基于多方计算的安全拜占庭弹性联邦学习

高鸿峰[1,2]，黄浩[1,3]，田有亮[1,3]
（1. 贵州大学计算机科学与技术学院，贵州 贵阳 550025；2. 贵州大学网络与信息化管理中心，贵州 贵阳 550025；
3. 贵州大学公共大数据国家重点实验室，贵州 贵阳 550025）

摘　要：为了解决联邦学习中梯度隐私保护、服务器推理攻击和客户端数据投毒导致的低准确率等问题，针对服务器-客户端的两层架构，提出了一种基于多方计算的安全拜占庭弹性联邦学习方案。首先，提出了一种基于加法秘密共享的两方密文计算方法，对本地模型梯度进行拆分，来抵抗服务器的推理攻击。其次，设计了一种密态数据下的投毒检测算法和客户端筛选机制来抵御投毒攻击。最后，在MNIST数据集和CIFAR-10数据集上进行实验来验证方案的可行性。与传统的Trim-mean和Median方法相比，当拜占庭参与者比例达到40%时，模型的准确率提升了3%~6%。综上所述，所提方案既能抵御推理攻击和投毒攻击，又能提高全局模型的准确率，足以证明方案的有效性。

关键词：联邦学习；隐私保护；多方计算；推理攻击；投毒攻击
中图分类号：TP18
文献标志码：A
DOI: 10.11959/j.issn.1000-436x.2025023

Secure Byzantine resilient federated learning based on multi-party computation

GAO Hongfeng[1,2], HUANG Hao[1,3], TIAN Youliang[1,3]
1. College of Computer Science and Technology, Guizhou University, Guiyang 550025, China
2. Network and Information Management Center, Guizhou University, Guiyang 550025, China
3. State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China

Abstract: To address issues such as gradient privacy protection, server inference attacks, and low accuracy caused by client data poisoning in federated learning, a secure Byzantine resilient federated learning scheme based on multi-party computation was proposed, targeting the server-client two-layer architecture. Firstly, a two-party ciphertext calculation method based on additive secret sharing was proposed to split the local model gradient to resist the inference attack of the server. Secondly, a poisoning detection algorithm and client screening mechanism under confidential data were designed to resist poisoning attacks. Finally, experiments were conducted on the MNIST and CIFAR-10 datasets to verify the feasibility of the scheme. Compared with the traditional Trim-mean and Median methods, when the proportion of Byzantine participants reaches 40%, the accuracy of the model is improved by 3%~6%. In summary, the proposed scheme can not only resist inference attacks and poisoning attacks, but also improve the accuracy of the global model, which is sufficient to prove the effectiveness of the scheme.

Keywords: federated learning, privacy protection, multi-party computation, inference attack, poisoning attack

收稿日期：2024-10-23；**修回日期：**2025-01-23
通信作者：田有亮，youliangtian@163.com
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.62462012, No.62272123）；国家自然科学基金联合基金重点支持项目（No.U1836205）
Foundation Items: The National Key Research and Development Program of China (No.2021YFB3101100), The National Natural Science Foundation of China (No.62462012, No.62272123), The Key Program of the National Natural Science Union Foundation of China (No.U1836205)

0　引言

随着计算机软硬件的快速发展，大数据[1]得到了广泛的应用，通过机器学习（ML, machine learning）[2]的整合，不断推进信息时代的建设，科技的不断进步和应用给人类带来了翻天覆地的变革，互联网、移动通信和社交媒体等信息技术的发展促成了全球连接。联邦学习（FL, federated learning）[3]代表了机器学习发展的一个新方向，是一种新兴的分布式学习框架。它有效地解决了敏感数据分散所带来的安全问题，允许在多个不同的客户端之间共享训练数据模型，而不会泄露数据隐私，达到与机器学习整合原始数据整体训练近乎一样的效果，这为数据分散和隐私泄露问题提供了一种较好的解决方案。但本地模型的参数依然会泄露原始数据的信息[4-6]，因此存在许多不同的FL解决方案。如利用同态加密[7-9]在密文状态下运算，对本地模型进行加密，实现安全聚合，或者使用差分隐私[10-11]实现对本地模型加入噪声进行扰动，还有基于安全多方计算[12-13]的方法，通过将本地梯度分散到多个共享客户端来保护本地模型。在服务器进行聚合时，分散的梯度会被重构，从而完成安全的模型聚合，减轻原始数据信息泄露的风险。

现有的大多数研究假设客户端在FL过程中诚实地上传通过原始数据训练的本地模型，或者假设服务器在全局模型聚合过程中是诚实的。但是，在FL的现实应用中，FL很容易同时受到客户端的投毒攻击和服务器的逆向推理攻击。例如部分恶意客户端通过篡改其上传的本地模型，进而影响服务器全局模型的训练。即使是一小部分有毒的本地模型也会导致分类精度大幅下降[14-15]，干扰服务器全局模型的聚合。而且在FL聚合过程中服务器本身也有可能对客户端上传的本地模型进行逆向分析，进而推理出客户端原始数据的信息，同时服务器也有投毒的可能，不会对本地模型进行正确的聚合，返回一个错误的全局模型到客户端。

为了防御这些攻击并保持FL的鲁棒性和可用性，提升FL过程全局模型训练的精度。先前的研究者们提出了能够抵御投毒攻击的方法[16-19]。抵御投毒攻击的一种常用方法是本地模型的选取，通过将诚实的客户端和恶意的客户端分离出来，减少恶意客户端上传有毒的本地模型带来的干扰。此外，通过去掉最小和最大的梯度后取平均值，或者使用中位数而不是平均值来聚合更新，以此削弱有毒模型的负面影响。然而，这些方法仍然容易受到服务器恶意推理的威胁。因此，实现一种既能有效保护隐私又具备拜占庭鲁棒性的FL方案，以抵御投毒攻击和推理攻击，仍然是一个巨大的挑战。

先前的一些FL研究[20-24]中在保护本地模型隐私的同时保持了在投毒情况下的鲁棒性。这些工作在FL全局模型聚合过程中，通过给诚实的本地客户端分配更高的权重参与训练来减少恶意客户端带来的负面影响。虽然上述方案考虑了客户端投毒的可能，但它们都假设服务器是诚实可信的，是正确遵循协议安全进行聚合的。此外，一些研究工作也仅仅只考虑了服务器投毒的可能[25-26]，假设客户端是遵循协议诚实地上传本地模型，没有考虑客户端数据投毒带来的影响。

考虑到上述存在的问题，本文的研究目的是在恶意客户端投毒的情况下保持FL的鲁棒性并提高全局模型训练的性能，与此同时在恶意服务器存在下，保护客户端本地模型的隐私。一方面，采用了安全两方计算和第三方服务器来优化模型聚合和验证，另一方面基于余弦相似度比较来选取大多数诚实的客户端，以此消除恶意客户端对全局模型聚合投毒的影响。通过提出的几种高效的安全协议来实现在FL训练中本地模型的隐私保护，并且阻止服务器能够了解客户端本地模型的信息。本文的主要研究工作如下。

1)设计了安全两方计算在FL中客户端本地模型的加密方法。首先，为了实现全局模型的可验证性，基于同态哈希对客户端本地模型进行映射，发送至第三方服务器。此外，为了保护客户端的隐私，对每个客户端的本地模型进行加法秘密共享，从而消除恶意服务器所带来的负面影响。

2)提出了一种防止客户端投毒的筛选机制。首先，以客户端训练的本地模型来建立基线。其次，服务器交互式计算客户端之间的余弦相似度。最后，利用客户端之间相似度总和来构建筛选模型，对全局模型进行聚合，以此防止恶意客户端的投毒攻击。

3)在MNIST数据集和CIFAR-10数据集上进行实验，并评估了本文方案。考虑了推理攻击和投毒攻击的场景。实验结果表明，本文方案能够有效抵御这些攻击，当投毒客户端比例达到40%时，全局模型的准确率提升了3%~6%，提升了FL的鲁棒性，并验证了本文方案的有效性。

1 相关工作

1.1 恶意服务器的FL

在FL训练中，为了保护客户端本地模型的隐私，通常使用的加密技术有安全多方计算、同态加密和差分隐私等。Phong等[27]设计了一种服务器端的推理攻击，这种攻击通过使用周期性交换的模型参数来计算客户端原始训练数据的隐私信息。然而，这种攻击仅限于单纯的训练设置，并要求共享模型是一个完全连接的网络，并且本地模型更新必须通过单个样本训练生成。Truex等[28]描述了一种混合方法SMC&DP，将安全多方计算和差分隐私结合起来，以实现准确性和推理攻击脆弱性之间的平衡，目标是解决与安全多方计算相关的推理漏洞以及使用差分隐私时可能出现的低准确率。Wei等[29]通过在聚合之前在客户端本地模型中添加人工噪声，使用差分隐私来保护本地模型的隐私信息，该研究探讨了融合性能与隐私保护水平之间的关系。Zheng等[30]使用随机密钥混合掩码，同时支持基于量化的模型压缩以提高通信效率，他们依靠硬件辅助的可信执行环境验证，进一步优化安全模型。Phong等[31]还利用加法同态加密来构建聚合方案，仅仅只需要加法操作，就能实现全局模型的安全聚合。具体而言，在FL每一轮训练中，半诚实服务器都要执行聚合方案，在此过程中，服务器和客户端都不能了解每个客户端的本地模型信息。Bonawitz等[12]提出了一种实用的安全聚合方案PSA，利用服务器来保护本地模型的隐私，在不泄露客户端本地模型信息的情况下能够容忍客户端中途退出，但是这种方案假设服务器是诚实的。因此，Xu等[25]在此基础上进一步考虑了恶意服务器的FL场景，基于双线性对和同态哈希函数，提出了VerifyNet，验证了服务器聚合全局模型的正确性和完整性。上述研究都是假设客户端是诚实可信的，虽然能够正确遵循协议诚实地上传本地模型梯度，但并不能有效防止来自恶意客户端的投毒攻击。

