计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202330886
61(9): 2185−2198, 2024

基于梯度回溯的联邦学习搭便车攻击检测

洪 榛[1,2] 冯王磊[1] 温震宇[1,2] 吴 迪[3] 李涛涛[1] 伍一鸣[1,2] 王 聪[4] 纪守领[5]

[1](浙江工业大学信息工程学院 杭州 310023)

[2](浙江工业大学网络空间安全研究院 杭州 310023)

[3](圣安德鲁斯大学计算机学院 圣安德鲁斯 KY16 9AJ)

[4](浙江大学控制科学与工程学院 杭州 310007)

[5](浙江大学计算科学与技术学院 杭州 310007)

(zhong1983@zjut.edu.cn)

Detecting Free-Riding Attack in Federated Learning Based on Gradient Backtracking

Hong Zhen[1,2], Feng Wanglei[1], Wen Zhenyu[1,2], Wu Di[3], Li Taotao[1], Wu Yiming[1,2], Wang Cong[4], and Ji Shouling[5]

[1] (*College of Information Engineering*, *Zhejiang University of Technology*, *Hangzhou* 310023)

[2] (*Institute of Cyberspace Security*, *Zhejiang University of Technology*, *Hangzhou* 310023)

[3] (*College of Computer Science, University of St Andrews, St Andrews* KY16 9AJ)

[4] (*College of Control Science and Engineering*, *Zhejiang University*, *Hangzhou* 310007)

[5] (*College of Computer Science and Technology*, *Zhejiang University*, *Hangzhou* 310007)

Abstract With the development of the Internet of vehicles (IoV), the rapid growth of intelligent vehicles generates a massive amount of data. These data are invaluable for training intelligent IoV application models. Traditional model training requires the centralized collection of raw data through the cloud, consuming substantial communication resources and facing issues like privacy breaches and regulatory constraints. Federated learning (FL) offers a solution by using model transfer instead of data transfer to tackle these challenges. However, practical FL systems are confronted with the issue of malicious users attempting to deceive the server by uploading false local models, known as free-riding attacks. These attacks significantly undermine the fairness and effectiveness of FL. Current research assumes that free-riding attacks are limited to a small number of rational users. However, when there are multiple malicious free-riders, current research falls short in effectively detecting and defending against these attackers. To address this issue, we introduce a novel gradient backtracking based algorithm to identify free-riders. We introduce random testing rounds into standard FL and compare the similarity of user's gradient between the testing round and the comparison round. It overcomes the challenge of ineffective defense in scenarios involving multiple malicious free-riders. Experimental results on the MNIST and CIFAR-10 datasets demonstrate that the proposed detection algorithm achieves outstanding performance in various free-riding attack scenarios.

Key words federated learning; Internet of vehicles; free-riding attack; gradient similarity; free-riding attack detection

收稿日期: 2023-11-01; **修回日期**: 2024-05-20

基金项目: 国家自然科学基金项目(62072408, 62302454); 浙江省自然科学基金杰出青年科学基金项目 (LR24F020004); 浙江省自然科学基金重大项目(青年原创)(LDQ24F020001); 中国博士后科学基金项目(2023M743403)

This work was supported by the National Natural Science Foundation of China (62072408, 62302454), the Natural Science Foundation of Zhejiang Province for Distinguished Young Scholars (LR24F020004), the Major Program of the Natural Science Foundation of Zhejiang Province (Youth Original Project) (LDQ24F020001), and the China Postdoctoral Science Foundation (2023M743403).

通信作者: 伍一鸣(wyiming@zjut.edu.cn)

摘 要 随着车联网的发展，快速增长的智能汽车产生了海量的用户数据.这些海量的数据对训练智能化的车联网应用模型有极高的价值.传统的智能模型训练需要在云端集中式地收集原始数据，这将消耗大量通信资源并存在隐私泄露和监管限制等问题.联邦学习提供了一种模型传输代替数据传输的分布式训练范式用于解决此类问题.然而，在实际的联邦学习系统中，存在恶意用户通过伪造本地模型骗取服务器奖励的情况，即搭便车攻击.搭便车攻击严重破坏了联邦学习的公平性，影响联邦学习的训练效果.目前的研究假设搭便车攻击行为只存在于少量的理性用户中.然而，当存在多个恶意搭便车攻击者时，当前的研究无法有效地检测和防御这些攻击者.为此，提出了一种基于梯度回溯的搭便车攻击检测算法.该算法在正常的联邦学习中随机引入测试轮，通过对比单个用户在测试轮和基准轮模型梯度的相似度，解决了多个恶意搭便车用户场景中防御失效的问题.在 MNIST 和 CIFAR-10 数据集上的实验结果表明，提出的算法在多种搭便车攻击情境下都能实现出色的检测性能.

关键词 联邦学习；车联网；搭便车攻击；梯度相似度；搭便车攻击检测

中图法分类号 TP391

随着现代智能汽车的快速发展，以智能网联汽车为中心的车联网系统逐渐深入人们的生活.这产生了许多基于车联网系统的智能应用，例如自动驾驶、远程车辆管理和驾驶行为分析等.现代智能汽车配备了多功能传感器、计算设备和存储设备[1]，在日常使用中产生了海量的车联网数据，如摄像头视频流、雷达点云和车辆行驶记录等[2].这些数据具有重要的使用价值，可以利用这些海量的数据训练人工智能模型来进一步提高车联网智能化水平.传统的方法是将这些海量原始数据上传到云端，然后基于这些数据训练特定任务的机器学习模型，最后将模型部署到边缘设备.然而这种训练方式需要消耗大量通信带宽用于数据上传，而且存在着车辆和车主的个人隐私泄露、监管法规限制等不可忽视的问题[3-5].

联邦学习[6]作为一种新兴的分布式机器学习范式提供了上述问题的解决方案.在联邦学习中，智能汽车用户不需要直接上传海量的原始数据到车联网平台训练机器学习模型，而是使用本地的数据和计算资源独立地训练机器学习模型，在训练结束后上传本地训练的模型代替传统方法的原始数据.具体而言，车联网中联邦学习的训练由 3 部分构成[7]：车联网平台分发初始模型到每个智能汽车用户；智能汽车用户使用本地收集的真实世界数据对初始模型进行训练更新；训练完成后，每个智能汽车用户上传训练后的模型到车联网平台，平台聚合所有智能汽车用户上传的模型.整个联邦学习过程就是不断地重复上述 3 个步骤直到模型收敛或者精度达到预期要求.

联邦学习的隐私保护机制使得车联网平台在不暴露智能汽车用户隐私数据的前提下仍能训练全局的机器学习模型.车联网平台使用联邦学习进行智能模型训练时，为了激励更多的智能汽车用户参与联邦学习任务，通常会提供相应的奖励来弥补他们在模型训练时的资源消耗[8-9]，例如提供最终模型的使用权或给予金钱报酬等.然而，这种奖励形式可能导致部分智能汽车用户出现不诚信的欺诈行为.其中最容易实现的是搭便车攻击[10]，即不进行真实的本地数据训练，而是使用伪造的模型代替真实训练的模型骗取车联网平台的奖励[10-11].参与搭便车攻击的智能汽车用户没有相应的本地训练成本，但获得了与诚信智能汽车用户相同的奖励.这种欺诈行为会严重影响联邦学习在车联网应用中激励的公平性.更严重的是，恶意的搭便车用户可能导致其他智能汽车用户隐私信息的泄露，比如对全局模型进行模型反演[11].

搭便车问题引起了研究人员的广泛关注，当前的研究主要从激励和攻击检测 2 个角度展开.激励的方法通过设计有效的博弈模型，结合评估机制，使得贡献真实数据的用户将获得最佳收益，同时降低搭便车用户获得的收益，从而促使理性的用户放弃搭便车攻击行为[12].然而，当存在恶意的非理性用户时，激励的方法无法解决此类用户的搭便车行为.攻击检测大多基于离群值检测实现[10-11,13].通过对比当前轮次所有用户上传的模型，判定与多数模型差异较大的模型为离群值，从而确定搭便车攻击者.基于离群值检测的方法在应对多个搭便车用户时，无法区分离群值，导致检测效果欠佳.综上所述，当存在多个恶意搭便车攻击者同时参与联邦学习任务时，当前的研究不能有效地解决这个问题，这会严重影响使用联邦学习训练车联网应用模型的效果.

为了解决多个恶意搭便车用户攻击的问题，本

文提出了一种基于梯度回溯的联邦学习搭便车攻击检测算法. 车联网平台作为服务器首先确定 1 个训练轮为基准轮, 并保存智能汽车用户上传的模型梯度. 在后续的训练轮次中, 服务器将随机插入 1 个测试轮用于模型梯度的相似性检测. 在此测试轮中, 服务器将下发基准轮的初始全局模型, 即回溯模型, 并使用余弦相似度比较单个用户 2 次梯度更新的相似性. 根据余弦相似度的检测结果确定用户是否存在搭便车攻击行为. 当存在多个恶意搭便车用户时, 基于梯度回溯的算法仍然能精确地检测出搭便车攻击者.

本文的主要贡献包括 3 个方面:

1)提出基于梯度回溯的联邦学习搭便车攻击检测算法. 通过比较单个用户 2 次梯度更新的余弦相似度, 判断该用户是否为搭便车攻击者. 与现有工作相比, 提出的算法是基于单个用户在不同训练阶段的梯度更新对比实现的, 不受其他用户的影响. 因此能够在多个搭便车用户的场景中高效地检测恶意搭便车用户.

