# 基于更新质量检测和恶意客户端识别的联邦学习模型

雷诚，张琳

引用本文

雷诚，张琳. 基于更新质量检测和恶意客户端识别的联邦学习模型[J]. 计算机科学，2024，51(11)：368-378.

LEI Cheng, ZHANG Lin. Federated Learning Model Based on Update Quality Detection and Malicious Client Identification [J]. Computer Science, 2024, 51(11): 368-378.

---

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.231100044

# 基于更新质量检测和恶意客户端识别的联邦学习模型

雷 诚[1] 张 琳[1,2]
1 南京邮电大学计算机学院 南京 210003
2 江苏省无线传感网高技术研究重点实验室 南京 210003
(leicheng2021@163.com)

**摘 要** 作为分布式机器学习,联邦学习缓解了数据孤岛问题,其在不共享本地数据的情况下,仅在服务器和客户端之间传输模型参数,提高了训练数据的隐私性,但也因此使得联邦学习容易遭受恶意客户端的攻击。现有工作主要集中在拦截恶意客户端上传的更新。对此,研究了一种基于更新质量检测和恶意客户端识别的联邦学习模型 umFL,以提升全局模型的训练表现和联邦学习的鲁棒性。具体而言,通过获取每一轮客户端训练的损失值来计算客户端更新质量,进行更新质量检测,选择每一轮参与训练的客户端子集,计算更新的本地模型与上一轮全局模型的相似度,从而判定客户端是否做出积极更新,并过滤掉负面更新。同时,引入 beta 分布函数更新客户端信誉值,将信誉值过低的客户端标记为恶意客户端,拒绝其参与随后的训练。利用卷积神经网络,分别测试了所提算法在 MNIST 和 CIFAR10 数据集上的有效性。实验结果表明,在 20%~40%恶意客户端的攻击下,所提模型依旧是安全的,尤其是在 40%恶意客户端环境下,其相比传统联邦学习在 MNIST 和 CIFAR10 上分别提升了 40%和 20%的模型测试精度,同时分别提升了 25.6%和 22.8%的模型收敛速度。

**关键词**:联邦学习;客户端更新质量;客户端信誉值;恶意客户端识别;客户端选择
**中图分类号** TP393

# Federated Learning Model Based on Update Quality Detection and Malicious Client Identification

LEI Cheng[1] and ZHANG Lin[1,2]
1 College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
2 Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China

**Abstract** As a distributed machine learning, federated learning alleviates the problem of data islands, which only transmits model parameters between the server and the client without sharing local data and improves the privacy of training data, at the same time it also makes federated learning vulnerable to malicious client attacks. The existing research mainly focuses on intercepting updates uploaded by malicious clients. A federated learning model based on update quality detection and malicious client identification method, named umFL, is studied to improve the training performance of global models and the robustness of federated learning. Specifically, the client importance is calculated by obtaining the loss value of each round of client training. The subset of clients participating in each round of training is selected by update quality detection. The similarity between the updated local model and the previous round of global model is calculated to determine whether the client makes positive updates and the negative updates are filtered. Meanwhile, the beta distribution function is introduced to update the client reputation value. The clients with low reputation value are marked as malicious clients and excluded from participating in subsequent training. The effectiveness of the proposed algorithm on MNIST and CIFAR10 datasets is tested by using convolutional neural networks respectively. Experimental results show that under the attack of 20%~40% of malicious clients, the proposed model is still safe. Especially under the 40% malicious clients, the umFL model improves the model testing accuracy by 40% and 20% on MNIST and CIFAR10 respectively compared with traditional federated learning, and the model convergence speed is also improved by 25.6% and 22.8% respectively.

**Keywords** Federated learning, Client update quality, Client reputation value, Malicious user indentification, Client selection

到稿日期:2023-11-07 返修日期:2024-04-14
基金项目:国家自然科学基金(61872196,61872194);江苏省科技支撑计划(BE2017166);南京邮电大学自然科学基金(NY222142)
This work was supported by the National Natural Science Foundation of China(61872196,61872194), Scientific & Technological Support Project of Jiangsu Province(BE2017166) and Natural Science Foundation of Nanjing University of Posts and Telecommunications(NY222142).
通信作者:张琳(zhangl@njupt.edu.cn)

## 1　引言

随着物联网技术的迅猛发展，越来越多的边缘设备进入了人们的生活，如智能手机、平板电脑、可穿戴设备等。这些设备每天产生的数据数以亿计，随之而来的是不断增长的对数据处理的需求。传统的方法是利用机器学习对客户端数据进行处理，以此来满足人们对数据处理的需求亦或者为客户端提供个性化的服务。但是，在传统的机器学习环境下，受限于客户端设备自身的计算资源，客户端通常需要将数据上传到云服务器进行机器学习。然而，随着通用数据保护条例等新兴隐私法规的颁布和客户端对自身隐私数据的重视程度日益增长[1]，客户端通常不愿意将自身数据上传到云服务器，传统的机器学习遇到了瓶颈，数据孤岛问题随之产生。

为了应对数据孤岛问题，McMahan 等[2] 于 2017 年提出了联邦学习的概念。联邦学习是一种分散数据的新兴分布式机器学习，允许多个客户端基于本地数据集协同训练共享全局模型。具体来说，在联邦学习环境下，存在一个中心服务器和多个客户端，每个客户端都持有着各自的数据集，服务器首先对全局模型参数进行随机初始化，然后将初始化好的全局模型参数下发到各个客户端，客户端在接收到服务器下发的全局模型参数后，利用本地数据集进行本地训练，然后将训练之后的本地模型参数上传到服务器，服务器端在接收到所有客户端上传的本地模型参数后进行聚合，得到新一轮的全局模型，重复此迭代过程，直至全局模型收敛。由于联邦学习环境下服务器和客户端之间只会进行模型参数的传递而非原始数据的传递，因此联邦学习不仅大大节省了通信开销，还极大程度上保护了客户端的数据隐私。然而，由于联邦学习的特性，服务器端并不了解客户端的训练过程，因为客户端的资源异构性，各个客户端上传的参数可能良莠不齐，甚至会出现恶意客户端。Zhang 等[3] 提出了一种基于生成对抗网络(GANs)的毒数据生成方法 Data_Gen，其依赖于迭代更新的全局模型参数来再生感兴趣的受害者的样本，指出了联邦学习在面临注毒攻击时的脆弱性。传统联邦学习在面临注毒攻击时十分脆弱，仅仅一个恶意客户端就足以使得全局模型中毒，进而导致全局模型收敛速度变慢甚至不收敛。因此，智能地选择高质量的客户端，使联邦学习在面临注毒攻击时可以有效地进行防御，对提高联邦学习的可靠性和加快全局模型的收敛速度具有重要的意义。

提高联邦学习的可靠性和加快全局模型的收敛速度研究的主要难点在于：1)由于客户端数据的不可见，服务器难以有效地选择拥有较高质量的本地数据的客户端；2)当客户端中存在恶意客户端进行注毒攻击时，服务器难以识别，进而导致全局训练失败；3)对于恶意客户端，服务器难以有效地对其进行标记与跟踪。

针对上述问题，本文提出了基于客户端更新质量检测和恶意客户端识别的安全联邦学习模型，通过收集客户端本地数据训练的损失值 $Loss(k)$ 对客户端上传的本地模型参数进行质量评估；对于客户端每轮训练提交的本地模型参数，计算其收敛方向并与全局收敛方向进行比对，摒弃收敛方向差异较大的客户端模型参数，以此来保证聚合的客户端参数的可靠性；考虑到本地模型训练差异的随机性，引入客户端信誉值，结合 beta 分布函数对其进行计算；标记长期更新不合格的客户端。本文的主要贡献如下：

1)引入了客户端更新质量，为每个参与训练的客户端计算各自的更新质量得分，以此来选择更新质量得分较高的客户端，从而提升全局模型的收敛速度和最终精度。

2)结合本地模型和全局模型之间的模型相似度，判断本地模型和全局模型的收敛方向是否相近，对于相似度较低的客户端模型，拒绝其参与本轮聚合。

3)引入客户端信誉值，结合 beta 分布函数，使用 beta 分布函数的期望来计算每一个客户端的信誉值，将客户端行为分为正常行为和作恶行为两类，并结合模型相似度来判定一轮训练中每一个上传本地模型参数的客户端是否作恶。对于作恶客户端，降低其信誉值，并在最终聚合中根据信誉值来决定每个客户端上传的本地模型参数所占的比重。若是客户端信誉值低于阈值，则将其标记为恶意客户端，不再参与之后的训练。