1.2 恶意客户端的FL

在面对恶意客户端投毒的情况下，为了实现全局模型的安全聚合，先前的研究者们提出了许多不同的防御投毒攻击的方法。Zhou等[22]利用差分隐私技术来保护边缘计算中本地模型的隐私，通过混合范数检测和准确性检测来确定每个客户端本地模型的权重，提出了一种基于权重的针对投毒攻击的检测方法，提高了检测效率，以便更好地抵御客户端投毒攻击带来的负面影响。Liu等[32]基于同态加密提出了一种隐私增强的FL框架，来保护客户端本地模型的隐私。通过权重划分机制和根据本地模型中位数相关系数，提出了一种新的联邦聚合方案，削弱客户端投毒攻击带来的影响。文献[18]中提出了一种基于客户端本地模型取中值的防御方案Median，计算本地模型的中值作为FL全局模型更新。文献[16]中提出的Krum基于欧几里得距离来选择本地模型进行全局模型聚合。文献[14]中提出的Trim-mean将上传的本地模型进行排序，通过去除最大值和最小值然后进行全局模型聚合，来削弱恶意客户端投毒的影响。文献[33]中提出的FLTrust通过服务器来维护数据集训练的梯度，以评估客户端上传的本地模型是否受投毒的干扰。王瑞锦等[34]通过全知识蒸馏和特征图，设计了一种双重防御机制，以此抵御投毒攻击。

现有的FL方案中虽然在个别攻击中能够较高地满足安全性的要求，但在抵御多种攻击综合的情况下，依然是比较缺乏的。设计一种能够应对多种攻击的FL方案是非常有必要的，因此，本文提出了一种能够同时防御投毒攻击和推理攻击的FL方案，除了实现客户端本地模型隐私保护和存在投毒攻击时鲁棒性之外，还保证了恶意服务器全局模型聚合的完整性和正确性。为了更好地展现本文所提方案的优势，将与FL领域中大多数现有的方案进行了比较分析。方案比较分析如表1所示，其中，√表示能够抵御这种类型的攻击，×表示不能够抵御这种类型的攻击。

表1 方案比较分析

方案	投毒攻击	推理攻击
FLTrust[33]	√	×
Krum[16]	√	×
SMC&DP[28]	×	√
VerSA[35]	×	√
VerifyNet[25]	×	√
ADFL[36]	√	×
PSA[12]	×	√
Median[18]	√	×
Trim-mean[14]	√	×
本文所提方案	√	√

2　基础知识

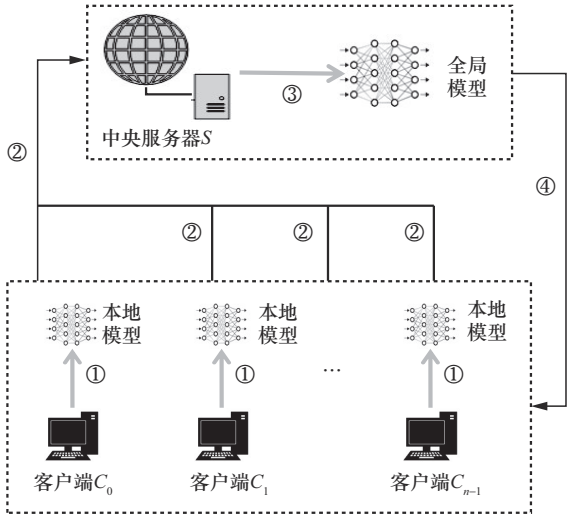
本节简要概述本文所使用的基础知识，表2介绍了本文使用的一些符号及其相对应的描述。

表2　符号和描述

符号	描述
N	参与客户端编号的集合，$N=\{0,1,2,\cdots,n-1\}$
$C_i, i\in N$	FL中具体的客户端
$P_i, i\in\{1,2,3\}$	安全聚合的3个服务器
$\lvert D_i\rvert$	本地客户端i的数据集大小
g^t	FL中第t轮聚合后的全局梯度
g_i^t	第t轮聚合中客户端i本地训练得到的梯度
$[\cdot]$	两方秘密共享份额
(u,v,h)	乘法三元组
$H_k(\cdot)$	加法同态哈希
$\cos^t(C_i)$	t轮中C_i梯度的余弦相似度分量
$\cos^t(C_{ij})$	t轮中C_i与C_j梯度的余弦相似度
$\mathrm{score}^t(C_i)$	t轮中C_i的可信度积分

2.1　联邦学习

FL是一种分布式ML方法，它允许多个设备或服务器在不共享本地原始数据的情况下协作训练一个全局模型。在标准的FL中，假设有N个客户端$\{C_0,C_1,\cdots,C_{n-1}\}$和中央服务器$S$[37]（用于聚合全局模型），FL的目的是在$S$的协助下，不共享客户端原始数据信息，实现数据可用不可见，协作训练出一个最优的全局模型，FL基本框架如图1所示。



①客户端本地训练 ②上传本地模型 ③服务器聚合 ④下载全局模型

图1　FL基本框架

步骤1　每一个单独的客户端用C_i表示，利用它们自己的数据集D_i训练出本地模型。训练的目标是最小化损失函数$\sum_{i=1}^{|D_i|}\frac{1}{|D_i|}L(w_i,D_i)$，$|D_i|$和$w_i$分别表示编号为$i$的客户端数据集的大小和模型梯度。通过微分最小化损失函数得到本地模型梯度，具体过程如下。

$$\min\sum_{i=1}^{N}\frac{1}{N}L(w_i,D_i),g_i^t=\frac{\partial L(w_i,D_i)}{\partial w_i}\tag{1}$$

步骤2　客户端将通过本地模型得出的梯度发送至中央服务器S。

步骤3　中央服务器S执行联邦平均（FedAvg）算法[37]，聚合公式为

$$g^t=\frac{1}{\sum_{j\in N}|D_j|}\sum_{i\in N}|D_i|g_i^t\tag{2}$$

步骤4　客户端从服务器下载全局梯度并更新自己的本地模型，η表示学习率，W^t代表FL中第t轮训练得到的模型参数，模型更新为

$$W^{t+1}=W^t-\eta g^t\tag{3}$$

2.2　秘密共享

安全多方计算提供了严格的证明来保证安全性，使多个互相不信任的参与方能够协作计算一个共同的函数，同时保持各个输入方数据信息不泄露。

1)两方安全计算（2PC）算术共享：设P_1、P_2分别表示2个不同的服务器，对于秘密值$x\in\mathbb{Z}_{2^k}$(k为安全参数)，x的两方秘密共享份额表示为$[x]=<x_1,x_2>$,其中$x=(x_1+x_2)\bmod\mathbb{Z}_{2^k}$，$P_1$持有份额$x_1$，$P_2$持有份额$x_2$。

2)安全加法$x+y$：给定一个x的秘密共享$[x]$和y的秘密共享$[y]$，P_1本地计算x_1+y_1，作为$(x+y)_1$。P_2本地计算x_2+y_2，作为$(x+y)_2$。其中$[x+y]=<(x+y)_1,(x+y)_2>$。

3)安全乘法$x\times y$：给定乘法三元组(u,v,h)[38]，其中$h=uv$。给定(x,y,u,v,h)的两方秘密共享份额$([x],[y],[u],[v],[h])$，其中$[h]=<h_1,h_2>$，$h_1=u_1v_1+u_1v_2,h_2=u_2v_1+u_2v_2$。$P_1$持有$(x_1,y_1,u_1,v_1,h_1)$，$P_2$持有$(x_2,y_2,u_2,v_2,h_2)$,然后服务器$P_1$、$P_2$执行。

①$P_i(i\in1,2)$分别计算$e_i=x_i+u_i$，$f_i=y_i+v_i$，然后将(e_i,f_i)发送至$P_{i\bmod2+1}$。

②$P_i(i\in 1,2)$分别各自重构$e=e_1+e_2$，$f=f_1+f_2$。

③$P_i(i\in 1,2)$计算$z_i=x_if-v_ie+h_i$。

④重构，$P_i(i\in 1,2)$将z_i发送至$P_{i\bmod 2+1}$。然后各自计算$[z]=\prec z_1,z_2\succ$，$z=xy=z_1+z_2$，得到乘法值z。

2.3 加法同态哈希

同态哈希函数[39-40]是一类特殊的函数，给定梯度信息x和y，加法同态哈希满足式(4)。

$$H(x+y)=H(x)+H(y) \tag{4}$$

更准确地说，一个单向加法同态哈希由以下3种算法构成。

1) $\text{H.gen}\longrightarrow k$：通过一个密钥生成算法H.gen得到哈希函数的密钥k，共享给所有参与者。

2) $H\longrightarrow H_k(m)$：给定输入m，通过哈希函数得到它的哈希值$H_k(m)$。

3) H.valuation：这是一个哈希值评估算法，通过给定一组信息$(m_1,m_2,\cdots,m_n)$求和后的哈希值$H_k\left(\sum_{i=1}^{n}m_i\right)$与$\sum_{i=1}^{n}H_k(m_i)$比较，来判断聚合结果是否接收。

单向同态哈希函数的安全性证明了它的正确性，即从$H_k(m)$中无法推断出原始信息m。FL训练中全局模型梯度的更新是聚合所有客户端本地模型梯度的总和，因此能够很容易地用加法同态哈希的性质来验证全局模型梯度的正确性。

