面向无人机集群的鲁棒协作式层次联邦学习

梁梦晴[1,2], 王 健[1,2], 江文彬[1,2], 王雪微[1,2], 刘吉强[1,2]

[1] 北京交通大学智能交通数据安全与隐私保护北京市重点实验室 北京 中国 100044
[2] 北京交通大学网络空间安全学院 北京 中国 100044

摘要 在无人机集群环境中, 联邦学习可支持无人机间协作学习, 实现灵活处理应急管理、智能交通监管等强实时性任务, 同时, 联邦学习也为无人机集群提供数据隐私保护, 并可提升数据智能处理效率。但由于工作环境的动态性, 无人机通信稳定性较差, 联邦学习的性能会受到这种间歇性连接的影响而降低; 而且联邦学习系统极易受到由恶意的内部参与者发起的投毒攻击, 攻击者通过共享错误的模型参数实现对全局模型预测的操纵, 而现有的防御方法在数据异构场景中适用性较低。针对上述挑战, 本文首先提出了一种基于 D2D(Device-to-Device)的协作式层次联邦学习算法(Collaborative Hierarchical Federated Learning, Col-HFL), 无人机集群通过 D2D 通信实现集群内的分布式边缘共识, 全局聚合时每个集群只需采样一个无人机上传模型, 从而实现掉线鲁棒性。其次, 对于 Col-HFL 中存在的投毒攻击威胁, 进一步设计了一种双阶段鲁棒聚合算法(Two-Stage Robust Aggregation, TSRA), 无人机和云服务器分别使用基于历史参数的鲁棒边缘共识算法和基于声誉系统的鲁棒全局聚合算法来保护全局模型; 其中, 历史参数和原谅机制的使用使得算法能够更好地区分异质数据带来的正常差异以及中毒参数带来的差异, 从而实现高准确度的异常检测。在不同数据集和场景下的实验结果表明, Col-HFL 在模型准确性和能耗方面显著优于已有层次联邦学习算法。当终端设备数据集统计异质时, TSRA 能够抵御高达 40%的恶意节点在不同阶段发起的多种投毒攻击, 并且防御效果优于其他鲁棒聚合方法, 有效提升 Col-HFL 的安全性。

关键词 联邦学习; 边缘计算; D2D 通信; 投毒攻击; 鲁棒聚合; 模型安全
中图法分类号 TP309 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2025.11.02

Robust Collaborative Hierarchical Federated Learning for UAV Clusters

LIANG Mengqing[1,2], WANG Jian[1,2], JIANG Wenbin[1,2], WANG Xuewei[1,2], LIU Jiqiang[1,2]

[1] Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100044, China
[2] School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China

Abstract In UAV cluster environment, federated learning can support collaborative learning among UAVs, enabling flexible processing of tasks with high real-time requirement, such as emergency management and intelligent traffic supervision. At the same time, federated learning also provides data privacy protection for UAV clusters, and can improve the efficiency of intelligent data processing. However, due to the dynamic nature of the working environment, UAV communication is less stable, and the performance of federated learning will be reduced by this intermittent connection. At the same time, federated learning systems are highly susceptible to poisoning attacks launched by malicious internal participants, where attackers can manipulate global model prediction results by sharing wrong model parameters, and existing defense methods have low applicability in data heterogeneous scenarios. In response to the above challenges, this paper first proposes a collaborative hierarchical federated learning algorithm (Col-HFL) based on Device-to-Device communication. UAV clusters achieve distributed edge consensus through D2D communication, and during global aggregation stages, only one UAV from each cluster needs to upload model parameters to the cloud server, thereby achieving dropout robustness. Furthermore, in response to the model poisoning attack threat faced by Col-HFL, a two-stage robust aggregation algorithm (TSRA) was designed. UAVs and the cloud servers use a robust edge consensus algorithm based on historical parameters and a robust global aggregation algorithm based on reputation system to protect the global model, respectively. The use of historical parameters and forgiveness mechanism enables the algorithm to better distinguish between normal differences caused by heterogeneous data and differences caused by poisoning parameters, thereby achieving highly accurate anomaly detection. Experimental results across various datasets and scenarios indicate that Col-HFL substantially surpasses current hierarchical federated learning algorithms in terms of model accuracy and energy consumption. When terminal device datasets are statistically heterogeneous, TSRA can withstand various poisoning attacks launched by up to