2)全面地分析了不同攻击场景中攻击者与诚信用户梯度更新相似性的差异, 并确定了诚信用户与搭便车用户梯度余弦相似度值的区分边界. 设置了合理的阈值, 能够精确检测出搭便车用户.

3)系统性地考虑了 4 种不同的搭便车攻击方法. 在 MNIST 数据集和 CIFAR-10 数据集上的评估结果表明, 我们的算法与基线算法 Delta-DAGMM[13] 相比, 总体 $F1$ 分数分别提高了 86.59 个百分点和 83.48 个百分点; 特别是在多个搭便车用户存在的情况下, 本文的算法实现了出色的检测成功率.

1 相关工作

近年来, 随着联邦学习研究的深入, 搭便车攻击在该领域也引起了研究人员的广泛关注. 搭便车攻击是指用户通过上传伪造的模型替代使用真实数据训练的模型, 以欺骗服务器获取奖励的行为. Lin 等人[10] 考虑了联邦学习中的搭便车攻击方法, 结合联邦学习的特点提出了 3 种攻击方法. Fraboni 等人[11] 基于 Lin 等人[10] 提出的攻击方法, 分析了搭便车攻击对全局模型的收敛性造成的影响, 并证明了在联邦学习中搭便车攻击不容易被服务器察觉. 防御搭便车攻击的研究主要集中于 3 个方面: 1)通过博弈论与激励机制的方法减少搭便车用户数量[9,12]; 2)从模型聚合策略的角度研究有效的防御方法[14-16]; 3)关注高效检测搭便车攻击者[10-11,13].

一些研究采用博弈论与激励机制的方法, 以减少搭便车攻击者的数量. Karimireddy 等人[12] 提出了一种基于用户贡献度的最优机制, 理性用户在这种机制下不会选择搭便车攻击以获取利益, 但其建立在用户数据成本已知的条件下. Zhang 等人[9] 将用户之间的关系建模为长期合作博弈的过程, 并引入惩罚策略. 他们证明了在此设置下存在最优的纳什均衡点, 可以激励用户增加数据贡献, 从而减少搭便车用户的数量. 虽然基于博弈论与激励机制的方法降低了理性用户参与搭便车行为的概率, 但在面对非理性的恶意攻击者时效果有限. 因此需要更有效的防御方法来解决这个问题.

修改模型的聚合策略被证明可以有效防御搭便车攻击. 文献 [14–15] 通过评估每轮用户的贡献, 分配不同精度的全局模型, 使搭便车用户难以获取与诚实用户相同的奖励. 然而, 这也可能影响到一些低质量的诚信用户, 导致他们获得较低的奖励. Yin 等人[16] 通过修改聚合策略, 防止恶意搭便车攻击者破坏全局模型, 但同时也影响了全局模型的精度. 这些防御方法降低了搭便车用户的奖励或影响, 但没有完全将攻击者消除.

当前的搭便车攻击检测通常基于模型参数离群值检测[10,13]. 机器学习和深度学习的模型参数通常是多维甚至高维的. Zong 等人[17] 提出了深度自编码高斯混合模型(DAGMM), 用于检测高维数据中的离群值. Lin 等人[10] 将 DAGMM 用于搭便车攻击检测, 在检测简单的攻击时取得了显著的效果, 但在其他高级攻击中的表现较差. 为此, 研究人员结合模型参数更新的标准偏差(standard deviation, STD)提出了 STD-DAGMM 算法, 这种算法能够在多种攻击中取得明显的检测效果. Huang 等人[13] 则将 DAGMM 直接与模型更新结合, 提出了 Delta-DAGMM 算法, 该算法与 DAGMM 相比取得了更高的检测精度. 文献 [10,13] 的算法需要使用每轮的模型参数训练自编码器模型. 然而, 当模型参数量巨大时, 会消耗大量的计算资源. 此外, 这些检测方法没有考虑在小规模联邦学习中由于用户数量有限、数据量不足以训练自编码器模型的问题. 尤其是当存在多个搭便车用户时, 离群值检测方法对搭便车攻击的检测效果会明显下降.

2 基础知识

在本节中, 主要介绍联邦学习中的搭便车攻击以及本文考虑的威胁模型.

2.1 联邦学习中的搭便车攻击

联邦学习通常包含1组用户 I 和1个中央服务器. 如图1所示，每个用户分别拥有 M_i 个样本组成的本地数据集 D_i. 联邦学习训练过程中，每个用户接收到服务器下发的全局模型后，使用其本地数据集 D_i 训练局部模型 $\boldsymbol{\theta}_i^t$，并将此局部模型上传至中央服务器，中央服务器负责聚合所有用户上传的局部模型，并生成新的全局模型 $\boldsymbol{\theta}^t$，这就完成了1轮联邦学习训练. 重复这个过程直到模型收敛，就完成了整个联邦学习训练. 本文采用了常用的FedAvg[18]聚合算法，该算法在模型聚合时要求用户提供其训练的数据量 M_i，服务器汇总所有用户提供的训练数据量，得到总训练数据量 $N=\sum_{i\in I}M_i$. 每轮模型聚合过程可以表示为

$$\boldsymbol{\theta}^t=\sum_{i\in I}\frac{M_i}{N}\boldsymbol{\theta}_i^t. \tag{1}$$

搭便车用户 a 没有使用本地数据进行训练，而是将伪造的模型 $\boldsymbol{\theta}_a^t$ 作为其训练后的局部模型，同时上传其真实拥有的数据量 M_a. 聚合后的全局模型为

$$\boldsymbol{\theta}^t=\sum_{i\in I/\{a\}}\frac{M_i}{N}\boldsymbol{\theta}_i^t+\frac{M_a}{N}\boldsymbol{\theta}_a^t. \tag{2}$$

当伪造的模型 $\boldsymbol{\theta}_a^t$ 与诚信用户的局部模型 $\boldsymbol{\theta}_i^t$ 相似时，服务器无法察觉用户 a 的搭便车行为，从而搭便车用户 a 将获得与诚信用户相同的奖励.

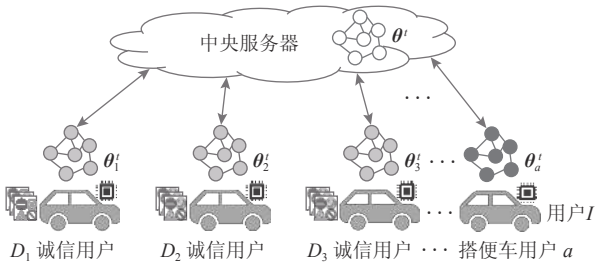

Fig. 1 Free-riding attack in federated learning

图1 联邦学习中的搭便车攻击

2.2 威胁模型

搭便车用户的目标是在不贡献本地数据和计算资源的情况下获取服务器的奖励. 因此，搭便车用户通常考虑以较低的计算成本实现较好的攻击效果. “较好的攻击效果”指在搭便车用户参与联邦学习时，全局模型的收敛性没有明显变化，服务器不易察觉. 当前研究主要涉及4种搭便车攻击方法，包括基础攻击[11]、伪装攻击[10]、Delta攻击以及高斯攻击[10,13]. 本文使用 H 表示诚信用户集合，K 表示搭便车用户集合. 下面将详细介绍这4种不同的攻击方式，各攻击方法的特点如表1所示.

Table 1 Characteristics of Different Attack Methods

表1 不同攻击方法的特点

攻击方法	特点
基础攻击	没有梯度更新，易被检测
伪装攻击	随机生成梯度，梯度变化无规律
Delta攻击	使用前一轮全局模型梯度，接近真实更新
高斯攻击	添加高斯噪声到Delta攻击，接近真实更新

1)基础攻击. 搭便车用户接收到全局模型 $\boldsymbol{\theta}^{t-1}$ 后，不进行任何修改，直接将 $\boldsymbol{\theta}^{t-1}$ 作为本地训练后的模型上传至服务器. 即 $\forall k\in K,\boldsymbol{\theta}_k^t=\boldsymbol{\theta}^{t-1}$，聚合后的全局模型表示为

$$\boldsymbol{\theta}^t=\sum_{j\in H}\frac{M_j}{N}\boldsymbol{\theta}_j^t+\sum_{k\in K}\frac{M_k}{N}\boldsymbol{\theta}^{t-1}. \tag{3}$$

由于该过程中模型未发生变化，因此服务器容易检测到用户的搭便车行为.