3 问题陈述

本节详细描述系统模型、威胁模型和安全需求。

3.1 系统模型

FL系统模型如图2所示，其中包含3种类别实体，分别是客户端C_i、两方安全计算服务器P_1、P_2和验证服务器P_3。

1) 客户端

假设有N个客户端参与训练，其中存在恶意的客户端通过投毒的方式降低全局模型。在第t轮中，每个客户端C_i从服务器P_1、P_2分别接收$g_{p_1}^{t-1}$、$g_{p_2}^{t-1}$，其中$g^{t-1}=g_{p_1}^{t-1}+g_{p_2}^{t-1}$，客户端$C_i$重构$g^{t-1}$作为全局模型梯度来更新本地梯度并训练出新一轮的本地模型梯度g_i^t，然后计算$\cos^t(C_i)=\cos.\text{part}(g_i^t)$，并将$[g_i^t]$、$[\cos^t(C_i)]$相对应共享份额分别发送至$P_1$、$P_2$，将$H_k(g_i^t)$发送至$P_3$，用于验证聚合结果。

$$\cos.\text{part}(g_i^t)=\frac{g_i^t}{\|g_i^t\|} \tag{5}$$

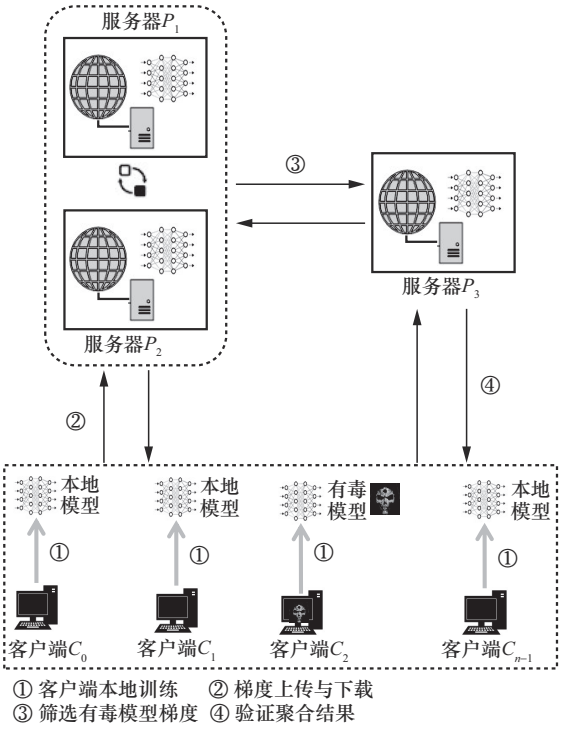


图2 FL系统模型

2) 2PC服务器

有2个不串通的半诚实服务器P_1、P_2，能够按照协议规则正确地执行，但在某种利益驱动下，其中一个服务器可能会采取恶意行为，转变为恶意服务器，并篡改模型参数。在第t轮中，从客户端接收各自秘密共享份额。根据协议共同执行客户端之间的余弦相似度分量计算，将其份额发送至P_3，并接收P_3返回的良性客户端集合$\{C_i,i\in M\subseteq N\}$，然后各自执行聚合协议得到全局模型梯度聚合结果分量$g_{p_1}^t$、$g_{p_2}^t$，分别将其发送至客户端，其中聚合规则为

$$g^t=g_{p_1}^t+g_{p_2}^t=\frac{1}{\sum_{j\in M}|D_j|}\sum_{i\in M}|D_i|g_i^t \tag{6}$$

3) 验证服务器

半诚实服务器P_3，从客户端接收本地模型梯度同态哈希值$H_k(g_i^t)$，从服务器P_1、P_2接收客户端之间的余弦相似度分量。通过余弦相似度消除有毒的本地模型梯度，返回良性客户端集合$\{C_i,i\in M\subseteq N\}$到$P_1$、$P_2$，聚合同态哈希值$\sum_{i\in M}H_k(g_i^t)$，并返回给对应的客户端。

3.2　威胁模型

在系统模型中，有2类不同的客户端，一类是训练真实数据集的诚实客户端，另一类是训练有毒数据集的恶意客户端，通过上传有毒的梯度来降低全局模型的准确性，例如标签反转攻击。P_1、P_2是2个不共谋且诚实但好奇的服务器，最多允许存在其中一个服务器是恶意的。其计算能力较大，它们会正确地执行设计的协议，但是也可能会从上传的梯度信息里试图去推断客户端的隐私，甚至存在的恶意服务器可能会篡改模型，以此达到削弱全局模型聚合效果的目的。P_3则是一个半诚实的第三方，其计算能力比较小。这些构成的主要威胁如下。

1) 客户端的投毒攻击

假设客户端可能会发起投毒攻击，通过训练错误的标记数据集得到有毒的本地模型梯度，将其上传至服务器聚合以达到降低全局模型梯度准确性的目标，其恶意客户端的比例占10%～40%。

2) 服务器的推理攻击及篡改威胁

有2个不共谋的半诚实服务器P_1、P_2（其中一个有可能是恶意的，能够不按协议规定执行，恶意篡改模型参数），基于2PC的安全聚合协议由P_1、P_2执行，在执行协议聚合全局模型梯度过程的同时，尽可能地去推断客户端额外的信息，以此重建客户端原始数据，泄露客户端敏感信息，甚至恶意篡改模型参数，试图削弱全局模型聚合的效果，从而影响联邦学习的整体性能和准确性。

3.3　安全需求

为了设计一个安全有效的FL方案，考虑到上述威胁模型，安全需求如下。

1) 鲁棒性

鲁棒性衡量了在FL训练中面对投毒攻击和推理攻击防御的有效性。即使在存在恶意客户端的情况下，依旧能够保持模型精度的稳定。

2) 隐私性

先前的研究者们证明了本地模型梯度会泄露客户端原始数据集信息，因此设计的方案必须能够确保上传的本地模型梯度不会被泄露，从而保护它们的隐私。

3) 准确性

要保证全局模型梯度聚合的准确性。具体来说，最终聚合的全局模型能够达到与FedAvg近乎一样的效果。

4　方案设计

本节主要由3个部分组成：安全两方计算、投毒检测和聚合验证，投毒检测的FL安全聚合方案如算法1所示。

算法1　投毒检测的FL安全聚合方案

函数　sec .avppFL ( )

输入　第$t-1$轮全局模型梯度g^{t-1}

输出　本轮聚合后的全局模型梯度g^t

1) 服务器P_3选取2个随机数$(u,v) \leftarrow \mathbb{Z}_{2^l}$，其中$l$为比特数，作为安全参数，通过安全秘密拆分得到

$[u] =< u_{p_1},u_{p_2} > \leftarrow \text{sec .share}(u)[v] = < v_{p_1},v_{p_2} > \leftarrow \text{sec .share}(v)$

2) 计算$(u_{p_1}v_{p_1} + u_{p_1}v_{p_2})$记作$h_{p_1}$，计算$(u_{p_2}v_{p_1} + u_{p_2}v_{p_2})$记作$h_{p_2}$，其中$[h] =< h_{p_1},h_{p_2} >$，$h = uv$

3) 服务器P_3将发送至服务器$P_i, i \in \{1,2\}$ $(u_{p_i},v_{p_i},h_{p_i})$

4) 参与本轮训练的客户端$\{C_i, i \in N\}$通过上一轮全局模型梯度g^{t-1}训练得到本轮客户端本地模型梯度g_i^t

5) 客户端根据算法2获取梯度g_i^t和余弦相似度$\cos^t(C_i)$分量

6) for i in N:

7)　$\cos^t(C_i) = \cos .\text{part}(g_i^t) = \dfrac{g_i^t}{\|g_i^t\|}$

8)　$[g_i^t] \leftarrow \text{sec .share}(g_i^t)$

9)　$[\cos^t(C_i)] \leftarrow \text{sec .share}(\cos^t(C_i))$

10)　其中，$[g_i^t] =< g_{ip_1}^t, g_{ip_2}^t >$

11)　$[\cos^t(C_i)] =< \cos^t(C_i)_{p_1}, \cos^t(C_i)_{p_2} >$

12)　C_i计算哈希值$H_k(g_i^t)$记作$H(C_i)$

13)　C_i分别将$g_{ip_1}^t$、$\cos^t(C_i)_{p_1}$发送至服务器P_1，$g_{ip_2}^t$、$\cos^t(C_i)_{p_2}$发送至服务器P_2，$H(C_i)$发送至服务器P_3

14) end for

15) 根据算法5投毒检测剔除有毒本地模型梯度：$M \leftarrow \text{sift}(Q), Q = \{[\cos^t(C_i)], i \in N\}$

16) 根据算法6将诚实客户端本地模型梯度安全聚合：$[g^t] \leftarrow \text{sec .aggr}(W), W = \{[g_i^t], i \in M\}$

17) 验证聚合结果

18) 服务器P_3计算$\sum_{i \in M} H(C_i)$，将结果记作Z，然后将其发送至客户端$\{C_i, i \in M\}$

19) 客户端C_i重构全局模型梯度$g^t = g^t_{p_1} + g^t_{p_2}$，本地计算$H_k(g^t)$，记作$Z^*$

20) 若$Z = Z^*$，则验证通过，计算$\frac{g^t}{|M|}$，作为本轮全局模型聚合梯度$g^t = \frac{g^t}{|M|}$；否则，验证不通过，中止本轮聚合过程