4)将客户端更新质量和恶意客户端识别结合，利用恶意客户端的天然特性。恶意客户端作恶越频繁，其被识别出来的概率就越高，这就加快了识别恶意客户端的速度，使得恶意客户端作恶的机会大大降低。

5)通过实验验证本文提出的客户端更新质量和恶意客户端识别结合的可行性，证明了在不同攻击频率的环境下本文所提算法的鲁棒性与保真性。相较于传统联邦学习，所提算法提高了全局模型的收敛速度和最终模型精度。

## 2　背景及相关工作

### 2.1　联邦学习

联邦学习作为一种分布式机器学习，允许多个客户端在服务器的帮助下协同训练一个全局模型。传统的联邦学习算法为 FedAvg[2]，假设一共有 $N$ 个客户端，每个客户端都持有各自的本地数据集 $D_i$，$i=1,2,\cdots,N$，使用 $D=\bigcup_{i=1}^{N} D_i$ 表示总数据集。传统的联邦学习可以概括为以下 3 个步骤。

1)全局模型参数分发：全局模型参数初始化完成后，服务器将全局模型参数 $\omega$ 下发给所有客户端或者一部分被选中的客户端。

2)本地模型训练以及上传本地模型参数：客户端 $i$ 在接收到服务器下发的全局模型参数 $\omega_G^t$ 后，使用全局模型参数 $\omega_G^t$ 替换本地模型参数，然后利用本地数据集 $D_i$ 执行本地模型训练，其本质上是最小化本地损失函数 $L(\omega_i^t)$，即：

$$\omega_i^{t^*}=\arg\min_{\omega_i^t} L(\omega_i^t) \tag{1}$$

随后，将更新的本地模型参数上传到服务器。

3)全局模型聚合：服务器在接收到所有客户端上传的模型参数后，对本地模型参数进行聚合并更新全局模型参数至 $\omega_G^{t+1}$。

联邦学习的最终目标可以描述为最小化全局损失函数 $L(\omega_G^t)$，即：

$$L(\omega_G^t)=\frac{1}{N}\sum_{i=1}^{N} L(\omega_i^t) \tag{2}$$

重复上述迭代过程，直至模型收敛或者达到预先设置的训练轮数。算法 1 描述了传统联邦学习的执行过程。

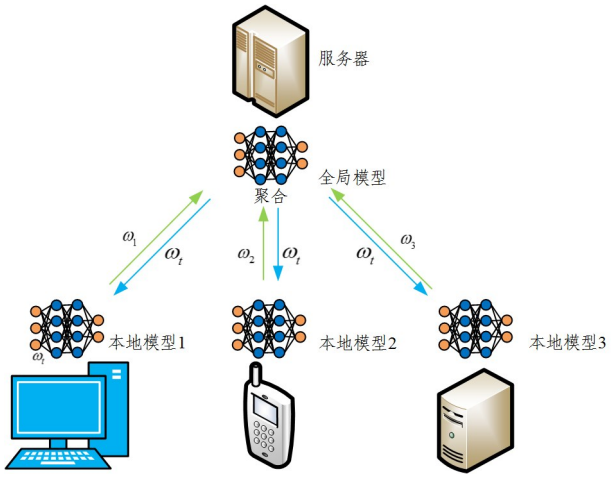



图 1 传统联邦学习

Fig. 1 Traditional federated learning

**算法 1** 传统联邦学习

输入：客户端集合 U，聚合客户端集合 S，选择比例 m，客户端数据集 $D_i(i=1,2,\cdots,K)$，学习率 $\eta$，全局模型初始参数 $\omega_G^0$，全局轮数 T

输出：全局模型 G

1. FOR epoch=0,1…,DO:
2. S←RandomChioce(U,m)
3. FOR each $U_i \in S$ in parallel DO
4. $\omega_i^{t+1}=SGD(\omega_i^t)$
5. Transmit $\omega_i^{t+1}$ to Server
6. END FOR
7. $\omega_{t+1}=AVG(\sum_{i\in S}\omega_i^{t+1})$
8. 广播 $\omega_{t+1}$ 到各个客户端
9. END FOR

## 2.2 相关工作

针对联邦学习，近年来国内外许多研究人员进行了广泛的研究[4]。Sattler 等[5]为了进一步提高全局模型与客户端的适配性，提出了个性化联邦学习，在所有客户端协同训练全局模型至最优解后，利用联邦学习损失面的几何特性，将客户群体分组为具有联合可训练数据分布的集群，并在集群中继续执行迭代算法，直至新的最优解产生。Fraboni 等[6]证明了聚类抽样可以更好地代表客户，提出了基于样本大小的客户机聚合和基于模型相似性的客户机聚合两种不同的聚类方法，证明了与标准抽样方法相比，通过聚类抽样进行的模型聚合始终能获得更好的训练收敛性和可变性。Chai 等[7]基于联邦学习中不同客户端之间存在计算能力和资源的异质性，指出计算能力和资源的异质性会对训练时间和模型精度造成明显的影响，提出根据客户端的训练表现，将客户端分类为不同的客户端组，每次进行组间客户端选择，以此来减少计算能力和资源的异质性对训练时间和模型精度造成的影响。Mhaisen 等[8]证明了在联邦学习中造成全局模型性能下降的一个主要原因是客户端设备上的数据的分布与全局分布之间的加权距离，提出了一种与边缘云计算相结合的方法，根据数据集分布和地理位置将客户端集切割开，相当于将一个大的联邦学习切割成一个个局部的联邦学习，最后利用云计算将边缘节点的模型聚合成全局模型，直至全局模型收敛。综合考虑模型精度、通信资源分配和能源消耗，Li 等[9]提出了一个精度-成本权衡优化问题，采用基于 DRL 的边缘关联方法，在不了解模型参数的情况下实现聚类。通过联合优化剪枝率、设备选择和无线资源分配的方法，Liu 等[10]提出了一个在给定的学习延迟预算下最大化收敛率的优化问题，通过解决该问题，导出了选择最优剪枝率和分配无线资源的封闭形式解，提出了基于阈值的客户端选择策略。Zou 等[11]提出了一个基于动态客户端选择的联邦学习框架，利用参数估计算法选择最优客户端加入协作，最终获得了更好的全局机器学习模型，但是其要求每个客户端在本地运行评估算法以对自身的参数进行评估，即便暂且不考虑客户端的计算能力的异构性，若存在恶意客户端，则会导致算法失效，最终造成全局模型精度下降乃至不收敛。Lai 等[12]提出客户端重要性的概念，根据本地模型的训练质量和训练时间将客户端重要性细分为统计重要性和系统重要性，综合考虑以上两者，最终对客户端进行选择，以此来提高全局模型的表现。考虑到终端设备的不可靠性，Wu 等[13]提出了一种新的客户端选择方案，相较于传统的联邦学习客户端选择方案[2]，该方案虽然依旧保留了客户端选择比例超参数 $C$，但不再将它作为硬约束应用，而是允许所有客户端在他们愿意的情况下参与，并允许中央服务器在收到 $C$ 部分更新后结束一轮交易，有效地将服务器与选定的客户端解耦，以减轻掉队者、崩溃和模型陈旧的影响，从而优化全局模型的收敛速度。为了应对联邦学习中客户端资源的异构性，Nishio 等[14]提出允许服务器在指定的最后期限内聚合尽可能多的客户端更新，希望通过每次选择较大比例的客户端来加快全局模型的收敛。Zhao 等[15]提出了一种基于工业大数据的匿名隐私保护联邦学习算法，在每轮训练时，只选取部分客户端节点中的部分参数进行上传，从而减少了隐私数据泄露的风险。此外，引入代理服务器作为中间层，以实现匿名机制，服务器无法将参数与客户端绑定起来，从而防止服务器作恶。Per-FedAvg 是 FedAvg 的变体，它利用与模型无关的元学习来实现个性化，但是 Per-FedAvg 的计算要求很高，这可能会导致严重的落后效应。Zhan 等[16]提出了一种策略，允许资源受限的客户端使用 FedAvg 的本地更新作为 Per-FedAvg 本地更新的近似值，以此来降低算法的计算量。Feng 等[17]提出了一种联邦强化学习框架，该框架利用多个客户端生成数据以同时训练模型。具体而言，所提出的框架允许表现最好的客户端与其他客户端分享其学习经验，以提高客户端的学习性能并保护客户端的隐私。Li 等[18]从社会学习的角度提出并充分阐述了一种可靠的个性化联邦学习方法，称为 RIPFL。RIPFL 可靠地选择和划分参与培训的客户端，以便每个客户端可以使用不同数量的社交信息，且更有效地与其他客户端沟通。Jeong 等[19]提出了一种个性化、完全去中心化的联邦学习算法，利用知识蒸馏技术对每个设备进行赋能，从而辨别局部模型之间的统计距离。每个客户端设备都可以在不共享本地数据的情况下提高其性能。Qin 等[20]提出了一种基于簇内训练和 top-$k$ 梯度稀疏化的方法，客户端采用集群内训练策略来减轻非 IID 数据的负面影响，