2)伪装攻击. 由于基础攻击没有产生梯度更新，因此容易被服务器察觉异常. 对此，搭便车用户考虑对全局模型 $\boldsymbol{\theta}^{t-1}$ 添加噪声扰动，使用随机噪声模拟梯度更新达到伪装的效果. 模型梯度中各元素的值是在有限域内变化的[10]，当噪声扰动超出此有限域时，可能导致梯度消失或梯度爆炸，从而严重影响全局模型的收敛性. 为了提高攻击效果，攻击者通常从均匀分布 $(-R,R)$ 中随机采样作为梯度各元素的值. 其中 R 表示梯度更新的阈值，与模型类型相关(例如，CNN与Resnet网络的梯度元素更新范围不同). 更新过程可以表示为

$$\forall k\in K,\boldsymbol{\theta}_k^t=\boldsymbol{\theta}^{t-1}+\boldsymbol{\phi},\boldsymbol{\phi}\sim\cup(-R,R), \tag{4}$$

聚合后的全局模型为

$$\boldsymbol{\theta}^t=\sum_{j\in H}\frac{M_j}{N}\boldsymbol{\theta}_j^t+\sum_{k\in K}\frac{M_k}{N}\left(\boldsymbol{\theta}^{t-1}+\boldsymbol{\phi}\right). \tag{5}$$

3)Delta攻击. 搭便车用户使用上一轮接收的全局模型与本轮接收的全局模型之间的差值，作为本地的模型梯度更新，以模拟真实的梯度下降过程. 本质上是将前一轮全局模型的梯度应用到当前训练轮的局部模型中，即 $\forall k\in K,\boldsymbol{\theta}_k^t=\boldsymbol{\theta}^{t-1}-(\boldsymbol{\theta}^{t-2}-\boldsymbol{\theta}^{t-1})$，聚合后的全局模型为

$$\boldsymbol{\theta}^t=\sum_{j\in H}\frac{M_j}{N}\boldsymbol{\theta}_j^t+\sum_{k\in K}\frac{M_k}{N}\left(\boldsymbol{\theta}^{t-1}-(\boldsymbol{\theta}^{t-2}-\boldsymbol{\theta}^{t-1})\right). \tag{6}$$

由于采用了真实的梯度更新，因此梯度的变化规律接近真实的模型训练，能够实现较好的攻击效果.

4)高斯攻击. 当有多个搭便车用户采用Delta攻击时，攻击者的梯度将完全一致. 为了避免多个用户产生相同的梯度，搭便车用户考虑在Delta攻击的基

础上添加均值为 0 且方差为σ^2的高斯噪声. 更新过程表示为

$$\forall k \in K, \boldsymbol{\theta}_k^t = \boldsymbol{\theta}^{t-1} - \left(\boldsymbol{\theta}^{t-2} - \boldsymbol{\theta}^{t-1} + \boldsymbol{\mu}\right), \boldsymbol{\mu} \sim N\left(0, \sigma^2\right). \quad (7)$$

高斯攻击生成的梯度各不相同, 聚合后的全局模型为

$$\boldsymbol{\theta}^t = \sum_{j \in H} \frac{M_j}{N} \boldsymbol{\theta}_j^t + \sum_{k \in K} \frac{M_k}{N} \left(\boldsymbol{\theta}^{t-1} - \left(\boldsymbol{\theta}^{t-2} - \boldsymbol{\theta}^{t-1} + \boldsymbol{\mu}\right)\right). \quad (8)$$

多个搭便车用户使用高斯攻击参与联邦学习训练时, 也能取得较好的攻击效果.

由于服务器无法直接访问用户的本地数据集, 因此无法判断用户上传的局部模型是否由本地数据集训练生成. 当搭便车用户使用上述 4 种攻击方式参与训练时, 由于对训练过程的收敛性影响很小, 因此服务器难以察觉. 没有检测方法参与时, 搭便车用户将获得与诚信用户相同的奖励, 这严重破坏了联邦学习的公平性.

本文考虑的攻击场景如下: 搭便车用户在训练开始时就使用搭便车攻击的方式参与训练, 在 1 个联邦学习任务中, 他们仅使用上述 4 种攻击方式中的 1 种进行搭便车攻击. 当存在多个搭便车用户时, 假设他们都将采用相同的搭便车攻击方式, 并且用户之间是独立的, 没有合作结盟关系.

3 方案设计

本节基于梯度回溯, 提出了一种用于检测上述 4 种搭便车攻击行为的算法. 该算法能够在多个搭便车用户参与的情境中准确识别搭便车用户, 从而保证联邦学习的公平性.

3.1 梯度回溯

本文将基于梯度回溯检测搭便车攻击者, 定义梯度回溯由以下 2 部分构成.

1)基准轮. 在联邦学习训练开始阶段, 随机选择 1 个全局模型未收敛的训练轮作为基准轮, 将此轮服务器下发的全局模型作为回溯模型, 定义此轮用户的梯度更新为基准梯度.

2)测试轮. 在基准轮的后续训练中, 随机插入 1 个测试轮, 在此轮服务器再次下发回溯模型, 定义用户在此轮对回溯模型的梯度更新为测试梯度.

设置回溯模型用于对比用户在不同训练轮对相同模型的梯度更新相似度. 联邦学习中用户对未收敛的全局模型梯度更新更明显, 因此选取训练开始阶段全局模型未收敛的训练轮为基准轮. 测试轮需要回溯模型作为下发模型, 因此将之设置于基准轮之后.

本文使用基于梯度回溯的方法检测搭便车攻击者时, 通过比较同一用户基准梯度与测试梯度的余弦相似度, 判断该用户是否为搭便车攻击者.

余弦相似度通常用于衡量向量的方向相似性, 通过计算 2 个向量之间夹角的余弦值来衡量它们的方向相似度, 与向量的幅值大小无关, 仅与向量的方向有关[19]. 余弦相似度的取值范围在 [–1, 1] 之间. 余弦相似度值越大, 表示 2 个向量之间的夹角越小, 方向越接近. 当余弦值为 0 时, 表示 2 个向量相互正交. 向量$\boldsymbol{V}_1, \boldsymbol{V}_2$的余弦相似度计算为

$$\cos(\boldsymbol{V}_1, \boldsymbol{V}_2) = \frac{\boldsymbol{V}_1 \cdot \boldsymbol{V}_2}{\|\boldsymbol{V}_1\| \times \|\boldsymbol{V}_2\|}, \quad (9)$$

其中$\|\cdot\|$表示向量的 L2 范数.

模型梯度是一个多维的张量, 为了准确衡量基准梯度与测试梯度的余弦相似度, 本文将此张量展开为一个多维向量以进行余弦相似度计算. 将 2 次梯度更新$\boldsymbol{g}_i^r$, $\boldsymbol{g}_i^c$展开为向量$\boldsymbol{V}_i^r, \boldsymbol{V}_i^c$. 根据式(9)计算 2 次梯度更新余弦相似度$\cos(\boldsymbol{g}_i^r, \boldsymbol{g}_i^c) = \cos\left(\boldsymbol{V}_i^r, \boldsymbol{V}_i^c\right)$.

3.2 算法概述

基于梯度回溯的搭便车攻击检测算法整体流程如图 2 所示, 算法核心思想是在正常的联邦学习过程中, 引入测试轮检测用户是否存在搭便车行为.

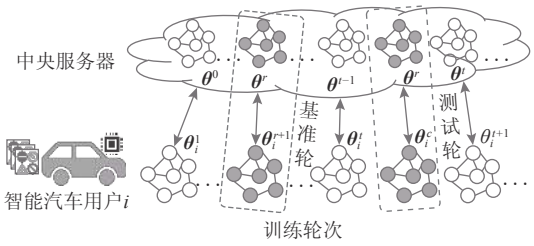

Fig. 2 Overview of free-riders detection

图 2 搭便车用户检测流程

联邦学习开始时, 中央服务器向所有参与联邦学习任务的智能汽车用户发布初始模型$\boldsymbol{\theta}^0$, 用户使用私有数据进行模型训练. 联邦学习开始后, 服务器保存第 r 轮下发的全局模型$\boldsymbol{\theta}^r$作为回溯模型. 当所有用户完成本地训练后, 服务器记录用户返回的局部模型$\boldsymbol{\theta}^{r+1}$, 计算得到用户此轮梯度更新$\boldsymbol{g}_i^r = \boldsymbol{\theta}^r - \boldsymbol{\theta}_i^{r+1}$. 服务器将梯度更新$\boldsymbol{g}_i^r$记为基准梯度并保存. 后续训练过程中, 服务器随机插入测试轮, 再次下发保存的回溯模型$\boldsymbol{\theta}^r$. 服务器在测试轮收集用户 i 基于回溯模型$\boldsymbol{\theta}^r$训练的局部模型$\boldsymbol{\theta}_i^c$, 计算得到测试梯度更新$\boldsymbol{g}_i^c = \boldsymbol{\theta}^r - \boldsymbol{\theta}_i^c$并保存. 测试轮上传的局部模型仅用于检

测用户的搭便车行为, 服务器不进行模型聚合. 当测试轮通信结束后, 服务器下发测试轮之前聚合的全局模型, 即图 2 中的$\boldsymbol{\theta}^t$, 继续正常的联邦学习模型训练.

在测试轮中, 服务器将用户 i 的基准梯度$\boldsymbol{g}_i^r$与测试梯度$\boldsymbol{g}_i^c$进行余弦相似度计算, 并将余弦相似度值与预设阈值 Th 比较. 当余弦相似度值大于 Th 时, 该用户被认为是诚信用户; 当余弦相似度值小于 Th 时, 该用户被判定为搭便车用户. 具体检测算法如算法 1 所示.

算法 1. 基于梯度回溯的搭便车检测算法.

输入: 初始全局模型$\boldsymbol{\theta}^0$,训练轮次 t, 基准轮次 r, 测试轮次 c, 阈值 Th, 用户集合 I={1, 2, ⋯, N}, 各用户提供的数据量M_i;

输出: 搭便车攻击用户集合 A.