通信作者: 王健, 博士, 副教授, Email: wangjian@bjtu.edu.cn。
本课题得到中国国家铁路集团有限公司科技研究开发计划项目(No. N2024W007), 中央高校基本科研业务费专项资金(No. 2025JBZY025), 国家重点研发计划项目(No. 2023YFB2703700)资助。
收稿日期: 2024-01-23; 修改日期: 2024-05-07; 定稿日期: 2025-10-14

40% of malicious UAVs at different stages, and its defense effect is better than other robust aggregation methods, effectively improving the security of Col-HFL.

Key words federated learning; edge computing; device-to-device communication; poisoning attack; robust aggregation; model security

1 引言

近年来, 无人机集群协同边缘计算在应急管理、智能交通监管等领域成为发展趋势。与传统的安装在地面基站上的边缘服务器相比, 搭载边缘服务器的无人机有着低成本、灵活部署等优点, 能够更好地为智能终端设备提供通信和计算服务。然而, 将原始数据上传至无人机进行处理会造成极大的传输开销和数据隐私泄露。针对该问题, 联合利用无人机边缘计算网络和联邦学习[1]设计解决方案成为研究热点。

如图 1 所示, 在联邦学习模式下, 无人机边缘服务器对终端设备的本地模型进行收集与聚合, 并将聚合结果上传至云服务器进行全局模型聚合, 从而以保护隐私的方式实现训练模型的共享。然而, 无人机机载电池有限, 另外由于通信限制(如城市中心、山区、海上等地区), 无人机与云服务器之间的连接会受到间歇性阻碍, 部分模型参数不能及时被云服务器聚合。设备数据异质时, 这种参与不平等会导致全局模型收敛速度变慢甚至不收敛[2-6], 并增加泛化差距。同时无人机的频繁掉线与加入又给攻击者带来可乘之机, 模型的安全性会受到一系列被动和主动攻击的威胁[7-8]。其中, 最常见的是投毒攻击[9-10], 恶意参与者可以对训练数据或局部模型进行投毒, 从而达到破坏全局模型的目的。

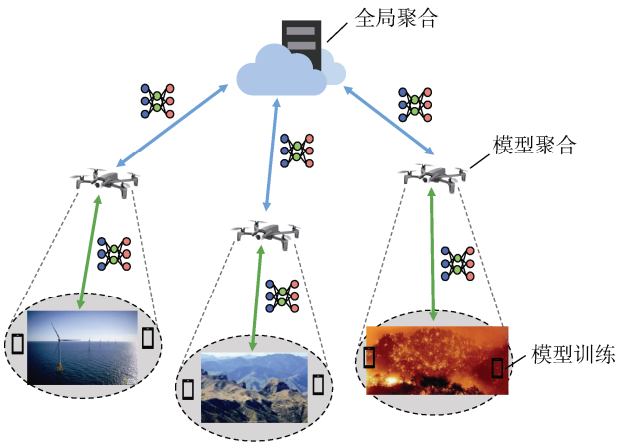


图 1 面向无人机边缘计算的联邦学习应用场景
Figure 1 Federated learning application scenarios for UAV edge computing

当前, 一些研究者通过设计节点选择机制[11-14]来缓解联邦学习系统中节点的间歇性连接导致的模型聚合延迟, 但其中不满足选择条件的节点会被排除在训练过程外, 因此参与不平等问题没有得到有效解决。其次, 现有的投毒攻击防护方案[15-17]大都基于单轮模型信息实现良性和恶意节点的区分, 而数据异质带来的随机性增加了检测的困难程度。