同时在客户端和服务器上实现稀疏梯度，以降低通信成本。然而，上述研究都未考虑存在恶意客户端的情况，因此服务器端很可能会遭遇恶意客户端的攻击，若不对恶意客户端上传的恶意模型参数加以识别，最终将导致模型收敛变慢甚至不收敛。

Song 等[21]利用局部模型的损失函数值与当前全局模型值的差值来评估接收到的本地模型参数更新的质量，以此来抵御恶意客户端的攻击，同时计算客户端的可信度，最终根据客户端可信度提出了一种新的客户端调度策略，但是并未对恶意客户端进行处理，使得恶意客户端的参与贯穿始终。Cao 等[22]为了更加可靠地识别出联邦学习中的恶意客户端，在服务器端也设置了一个服务器本地模型，用服务器提前收集的可信数据集进行训练，以此为全局模型提供一个标准的收敛方向，但是由于服务器数据集小且不一定与多数客户端数据集分布相符合，使用这种方法有一定的风险，并且很多情况下，服务器无法提前收集数据集。为了防止全局模型被污染，Liu 等[23]提出 eFL 模型，对比全局模型和本地模型的相似度，当相似度低于阈值时，本地模型上传的参数将被忽略，但是该方法并不能对恶意客户端进行识别，恶意客户端可以无限次重复作恶。实验表明，在攻击强度足够大的情况下，当全局模型接近收敛时，恶意客户端有可能突破防线，进而导致全局模型质量下降。本文提出基于更新质量检测和恶意客户端识别的联邦学习模型，在有效抵御恶意客户端攻击的前提下，一方面，对客户端进行更新质量检测，加快了全局模型的收敛速度；另一方面，与更新质量检测相结合，能够快速识别出恶意客户端。对于标记的恶意客户端，拒绝其参与后续训练，防止恶意客户端重复作恶，节省了计算成本。

## 3　UmFL 模型

本文提出了基于更新质量检测和恶意客户端识别的联邦学习模型(Federated Learning Model Based on Update Quality Detection and Malicious Client Identification，简称 umFL)。每次挑选高质量的客户端参与训练，关键是要对客户端上传的本地模型参数进行质量评估。另一方面，抵御恶意客户端攻击的本质是阻止恶意客户端上传的本地模型参数参与聚合，换言之，同样是对客户端上传的本地模型参数进行质量评估。因此，umFL 模型同时维护了两个值——客户端本地数据更新质量得分和自身信誉值，前者衡量了本地数据集的可训练价值，后者根据客户端每次上传的本地模型参数质量，衡量客户端的可信度。模型架构如图 2 所示。

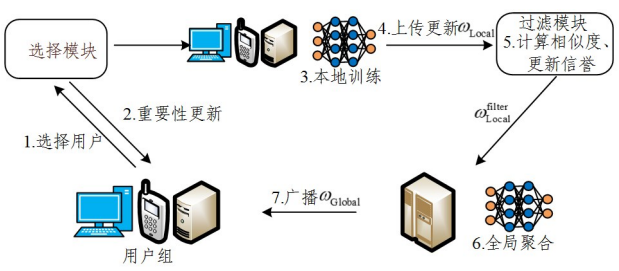



图 2　基于更新质量检测和恶意客户端识别的联邦学习模型

Fig. 2　Federated learning model based on update quality detection and malicious client identification

### 3.1　客户端选择

#### 3.1.1　客户端更新质量得分计算

文献[24-25]指出，在传统机器学习中，不同的样本具有不同的更新质量得分，主要思想可以描述为：假设在每个客户端本地都存在一个用来存放训练样本的数据集 $B_i$，那么，为了提高训练精度，可以根据其更新质量得分 $|B_i|\sqrt{\frac{1}{|B_i|}\sum_{k\in B_i}\|\nabla f(k)\|^2}$ 来选择训练样本，其中 $\|\nabla f(k)\|$ 表示样本 $k$ 的梯度二范数。但是，在联邦学习中，每个客户端需要每轮更新 $\|\nabla f(k)\|$，因此每轮都需要重新计算 $\|\nabla f(k)\|$，这无疑将会带来巨大的计算成本，并且这种计算需要在本地进行，而参与联邦学习的客户端本地计算能力参差不齐，因此将这一想法直接用于联邦学习环境下是不现实的。由于梯度是通过取训练损失对当前模型权重求导得到的，并且损失值的含义是目前训练结果与真实结果之间的误差，这意味着较大的梯度范数通常是由较大的损失值导致的，因此考虑使用 $Loss(k)$ 替代 $\|\nabla f(k)\|$，即每个客户端的更新质量得分 $U(i)$ 可以表示为：

$$U(i)=|B_i|\sqrt{\frac{1}{|B_i|}\sum_{k\in B_i}Loss\ (k)^2} \tag{3}$$

#### 3.1.2　公平性

联邦学习作为一种分布式的机器学习模型，拥有大量可获取的样本是其一大优势。如果联邦训练一直聚焦在少数几个客户端上，那么这不仅违背了联邦学习设计的初衷，也会导致最终产生的全局模型误差较大，泛化能力不足。因此，有必要在计算客户端更新质量得分时考虑训练公平性，对于那些多轮未被选中参与训练的客户端，可以适当提高其更新质量得分，以提升其被选中的概率，即：

$$U_i^t=|B_i|\sqrt{\frac{1}{|B_i|}\sum_{k\in B_i}Loss\ (k)^2}+U(i)\times\sqrt{R-L(i)}\times 0.1 \tag{4}$$

其中，$R$ 表示当前 epoch；$L(i)$ 表示客户端 $i$ 最后一次参与训练的 epoch，即对那些长时间未参与训练的客户端做出补偿，以提高其被选中的概率。

#### 3.1.3　首轮聚合

所有客户端上传的本地模型参数都需要与上一轮全局模型参数进行对比，即进行相似度计算。全局模型作为标准，一旦全局模型偏向恶意客户端，将会导致算法失效，进而导致最终的全局模型无法收敛，视为训练失败。而全局模型的更新来自与本地模型的聚合，由于全局模型的初始参数是随机生成的，因此第一次模型聚合十分重要。第一轮为广泛训练，即选中所有可用的客户端，被选中的客户端的本地模型参数与全局模型的初始参数进行相似度计算。值得注意的是，此时的全局模型参数是随机初始化的，没有任何实际意义，只是作为一个标准参与计算，以衡量出首轮被选中的客户端上传的本地模型参数之间的差异大小，计算出相似度均值，摒弃本轮相似度低于均值的本地模型参数，首轮执行平均聚合，即：

$$\omega_G^1=\sum_{i=1}^{N}\frac{D_i}{D}\omega_G^0 \tag{5}$$

#### 3.1.4　更新质量检测

假设多个木盒并排放在我们面前，首先给它们编号，每一轮

可以选择一个木盒打开，同时记录每个木盒开出的奖励。假设各个木盒不是完全相同的，那么经过多轮操作后，就可以勘探出木盒的部分统计信息，然后选择看起来奖励最高的木盒。在多臂赌博机中，我们把木盒称为臂。在多个客户端中选择参与的客户端子集可以建模为多臂赌博机问题[26]，每个客户端即赌博机的“臂”，客户端更新质量得分可以理解为“奖励”。即使可选择空间会随着时间变化，赌博机模型也可以灵活可扩展。因此，设计了一种自适应的对不同“臂”的探索和开发方案，以达到长期收益最大化。与多臂赌博机问题设置类似，本文的客户端选择方案可以在客户端集合中探索那些潜在参与者，并自适应地选择高更新质量得分的客户端。对于已经被选中参与过训练的客户端，可以根据其更新质量得分将选择空间缩小至某些高更新质量得分客户端。同时，对于还未被选中参与训练的客户端，设置 $\kappa\in[0,1]$ 作为本轮被选择的比例，进行随机选择。