① for each round t

② if $t=r$

③ 下发全局模型$\boldsymbol{\theta}^r$至 I 中各用户;

④ 收集基准轮的局部模型, 并计算梯度

$\vartheta_{\text{client}}^{r+1} \leftarrow \left\{\boldsymbol{\theta}_1^{r+1}, \boldsymbol{\theta}_2^{r+1}, \cdots, \boldsymbol{\theta}_N^{r+1}\right\}$,

$G_{\text{client}}^{r} \leftarrow \left\{\boldsymbol{g}_1^{r}, \boldsymbol{g}_2^{r}, \cdots, \boldsymbol{g}_N^{r}\right\}$;

⑤ else if $t=c$

⑥ 下发回溯模型$\boldsymbol{\theta}^r$至 I 中各用户;

⑦ 收集用户测试轮的局部模型并计算梯度

$\vartheta_{\text{client}}^{c} \leftarrow \left\{\boldsymbol{\theta}_1^{c}, \boldsymbol{\theta}_2^{c}, \cdots, \boldsymbol{\theta}_N^{c}\right\}$,

$G_{\text{client}}^{c} \leftarrow \left\{\boldsymbol{g}_1^{c}, \boldsymbol{g}_2^{c}, \cdots, \boldsymbol{g}_N^{c}\right\}$;

⑧ 计算基准梯度和测试梯度的余弦相似度

$S \leftarrow \left\{\cos\left(\boldsymbol{g}_i^r, \boldsymbol{g}_i^c\right)\right\}, i \in I$;

⑨ for s_i in S

⑩ if $s_i < Th$

⑪ 记录余弦相似度小于阈值的用户为搭便车用户$A \leftarrow A \cup \{i\}$;

⑫ end if

⑬ end for

⑭ 下发全局模型$\boldsymbol{\theta}^t$至各用户进行正常模型训练并收集用户梯度更新$\vartheta_{\text{client}}^{t+1} \leftarrow \left\{\boldsymbol{\theta}_1^{t+1}, \boldsymbol{\theta}_2^{t+1}, \cdots, \boldsymbol{\theta}_N^{t+1}\right\}$;

⑮ else

⑯ 下发全局模型$\boldsymbol{\theta}^t$至各用户;

⑰ $\vartheta_{\text{client}}^{t+1} \leftarrow \left\{\boldsymbol{\theta}_1^{t+1}, \boldsymbol{\theta}_2^{t+1}, \cdots, \boldsymbol{\theta}_N^{t+1}\right\}$;

⑱ end if

⑲ 聚合所有局部模型$\boldsymbol{\theta}^{t+1} = \sum_{i \in I} \frac{M_i}{N} \boldsymbol{\theta}_i^{t+1}$;

⑳ end for

3.3 算法分析

诚信用户使用本地私有数据进行正常的模型训练, 因此在基准轮和测试轮, 模型使用相同的数据训练. 不同训练轮次, 模型训练的随机参数不同, 因此对于同一初始模型的梯度更新并不完全相同. 但 2 轮更新都向损失值减小的方向更新. 如图 3 所示, 尽管基准轮和测试轮的梯度方向并不完全一致, 但它们都朝着损失值降低的方向更新, 因此诚信用户的 2 次梯度更新在方向上具有相似性. 在后续 4.2 节的实验中, 诚信用户的模型更新余弦相似度都位于大于 0.1 的区间内.

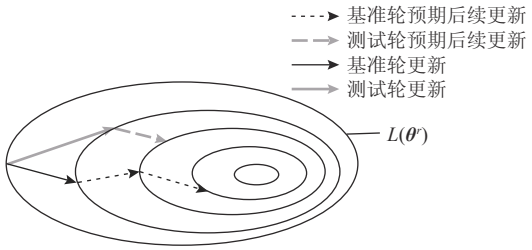

Fig. 3 Direction of the model update in honest users

图 3 诚信用户模型更新方向

攻击者将采用伪造模型的方法参与联邦学习训练, 伪造模型是基于正常联邦学习的梯度更新规律设计的. 对用户而言, 当测试轮使用回溯模型作为初始化全局模型时, 联邦学习的梯度更新规律就被改变了. 因此, 攻击者在测试轮对回溯模型的更新与正常联邦学习训练不同, 在测试轮产生的梯度与基准轮产生的梯度差异较大. 下面将详细分析不同攻击方式下, 搭便车用户与诚信用户在基准轮和测试轮模型梯度相似性的区别.

搭便车用户采用基础攻击时, 由于每一轮都将全局模型直接上传, 因此每一轮的梯度$\boldsymbol{g}=\boldsymbol{\theta}^t-\boldsymbol{\theta}^t=\boldsymbol{0}$. 因此梯度的 L2 范数始终为 0, 这将无法使用式(9)计算余弦相似度. 然而, 由于梯度计算在中央服务器端完成, 本文在实际操作中将无法计算余弦相似度的值标记为 0. 这使得余弦相似度与诚信用户有所不同, 从而识别使用基础攻击的搭便车用户.

搭便车用户采用伪装攻击时, 他们对每个模型参数都添加了独立采样的随机噪声, 计算得到梯度$\boldsymbol{g}=\boldsymbol{\phi}$. 梯度展开后是一个由随机噪声构成的高维向量, 其中向量的每个元素都是从均匀分布($-R$, R)中独立采样得到的. 文献 [20–21] 对于高维随机向量的正交性进行了深入的探索, 并指出 2 个随机独立采样得到的高维向量接近正交. 由于机器学习模型中的梯度参数高达数千甚至数万维, 因此, 采用伪装攻

击生成的梯度余弦相似度接近 0. 为了验证这一理论，本文在不同区间的均匀分布中，R=10, 1, 0.1, 0.01, 0.001 随机采样生成了 100, 1 000, 10 000 维的随机向量. 在每种设置中生成了 1 000 对随机向量，并计算每对向量的余弦相似度，结果如图 4 所示. 向量的维度越高，其余弦相似度值接近 0 的概率越大，与上述理论相符. 4.2 节的实验结果进一步证明了可以用余弦相似度值区分采用伪装攻击的搭便车用户与诚信用户.

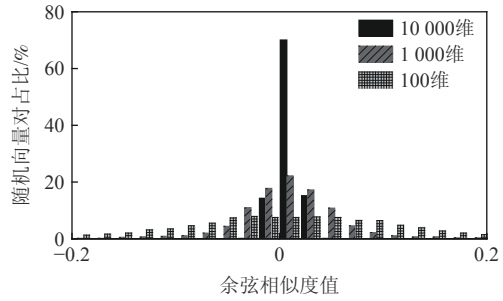

Fig. 4 Cosine similarity distribution of randomly sampled vectors

图 4 随机采样向量的余弦相似度分布

采用 Delta 攻击时，搭便车用户的梯度是根据当前训练轮与前一轮下发的全局模型确定的. 在基准轮，用户的梯度更新为 $\boldsymbol{\theta}^{r-1}-\boldsymbol{\theta}^{r}$. 由于 $\boldsymbol{\theta}^{r-1}$ 与 $\boldsymbol{\theta}^{r}$ 都是全局模型，因此 $\boldsymbol{\theta}^{r-1}-\boldsymbol{\theta}^{r}$ 遵循正常梯度下降的更新规律，即基准轮模型更新的方向是模型损失值下降的方向. 在测试轮将回溯模型再次下发时，用户梯度更新为 $\boldsymbol{\theta}^{t-1}-\boldsymbol{\theta}^{r}$. 然而，回溯模型 $\boldsymbol{\theta}^{r}$ 是基准轮的全局模型，因此其模型损失值 $L(\boldsymbol{\theta}^{r})$ 大于 $\boldsymbol{\theta}^{t-1}$ 的模型损失值 $L(\boldsymbol{\theta}^{t-1})$，这使得用户在测试轮的梯度方向是损失值增加的方向. 图 5 演示了 2 次梯度更新的区别，由于 2 次梯度更新的方向相反，因此余弦相似度值更接近负值，这与诚信用户有明显区别.

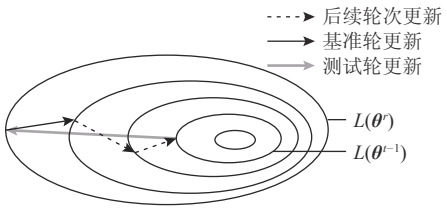

Fig. 5 Direction of model update in free-riders using Delta attack

图 5 采用 Delta 攻击的搭便车用户模型更新方向

类似于 Delta 攻击的结果，高斯攻击在 Delta 梯度的基础上添加了高斯噪声. 添加高斯噪声是为了防止使用相同攻击方式的 2 个搭便车用户生成相同的梯度. 确定高斯噪声的大小时，攻击者考虑对训练收敛性影响较小的高斯噪声，因此，高斯攻击产生的梯度更新与 Delta 攻击是近似相等的. 与 Delta 攻击类似，高斯攻击 2 次模型更新的方向相似性也接近相反，导致余弦相似度值更接近负值.

综合上述 4 种攻击分析，在不同攻击方法中，诚信用户的余弦相似度值与搭便车用户的余弦相似度值都有明显的区分边界，可以设置合适的阈值作为边界值来检测用户的搭便车行为. 由于该算法是基于单个用户的时序梯度相似性比较，无需考虑其他参与训练的用户表现. 因此，算法的有效性与搭便车用户的数量无关.