为解决上述问题, 本文提出了一种基于 D2D 的鲁棒协作式层次联邦学习架构, 解决了在无人机边缘计算场景下通信不稳定带来的全局模型性能下降问题以及投毒攻击对全局模型安全性的潜在威胁。本文的主要贡献如下。

(1) 提出了一种协作式层次联邦学习算法(Collaborative Hierarchical Federated Learning, Col-HFL), 根据所处的地理位置将终端设备和无人机边缘服务器划分为多个集群, 每个无人机都负责一个终端设备集群的参数聚合, 并将聚合后的参数通过 D2D 通信与相邻无人机交换, 从而实现集群内的分布式边缘共识。在全局聚合时, 根据连接状态从每个边缘集群采样一个无人机将模型上传到云服务器。这样不仅缓解了通信状态不稳定无人机对全局模型的影响, 同时显著减少了无人机上行链路传输的数据量。

(2) 在 Col-HFL 算法的基础上, 针对无人机在边缘共识和全局聚合阶段中发起的模型投毒攻击, 本文提出了双阶段鲁棒聚合算法(Two-Stage Robust Aggregation, TSRA)。在边缘共识阶段, 不同于已有方案, 利用邻域内其他无人机在前几轮 D2D 通信中的历史参数来进行恶意节点的检测, 从而减少异质数据带来的偶然性。在全局聚合阶段, 根据无人机上传的参数对全局模型的贡献度在云服务器侧维护一个声誉系统, 声誉越高的无人机在接下来的训练过程中被选中的可能性越高, 从而降低了恶意节点对全局模型安全性的影响。

(3) 在真实数据集上的实验结果证明了 Col-HFL 算法能够在保证全局模型高性能的同时减少长距离数据传输带来的延迟和通信成本。Col-HFL 通过 D2D 通信在异构数据集上实现更多的分布式处理, 能够在更少的全局聚合中实现更快的收敛。同时, TSRA 能够抵御集群内高达 40%的恶意无人机在不同阶段发起的多种投毒攻击, 模型准确率的降低可以忽略不计(1%～3%), 并且防御效果优于所有基准方案, 有效提升架构的安全性。

2 相关工作

数据孤岛以及隐私泄露是当前人工智能领域面临的主要挑战, 联邦学习[1]作为一个可行的解决方案于 2016 年被谷歌提出。联邦学习是一种分布式的机器学习框架, 数据拥有者的数据都保留在本地, 通过与云服务器交换模型信息进行协作训练。为了提高联邦学习的效率和可扩展性, Liu 等人[18]提出了基于云-边-端的层次联邦学习(Hierarchical Federated Learning, HFL), 采用多个边缘服务器来缓解与云服务器频繁交换模型信息带来的高延迟和带宽压力[19]。HFL 的训练流程可总结如下: ①终端设备使用本地数据集对其本地模型进行多轮更新并将模型更新汇总到选定的边缘服务器(例如基站); ②边缘服务器计算聚合模型并下传; ③上述 2 步迭代特定次数后, 边缘服务器将聚合模型汇总到云服务器; ④云服务器通过特定算法聚合数据, 并更新全局模型; ⑤迭代执行上述 4 步直到全局模型收敛至期望值。

设备的间歇性连接是联邦学习中一个特别重大的挑战。由于设备自身情况及物理环境的不同, 设备与云服务器之间连接的可靠度可能不同, 个别设备不能及时上传模型参数给云服务器进行聚合。最近, 一些研究工作通过考虑定制的节点选择机制[11-14]来加快学习过程。