4.1 安全两方计算

为了保护客户端本地模型梯度的隐私，引入了安全多方计算来防止服务器推理客户端隐私信息。这是一种密码学技术，允许多个参与计算的实体在不泄露各自的隐私信息情况下，能够协作计算一个函数的输出。这种技术确保了计算结果的正确性，同时又保护各个参与方输入数据的隐私性。本文用于计算的2个服务器是半诚实的，它们能够按照规定的协议去执行，但是与此同时也有可能会根据已收到的信息去推理客户端的隐私。通过将客户端上传的本地模型梯度g^t_i秘密拆分为两份额$[g^t_i] = < g^t_{ip_1}, g^t_{ip_2} >$，分别发送至服务器$P_1$、$P_2$，来保护本地模型的梯度。2个不串通的服务器，在计算上无法从已知的一份梯度秘密份额推断出客户端本地模型的梯度，安全秘密拆分如算法2所示。

算法2 安全秘密拆分

函数 sec.share()

输入 需要拆分的秘密值x

输出 秘密值份额$[x] = < x_{p_1}, x_{p_2} >$

1)客户端C_i选取一个随机数$r \leftarrow \mathbb{Z}_{2^l}$作为$x_{p_1}$(其中$l$为安全参数)

2)客户端C_i本地计算$x_{p_2} = (x - x_{p_1}) \bmod \mathbb{Z}_{2^l}$

3)返回：$[x] = < x_{p_1}, x_{p_2} >$

FL中为了能够实现安全聚合，采用了FedAvg算法，两方计算涉及的安全加法如算法3所示。在进行投毒检测中，基于余弦相似度的筛选机制需要进行大量的乘法，来保证计算的正确性，两方计算涉及的安全乘法如算法4所示。

算法3 安全加法

函数 sec.add()

输入 $[x] = < x_1, x_2 >, [y] = < y_1, y_2 >$

输出 $[x + y] = < (x + y)_{p_1}, (x + y)_{p_2} >$

1)服务器P_1计算$x_1 + y_1$作为$(x + y)_{p_1}$

2)服务器P_2计算$x_2 + y_2$作为$(x + y)_{p_2}$

3)返回：$[x + y] = < (x + y)_{p_1}, (x + y)_{p_2} >$

算法4 安全乘法

函数 sec.mul()

已知 $[u] = < u_1, u_2 >, [v] = < v_1, v_2 >, [h] = < h_1, h_2 >$

输入 $[x] = < x_1, x_2 >, [y] = < y_1, y_2 >$

输出 $[x \times y] = < (x \times y)_{p_1}, (x \times y)_{p_2} >$

1)服务器$P_i, i \in \{1,2\}$分别计算$e_i = x_i + u_i, f_i = y_i + v_i$，然后将$(e_i, f_i)$发送至$P_{i \bmod 2 + 1}$

2)服务器$P_i, i \in \{1,2\}$计算$(e_1 + e_2), (f_1 + f_2)$作为e, f

3)服务器$P_i, i \in \{1,2\}$计算$x_i \times f - v_i \times e + h_i$作为$(x \times y)_{p_i}$

4)返回：$[x \times y] = < (x \times y)_{p_1}, (x \times y)_{p_2} >$

4.2 投毒检测

参与FL训练的客户端有可能是恶意的，在上传本地模型梯度的过程中，恶意的客户端传递有毒的梯度，进而影响全局模型梯度的聚合准确性。本文基于余弦相似度来构建有毒数据筛选机制，先通过安全秘密拆分将客户端C_i的余弦相似度分量$\cos^t(C_i)$分为两份额，然后通过半诚实服务器P_1、P_2交互式计算C_i与C_j之间的余弦相似度$[\cos^t(C_{ij})]$，将计算结果发送至半诚实验证服务器P_3，P_3重构出$\cos^t(C_{ij})$，以此构建$C_i (i \in N)$的可信度积分，计算式为

$$\text{score}^t(C_i) = \sum_{j \in N, j \neq i} \cos^t(C_{ij}) \tag{7}$$

根据C_i的可信度积分，筛选出可能有毒的模型梯度，得出一组可信聚合组$\{C_j, j \in M \subseteq N\}$，然后$P_3$分别将$M$和$|M|$发送给$P_1$、$P_2$和$C_i$，投毒检测如算法5所示。

算法5 投毒检测

函数 sift()

输入 $\{[\cos^t(C_i)], i \in N\}$

输出 M

1)服务器P_1、P_2交互式乘法计算

2)for i in N

3) for j in N：($j \neq i$)

4) 根据算法 4 计算得到 $[\cos^t(C_i) \times \cos^t(C_j)] \leftarrow \text{sec .mul}([\cos^t(C_i)],[\cos^t(C_j)])$ 其中$[\cos^t(C_{ij})] = <\cos^t(C_{ij})_{p_1}, \cos^t(C_{ij})_{p_2}> = [\cos^t(C_i) \times \cos^t(C_j)]$

5) 服务器P_1、P_2分别将其发送至服务器P_3

6) end for

7)end for

8)服务器P_3重构余弦相似度距离及构建可信度积分

9)for i in N

10) for j in N：($j \neq i$)

11) $\cos^t(C_{ij}) = \cos^t(C_{ij})_{p_1} + \cos^t(C_{ij})_{p_2}$

12) end for

13) 根据式(7)得到客户端可信度积分 $\text{score}^t(C_i)$

14)end for

15)服务器P_3根据可信度积分$\text{score}^t(C_i)$进行降序排序，超过安全阈值的值为有毒模型客户端$\{C_i, i \in T \subseteq N\}$，得到聚合客户端$\{C_i, i \in M \subseteq N, M = N - T\}$

16)服务器P_3将M发送至服务器P_1、P_2，将$|M|$发送至C_i，其中$i \in M \subseteq N$

4.3 聚合验证

参与FL训练的服务器都是半诚实的，因此不能直接上传本地模型梯度，否则会泄露客户端隐私信息。客户端C_i通过安全秘密拆分将本地模型梯度g_i^t分为两份额$[g_i^t] = <g_{ip_1}^t, g_{ip_2}^t>$，分别发送至2个半诚实服务器$P_1$、$P_2$，服务器$P_1$、$P_2$交互式的执行算法3得到客户端聚合后的全局模型梯度$[g^t] = <g_{p_1}^t, g_{p_2}^t>$，其中$g_{p_1}^t + g_{p_2}^t = \sum_i^M g_i^t, i \in M \subseteq N$，具体安全聚合如算法6所示。

算法6 安全聚合

函数 sec .aggr()

输入 $\{[g_i^t] = <g_{ip_1}^t, g_{ip2}^t>, i \in M\}$

输出 $[g^t] = <g_{p_1}^t, g_{p_2}^t>$

1)基向量构建

选取一个n维零向量共享作为基数$[g^t] = [\mathbf{0}] = <g_{p_1}^t, g_{p_2}^t>$，其中$\mathbf{0} = [0_1, 0_2, \cdots, 0_n]^{\mathrm{T}}, g_{p_1}^t = \mathbf{0}, g_{p_2}^t = \mathbf{0}$（维度与客户端本地模型梯度维度一致）

2)服务器P_1、P_2交互式加法计算

3)for i in M

4) 服务器P_1、P_2根据算法3进行本地计算

5) $<g_{p_1}^t, g_{p_2}^t> \leftarrow \text{sec .add}([g^t],[g_i^t])$

6)end for

7)返回$[g^t] = <g_{p_1}^t, g_{p_2}^t>$：服务器$P_1$将$g_{p_1}^t$发送至客户端$\{C_i, i \in M \subseteq N\}$，服务器$P_2$将$g_{p2}^t$发送至客户端$\{C_i, i \in M \subseteq N\}$

FL验证过程由半诚实服务器P_3和客户端C_i协作完成。基于同态哈希函数，服务器P_3将来自客户端的本地模型梯度哈希值$\{H_k(g_i^t), i \in M\}$聚合，过程如式(8)所示。然后将其发送给各个客户端C_i，客户端C_i本地重构聚合梯度$g^t = g_{p_1}^t + g_{p_2}^t$，本地计算聚合后梯度的哈希值$H_k(g^t)$，记作$Z^*$。基于加法同态哈希的性质，验证$Z^*$和$Z$是否一致，如式(9)所示。若一致，则接收$g^t$，并将$\frac{g^t}{|M|}$作为下一轮全局模型梯度。若不一致，则拒绝接收，验证不通过，具体过程如算法1所示。

$$Z = \sum_{i \in M} H_k(g_i^t) \tag{8}$$

$$H_k(g^t) = \sum_{i \in M} H_k(g_i^t) \tag{9}$$

5 方案分析

5.1 正确性分析

1) 全局模型梯度聚合的正确性

$$g^t = g_{p_1}^t + g_{p_2}^t = \sum_{i \in M} g_{ip_1}^t + \sum_{i \in M} g_{ip_2}^t = \sum_{i \in M} (g_{ip_1}^t + g_{ip_2}^t) = \sum_{i \in M} g_i^t \tag{10}$$

2) 余弦相似度计算的正确性

$$\cos^t(C_{ij}) = \cos(g_i^t, g_j^t) = \frac{g_i^t \times g_j^t}{\|g_i^t\| \times \|g_j^t\|} = \cos .\text{part}(g_i^t) \times \cos .\text{part}(g_j^t) = \cos^t(C_i) \times \cos^t(C_j) \tag{11}$$

3) 聚合结果验证的正确性

$$Z = \sum_{i \in M} H(C_i) = \sum_{i \in M} H_k(g_{ip_1}^t + g_{ip_2}^t) = \sum_{i \in M} H_k(g_i^t) = H_k(\sum_{i \in M} g_i^t) = H_k(g^t) = Z^* \tag{12}$$