3.4 收敛性分析

如算法 1 所示，本文提出的算法仅在正常联邦学习中加入了 1 个测试轮，没有改变联邦学习过程的其他环节. 当检测算法启动时，服务器在测试轮下发回溯模型，服务器并未聚合基于该模型更新的局部模型，仅用于梯度计算与相似性比较. 检测结束后，服务器在该轮下发测试轮前正常聚合全局模型，并聚合用户对该模型更新的局部模型用于后续正常的联邦学习训练. 因此，后续的联邦学习过程与未添加测试轮的联邦学习一致. 综上所述，基于梯度回溯的搭便车攻击检测算法不会改变联邦学习的收敛性，模型训练能够保持正常的收敛状态.

4 实验与结果

4.1 实验设置

本文在 MNIST 数据集和 CIFAR-10 数据集中进行实验，分别使用逻辑回归模型（Logistic[22]）和残差网络（Resnet56[23]）进行模型训练. MNIST 是包含了 10 个类别、60 000 张训练数据和 10 000 张测试数据的手写数字集，数据类型为 28×28 的灰度图[24]. CIFAR-10 是一种自然图片数据集，包含来自 10 个不同类别的 60 000 张彩色图片，每个类别都包含 6 000 张图片. 图片尺寸为 3×32×32，其中 50 000 张作为训练集，10 000 张作为测试集[25]. 逻辑回归和残差网络模型常用于分类任务，可以使用梯度下降方法进行训练.

本文使用 FedML[26] 联邦学习框架进行联邦学习训练，并模拟搭便车攻击的情况. 为了接近现实世界的数据分布的情况，本文在独立同分布（independently identically distributed, IID）和非独立同分布（non-independently identically distributed, non-IID）这 2 种数据分布场景下进行实验. 本文使用狄利克雷分布[27]

的方法实现 non-IID 场景，将完整数据集洗牌后发送至参与训练的用户并设置狄利克雷分布系数$\alpha=0.1$. 为了验证不同数量用户和不同比例搭便车攻击者对实验的影响，本文在 20 个用户和 50 个用户的场景中进行实验，并在 20 个用户场景中设置了 1, 10, 18 个搭便车攻击者，在 50 个用户场景中设置了 1, 25, 40 个搭便车攻击者.

搭便车攻击者以不贡献实际训练资源骗取奖励为目标，因此研究普遍考虑在训练开始阶段开始攻击行为[10-11,13]. 为了达到较好的检测效果，本文设置模型未收敛的第 6 轮作为基准轮. 算法评估中，使用 Delta 攻击和高斯攻击时需要正常训练至少 1 轮. 因此，与其他研究评估方法类似，本文设置第 5 轮开始搭便车攻击，并在基准轮后随机设置测试轮进行实验. 为了评估检测的性能，每次实验都选择不同的测试轮，并记录检测结果. 本文选择搭便车检测算法 Delta-DAGMM[13] 作为基线算法，在相同的实验设置下进行了对比分析.

本文的实验在 Linux 服务器上进行，硬件配置为 40 核 4.0 GHz Intel Xeon CPU 和 RTX3070Ti 显卡. 本文采用 Pytorch 作为机器学习训练库，基于 Python3 实现本文攻击与检测. 在实验中，使用 MNIST 数据集训练逻辑回归模型，使用 CIFAR-10 数据集训练 Resnet 模型，分别进行了 200 轮迭代和 400 轮迭代. 每当用户完成 1 个 epoch 训练后，便提交 1 次模型作为 1 轮联邦学习训练.

4.2 攻击参数选择

伪装攻击与高斯攻击的参数 R 和σ^2需要进行合理的设置，以保证模型训练的收敛性. 在参数选择时，本文选择 20 个用户参与联邦学习任务，并设定 1 个搭便车用户，分别采用不同 R 和σ^2在 IID 和 non-IID 环境下实验. 比较不同 R 和σ^2产生的最终全局模型的精度，确定最佳参数值进行后续实验.

当 R 取 0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1 时，训练后的模型精度如图 6 所示. 在 MNIST 数据集中，当 R=1 时，IID 环境下和 non-IID 环境下模型的精度与 R=0.1 时相比发生明显下降. 在 CIFAR-10 数据集中，当 R=10^{-3} 时，IID 环境下和 non-IID 环境下模型的精度与 R=10^{-4} 时相比发生明显下降. 因此，本文选择表现较好的 R=0.1 作为 MNIST 数据集的攻击参数，R=10^{-4} 作为 CIFAR-10 数据集的攻击参数.

图 7 是当σ^2取 0, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.2 时全局模型的精度变化情况. 在 MNIST 数据集中，当σ^2=0.1 时，IID 场景下全局模型精度明显降低；当σ^2=0.2 时，non-IID 场景下的全局模型也发生明显下降. 综合不同分布场景的变化趋势，本文取σ^2=10^{-2} 作为 MNIST 数据集的攻击参数. 同理取σ^2=10^{-4} 作为 CIFAR-10 数据集的攻击参数.

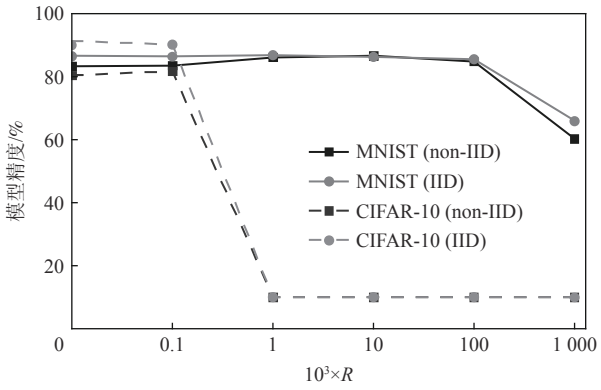

Fig. 6 Curve of model accuracy varying with R

图 6 模型精度随 R 值的变化曲线

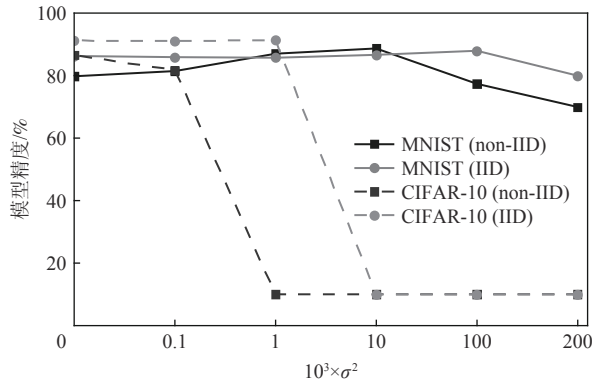

Fig. 7 Curve of model accuracy varying with σ^2

图 7 模型精度随σ^2的变化曲线

4.3 阈值确定

为了证明诚信用户与搭便车用户在基准轮和测试轮模型梯度的余弦相似度不同. 本文在 MNIST 数据集的场景下，选取 20 个用户参与联邦学习并随机指定 10 个用户为搭便车用户，重复 200 次实验，每次选取不同轮次作为测试轮，记录用户在每次实验中基准梯度与测试梯度的余弦相似度. 结果如图 8 所示.

图 8 直观地反映了在不同攻击方式下诚信用户和搭便车用户的余弦相似度值分布情况. 诚信用户基准梯度与测试梯度的余弦相似度值在 non-IID 场景中主要分布在 (0.3, 0.6] 区间内，在 IID 场景中主要分布在 (0.1, 0.3] 区间内.

搭便车用户采用伪装攻击的结果如图 8(a)所示，余弦相似度值主要分布在 (–0.1, 0.1] 区间内，IID 与 non-IID 中值的分布没有差异. 从图 8(a)中可以明显观察到搭便车用户与诚信用户余弦相似度值分布的