Chen 等人[11]提出了一种启发式贪婪节点选择策略, 该策略根据节点本地的计算和通信资源迭代选择异构物联网节点参与训练。Hao 等人[12]根据每个节点上本地模型的计算能力和模型精度变化, 设计了优先节点选择函数, 其他未被选中的节点同时在本地继续迭代。然而, 这种方法没有考虑到设备的不可靠性。因此, Imteaj 等人[13]提出了一种机制, 根据每个节点的行为为其分配信任分数, 与此同时, 具有资源要求和最低信任分数的机器学习任务在联邦学习网络中发布, 不符合任务要求的节点会在训练开始前被过滤掉。完成任务的客户将获得奖励, 而未完成任务的客户的信任分数会降低。类似地, Wu 等人[14]在每次迭代中选择崩溃概率较低的节点, 落后节点将被标记为已弃用并被迫与服务器同步。在收到来自一部分节点的更新后, 中央服务器结束一轮训练并更新全局模型, 从而减少了计算成本和通信开销。

尽管上述基于节点选择的方案能够减少服务器在聚合时等待的时间, 但仍存在一个问题: 想要训练但不满足选择条件的节点会被排除, 不同参与方的参与程度存在差距, 因而在数据异质场景下, 最终得到的全局模型在不同参与方上的表现会出现较大差异。不同于上述工作, 本文提出了一种新的联邦学习范式, 通过节点之间的合作来减轻间歇性连接带来的负面影响。

投毒攻击是机器学习领域面临的一个严重挑战, 其中恶意参与者会故意向机器学习模型引入错误数据或干扰其训练过程, 进而导致模型性能下降、泛化能力减弱甚至失效, 对全局模型的安全性和可用性造成严重影响。随着联邦学习的推广应用, 越来越多的研究人员聚焦于研究联邦学习框架中的模型投毒攻击问题。根据攻击者的目标, 模型投毒攻击可以是有目标的[20-25]或无目标的[26-29]。前者的目标是最小化特定测试输入的准确性, 同时保持其余测试输入的高精度。而后者旨在针对任何测试输入不加区别地最小化全局模型的准确性, 因此对联邦学习构成更严重的威胁。在本文中, 我们重点关注无目标模型投毒攻击。

当前, 研究人员提出了许多鲁棒聚合方法来减轻投毒攻击对联邦学习系统的影响。Median[30]将平均替换为模型更新的中位数, 从而选择代表分布中心的值。Geometric Median[31]则使用所有模型更新的几何中位数替代平均。基于恶意客户端通常生成比良性客户端具有更大方差和范数的更新这一观察结果, Norm Bound[20]忽略本地更新范数高于特定阈值的客户端。FoolsGold[15]假设可以通过观察恶意客户端本地更新之间的相似性来区分良性客户端和攻击者, 类似地, Krum 和 Multi-Krum[16,26]利用良性客户端本地更新的相似性来进行区分。具体来说, Krum 首先根据客户端模型更新分布的几何距离对其进行排序, 并选择一个与其他客户端参数最为相似的客户端, 将该客户端的模型参数作为全局模型参数。Multi-Krum 则包含一个参数 f, 用于指定要聚合的客户端数量(排序后的前 f 个), 并将聚合结果作为全局模型参数。FLTrust[17]利用服务器上的附加验证数据集, 如果本地模型更新的更新方向与基于验证数据集计算的服务器模型更新的更新方向偏差较大, 则本地模型更新的信任分数较低。

上述方案大都利用不同客户端单轮模型之间的差异来进行恶意客户端的检测, 然而, 数据异质带来的正常模型差异极大地增加了检测难度。本文通过使用历史参数来摆脱异质数据带来的影响。

3 基于 D2D 的鲁棒协作式层次联邦学习

本节介绍基于 D2D 的鲁棒协作式层次联邦学习架构, 与传统的层次联邦学习架构相比, 本文提出的架构对节点掉线及由恶意内部节点发起的投毒攻

击具有鲁棒性。3.1 介绍了该协作式层次联邦学习架构的系统模型与威胁模型。3.2 节和 3.3 节分别提出了 Col-HFL 算法和 TSRA 算法。Col-HFL 算法描述了架构的基本训练过程, 无人机之间通过协作式 D2D 通信实现集群内的模型参数共识, 在全局聚合时, 可以根据每个无人机的实际情况选择上传参数的无人机, 在显著减小通信成本的同时解决了无人机掉线导致的联邦学习性能下降问题。针对架构中存在的投毒攻击威胁, 在 Col-HFL 算法的基础上提出了 TSRA 算法, 使得协作式层次联邦学习架构能够在训练过程中剔除掉恶意无人机的中毒参数, 从而实现对全局模型的保护。