### 3.2 异常更新检测

#### 3.2.1 相似度计算

由于联邦学习对隐私数据保护的天然特性，客户端与服务器端只会进行模型参数的交换，这确实为客户端隐私数据提供了良好的保护性，但是服务器也因此无法确认客户端提交的更新质量。如果客户中存在恶意客户端，故意上传毒数据，那么将会影响全局模型的质量，甚至会导致全局模型无法聚合。因此，对客户端上传的参数进行有目的的排查显得非常有必要。文献[5]指出，可以通过计算两个模型之间的参数相似度判断出两个模型是否相似，即收敛方向是否一致。目前可用的相似度度量方法包括欧氏距离、余弦相似度、马氏距离等。考虑到我们最终的目的是判断两个模型的收敛方向是否一致，而余弦相似度作为度量两个向量之间夹角的余弦值的方法，其本身就可以用来度量两个向量的方向，因此本文使用余弦相似度来计算全局模型和客户端本地模型之间的相似度。本文使用模型权重计算模型相似度，使用 $w_G^t$ 表示第 $t$ 轮全局模型权重，$w_i^t$ 表示第 $i$ 个客户端在第 $t$ 轮的模型权重。两个模型的相似度可以表示为：

$$sim_i^t=\cos(\omega_G^{t-1},\omega_i^t)=\frac{\omega_G^{t-1}\cdot\omega_i^t}{\|\omega_G^{t-1}\|\times\|\omega_i^t\|} \tag{6}$$

其中，$sim_i^t\in[-1,1]$，当本地模型参数更新方向与全局模型相反时，$sim_i^t$ 就会取到负值。为了表达更加直观，本文对模型相似度进行归一化，映射到 $[0,1]$，即：

$$sim_i^t=\frac{sim_i^t-sim_{\min}}{sim_{\max}-sim_{\min}}=\frac{sim_i^t-(-1)}{2} \tag{7}$$

#### 3.2.2 动态自适应阈值

由于 umFL 是零知识启动，没有任何先验知识和预训练，全局模型的参数是随机初始化的，这也就意味着在联邦训练的前期，本地客户端模型和全局模型的相似度并不会太高，因此，如果一开始就设置较高的阈值，那么在联邦训练的前期甚至会阻碍正常客户端参数的上传，最终导致全局模型的收敛速度变慢甚至不收敛；而将阈值设置得较低，那么恶意客户端将有机会将有毒模型上传到服务器进行聚合，从而成功向全局模型注毒，最终导致全局模型的收敛速度变慢甚至不收敛。综上所述，本文提出动态阈值的概念，考虑到正常本地客户端模型与全局模型的相似度将会随着全局训练轮数的上升而上升，本文将阈值定义为：

$$\alpha_t=\min[(\lambda+epoch\times\tau),\lambda_{\max}] \tag{8}$$

其中，epoch 表示当前全局训练轮数，$\lambda$ 表示在不同数据集上训练的初始阈值，$\tau$ 为成长系数。

### 3.3 恶意客户端识别

#### 3.3.1 客户端信誉值计算

为了表示客户端信誉值，本文考虑引入贝叶斯公式，其一般形式可以表示为：

$$P(H|D)=\frac{P(D|H)P(H)}{P(D)} \tag{9}$$

即计算在给定观测数据 $D$ 的情况下，假设 $H$ 的概率。由于本文是对离散型变量进行计算，因此有：

$$P(D)=\sum_i P(D|H_i)P(H_i) \tag{10}$$

即：

$$P(H|D)=\frac{P(D|H)P(H)}{\sum_i P(D|H_i)P(H_i)} \tag{11}$$

本文使用假设 $H$ 和观测数据 $D$ 分别表示客户端 $i$ 的信誉 $R_i$ 和服务器的观测数据 $D_i$，因此客户端信誉计算式可以表示为：

$$R_i=\frac{P(D_i|R_i)R_i}{\sum_i P(D_i|R_i)R_i} \tag{12}$$

为了便于客户端信誉 $R_i$ 的计算和更新，使用 $\beta$-分布来表示客户端 $i$ 的信誉，使用 gamma 函数表述为：

$$P(x)=\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}x^{a-1}(1-x)^{b-1} \tag{13}$$

其中，$a>0$，$b>0$ 且 $x\in(0,1)$。为了准确更新客户端信誉，快速识别出恶意模型上传的参数，从而最小化有毒参数对全局模型造成的影响，本文基于相似度将每轮客户端上传的模型分为两类：正面模型参数和负面模型参数。假设一个客户端 $U_i$ 向全局模型共上传 $r+w$ 次模型参数，其中 $r$ 次为正面模型参数，$w$ 次为负面模型参数。考虑到 umFL 为零知识启动，不需要任何先验知识，因此所有客户端的信誉初始化服从均匀分布，即：

$$P(x)=uni(0,1)=beta(1,1) \tag{14}$$

可以得到后验分布：

$$P(x)=\frac{Bin(r+w,r)beta(1,1)}{a+b+1}=beta(r+1,w+1) \tag{15}$$

容易看出，$x$ 的后验分布也服从 $\beta$-分布。因此，客户端信誉 $R_i$ 可以表示为：

$$R_i=beta(a_i+1,b_i+1) \tag{16}$$

其中，$a_i$ 表示客户端 $U_i$ 上传正面模型参数的次数，$b_i$ 表示客户端 $U_i$ 上传负面模型参数的次数。注意到，当前的客户端信誉 $R_i$ 只是一个概率而不是一个确切的值，因此使用对 $R_i$ 取期望的值来表示具体客户端信誉，即：

$$C_i=E[beta(a_i+1,b_i+1)]=\frac{a_i+1}{a_i+b_i+2} \tag{17}$$

本文使用模型之间的相似度来计算一个客户端的信誉值，从而判断一个客户端是否为恶意客户端，那么就需要对

客户端 $U_i$ 的信誉值进行更新，即对参数 $a_i$，$b_i$ 进行更新：

若 $sim_i^t \geqslant \alpha_t$：

$$a_i^{new} = a_i + \gamma(sim_i^t - \alpha_t) \tag{18}$$

若 $sim_i^t < \alpha_t$：

$$b_i^{new} = b_i + \delta(\alpha_t - sim_i^t) \tag{19}$$

其中，$\gamma$ 和 $\delta$ 分别表示奖励系数和惩罚系数。当一个客户端 $i$ 的信誉值 $C_i^t$ 低于信誉阈值 $\beta$ 时，客户端 $i$ 将会被标记为恶意客户端，将无法参与之后的训练。

3.3.2　信誉衰减

考虑到全局模型随着全局训练轮数的增加在不断更新，而客户端信誉值的计算是基于本地模型和全局模型的相似度，因此有必要在更新客户端信誉时考虑时间衰减，即：

$$\alpha_i^t{}' = \alpha_i^t \times time(t, \xi, L(i)) \tag{20}$$

其中，$time(t, \xi, L(i)) = Exp(-\xi \times (t - L(i)))$，$t$ 表示当前 epoch，$\xi$ 表示衰减系数，$L(i)$ 表示客户端 $i$ 最后一次参与训练的 epoch，若 $(t - L(i))$ 大于 3 则按照 3 计算。

umFL 首先基于更新质量检测算法选择出参与本轮训练的客户端，被选中的客户端随即开始训练本地模型，并将更新之后的本地模型参数发送给服务器，服务器在接收到本轮所有客户端上传的参数之后运行相似度计算模块得出每个客户端模型与上一轮全局模型的相似度，随后根据得到的相似度对 beta 参数 $a_i^{new}$，$b_i^{new}$ 进行更新，并计算每个客户端的信誉值。对于信誉值低于 $\beta$ 的客户端，将其标记为恶意客户端，拒绝其参与之后的训练；对于相似度低于 $\alpha_t$ 的客户端，将其本轮模型更新 $\omega_i^{t+1}$ 设置为 NULL。服务器以本轮最终参与聚合的客户端的信誉值作为权重进行全局模型聚合，即：

$$\omega_G^{t+1} = \sum_{i \in SUC} \mu_i^{t+1} \omega_i^{t+1}, \mu_i^{t+1} = \frac{C_i^t}{\sum_{i \in SUC} C_i^t} \tag{21}$$