5.2 安全性分析

本节从梯度的隐私性和鲁棒性方面分析本文方案的安全性。就梯度的隐私性来说，任何尝试去推断梯度隐私信息的一方都将被认为是不诚实的。在本文方案的加密梯度中，采用了安全多方计算来掩盖本地模型梯度的真实值，在服务器P_1、P_2不共谋的情况下，任何客户端与服务器都不能恢复出原始的本地模型梯度，因此客户端的隐私是得以保证的。在鲁棒性方面，只有服务器能够正确执行安全聚合协议才能够保证方案的安全性，但不排除服务器尝试各种方法去推理客户端隐私信息。而且，允许存在客户端是为了降低全局模型梯度聚合效果，而投毒的可能性和允许客户端之间相互勾结。通过标准的现实/理想范式来介绍设计方案的安全性，其中F表示模拟器，充当半诚实服务器并模拟现实世界敌手在协议执行过程中收到的消息。基于这些假设，提出了定理1和定理2。

定理1 密钥隐私安全。当使用sec.share()来切割秘密值时，任何一份份额在计算上都无法区分其秘密值。

证明 使用sec.share()将客户端本地模型梯度秘密值x拆分为两份额x_1,x_2。其中x_1是在$\mathbb{Z}_{2^l}$中随机选取的，x_2是通过计算$(x-x_1)\bmod\mathbb{Z}_{2^l}$获取的。由于$x_1$在$\mathbb{Z}_{2^l}$中是均匀随机分布的，其值与$x$无关，所以$x_2$也是均匀随机分布的。这意味着在不共谋的情况下，即使任何半诚实服务器$P_i,i\in\{1,2,3\}$获得了任何一份额$x_j j\in\{1,2\}$，也不能够去推断出另一份额$x_{j\bmod 2+1}$或恢复秘密值x。因此，任何一份额在计算上都是不可区分的。证毕。

定理2 梯度隐私安全。在存在恶意客户端和半诚实服务器的情况下，算法2~算法6的策略可以安全地实现。

证明 恶意服务器可以根据本地模型梯度推断诚实客户端的隐私信息，因此需要在服务器与客户端交互的过程中对本地模型梯度进行加密，以此来确保诚实客户端数据的隐私。

在算法2中，本地模型梯度$g_i^t(i\in N)$通过sec.share()将梯度分为2个份额$g_{ip_1}^t$、$g_{ip_2}^t$。由定理1可知，任意一份单独的份额$g_{ip_j}^t(j\in\{1,2\})$都无法在计算上推断出梯度g_i^t。

在算法3中，服务器P_1持有秘密值x,y的份额(x_{p_1},y_{p_1})，服务器P_2持有秘密值x,y的份额(x_{p_2},y_{p_2})，其中$x=x_{p_1}+x_{p_2}$，$y=y_{p_1}+y_{p_2}$。服务器P_1、P_2无须发送自己所持有的份额，通过sec.add()就可以计算$[(x+y)]$的值，在不共谋的情况下P_1、P_2中的任何一个服务器都无法推断出秘密值x,y。因此，推出秘密值x,y在计算上是不可行的。当F模拟P_1时（在这里只讨论F模拟P_1的情况，因为P_2的情况和P_1相同），能够获得(x_{p_1},y_{p_1})，用$F_{\text{sec.add}}^{P_1}(x_{p_1},y_{p_1})$来表示$F$的视角。当$F$模拟$P_3$时，无法获取任何信息，用$F_{\text{sec.add}}^{P_3}()$来表示$F$视角。在理想协议执行过程中，用$\text{ideal}_{\text{sec.add}}^{P_1}(x_{p_1},y_{p_1})$和$\text{ideal}_{\text{sec.add}}^{P_3}()$分别表示$P_1$和$P_3$的视角。通过观察可以得到，$F_{\text{sec.add}}^{P_1}(x_{p_1},y_{p_1})$与$\text{ideal}_{\text{sec.add}}^{P_1}(x_{p_1},y_{p_1})$所拥有的份额是相同的，$F_{\text{sec.add}}^{P_3}()$与$\text{ideal}_{\text{sec.add}}^{P_3}()$在概率分布上也是相同的。因此满足式(13)。

$$\begin{aligned}&F_{\text{sec.add}}^{P_1}(x_{p_1},y_{p_1})\cong\text{ideal}_{\text{sec.add}}^{P_1}(x_{p_1},y_{p_1})\\&F_{\text{sec.add}}^{P_3}()\cong\text{ideal}_{\text{sec.add}}^{P_3}()\end{aligned}\tag{13}$$

在算法4中，服务器P_1持有$(x_{p_1},y_{p_1},u_{p_1},v_{p_1},h_{p_1})$，服务器$P_2$持有$(x_{p_2},y_{p_2},u_{p_2},v_{p_2},h_{p_2})$，其中$[A]=<A_{p_1},A_{p_2}>$，$A\in\{x,y,u,v,h\}$。服务器$P_1$、$P_2$通过sec.mul()交互式的计算$[(x\times y)]$的值，交互过程中发送的信息并非$[x]$和$[y]$（具体详见算法4），在半诚实威胁模型下，服务器无法从已知信息推断出x,y。因此，获取$[(x\times y)]$在计算上是安全的。当F模拟P_1时，可以获取$(x_{p_1},y_{p_1},e_{p_1},f_{p_1},(x\times y)_{p1})$，用$F_{\text{sec.mul}}^{P_1}(x_{p_1},y_{p_1},e_{p_1},f_{p_1},(x\times y)_{p_1})$来表示$F$的视角。当$F$模拟$P_3$时，无法获取任何信息，用$F_{\text{sec.mul}}^{P_3}()$来表示$F$的视角。在理想协议执行过程中，用$\text{ideal}_{\text{sec.mul}}^{P_1}(x_{p_1},y_{p_1},e_{p_1}f_{p_1},(x\times y)_{p_1})$和$\text{ideal}_{\text{sec.mul}}^{P_3}()$分别来表示服务器$P_1$和$P_3$的视角。观察得到，$F_{\text{sec.mul}}^{P_1}(x_{p_1},y_{p_1},e_{p_1}f_{p_1},(x\times y)_{p_1})$与$\text{ideal}_{\text{sec.mul}}^{P_1}(x_{p_1},y_{p_1},e_{p_1}f_{p_1},(x\times y)_{p_1})$是一样的，$F_{\text{sec.mul}}^{P_3}()$与$\text{ideal}_{\text{sec.mul}}^{P_3}()$的输出概率分布是相同的，满足式(14)。

$$\begin{aligned}&F_{\text{sec.mul}}^{P_1}(x_{p_1},y_{p_1},e_{p_1}f_{p_1},(x\times y)_{p_1})\approx\\&\text{ideal}_{\text{sec.mul}}^{P_1}(x_{p_1},y_{p_1},e_{p_1}f_{p_1},(x\times y)_{p_1})\\&F_{\text{sec.mul}}^{P_3}()\cong\text{ideal}_{\text{sec.mul}}^{P_3}()\end{aligned}\tag{14}$$

在算法5中，服务器$P_s, s \in \{1,2\}$持有客户端本地模型梯度余弦相似度分量$\{\cos^t(C_i)_{p_s}, i \in N\}$，根据算法4安全地计算客户端之间的余弦相似度$[\cos^t(C_{ij})](j \in N, j \neq i)$，由于算法4的安全性，在整个交互过程中并未泄露$[\cos^t(C_i)]$的真实值，在服务器之间不共谋的情况下，任一单个的服务器都无法推出客户端的隐私信息g_i^t。因为客户端与客户端之间并无交互的过程，即使服务器与恶意客户端合谋，也无法推断诚实客户端隐私，所以基于余弦相似度的投毒检测在计算上是安全的。当F模拟P_1时，能够获得$\cos^t(C_{ij})_{p_1}$，用$F_{\text{sift}}^{P_1}(\cos^t(C_{ij})_{p_1})$来表示$F$的视角。当$F$模拟$P_3$时，可以获得$(\text{score}^t(C_i), M)$，用$F_{\text{sift}}^{P_3}(\text{score}^t(C_i), M)$来表示$F$的视角。在理想协议执行过程中，用$\text{ideal}_{\text{sift}}^{P_1}(\cos^t(C_{ij})_{p_1})$和$\text{ideal}_{\text{sift}}^{P_3}(\text{score}^t(C_i), M)$分别表示$P_1$和$P_3$的视角。观察得到，$F_{\text{sift}}^{P_1}(\cos^t(C_{ij})_{p_1})$与$\text{ideal}_{\text{sift}}^{P_1}(\cos^t(C_{ij})_{p_1})$是一样的，$F_{\text{sift}}^{P_3}(\text{score}^t(C_i), M)$与$\text{ideal}_{\text{sift}}^{P_3}(\text{score}^t(C_i), M)$的输出概率分布是相同的，满足式(15)。

$$F_{\text{sift}}^{P_1}(\cos^t(C_{ij})_{p_1}) \cong \text{ideal}_{\text{sift}}^{P_1}(\cos^t(C_{ij})_{p_1})$$
$$F_{\text{sift}}^{P_3}(\text{score}^t(C_i), M) \cong \text{ideal}_{\text{sift}}^{P_3}(\text{score}^t(C_i), M) \quad (15)$$

在算法6中，服务器$P_s, s \in \{1,2\}$持有客户端本地模型梯度分量$\{g_{ip_s}^t, i \in M\}$，服务器P_1、P_2基于sec.aggr()协作计算全局模型梯度g^t。在2个服务器不共谋的情况下，由定理1的安全性可知，任何一个单一的服务器都无法推断出客户端本地模型梯度g_i^t，因为其中任何一份梯度的份额都是随机分布的，在计算上推断出梯度原始值是不可行的。由于客户端之间并无直接交互过程，即使恶意客户端与半诚实服务器共谋，也推断不出诚实客户端任何额外的隐私信息。证毕。