3.1 系统模型与威胁模型

3.1.1 系统模型

我们考虑在图 2 所示的网络结构上进行模型学习, 该网络由 1 个云服务器, C 个无人机(作为边缘服务器, 以集合 $\mathcal{C}=\{1,2,\cdots,C\}$ 表示)以及 L 个终端设备(以集合 $\mathcal{L}=\{1,2,\cdots,L\}$ 表示)组成。每个终端设备 l 拥有一个样本数为 D_l 的本地数据集 $\mathcal{D}_l$, 目的是使用设备的本地数据集协作训练一个机器学习模型。

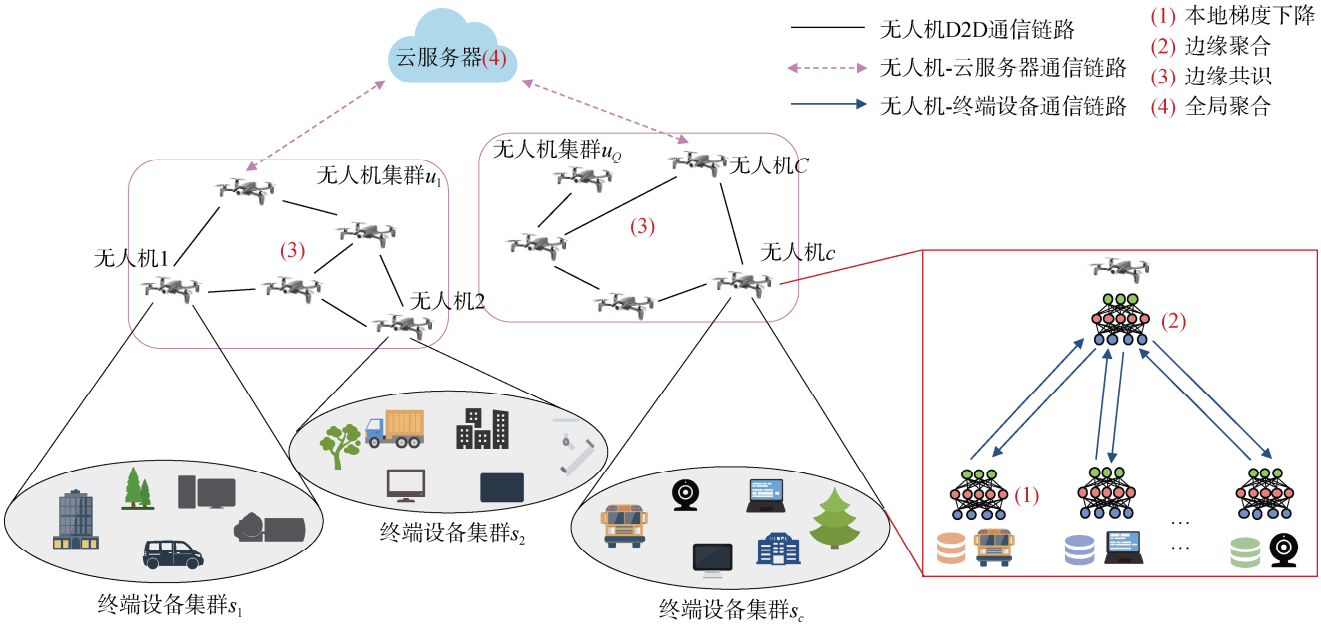


图 2 基于 D2D 的协作式层次联邦学习架构

Figure 2 Framework of Collaborative Hierarchical Federated Learning

终端设备被划分为 C 个两两不相交的集群 $\mathcal{S}_1,\mathcal{S}_2,\cdots,\mathcal{S}_C$, 集群 $\mathcal{S}_c$ 包含 s_c 个底层设备, 并且 $\sum_{c=1}^{C}s_c=L$。每个集群 $\mathcal{S}_c$ 都有一个对应的无人机边缘节点 $c\in\mathcal{C}$, 集群中的所有终端设备只与无人机 c 进行通信。