随后将新的全局模型参数广播给所有客户端，重复上述迭代过程，直至模型收敛或者达到预先设置的训练轮数。算法 2 详细说明了 umFL 的执行流程。

**算法 2**　umFL 模型

输入：客户端集合 U，聚合客户端集合 S，选择比例 m，客户端数据集 $D_i$ $(i=1,2,\cdots,K)$，学习率 $\eta$，全局模型初始参数 $\omega_0$，全局轮数 T，客户端更新质量得分 Imp，相似度 $sim_i^t$ $(i=1,2,\cdots,k)$，客户端信誉 $C_i^t$ $(i=1,2,\cdots,k)$，相似度阈值 $\alpha_t$，信誉阈值 $\beta$

输出：全局模型 G，恶意客户端集 $E_v$

```
FOR epoch=0,1,… DO:
    S←ImpChioce(U,m)//以及质量检测
    FOR each U_i∈S in parallel DO
        ω_i^{t+1}=SGD(ω_i^t)
        Imp←GET_Imp(loss_i)//更新客户端的更新质量得分
        sim_i^t←GET_Sim(ω_i^{t+1},ω_t)//计算相似度
        a_i^new,b_i^new←GET_ab(sim_i^t,γ,δ)
        C_i^t←GET_Cred(a_i^new,b_i^new)//更新客户端信誉
        IF C_i^t<β DO
            Ev←U_i
        IF sim_i^t<α_t DO
            ω_i^{t+1}=NULL
        ELSE DO
            Transmit ω_i^{t+1} to Server
            更新聚合客户端集合 SUC←U_i
    END FOR
    ω_G^{t+1}=Σ_{i∈SUC} μ_i^{t+1} ω_i^{t+1}, μ_i^{t+1}=C_i^t / Σ_{i∈SUC} C_i^t
    广播ω_{t+1}到各个客户端
End For
```

## 4　实验分析

### 4.1　数据集和实验准备

本文分别在 MNIST 和 CIFAR10 两个数据集上评估联邦学习模型的训练结果，模型使用卷积神经网络（CNN）模型，所有实验基于 Pytorch 框架实现。

MNIST：一个手写数字数据集。其分为训练集和测试集，其中训练集包含 60 000 个样本，测试集包含 10 000 个样本，有 0～9 共 10 个类别，每幅图像由 28×28 个像素点组成。

CIFAR10：一个更加接近普适物体的图像数据集。其分为训练集和测试集，其中训练集包含 50000 个样本，测试集包含 10000 个训练样本，共有 10 个类别的 RGB 彩色图片，每幅图像由 32×32 个像素组成。相比 MNIST，CIFAR10 更加贴近现实生活中的真实物体，噪声更大。

CNN：包括输入层、两个卷积层、一个池化层、两个全连接层。特别地，采用 CIFAER10 进行训练时设置 3 个全连接层，卷积核大小为 5×5，激活函数为 ReLU。

本文设置 50 个客户端，将训练集样本平均分配到 50 个客户端中，即在 MNIST 环境下，每个客户端包含 1 200 个训练样本，在 CIFAR10 环境下，每个客户端包含 1000 个训练样本。本地模型迭代轮数设置为 5，local_batch 设置为 10。为了更加贴近联邦学习的真实环境，实验分别从 IID 和 No-IID 两个角度进行验证。

相似度阈值 $\alpha_t = \min[(\lambda + epoch * \tau), \lambda_{max}]$、信誉阈值 $\beta$ 以及衰减系数 $\xi$ 等参数设置如表 1 所列。

表 1　各类参数设置

Table 1　Various parameter settings

| | $\lambda$ | $\tau$ | $\lambda_{max}$ | $\beta$ | $\xi$ |
|---|---|---|---|---|---|
| MNIST | 0.93 | 0.0075 | 0.993 | 0.3 | 0.03 |
| CIFAR10 | 0.97 | 0.0025 | 0.997 | 0.32 | 0.03 |

注毒攻击通常指对机器学习的训练阶段进行攻击。其中一种数据注毒攻击方法是污染训练数据，从而破坏学习到的模型。数据注毒攻击已经被证明会对许多机器学习系统造成强大的破坏性，如推荐系统[27]、支持向量机 SVM[28]、神经网络[29-33]。本文采用标签翻转来生成有毒数据，先从训练集上分离出一部分数据 $D_{poison} \in D$，然后修改训练样本 $data \in D_{poison}$ 的标签 label。例如，在一个训练样本上的原始标签为 1，可以将其重新标记为 5，原始标签为 9，可以重新标记为 4，以此类推，最后将有毒数据分配给特定比例的客户端以模拟生成恶意客户端。

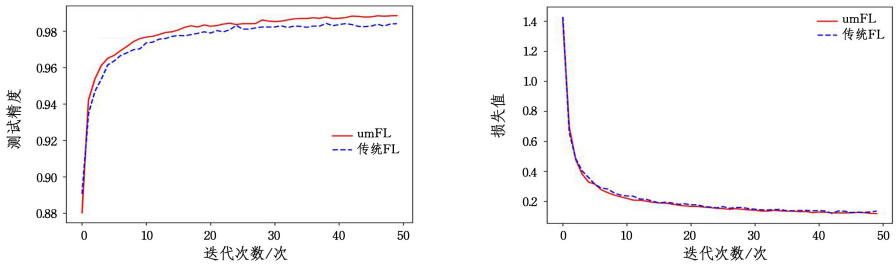
### 4.2 实验结果分析

为了模拟不同的攻击强度，本节分别设置无恶意客户端、0.2 比例恶意客户端、0.3 比例恶意客户端和 0.4 比例恶意客户端的环境来评估 umFL，另外训练采用 eFL[18] 和传统联邦学习[2]作为实验对照组。

#### 4.2.1 无恶意客户端环境下的模型表现

umFL 的目标可以归结为 3 个：保真性、鲁棒性和效率。首先，当不存在恶意客户端攻击时，我们的全局模型精度不应低于 FedAvg，但是这只是最基本的保真目标。进一步地，为了测试 umFL 模型在不存在恶意客户端攻击时是否还能体现出更新质量检测的优势，我们不进行标签翻转，而将所有训练样本平均分配到 50 个客户端中，每次训练选取 10 个客户端参与聚合，设置模型学习率为 0.01，如非特别说明，以下所有实验设置遵循上述说明。

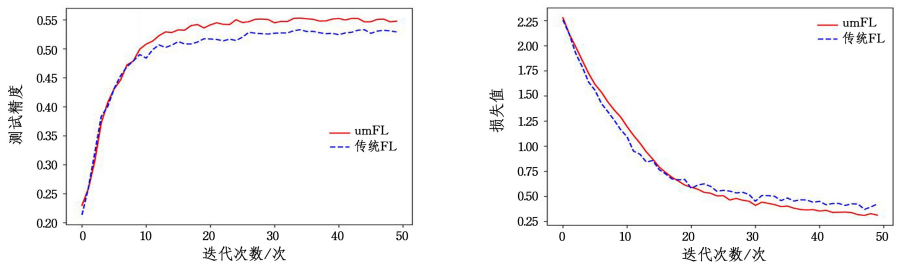
本文分别在 MNIST 和 CIFAR10 数据集上测量全局迭代 50 轮的全局模型准确率和训练损失，将所提方法与传统的联邦学习进行对比，并在图 3 和图 4 中描述了它们的关系。