6　实验分析

本节通过实验来分析本文方案。首先对实验的基本设置做出描述，然后在MNIST[41]数据集和CIFAR-10[42]数据集上对本文方案进行全面的分析，以此评估方案的有效性。

6.1　数据集和实验设置

为了评估本文方案，使用2个常见的数据集：MNIST数据集和CIFAR-10数据集。MNIST数据集是机器学习和计算机视觉领域中广泛使用的基准数据集之一，包含60 000张训练集和10 000张测试集。每张图像都是手写的数字，类别为0~9，图像大小为28像素×28像素。CIFAR-10数据集常用于训练和评估图像分类算法，有10类不同的物体，如狗、鸟和飞机等，包含50 000张训练集和10 000张测试集。每张图像都是一个32像素×32像素的彩色图像，标签是0~9的整数。对于客户端本地模型训练，使用LeNet[43]架构。其中CIFAR-10数据集是彩色图像，由于LeNet神经网络输入的特性，将CIFAR-10数据集设置为灰度图像，即从三通道（RGB）改为一通道（灰度图像）。在实验中，将每个数据集平均分给10个客户端，协作训练全局模型。

实验在具有Intel(R) Core(TM) i5-12400 2.50 GH、16 GB RAM和NVIDIA GeForce RTX 2080 GPU的设备上实现。客户端本地模型训练使用GPU，加密协议使用CPU。FL实验使用Python语言进行编写，基于PyTorch库[44]来构建FL框架。为了进行基准测试和比较，方案使用了LeNet架构来进行客户端本地模型训练，第一层是一个2维卷积层，6个卷积核，大小为5×5，然后是激活函数ReLU，输出6个28像素×28像素的特征图，接下来是一个2×2的池化层，输出6个14×14的特征图。第二层是一个2维卷积层，16个卷积核，大小为5×5，激活函数同第一层，还有一个2×2的池化层，输出16个5×5的特征图。第三层是一个全连接层，输出120个神经元，使用ReLU激活函数。第四层是一个全连接层，输出84个神经元，使用激活函数ReLU。第五层是输出层，输出10个神经元，对应数据集的10个类别，具体LeNet神经网络模型参数如表3所示。此外，将CIFAR-10数据集从三通道（RGB）改为一通道（灰度图像），并将其大小裁剪为28像素×28像素。设置FL全局总训练轮数为50，本地客户端训练轮数为3。采用独立同分布的数据分布方式，为每个参与客户端分配相同的数量的训练数据集。为了模拟现实中投毒的场景，将分配至恶意客户端的训练数据集进行更改。具体来说，将恶意客户端数据集的标签改为另一个标签。例如，在CIFAR-10数据集中将飞机改为汽车、狗改为鸟等。为了体现防御方案的有效性，将恶意客户端比例调到40%来进行训练，加大投毒攻击对全局模型训练的影响。

表3　LeNet神经网络模型参数

层	输入	输出
第一层	1 × 28 × 28	6 × 14 × 14
第二层	6 × 14 × 14	16 × 5 × 5
第三层	16 × 5 × 5	120
第四层	120	84
第五层	84	10

6.2　实验结果

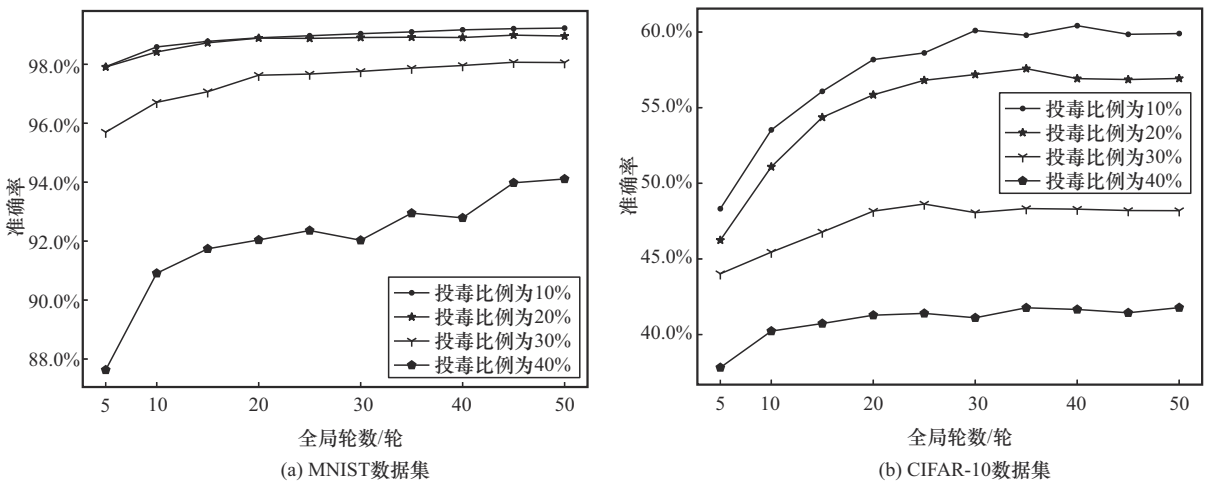
本文针对推理攻击与投毒攻击进行了实验验证，相较于Trim-mean和Median方案，本文方案仅需增加少部分的额外计算成本。即便在恶意客户端与半诚实服务器并存的不利环境下，本文方案仍能确保全局模型的安全协作训练，同时维护训练数据的准确性、鲁棒性及隐私性。这一结果有力地证明了本文方案的有效性和实用性。

1) 准确性

在FL场景中，准确性衡量的是模型在全局测试集上做出正确预测的能力。全局模型的准确性受多个因素的影响，如训练的次数、客户端用于训练的数据集大小、恶意客户端数量等。现有的研究显示，即使只有一个恶意的客户端，在对FL训练中不加以任何防护策略的情况下，全局模型的准确率也会受到较大的影响。在MNIST数据集和CIFAR-10数据集上评估了本文方案的准确性，在存在恶意客户端的情况下，研究了其他防御方案的差异，其中Median是选取中值来作为全局模型梯度的更新，从而减少极端值和噪声数据对模型的影响；Trim-mean是去掉最大最小值，然后求平均值来进行梯度更新。本文基于LeNet神经网络训练MNIST数据集和CIFAR-10数据集，通过数据投毒的方式，来衡量不同恶意客户端比例下全局模型的准确率。图3和图4分别显示了在投毒比例为10%、20%、30%和40%下安全聚合算法Trim-mean和Median的准确率变化。对于Trim-mean，当恶意客户端比例达到40%时，在MNIST数据集中准确率下降了5%，随着恶意客户端的比例不断增加，模型的准确率也随之下降。对于Median，当恶意客户端比例达到40%时，其模型准确率也会下降2%~4%，在CIFAR-10数据集中更为显著。图5显示了本文方案在面对不同恶意客户端比例下模型准确率的变化。与Trim-mean和Median相比，即使恶意客户端比例达到40%，本文方案依然高于前2种防御方法，在一定程度上能够保持模型的精度稳定。

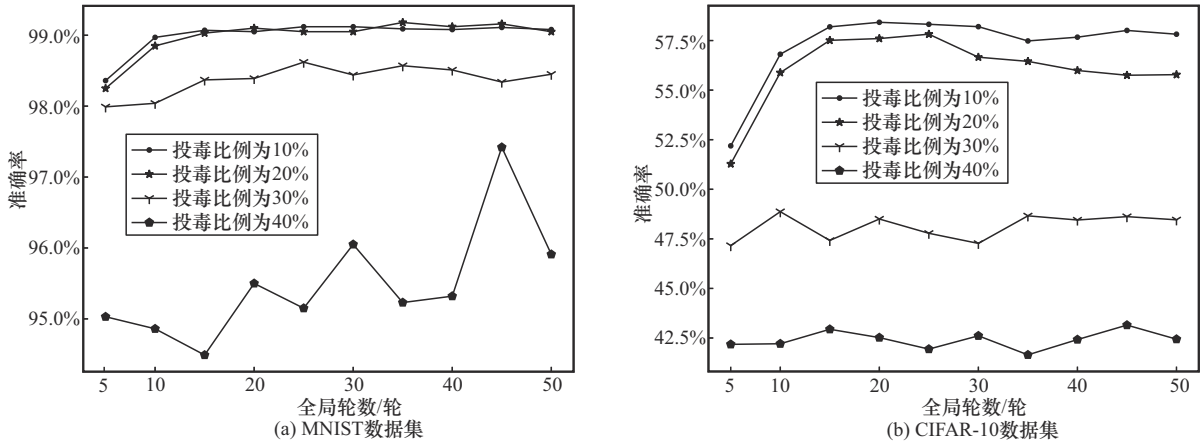
2) 鲁棒性

鲁棒性指的是FL系统在面对恶意客户端或非正常数据时，仍然能够保持模型性能的能力。在FL协作训练中，客户端通过有毒的数据集训练出有毒的本地模型梯度，以此影响全局模型聚合的准确率。在此有毒的场景下，以FedAvg方案为基准（不设置投毒，用于比较分析），分别设置恶意客户端比例为10%、20%、30%和40%，在MNIST数据集和CIFAR-10数据集上进行了实验。图6和图7分别显示了恶意客户端比例为10%和20%时Trim-mean、Median以及本文方案的准确率比较分析。当恶意客户端比例为10%时，准确率趋于稳定，Trim-mean、Median和本文方案的准确率没有较大变化。