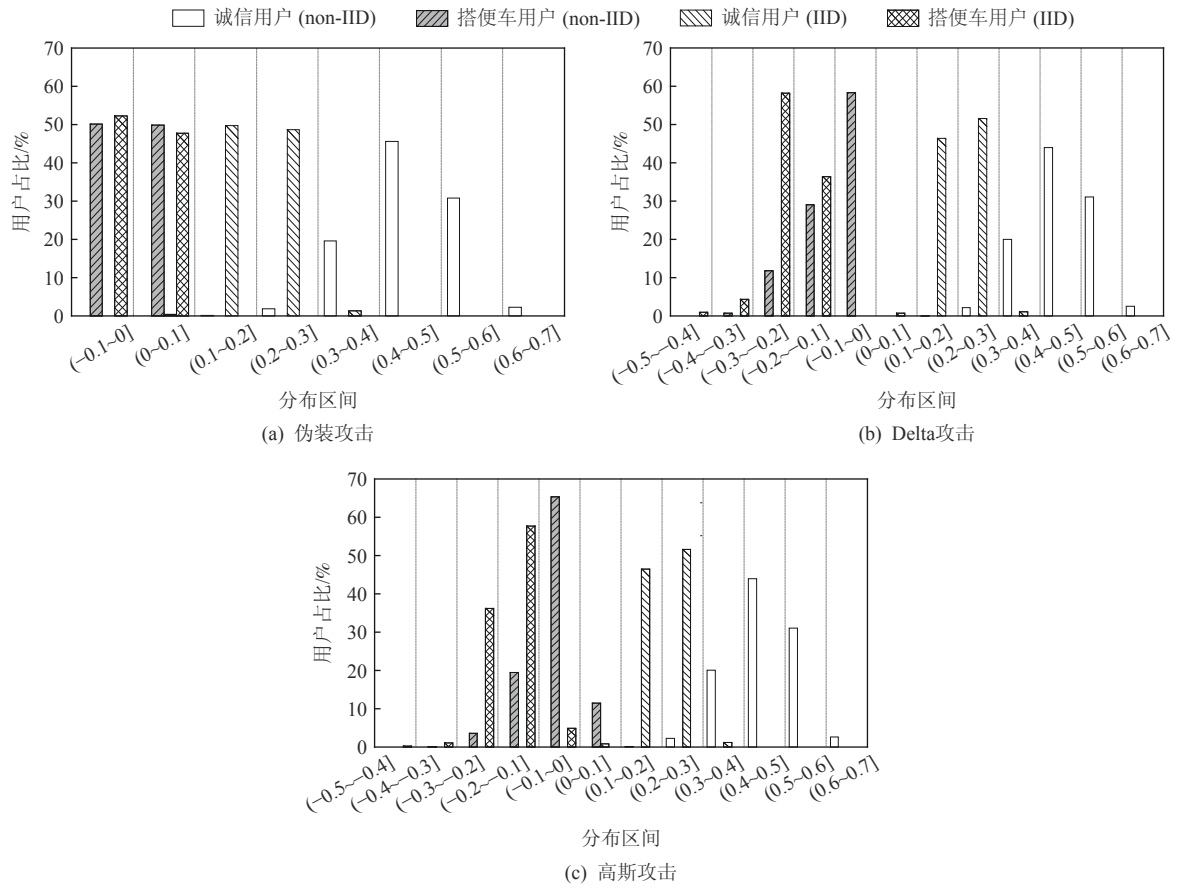

Fig. 8 Distribution of model update cosine similarity values under different attacks

图 8 不同攻击方式下模型更新余弦相似度值的分布

区别，诚信用户的余弦相似度都分布在大于 0.1 的区间，而搭便车用户的余弦相似度都分布在小于 0.1 的区间. 图 8(b)演示了采用 Delta 攻击的结果. 与伪装攻击结果类似，采用 Delta 攻击方式的搭便车用户的余弦相似度值分布在 (–0.3, 0] 区间内，可以明显观察到与诚信用户的余弦相似度值的区别. 与伪装攻击相比，使用 Delta 攻击的搭便车用户与诚信用户的余弦相似度区分边界更明显. 高斯攻击的结果如图 8(c)所示，受高斯噪声影响，与 Delta 攻击相比，采用高斯攻击的搭便车用户余弦相似度值的分布向右偏移. 在 non-IID 场景中小部分搭便车用户的余弦相似度值偏移至 (0, 0.1] 区间内. 总体而言，诚信用户与搭便车用户仍存在明显的区分边界.

综合上述 3 种搭便车用户余弦相似度值的分布，结合 3.3 节的算法分析，本文确定将阈值 Th=0.1 作为 MNIST 数据集场景下搭便车用户与诚信用户的区分边界. 本文也将该阈值应用到 CIFAR-10 数据集场景中，在后续的检测实验中将采用该阈值识别攻击者.

4.4 结果分析

为了评估检测算法在不同用户数量和不同比例攻击者场景中的检测性能，本文根据 4.2 节中确定的攻击参数和 4.3 节中确定的阈值进行实验，并与基线算法 Delta-DAGMM 进行对比. 本文使用检测成功率(detection success rate, *DS*)、误报率(false positive rate, *FR*)和 *F*1 分数作为评估指标，全面评估算法的检测性能. 定义搭便车用户被成功检测的比例为算法的检测成功率(*DS*)：

$$DS = \frac{\text{被检测到的搭便车用户数量}}{\text{总搭便车用户数量}} \times 100\%. \quad (10)$$

定义诚信用户被检测为搭便车用户的比例为算法的误报率：

$$FR = \frac{\text{诚信用户被检测为搭便车数量}}{\text{总诚信用户数量}} \times 100\%. \quad (11)$$

*F*1 分数可以用来衡量检测算法的总体表现，是分类模型中常用的评估指标，本文按 *F*1 分数的常规定义计算：

$$F1 = \frac{2 \times \frac{TP}{TP+FP} \times \frac{TP}{TP+FN}}{\frac{TP}{TP+FP} + \frac{TP}{TP+FN}} \times 100\%, \quad (12)$$

其中 TP 表示被成功检测的搭便车用户数量，FP 表

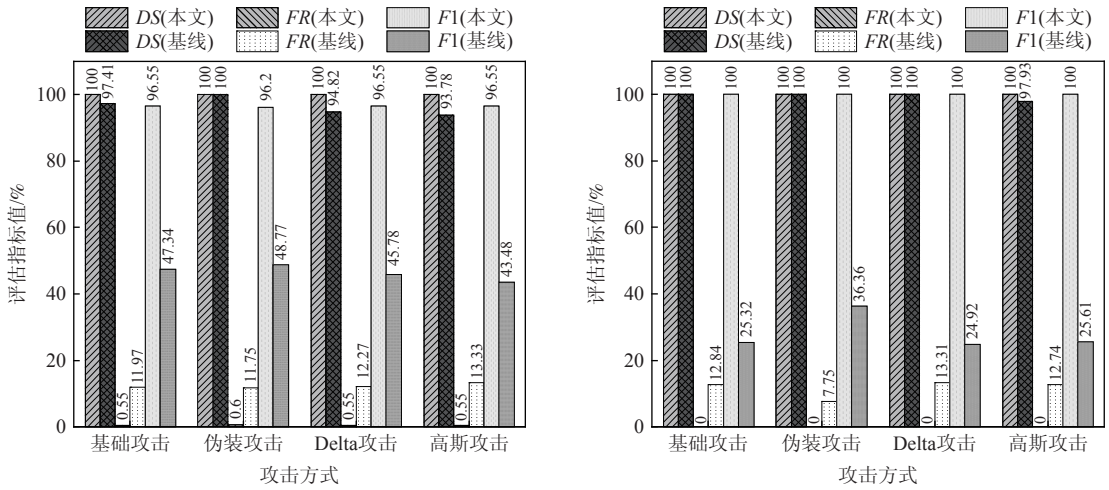

Fig. 9 Detection effects of a single free-riders on MNIST dataset in IID

图 9 MNIST 数据集单个搭便车用户在 IID 场景下的检测效果

示将诚信用户检测为搭便车用户的数量, *FN* 表示搭便车用户未被检测到的数量. 总体而言, 算法的检测效果越好, *DS* 越大, *FR* 越小, *F*1 分数越大. 在后续评估中, 将同一场景中不同测试轮的平均值作为该场景下的评估值.

图 9 是单个搭便车用户在 MNIST 数据集的 IID 场景下进行实验的评估结果. 20 个用户参与的场景中, 对使用 4 种不同攻击方法的搭便车检测结果如图 9(a)所示. 本文算法与基线算法 Delta-DAGMM 都到达了接近 100 个百分点的检测成功率. 但在误报率指标上, 本文算法在 4 种攻击方式下比基线算法分别低 11.42 个百分点、11.15 个百分点、11.72 个百分点、12.78 个百分点, 效果更好. *F*1 分数方面, 本文算法在 4 种攻击下的平均 *F*1 分数高达 96.46%, 对比基线算法的 46.34% 提高了 50.12 个百分点. 本文算法的总体表现更优.

图 9(b)是用户规模扩大至 50 的实验结果, 与 20 个用户参与时类似, 2 种检测算法在检测成功率上表现优异, 在大部分攻击方式下都达到了 100% 的检测成功率. 误报率指标上, 本文算法在 4 种攻击方式下都没有将诚信用户误检测为搭便车用户. 基线算法的平均误报率为 11.66%, 与 20 个用户参与时的平均值 12.33% 相比下降了 0.67 个百分点, 但仍高于本文算法的误报率. 本文算法在 *F*1 分数指标上达到了 100%, 比 20 用户参与时提高了 3.54 个百分点. 基线算法的平均 *F*1 分数为 28.07%, 远低于本文算法.

综合 2 种不同规模场景中的算法表现, 当仅有 1 个搭便车用户参与时, 本文算法具有较高的检测成功率, 能够高精度地检测出所有搭便车用户且误报率较低, 不易将诚信用户检测为搭便车用户. 与基线算法相比, 本文算法的误报率平均下降了 11.72 个百分点, *F*1 分数平均提高了 61 个百分点, 在检测成功率方面保持优异性能, 全面提升了算法的检测性能.

IID 场景下多个搭便车用户参与的检测结果如表 2 所示. 本文算法在不同攻击下都达到了接近 100% 的检测成功率. 20 个用户参与的场景中, 在多种攻击、不同比例攻击者的情况下平均误报率为 0.46%, 比基线算法 Delta-DAGMM 低 15.68 个百分点. 算法的 *F*1 分数在所有场景中都超过了 96%, 总体检测效果优异. 从表 2 中可以观察到当搭便车用户占比大于 50% 时, 基线算法 Delta-DAGMM 的检测效果严重下降, 无法精确检测搭便车用户. 50 个用户参与的场景中, 本文算法的检测成功率和 *F*1 分数都达到 100%, 误报率都为 0. 总体而言, 本文的算法在多个搭便车用户的场景下, 仍可以取得与单个搭便车用户相当的检测效果, 在所有场景中都取得优异的检测性能. 与基线算法对比, 本文算法可以弥补基线算法在多个搭便车用户场景下检测失效的不足, 同时降低误报率.