(a)无攻击 MNIST 数据集下的模型精度　　(b)无攻击 MNIST 数据集下的训练损失值

图 3　MNIST 数据集下的训练表现

Fig. 3　Training performance on MNIST




(a)无攻击 CIFAR10 数据集下的模型精度　　(b)无攻击 CIFAR10 数据集下的训练损失值

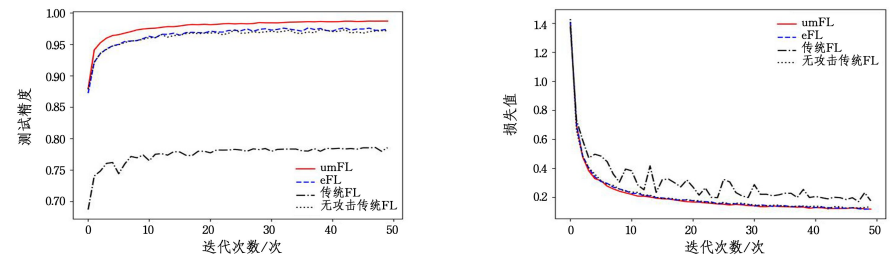
图 4　CIFAR10 数据集下的训练表现

Fig. 4　Training performance on CIFAR10

可以看出，针对正常环境下的联邦学习需求，umFL 在全局模型精度方面不仅没有落后于传统 FL(这说明 umFL 达到了保真度的目标)，反而在 MNIST 上提升了约 0.5%，在 CIFAR10 上提升了约 3%，说明基于更新质量检测的方法是有效的。

#### 4.2.2 不同比例恶意客户端环境下的模型表现

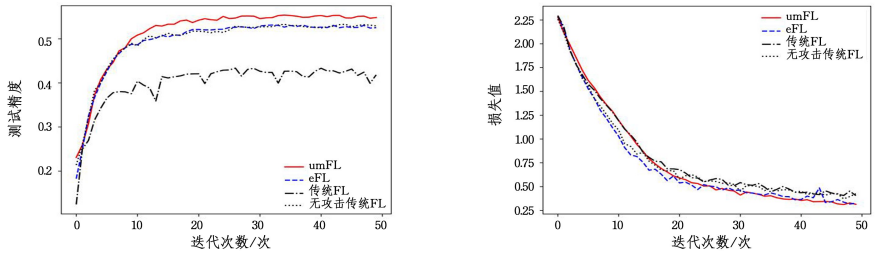
联邦学习由于其天然特性，在遭遇恶意客户端攻击时非常脆弱，因此，为了验证所提模型在遭遇恶意客户端攻击时是否能够保持全局模型精度，本文设置了 3 种强度的攻击，分别是占比为 0.2，0.3，0.4 的恶意客户端。特别地，为了更加贴近真实联邦学习环境，在恶意客户端占比为 0.3 的情况下同时对 IID 和 No-IID 两种数据分布模式进行验证，分别测试了在面临不同攻击强度下 umFL，eFL 和传统 FL 的全局模型测试精度和损失值，实验分别在 MNIST 和 CIFAR10 两个数据集上进行，结果如图 5—图 12 所示。




(a)0.2 攻击强度 MNIST 数据集下模型精度对比　　(b)0.2 攻击强度 MNIST 数据集下损失值对比

图 5　MNIST 数据集下 0.2 恶意客户端训练表现

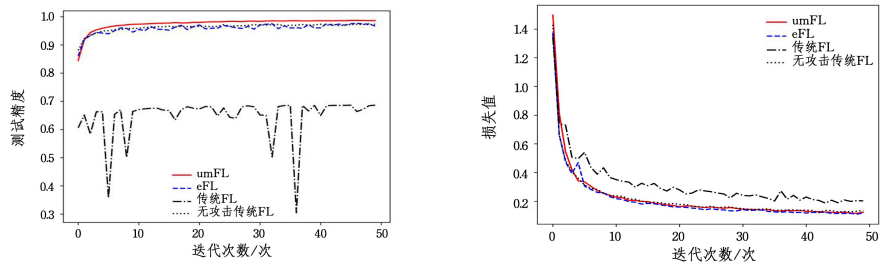
Fig. 5　Training performance under 0.2 attack intensity on MNIST




(a)0.2 攻击强度 CIFAR10 数据集下模型精度对比　(b)0.2 攻击强度 CIFAR10 数据集下损失值对比

图 6　CIFAR10 数据集下 0.2 恶意客户端训练表现

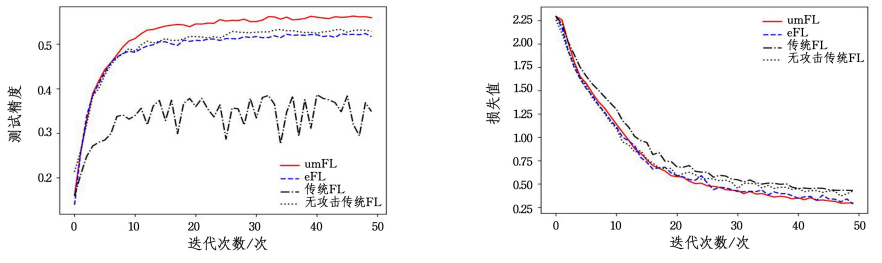
Fig. 6　Training performance under 0.2 attack intensity on CIFAR10




(a)0.3 攻击强度 MNIST 数据集下模型精度对比　(b)0.3 攻击强度 MNIST 数据集下损失值对比

图 7　MNIST 数据集下 0.3 恶意客户端训练表现(IID)

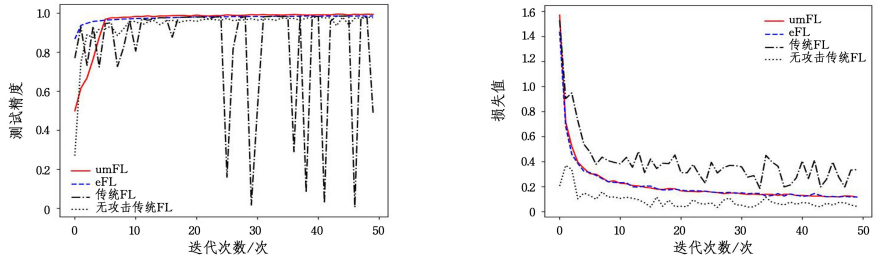
Fig. 7　Training performance under 0.3 attack intensity on MNIST(IID)




(a)0.3 攻击强度 CIFAR10 数据集下模型精度对比　(b)0.3 攻击强度 CIFAR10 数据集下损失值对比

图 8　CIFAR10 数据集下 0.3 恶意客户端训练表现(IID)

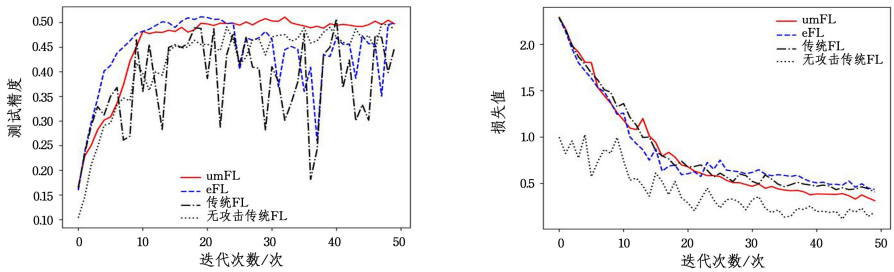
Fig. 8　Training performance under 0.3 attack intensity on CIFAR10(IID)




(a)0.3 攻击强度 MNIST 数据集下模型精度对比　(b)0.3 攻击强度 MNIST 数据集下损失值对比

图 9　MNIST 数据集下 0.3 恶意客户端训练表现(No-IID)

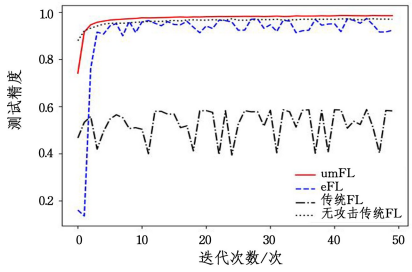
Fig. 9　Training performance under 0.3 attack intensity on MNIST(No-IID)