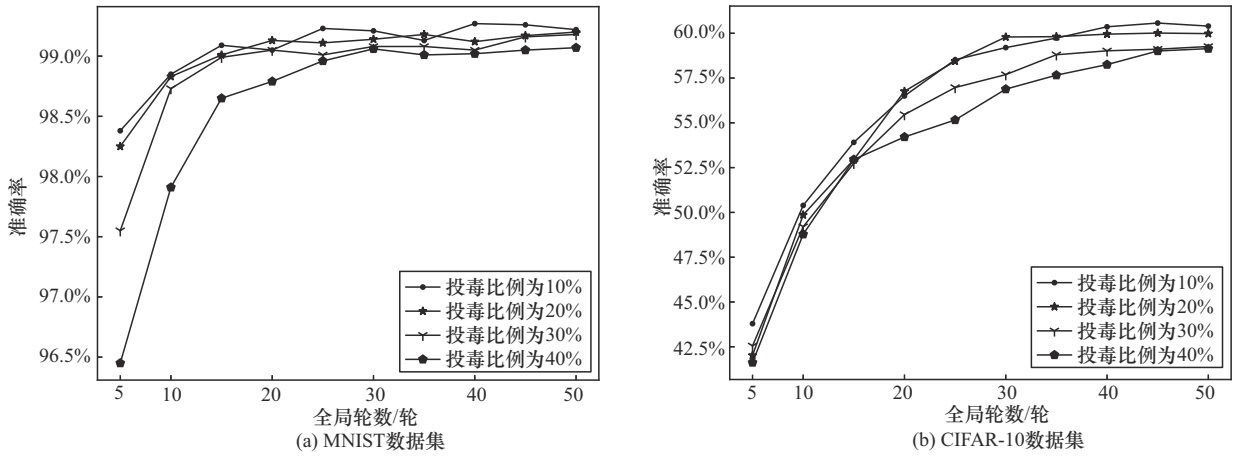
(a) MNIST数据集　　(b) CIFAR-10数据集

图3　Trim-mean方案中不同投毒比例下准确率的变化



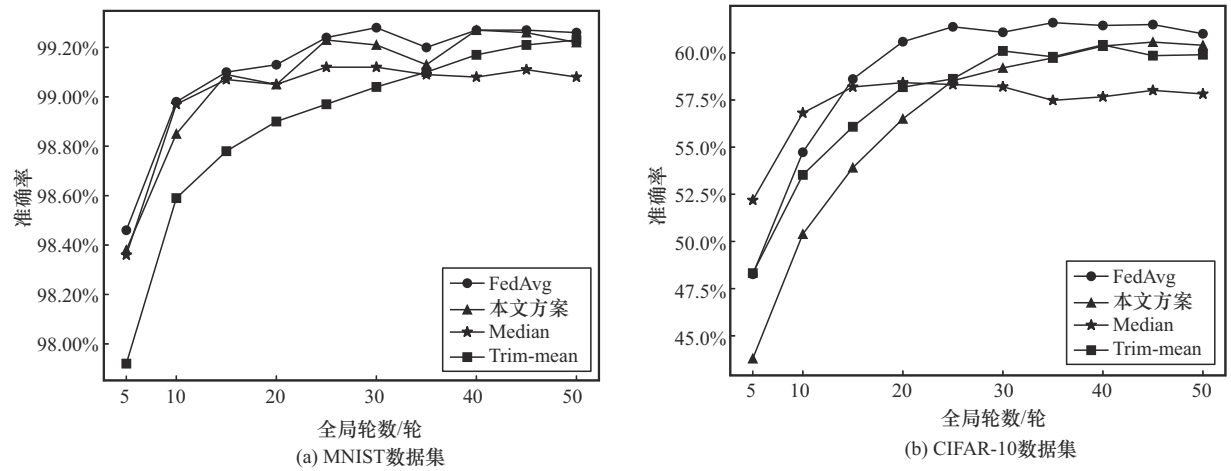
(a) MNIST数据集　　(b) CIFAR-10数据集

图4　Median方案中不同投毒比例下准确率的变化



(a) MNIST数据集　　(b) CIFAR-10数据集

图5　本文方案中不同投毒比例下准确率的变化



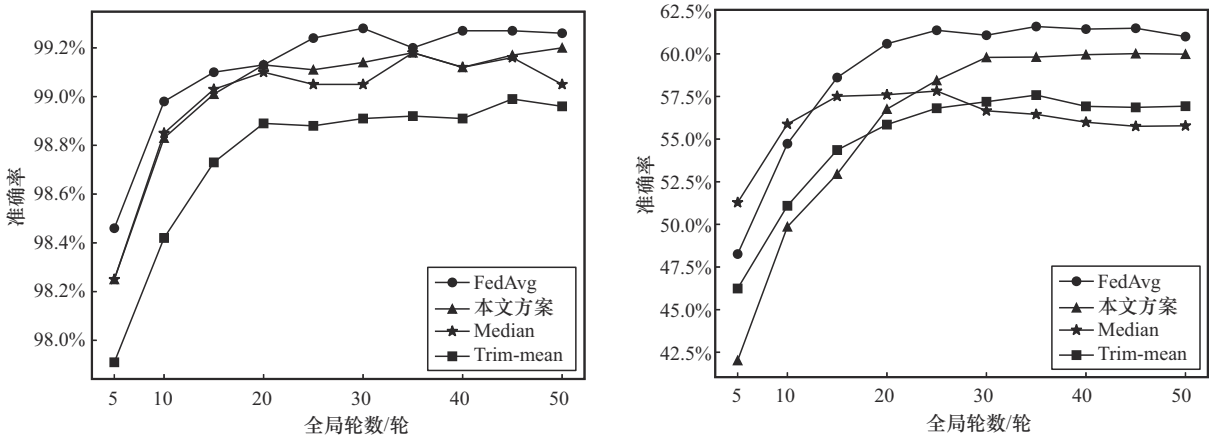
(a) MNIST数据集　　(b) CIFAR-10数据集

图6　恶意客户端比例为10%时3种方案的准确率比较

当恶意客户端比例为20%时，本文方案的准确率略高于Trim-mean和Median，能够较好抵御投毒攻击带来的影响。在恶意客户端比例为30%和40%时，本文方案显著优于Trim-mean和Median，在全局模型准确率上表现更佳，更接近FedAvg方案的性能，如图8和图9所示。总体而言，与Median和Trim-Mean相比，本文方案在抵御数据投毒方面表现更优，能够更有效地保持全局模型的性能。

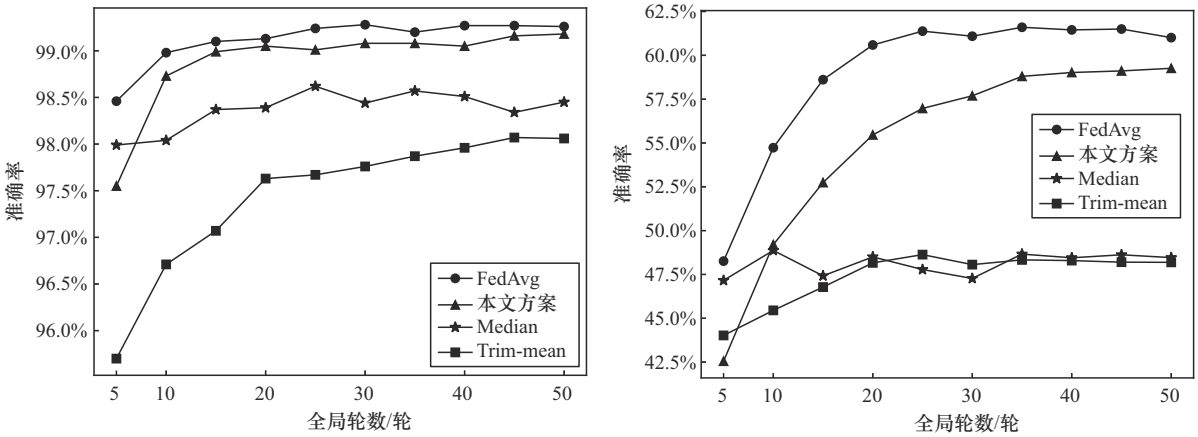
3) 隐私性

第5节中详细介绍了本文方案的安全性以及隐



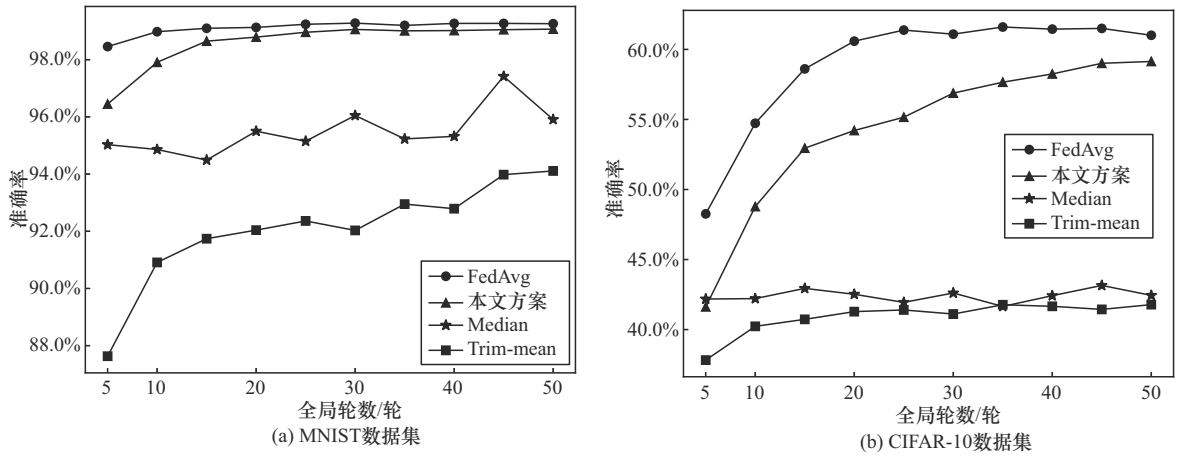
(a) MNIST数据集　(b) CIFAR-10数据集

图7　恶意客户端比例为20%时3种方案的准确率比较



(a) MNIST数据集　(b) CIFAR-10数据集

图8　恶意客户端比例为30%时3种方案的准确率比较



(a) MNIST数据集　(b) CIFAR-10数据集

图9　恶意客户端比例为40%时3种方案的准确率比较

私性。具体来说，通过安全多方计算来保证梯度的隐私性，任何一方服务器在不共谋的情况下都无法解密客户端原始本地模型梯度信息，达到了隐私保护的效果。

7　结束语

本文提出了基于多方计算的安全拜占庭弹性联邦学习方案，其目标是解决FL中隐私泄露问题和抵御投毒攻击等。具体来说，基于多方计算技术来

保护本地模型梯度隐私，抵抗恶意参与者的推理攻击。此外，本文设计了一种基于余弦相似度筛选机制来抵御客户端投毒攻击带来的负面影响，这种筛选机制可以较好地去除有毒的模型，从而更好地进行全局模型训练。为了验证本文方案的有效性，在2个常见的数据集上进行了实验。结果表明，所提方案不仅能够抵御推理攻击和投毒攻击，与此同时还能够保护客户端隐私。

