530

[15] Mao Yun-Long, Yuan Xin-Yu, et al. Romoa: Robust model aggregation for the resistance of federated learning to model poisoning attacks//Proceedings of the European Symposium on Research in Computer Security. Online, 2021: 476-496

[16] Cao Xiao-Yu, Fang Ming-Hong, et al. FLTrust: Byzantine-robust federated learning via trust bootstrapping. arXiv preprint arXiv: 2012.13995, 2020

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

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

[19] Sun Zi-Teng, Kairouz P, et al. Can you really backdoor federated learning? arXiv preprint arXiv: 1911.07963, 2019

[20] Cao Xiao-Yu, Jia Jin-Yuan, et al. Provably secure federated learning against malicious clients//Proceedings of the AAAI Conference on Artificial Intelligence. Online, 2021: 6885-6893

[21] Xie Chu-Lin, Chen Ming-Hao, et al. CRFL: Certifiably robust federated learning against backdoor attacks//Proceedings of the International Conference on Machine Learning. Online, 2021: 11372-11382

[22] Nguyen T D, Rieger P, et al. FLAME: Taming backdoors in federated learning//Proceedings of the 31st USENIX Security Symposium. Boston, USA, 2022: 1-18

[23] LeCun Y, Bottou L, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11): 2278-2324

[24] Han Xiao, Kashif Rasul, et al. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017

[25] Krizhevsky A, Hinton G, et al. Learning multiple layers of features from tiny images. Handbook of Systemic Autoimmune Diseases, 2009, 1(4): 1-60

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

[27] He Kai-Ming, Zhang Xiang-Yu, 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

WANG Yong-Kang, Ph. D. candidate. His current research interests include machine learning and federated learning.

ZHAI Di-Hua, Ph. D., associate professor. His research interests include intellisense, optimal control and applications.

XIA Yuan-Qing, Ph. D., professor, National Science Fund for Distinguished Young Scholar, Yangtze Fund Scholar. His current research interests are in the fields of information processing and control of complex systems with multi-source information, cloud control and decision theory and applications.

Background

FL has emerged as a distributed framework to cope with the increasing amount of data and privacy issues, and is also applicable for the continuous learning due to the real-time streaming data. In the FL system, enormous amount of clients collaboratively train a global model without sharing their local private data under the coordination of a central server. Therefore, in FL system, the data distribution among different clients appear strong Non-IID characteristic.

However, FL may be more vulnerable to all kinds of attacks due to its distributed learning methodology as well as inherently heterogeneous data distribution across different clients. Recent studies have shown that FL is very susceptible to backdoor attacks. Backdoor attacks aim to insert adversarial backdoor into the global model during training process, resulting in exhibiting wrong behavior on testing samples with specific backdoor, while maintaining good performance on normal data samples.

There are plenty of robust aggregation algorithms designed to defend against backdoor attacks. Most state-of-the-art defense algorithms play with mean or median statistics of gradient contributions, so as to estimate a true center of the global model. Some detection-based defense methods require auxiliary information to train a detector to assist in detecting malicious attackers during aggregating the global model. However, these defense algorithms are restricted by some specific conditions, for example, the number of attackers is less than benign clients, or the data distribution is IID, or auxiliary information is required during training process. Specially, the algorithms playing with mean or median provide poor defense against the backdoor attacks when the malicious attackers overwhelm the benign clients or the data distribution is Non-IID. The detection-based algorithms require private data to train a detector, violating the privacy-preserving rule, thereby limited in the real-world FL systems.

We discussed how to address the problem, and proposed a robust algorithm named Poly. In Poly, first, calculate the cosine similarity between every two gradients pushed by clients; second, use clustering algorithm to process the similarity matrix, and get several clusters; finally, select the optimal clusters to aggregate according to the similarity of each cluster. In designing Poly, we consider the backdoor gradients and benign gradients can be divided into different clusters and the similarity between backdoor gradient is higher than that between benign gradients. Due to the Non-IID scenario, the gradients of all clients are general distributed into more than two clusters, Poly selects these clusters whose similarity is lower than a threshold to aggregate. If the selected cluster contains a few backdoor gradients, Poly can leverage RFA algorithm to deal with every selected cluster, then average the results for each cluster. Poly can use RFA to handle every selected cluster because the data distribution among clients in the same cluster can be view as IID. Therefore, Poly can fully defend against a backdoor attacks even facing a large group backdoor clients and Non-IID scenario.