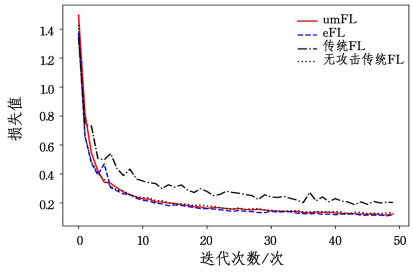

(a)0.3 攻击强度 CIFAR10 数据集下模型精度对比　(b)0.3 攻击强度 CIFAR10 数据集下损失值对比

图 10　CIFAR10 数据集下 0.3 恶意客户端训练表现(No-IID)

Fig. 10　Training performance under 0.3 attack intensity on CIFAR10(No-IID)



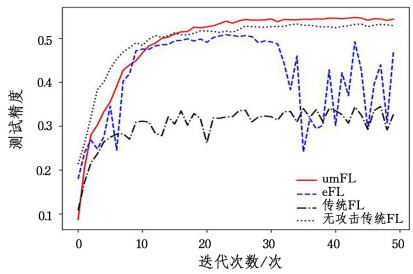

(a)0.4 攻击强度 MNIST 数据集下模型精度对比



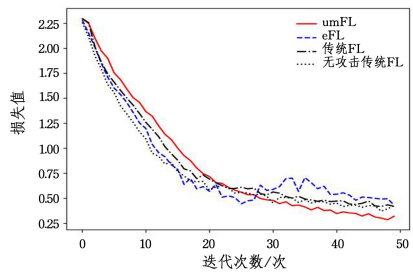

(b)0.4 攻击强度 MNIST 数据集下损失值对比

图 11 MNIST 数据集下 0.4 恶意客户端训练表现

Fig. 11 Training performance under 0.4 attack intensity on MNIST




(a)0.4 攻击强度 CIFAR10 数据集下模型精度对比




(b)0.4 攻击强度 CIFAR10 数据集下损失值对比

图 12 CIFAR10 数据集下 0.4 恶意客户端训练表现

Fig. 12 Training performance under 0.4 attack intensity on CIFAR10

可以看出，在 0.2 的攻击强度下，umFL 和 eFL 的表现都很稳定，全局模型精度都不低于无攻击下的传统联邦学习，损失值下降也表现得非常稳定；而传统联邦学习的全局模型受到了非常严重的影响，在 MNIST 数据集上精度下降了 20%，在 CIFAR10 数据集上精度下降了 10%，且损失值明显增大，波动频繁。然而，随着恶意客户端占比的增加，umFL 的表现依旧稳定，在 3 种强度的攻击下全局模型精度不仅不低于无攻击下的传统联邦学习，且由于结合了更新质量检测的方法，全局模型精度甚至高出无攻击下的传统联邦学习 0.5%～3%，损失值下降曲线下降稳定，且始终迭代至最低点。而 eFL 随着攻击强度的增大，全局模型受到的影响越来越大，可以看到，在遭遇攻击强度为 0.4 的恶意攻击时，eFL 的全局模型受到了非常大的影响。具体而言，在精度方面，精度曲线震荡明显，甚至在训练难度较大的 CIFAR10 数据集上出现了较大程度的下降。在损失值方面，eFL 的损失下降曲线也显得十分不稳定。这是由于随着全局模型接近收敛，本地数据集对于全局模型的修正会越来越小，而每一轮本地模型的参数都来源于上一轮聚合的全局模型，这就会导致本地模型与全局模型的相似度越来越高，恶意客户端与正常客户端也因此越来越相似(这也是本文提出动态自适应阈值概念的原因)。因此，本文提出恶意客户端识别，结合更新质量检测算法，迅速且准确地对恶意客户端进行识别，拒绝被标记的客户端参与之后的训练。图 5—图 10 证明了 umFL 的有效性。

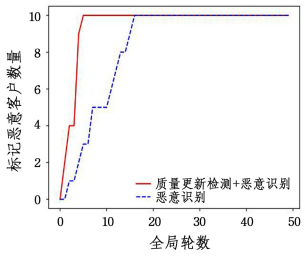
总体而言，无论是在 MNIST 上还是在 CIFAR10 上，umFL 都能够保证模型在遭遇不同强度的恶意攻击时最终依旧收敛至不存在攻击时的精度，精度提升稳定，训练损失值显著下降。而 eFL 在遭遇较大强度的攻击时，容易被突破，图 9 和图 10 非常直观地体现了这一点。反观传统联邦学习，在遭遇恶意客户端攻击时全局模型受到了非常严重的污染，由于攻击强度的不同，在 MNIST 上测试精度下降了 20%～30%，在 CIFAR10 上测试精度下降了 10%～20%，且精度曲线震荡明显，十分不稳定。显然，在面临不同强度的恶意攻击下，umFL 依然能够保证全局模型稳定收敛，这说明 umFL 达到了鲁棒性目标。

#### 4.2.3 更新质量检测和恶意客户端识别相结合

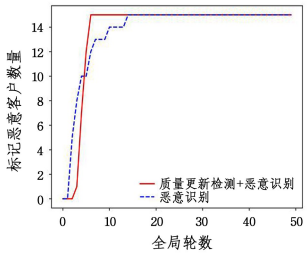
由于每一轮本地模型都是上一轮更新的全局模型的拷贝，而每一次全局模型的更新都需要进行相似度的对比。若恶意客户端进行数据投毒，由于恶意客户端拦截机制，恶意客户端本地模型将无法参与聚合，那么其训练的 loss 值就会被放大，因此恶意客户端的更新质量得分就会出现虚高的情况，从而更容易被选中。同时，引入客户端信誉值，那么客户端作恶的频率越高，就越容易被识别，这就加快了恶意客户端被识别的速度，使得恶意客户端作恶的机会大大降低。为了更直观地说明更新质量检测和恶意客户端识别方法相结合的优越性，本文设置了两组实验，一组为更新质量检测与恶意客户端识别相结合的方法，一组仅使用恶意客户端识别，在 MNIST 和 CIFAR10 上分别测试并记录了在 3 种不同攻击强度下 umFL 对恶意客户端的识别速度，如图 13 和图 14 所示。

可以看出，在不同攻击强度下，结合了更新质量检测的恶意客户端识别方法可以更加迅速地识别出恶意客户端。具体而言，在 MNIST 数据集上，结合了更新质量检测的恶意客户端识别方法都能够做到在 10 轮以内就将恶意客户端全部标记；而在仅使用恶意识别的情况下，需要经过超过 15 轮的训练才可以将恶意客户端全部标记。在 CIFAR10 数据集上，这种情况更加明显，结合了更新质量检测的恶意客户端识别方法

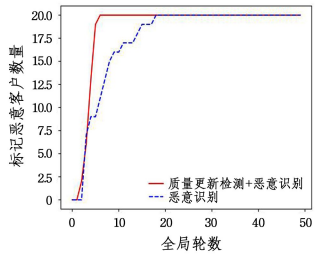
依旧能够做到在 10 轮以内就将恶意客户端全部标记，而仅使用恶意客户端识别的情况下，通常需要超过 20 轮的训练。




(a)0.2 攻击强度 MNIST 数据集下不同方式恶意客户端识别速度



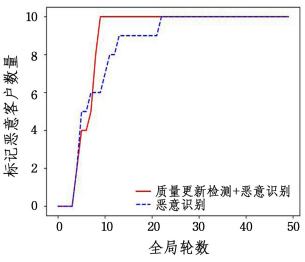

(b)0.3 攻击强度 MNIST 数据集下不同方式恶意客户端识别速度



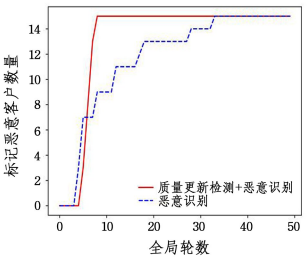

(c)0.4 攻击强度 MNIST 数据集下不同方式恶意客户端识别速度

图 13　不同攻击强度 MNIST 数据集下恶意客户端的识别速度

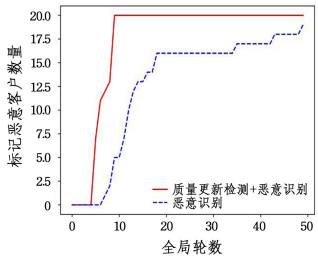
Fig. 13　Identification speed of malicious client under different attack intensities on MINST