长远来看，本文面临的一个主要局限性在于，服务器可能会因某种未知因素而掉线，这一潜在风险不仅会对全局模型的协同训练流程造成干扰，而且还可能引发训练数据的丢失或损毁，进而对模型的准确性、鲁棒性以及隐私保护能力产生不利的后果。针对这一挑战，未来的研究可以深入探索和实践有效的解决方案，旨在提升全局模型训练的稳定性和可靠性，同时确保训练数据的完整性和隐私性。

参考文献：

[1] SAGIROGLU S, SINANC D. Big data: a review[C]//Proceedings of the 2013 International Conference on Collaboration Technologies and Systems (CTS). Piscataway: IEEE Press, 2013: 42-47.

[2] JORDAN M I, MITCHELL T M. Machine learning: trends, perspectives, and prospects[J]. Science, 2015, 349(6245): 255-260.

[3] LI L, FAN Y X, TSE M, et al. A review of applications in federated learning[J]. Computers & Industrial Engineering, 2020, 149: 106854.

[4] SHOKRI R, STRONATI M, SONG C Z, et al. Membership inference attacks against machine learning models[C]//Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2017: 3-18.

[5] HITAJ B, ATENIESE G, PEREZ-CRUZ F, et al. Deep models under the GAN[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2017: 603-618.

[6] NASR M, SHOKRI R, HOUMANSADR A. Comprehensive privacy analysis of deep learning: passive and active white-box inference attacks against centralized and federated learning[C]//Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2019: 739-753.

[7] FANG C, GUO Y B, HU Y J, et al. Privacy-preserving and communication-efficient federated learning in Internet of things[J]. Computers & Security, 2021, 103: 102199.

[8] MA J, NAAS S A, SIGG S, et al. Privacy-preserving federated learning based on multi-key homomorphic encryption[J]. arXiv Preprint, arXiv: 2104.06824, 2021.

[9] JIA B, ZHANG X S, LIU J W, et al. Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT[J]. IEEE Transactions on Industrial Informatics, 2022, 18(6): 4049-4058.

[10] TRIASTCYN A, FALTINGS B. Federated learning with Bayesian differential privacy[C]//Proceedings of the 2019 IEEE International Conference on Big Data (Big Data). Piscataway: IEEE Press, 2019: 2587-2596.

[11] CHAMIKARA M A P, BERTOK P, KHALIL I, et al. Privacy preserving distributed machine learning with federated learning[J]. Computer Communications, 2021, 171: 112-125.

[12] BONAWITZ K, IVANOV V, KREUTER B, et al. Practical secure aggregation for privacy-preserving machine learning[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2017: 1175-1191.

[13] ABADI M, CHU A, GOODFELLOW I, et al. Deep learning with differential privacy[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2016: 308-318.

[14] FANG M H, CAO X Y, JIA J Y, et al. Local model poisoning attacks to Byzantine-robust federated learning[C]//Proceedings of the 29th USENIX Conference on Security Symposium. New York: ACM Press, 2020: 1623-1640.

[15] BHAGOJI A, CHAKRABORTY S, MITTAL P, et al. Analyzing federated learning through an adversarial lens[J]. arXiv Preprint, arXiv: 1811.12470, 2016.

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

[17] GUERRAOUI R, ROUAULT S. The hidden vulnerability of distributed learning in byzantium[J]. arXiv Preprint, arXiv: 1802.07927, 2018.

[18] YIN D, CHEN Y D, RAMCHANDRAN K, et al. Byzantine-robust distributed learning: towards optimal statistical rates[J]. arXiv Preprint, arXiv: 1803.01498, 2018.

[19] MUÑOZ-GONZÁLEZ L, CO K T, LUPU E C. Byzantine-robust federated machine learning through adaptive model averaging[J]. arXiv Preprint, arXiv: 1909.05125, 2019.

[20] ZHAO L C, WANG Q, ZOU Q, et al. Privacy-preserving collaborative deep learning with unreliable participants[J]. IEEE Transactions on Information Forensics and Security, 2019, 15: 1486-1500.

[21] XU G W, LI H W, ZHANG Y, et al. Privacy-preserving federated deep learning with irregular users[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(2): 1364-1381.

[22] ZHOU J, WU N, WANG Y S, et al. A differentially private federated learning model against poisoning attacks in edge computing[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(3): 1941-1958.

[23] WANG R J, LAI J S, LI X, et al. RPIFL: reliable and privacy-preserving federated learning for the Internet of things[J]. Journal of Network and Computer Applications, 2024, 221: 103768.

[24] WANG J B, WANG R J, XU G Q, et al. FedPKR: federated learning with non-IID data via periodic knowledge review in edge computing[J]. IEEE Transactions on Sustainable Computing, 2024, 9(6): 902-912.

[25] XU G W, LI H W, LIU S, et al. VerifyNet: secure and verifiable federated learning[J]. IEEE Transactions on Information Forensics and Security, 2019, 15: 911-926.

[26] ZHENG W T, POPA R A, GONZALEZ J E, et al. Helen: maliciously secure coopetitive learning for linear models[C]//Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2019: 724-738.

[27] PHONG L T, AONO Y, HAYASHI T, et al. Privacy-preserving deep learning: revisited and enhanced[C]//International Conference on Applications and Techniques in Information Security. Berlin: Springer , 2017: 100-110.

[28] TRUEX S, BARACALDO N, ANWAR A, et al. A hybrid approach to privacy-preserving federated learning[C]//Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security. New York:

ACM Press, 2019: 1-11.
[29] WEI K, LI J, DING M, et al. Federated learning with differential privacy: algorithms and performance analysis[J]. IEEE Transactions on Information Forensics and Security, 2020, 15: 3454-3469.
[30] ZHENG Y F, LAI S Q, LIU Y, et al. Aggregation service for federated learning: an efficient, secure, and more resilient realization[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(2): 988-1001.
[31] PHONG L T, AONO Y, HAYASHI T, et al. Privacy-preserving deep learning via additively homomorphic encryption[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(5): 1333-1345.
[32] LIU X Y, LI H W, XU G W, et al. Privacy-enhanced federated learning against poisoning adversaries[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 4574-4588.
[33] CAO X Y, FANG M H, LIU J, et al. FLTrust: Byzantine-robust federated learning via trust bootstrapping[J]. arXiv Preprint, arXiv: 2012.13995, 2020.
[34] 王瑞锦, 王金波, 张凤荔, 等. 联邦原型学习的特征图中毒攻击和双重防御机制[J]. 软件学报, 2025, 36(3): 1355-1374.
WANG R J, WANG J B, ZHANG F L,et al. Feature map poisoning attack and dual defense mechanism for federated prototype learning[J]. Journal of Software, 2025, 36(3): 1355-1374.
[35] HAHN C, KIM H, KIM M, et al. VerSA: verifiable secure aggregation for cross-device federated learning[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(1): 36-52.
[36] GUO J J, LI H Y, HUANG F R, et al. ADFL: a poisoning attack defense framework for horizontal federated learning[J]. IEEE Transactions on Industrial Informatics, 2022, 18(10): 6526-6536.
[37] MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[J]. arXiv Preprint, arXiv: 1602.05629, 2016.
[38] BEAVER D. Efficient multiparty protocols using circuit randomization[C]//Advances in Cryptology. Berlin: Springer, 2007: 420-432.
[39] KROHN M N, FREEDMAN M J, MAZIERES D. On-the-fly verification of rateless erasure codes for efficient content distribution[C]//Proceedings of the IEEE Symposium on Security and Privacy. Piscataway: IEEE Press, 2004: 226-240.
[40] FIORE D, GENNARO R, PASTRO V, et al. Efficiently verifiable computation on encrypted data[C]//Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2014: 844-855.
[41] LECUN Y, CORTES C, BURGES C J. The MNIST database of handwritten digits[R]. 2017.
[42] KRIZHEVSKY A, NAIR V, HINTON G. The CIFAR-10 dataset[R]. 2009.
[43] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86 (11): 2278-2324.
[44] PASZKE A, GROSS S, MASSA F, et al. Pytorch: an imperative style, high-performance deep learning library[J]. arXiv Preprint, arXiv: 1912.01703, 2019.

[作者简介]

高鸿峰（1975– ），男，贵州遵义人，贵州大学副教授、硕士生导师，主要研究方向为网络与信息安全。

黄浩（1999– ），男，侗族，贵州铜仁人，贵州大学硕士生，主要研究方向为联邦学习、隐私保护等。

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等。