在实际的车联网系统中, 用户的数据往往是 non-IID 分布的[27], 因此本文也在 non-IID 场景中对算法的检测效果进行评估. 如表 3 所示, 在 non-IID 场景中, 本文算法在不同比例搭便车用户参与时, 都未出现将诚信用户误报为搭便车攻击者的情况, 且在 22 个场景中取得了 100% 的检测成功率和 100% 的 *F*1 分数. 当有 40 个搭便车用户采用高斯攻击方式时, 本文算法的检测效果最差, 但仍达到 99.43% 的检测成功率和 99.71% 的 *F*1 分数. 与 IID 场景相比, 本文算法

Table 2 Detection Results of Multiple Free-riders Participation under MNIST Dataset in IID

表 2 在 MNIST 数据集 IID 场景中多个搭便车用户参与的检测结果 %

攻击方式	评估指标	20 个用户参与		50 个用户参与	
		10 攻击	18 攻击	25 攻击	40 攻击
基础攻击	*DS*（基线）	0	0	0	5.49
	DS（本文）	100.00	100.00	100.00	100.00
	FR（基线）	10.98	0	18.72	93.16
	FR（本文）	0.78	0.26	0.00	0.00
	*F*1（基线）	0	0	0	7.53
	*F*1（本文）	99.63	99.99	100.00	100.00
伪装攻击	*DS*（基线）	0.47	0.49	2.63	1.01
	DS（本文）	100.00	100.00	100.00	100.00
	FR（基线）	0	0	0	0
	FR（本文）	0.31	0.26	0.00	0.00
	*F*1（基线）	0.85	0.91	5.02	1.97
	*F*1（本文）	99.85	99.86	100.00	100.00
Delta 攻击	*DS*（基线）	0	0	0	10.51
	DS（本文）	100.00	100.00	100.00	100.00
	FR（基线）	12.33	0	20.10	98.39
	FR（本文）	0.78	0.26	0.00	0.00
	*F*1（基线）	0	0	0	14.48
	*F*1（本文）	99.63	99.86	100.00	100.00
高斯攻击	*DS*（基线）	3.21	4.92	5.68	5.09
	DS（本文）	100.00	99.91	100.00	100.00
	FR（基线）	0.05	0.78	0	0.10
	FR（本文）	0.78	0.26	0.00	0.00
	*F*1（基线）	5.64	8.75	10.51	9.51
	*F*1（本文）	99.63	99.94	100.00	100.00

在 non-IID 场景下的检测效果更优. 与基线算法相比, 算法的平均检测成功率提高了 67.98 个百分点, 平均误报率降低了 8.48 个百分点, 平均 *F*1 分数提升了 83.35 个百分点.

为了全面充分评估本文算法的有效性, 还使用了更加复杂的 CIFAR-10 数据集对算法的有效性进行评估. 在 IID 与 non-IID 场景中评估的结果如表 4 所示.

与 MNIST 数据集中的表现一致, 本文算法在 CIFAR-10 数据集中的检测效果仍然优异. 在所有场景中, 本文算法都取得了 100% 的检测成功率, 并且误报率为 0%, *F*1 分数达到 100%. 与基线算法相比, 平均检测成功率提高了 66.02 个百分点, 误报率降低了 33.61 个百分点, *F*1 分数提高了 83.48 个百分点.

Table 3 Detection Results of MNIST Dataset in non-IID

表 3 在 MNIST 数据集 non-IID 场景的检测结果 %

攻击方式	评估指标	20 个用户参与			50 个用户参与		
		1 攻击	10 攻击	18 攻击	1 攻击	25 攻击	40 攻击
基础攻击	*DS*（基线）	92.23	0	0	97.93	0	2.33
	DS（本文）	100.00	100.00	100.00	100.00	100.00	100.00
	FR（基线）	9.11	9.22	0	8.86	11.75	92.54
	FR（本文）	0	0	0	0	0	0
	*F*1（基线）	51.17	0	0	33.89	0	3.49
	*F*1（本文）	100.00	100.00	100.00	100.00	100.00	100.00
伪装攻击	*DS*（基线）	95.85	0.21	0.43	100	2.86	1.02
	DS（本文）	100.00	100.00	100.00	100.00	100.00	100.00
	FR（基线）	8.78	0	0	4.81	0	0
	FR（本文）	0	0	0	0	0	0
	*F*1（基线）	54.09	0.38	0.80	48.58	5.46	2.00
	*F*1（本文）	100.00	100.00	100.00	100.00	100.00	100.00
Delta 攻击	*DS*（基线）	91.71	0	0	98.96	0	1.02
	DS（本文）	100.00	100.00	100.00	100.00	100.00	100.00
	FR（基线）	9.05	7.46	0	9.37	12.39	0
	FR（本文）	0	0	0	0	0	0
	*F*1（基线）	51.40	0	0	32.62	0	2.00
	*F*1（本文）	100.00	100.00	100.00	100.00	100.00	100.00
高斯攻击	*DS*（基线）	74.09	4.15	6.10	87.05	6.05	5.78
	DS（本文）	100.00	100.00	99.94	100.00	100.00	99.43
	FR（基线）	8.75	0	2.33	8.37	0.12	0.52
	FR（本文）	0	0	0	0	0	0
	*F*1（基线）	41.85	7.20	10.85	31.91	11.14	10.70
	*F*1（本文）	100.00	100.00	99.97	100.00	100.00	99.71

由于本文使用了 Resnet 模型训练 CIFAR-10 数据集, 因此模型更加复杂, 这使得模型梯度的维度更广. 使用本文算法比较梯度回溯的相似度时, 诚信用户和搭便车用户的相似度区别更明显. 因此, 本文算法在 CIFAR-10 数据集中的表现更加优异.

综合算法在 2 个数据集中的表现, 与基线算法相比, 本文算法在多种场景中都取得了明显的优势, 且检测效果没有明显波动, 表现出了优异的稳定性. 在更接近现实的 non-IID 场景和更加复杂的 CIFAR-10 数据集中, 本文算法表现更加优异, 能够适用于车联网场景中联邦学习搭便车检测任务.

5 总 结

搭便车攻击极大地破坏了联邦学习的公平性, 是阻碍联邦学习在车联网系统中应用的重要瓶颈之

Table 4 Detection Result of Multiple Free-riders Participation under CIFAR-10 Dataset in IID and non-IID

表 4 CIFAR-10 数据集在 IID 和 non-IID 场景中多个搭便车用户参与的检测结果 %

攻击方式	评估指标	IID						non-IID					
		20 个用户参与			50 个用户参与			20 个用户参与			50 个用户参与		
		1 攻击	10 攻击	18 攻击	1 攻击	25 攻击	40 攻击	1 攻击	10 攻击	18 攻击	1 攻击	25 攻击	40 攻击
基础攻击	*DS*（基线）	93.00	4.20	0	98.25	6.51	8.36	92.00	6.22	0.18	96.50	1.15	7.90
	DS（本文）	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	FR（基线）	15.00	36.98	0	14.49	36.24	89.42	8.59	27.47	0.12	12.98	34.74	75.65
	FR（本文）	0	0	0	0	0	0	0	0	0	0	0	0
	*F*1（基线）	38.91	5.95	0	21.63	9.12	12.79	51.79	9.31	0.36	23.18	1.69	12.46
	*F*1（本文）	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
伪装攻击	*DS*（基线）	93.75	9.40	11.01	96.63	8.13	11.24	94.25	7.00	9.90	88.18	0.65	9.04
	DS（本文）	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	FR（基线）	14.74	32.07	49.25	14.37	25.08	74.62	8.62	27.95	52.88	13.55	32.33	73.01
	FR（本文）	0	0	0	0	0	0	0	0	0	0	0	0
	*F*1（基线）	39.58	13.29	18.91	21.46	12.21	17.31	52.65	10.37	17.11	20.69	0.98	14.20
	*F*1（本文）	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Delta 攻击	*DS*（基线）	93.03	5.59	0.59	97.57	7.77	9.63	69.98	6.16	0.43	84.49	1.55	8.02
	DS（本文）	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	FR（基线）	16.31	40.98	0.20	15.43	34.58	94.94	12.14	34.08	0.41	14.66	32.50	89.92
	FR（本文）	0	0	0	0	0	0	0	0	0	0	0	0
	*F*1（基线）	37.00	7.63	1.17	20.47	10.92	14.44	34.53	8.79	0.86	18.72	2.32	12.30
	*F*1（本文）	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
高斯攻击	*DS*（基线）	93.90	5.40	9.74	97.94	7.37	9.18	67.74	7.34	6.94	78.63	1.80	7.98
	DS（本文）	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	FR（基线）	15.90	40.04	40.12	15.07	34.56	95.91	11.84	31.05	33.87	14.64	32.67	91.10
	FR（本文）	0	0	0	0	0	0	0	0	0	0	0	0
	*F*1（基线）	37.87	7.43	17.07	20.92	10.39	13.79	34.50	10.61	12.55	17.55	2.68	12.21
	*F*1（本文）	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

一，并存在隐私泄露的风险. 本文在正常联邦学习训练中插入测试轮将基准轮的回溯模型再次下发. 通过计算用户对同一模型 2 次更新的余弦相似度判断用户是否为搭便车攻击者. 基于同一用户 2 次梯度更新的相似度比较代替之前工作的离群值检测，能够在多个搭便车用户的场景中精确检测出搭便车攻击者. 这对于相关研究的深入进行具有重要的参考意义. 本文设置了较为严苛的实验场景，对于间歇性参与攻击的场景没有考虑. 我们考虑在后续工作中结合模型参数多轮变化的规律，以提高算法的鲁棒性.