(a)0.2 攻击强度 CIFAR10 数据集下不同方式恶意客户端识别速度




(b)0.3 攻击强度 CIFAR10 数据集下不同方式恶意客户端识别速度




(c)0.4 攻击强度 CIFAR10 数据集下不同方式恶意客户端识别速度

图 14　不同攻击强度 MNIST 数据集下恶意客户端的识别速度

Fig. 14　Identification speed of malicious client under different attack intensities on MNIST

### 4.2.4　效率提升分析

umFL 在识别出恶意客户端后，会对其进行标记。被标记为恶意客户端的客户端，将被服务器拒绝参与将来的训练。这样做不仅能防止恶意客户端重复作恶，还可以避免不必要的计算，加速全局模型收敛，如避免评估恶意客户端上传的参数，或者等待其完成本地训练等。

**结束语**　本文提出了一种基于更新质量检测和恶意客户端识别的联邦学习模型 umFL，旨在提高全局模型的收敛速度以及在面临数据注毒攻击时联邦学习的鲁棒性。引入更新质量检测的思想对客户端进行选择，优先选取具有高更新质量得分的客户端，利用余弦相似度计算出本地模型与全局模型的收敛方向是否一致，拦截异常客户端参与聚合，结合相似度，提出客户端信誉值，并在全局范围内持续更新客户端信誉值，对于信誉值过低的客户端，将其标记为恶意客户端，拒绝与其进行参数交换。最后，在 MNIST 和 CIFAR10 两个数据集上进行实验，验证了 umFL 在提高和加速全局模型的收敛，以及在防御恶意客户端攻击时的有效性。

## 参 考 文 献

[1] ZHANG P C, JIN H Y. A Privacy-Oriented Prediction Method in Mobile Edge Environment [J]. Chinese Journal of Computers, 2020, 43(8): 1555-1571.

[2] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [C]// Artificial Intelligence and Statistics. PMLR, 2017: 1273-1282.

[3] ZHANG J, CHEN B, CHENG X, et al. Poisongan: Generative poisoning attacks against federated learning in edge computing systems[J]. IEEE Internet of Things Journal, 2020, 8(5): 3310-3322.

[4] LIU Y X, CHEN H, LIU Y H, et al. Privacy Protection Technology in Federated Learning[J]. Journal of Software, 2022, 33(3): 1057-1092.

[5] SATTLER F, MÜLLER K R, SAMEK W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints[J]. IEEE transactions on neural networks and learning systems, 2020, 32(8): 3710-3722.

[6] FRABONI Y, VIDAL R, KAMENI L, et al. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning[C]// International Conference on Machine Learning. PMLR, 2021: 3407-3416.

[7] CHAI Z, ALI A, ZAWAD S, et al. Tifl: A tier-based federated learning system [C] // Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing. 2020: 125-136.

[8] MHAISEN N, ABDELLATIF A A, MOHAMED A, et al. Optimal user-edge assignment in hierarchical federated learning based on statistical properties and network topology constraints [J]. IEEE Transactions on Network Science and Engineering, 2021, 9(1): 55-66.

[9] LI Y, QIN X, CHEN H, et al. Energy-Aware Edge Association for Cluster-based Personalized Federated Learning [J]. IEEE Transactions on Vehicular Technology, 2022, 71(6): 6756-6761.

[10] LIU S, YU G, YIN R, et al. Joint Model Pruning and Device Selection for Communication-Efficient Federated Edge Learning [J]. IEEE Transactions on Communications, 2021, 70(1): 231-244.

[11] ZOU S, XIAO M, XU Y, et al. FedDCS: Federated Learning

Framework based on Dynamic Client Selection[C]// 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems(MASS). IEEE, 2021: 627-632.

[12] LAI F, ZHU X, MADHYASTHA H V, et al. Oort: Efficient federated learning via guided participant selection[C]// 15th {USENIX} Symposium on Operating Systems Design and Implementation({OSDI} 21). 2021: 19-35.

[13] WU W, HE L, LIN W, et al. SAFA: A semi-asynchronous protocol for fast federated learning with low overhead[J]. IEEE Transactions on Computers, 2020, 70(5): 655-668.

[14] NISHIO T, YONETANI R. Client selection for federated learning with heterogeneous resources in mobile edge[C]// 2019 IEEE International Conference on Communications. IEEE, 2019: 1-7.

[15] ZHAO B, FAN K, YANG K, et al. Anonymous and privacy-preserving federated learning with industrial big data[J]. IEEE Transactions on Industrial Informatics, 2021, 17(9): 6314-6323.

[16] ZHAN Z, ZHANG X. Computation-Effective Personalized Federated Learning: A Meta Learning Approach[C]// 2023 IEEE 43rd International Conference on Distributed Computing Systems(ICDCS). IEEE, 2023: 957-958.

[17] FENG W, LIU H, PENG X. Federated Reinforcement Learning for Sharing Experiences Between Multiple Workers[C]// 2023 International Conference on Machine Learning and Cybernetics (ICMLC). IEEE, 2023: 440-445.

[18] LI Y, LIU Z, HUANG Y, et al. FedOES: An Efficient Federated Learning Approach[C]// 2023 3rd International Conference on Neural Networks, Information and Communication Engineering (NNICE). IEEE, 2023: 135-139.

[19] JEONG E, KOUNTOURIS M. Personalized Decentralized Federated Learning with Knowledge Distillation[J]. arXiv: 2302. 12156, 2023.

[20] QIN Z, YANG L, WANG Q, et al. Reliable and Interpretable Personalized Federated Learning[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 20422-20431.

[21] SONG Z, SUN H, YANG H H, et al. Reputation-based Federated Learning for Secure Wireless Networks[J]. IEEE Internet of Things Journal, 2021, 9(2): 1212-1226.

[22] CAO X, FANG M, LIU J, et al. Fltrust: Byzantine-robust federated learning via trust bootstrapping[J]. arXiv: 2012. 13995, 2020.

[23] LIU Y, WANG T, PENG S L, et al. Cleaning and Equipment Clustering Method of Federated Learning Model Based on Edge [J]. Chinese Journal of Computers, 2021, 12: 2515-2528.

[24] KATHAROPOULOS A, FLEURET F. Not all samples are created equal: Deep learning with importance sampling[C]// International Conference on Machine Learning. PMLR, 2018: 2525-2534.

[25] ZHAO P, ZHANG T. Stochastic optimization with importance sampling for regularized loss minimization[C]// International Conference on Machine Learning. PMLR, 2015: 1-9.

[26] SHI C, SHEN C. Federated multi-armed bandits[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2021: 9603-9611.

[27] FANG M, YANG G, GONG N Z, et al. Poisoning attacks to graph-based recommender systems[C]// Proceedings of the 34th Annual Computer Security Applications Conference. 2018: 381-392.

[28] BIGGIO B, NELSON B, LASKOV P. Poisoning attacks against support vector machines[J]. arXiv: 1206. 6389, 2012.

[29] CHEN X, LIU C, LI B, et al. Targeted backdoor attacks on deep learning systems using data poisoning[J]. arXiv: 1712. 05526, 2017.

[30] GU T, DOLAN-GAVITT B, GARG S. Badnets: Identifying vulnerabilities in the machine learning model supply chain[J]. arXiv: 1708. 06733, 2017.

[31] LIU Y, MA S, AAFER Y, et al. Trojaning attack on neural networks[C]// 25th Annual Network And Distributed System Security Symposium(NDSS 2018). Internet Soc, 2018.

[32] MUÑOZ-GONZÁLEZ L, BIGGIO B, DEMONTIS A, et al. Towards poisoning of deep learning algorithms with back-gradient optimization[C]// Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. 2017: 27-38.

[33] SHAFAHI A, HUANG W R, NAJIBI M, et al. Poison frogs! targeted clean-label poisoning attacks on neural networks[J]. arXiv: 1804. 00792, 2018.

**LEI Cheng**, born in 1999, postgraduate. His main research interests include federated learning and privacy protection.

**ZHANG Lin**, born in 1980, Ph. D, associate professor, postgraduate supervisor. Her main research interests include trusted computing, federated learning and privacy protection.

(责任编辑: 何杨)