作者贡献声明：洪榛负责方案设计及论文撰写；冯王磊负责文献资料整理和实验数据分析；温震宇参与方案设计和定稿讨论；吴迪和李涛涛负责论文修订；伍一鸣参与论文修改及定稿讨论；王聪和纪守领负责对研究提出意见并指导论文修改.

参 考 文 献

[1] Kuang Boyu, Li Yuze, Gu Fangming, et al. Review of Internet of vehicle security research: Threats, countermeasures, and future prospects[J]. Journal of Computer Research and Development, 2023, 60(10): 2304−2321 (in Chinese)
(况博裕，李雨泽，顾芳铭，等. 车联网安全研究综述：威胁、对策与未来展望[J]. 计算机研究与发展，2023，60(10)：2304−2321)

[2] Zheng Di, Wang Jun, Ben Kerong. Information processing for massive

sensorsin extended IOV applications[J]. Journal of Computer Research and Development, 2013, 50(S2): 257−266 (in Chinese)
(郑笛, 王俊, 贲可荣. 扩展车联网应用中的海量传感器信息处理技术[J]. 计算机研究与发展, 2013, 50(S2): 257−266)

[3] Jung K, Lee J, Park K, et al. PRIVATA: Differentially private data market framework using negotiation-based pricing mechanism[C]// Proc of the 28th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2019: 2897−2900

[4] Sun Jingwei, Li Ang, Wang Binghui, et al. Soteria: Provable defense against privacy leakage in federated learning from representation perspective[C]//Proc of IEEE/CVF Conf on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2021: 9311−9319

[5] Dong Ye, Hou Wei, Chen Xiaojun, et al. Efficient and secure federated learning based on secret sharing and gradients selection[J]. Journal of Computer Research and Development, 2020, 57(10): 2241−2250 (in Chinese)
(董业, 侯炜, 陈小军, 等. 基于秘密分享和梯度选择的高效安全联邦学习[J]. 计算机研究与发展, 2020, 57(10): 2241−2250)

[6] Cheng Yong, Liu Yang, Chen Tianjian, et al. Federated learning for privacy-preserving AI[J]. Communications of the ACM, 2020, 63(12): 33−36

[7] Deng Yongheng, Lyu F, Ren Ju, et al. AUCTION: Automated and quality-aware client selection framework for efficient federated learning[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 33(8): 1996−2009

[8] Chen Jinyin, Li Mingjun, Liu Tao, et al. Rethinking the defense against free-rider attack from the perspective of model weight evolving frequency[J]. arXiv preprint, arXiv: 2206.05406, 2022

[9] Zhang Ning, Ma Qian, Chen Xu. Enabling long-term cooperation in cross-silo federated learning: A repeated game perspective[J]. IEEE Transactions on Mobile Computing, 2023, 22(7): 3910−3924

[10] Lin Jierui, Du Min, Liu Jian. Free-riders in federated learning: Attacks and defenses[J]. arXiv preprint, arXiv: 1911.12560, 2019

[11] Fraboni Y, Vidal R, Lorenzi M. Free-rider attacks on model aggregation in federated learning[C]//Proc of the 24th Int Conf on Artificial Intelligence and Statistics. Brookline, MA: Microtome Publishing, 2021: 1846−1854

[12] Karimireddy S P, Guo Wenshuo, Jordan M I. Mechanisms that incentivize data sharing in federated learning[J]. arXiv preprint, arXiv: 2207.04557, 2022

[13] Huang Hai, Zhang Borong, Sun Yinggang, et al. Delta-DAGMM: A free rider attack detection model in horizontal federated learning[J]. Security and Communication Networks, 2022, 2022(1): 310−322

[14] Bernstein J, Zhao Jiawei, Azizzadenesheli K, et al. SignSGD with majority vote is communication efficient and fault tolerant[J]. arXiv preprint, arXiv: 1810.05291, 2018

[15] Xu Xinyi, Lyu Lingjuan. A reputation mechanism is all you need: Collaborative fairness and adversarial robustness in federated learning[J]. arXiv preprint, arXiv: 2011.10464, 2020

[16] Yin Dong, Chen Yudong, Kannan R, et al. Byzantine-robust distributed learning: Towards optimal statistical rates[C]//Proc of the 35th Int Conf on Machine Learning. New York: ACM, 2018: 5650−5659

[17] Zong Bo, Song Qi, Min M R, et al. Deep autoencoding gaussian mixture model for unsupervised anomaly detection[C/OL]//Proc of the 6th Int Conf on Learning Representations. Brookline, MA: Microtome Publishing, 2018[2023-10-31].https://openreview.net/forum?id=BJJLHbb0-

[18] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Proc of the 20th Artificial Intelligence and Statistics. Brookline, MA: Microtome Publishing, 2017: 1273−1282

[19] Wang Dong, Lu Huchuan, Bo Chunjuan. Visual tracking via weighted local cosine similarity[J]. IEEE Transactions on Cybernetics, 2014, 45(9): 1838−1850

[20] Zhang J, Qiao Guanxiong, Lopotenco A, et al. Understanding stochastic optimization behavior at the layer update level[C]//Proc of the 36th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2022: 13109−13110

[21] Makey G, Yavuz Ö, Kesim D K, et al. Breaking crosstalk limits to dynamic holography using orthogonality of high-dimensional random vectors[J]. Nature Photonics, 2019, 13(4): 251−256

[22] LaValley M P. Logistic regression[J]. Circulation, 2008, 117(18): 2395−2399

[23] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition[C]//Proc of IEEE/CVF Conf on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2016: 770−778

[24] Deng Li. The MNIST database of handwritten digit images for machine learning research[J]. IEEE Signal Processing Magazine, 2012, 29(6): 141−142

[25] Krizhevsky A , Hinton G. Learning multiple layers of features from tiny images[D]. Toronto, Canada: Department of Computer Science, University of Toronto, 2009

[26] He Chaoyang, Li Songze, So Jinhyun, et al. FedML: A research library and benchmark for federated machine learning[J]. arXiv preprint, arXiv: 2007.13518, 2020

[27] Li Qinbin, Diao Yiqun, Chen Quan, et al. Federated learning on non-IID data silos: An experimental study[C]//Proc of the 38th Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2022: 965−978

Hong Zhen, born in 1983. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include Internet of things/ information physical systems, intelligent systems security, big data analytics, and artificial intelligence.

洪　榛, 1983 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为物联网/信息物理系统、智能系统安全、大数据分析、人工智能.

Feng Wanglei, born in 1997. Master candidate. His main research interests include federated learning and distributed machine learning.

冯王磊, 1997 年生. 硕士研究生. 主要研究方向为联邦学习、分布式机器学习.

Wen Zhenyu, born in 1987. PhD, professor, PhD supervisor. Member of CCF. His main research interests include IoT, crowd sources, AI system, and cloud computing.

温震宇, 1987 年生. 博士，教授，博士生导师. CCF 会员. 主要研究方向为物联网、众包、人工智能系统、云计算.

Wu Di, born in 1993. PhD candidate. His main research interests include federated learning, distributed machine learning, edge computing, model compression, and Internet-of-Things.

吴　迪, 1993 年生. 博士研究生. 主要研究方向为联邦学习、分布式机器学习、边缘计算、模型压缩、物联网.

Li Taotao, born in 1996. PhD candidate. His main research interests include Web mining, information retrieval, machine learning.

李涛涛, 1996 年生. 博士研究生. 主要研究方向为网络挖掘、信息检索、机器学习.

Wu Yiming, born in 1996. PhD, associate professor, master supervisor. Member of CCF. Her main research interests include data-driven security, black industry mining, and cybercrime research.

伍一鸣, 1996 年生. 博士，副教授，硕士生导师. CCF 会员. 主要研究方向为数据驱动安全、黑灰产业挖掘、网络犯罪研究.

Wang Cong, born in 1985. PhD, professor, PhD supervisor. Member of CCF. His main research interests include addressing security and privacy challenges in mobile, cloud computing, IoT, and machine learning and system.

王　聪, 1985 年生. 博士，教授，博士生导师. CCF 会员. 主要研究方向为应对移动、云计算、物联网、机器学习和系统中的安全和隐私挑战.

Ji Shouling, born in 1986. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include data-driven security and privacy, AI security, and big data mining and analytics.

纪守领, 1986 年生. 博士，教授，博士生导师. CCF 高级会员. 主要研究方向为数据驱动安全和隐私、人工智能安全、大数据挖掘与分析.