无人机被划分为 Q 个两两不相交的集群 $\mathcal{U}_1,\mathcal{U}_2,\cdots,\mathcal{U}_Q$, 集群 $\mathcal{U}_q$ 包含 u_q 个无人机, 并且 $\sum_{q=1}^{Q}u_q=C$。使用 $G_q=(\mathcal{U}_q,\mathcal{E}_q)$ 代表无人机集群 $\mathcal{U}_q$ 的通信拓扑图, 其中 $\mathcal{E}_q$ 代表边集合, $c,c'\in\mathcal{E}$ 当且仅当 $c,c'\in\mathcal{U}_q$ 且 $c\in\mathcal{N}_{c'}$。对于无人机 $c\in\mathcal{U}_q$, 令 $\mathcal{N}_c\subseteq\mathcal{U}_q$ 表示该节点的 D2D 邻居无人机集合。我们假设 D2D 通信是双向的, 即对于任意 $c,c'\in\mathcal{U}_q$, $c\in\mathcal{N}_{c'}$ 当且仅当 $c'\in\mathcal{N}_c$。

模型训练是通过一系列由 $k=1,2,\ldots$ 索引的全局聚合来进行的, 具体细节在 3.2 节给出, 在两次相邻的全局聚合之间, 终端设备进行本地梯度下降更新本地模型, 无人机 c 会对其对应集群 $\mathcal{S}_c$ 中终端设备上传的本地模型进行边缘聚合, 然后参与与其邻居 $\mathcal{N}_c$ 的边缘共识过程。因为每个无人机的模型都反映了其集群的共识, 在全局聚合时, 每个无人机集群 $\mathcal{U}_q$ 只需一个无人机上传参数。与传统的层次联邦学习架构相比, 我们的架构将无人机层内的通信及多层(云服务器层、无人机层和终端设备层)之间的参数传输进行集成。我们将该架构称为协作式层次联邦学习架构, 因为它涉及无人机集群内多轮协作的 D2D 通信。

3.1.2 威胁模型

在上述的系统模型中, 我们假设云服务器和终

端设备都是诚实可信的, 而无人机节点的子集是恶意的或受恶意对手控制。恶意无人机的目标和能力如下。

恶意无人机的目标。恶意无人机通过生成中毒模型参数, 使云服务器在聚合每个集群的共识参数后得到的全局模型的准确性在任何测试输入上会不加区别地降低。恶意无人机可能在以下两个阶段发起投毒攻击以达到攻击目标。

(1) 在无人机集群通过 D2D 通信进行边缘共识时, 恶意无人机可以生成中毒的模型参数发送给所有的邻居无人机节点;

(2) 在全局聚合时, 被选中的恶意无人机节点可以生成中毒的模型参数发送给云服务器。

恶意无人机的能力如下。

(1) 恶意无人机无法获取终端设备本地数据集或操纵其模型训练过程;

(2) 恶意无人机只能操纵自己发送给云服务器及邻居无人机的模型参数, 但无法控制其他无人机;

(3) 在边缘共识阶段, 恶意无人机在进行集群内共识之前生成中毒参数, 接着在每轮 D2D 通信时将中毒参数发送给邻居无人机节点;

(4) 在全局聚合阶段, 恶意无人机只有被选中时才能将生成的中毒参数发送给云服务器。

在3.3 节中, 我们对不同阶段投毒攻击的防护关键进行了分析并提出 TSRA 算法对全局模型进行保护。

3.2 Col-HFL 算法

Col-HFL 由周期性全局聚合之间的一系列边缘共识间隔组成。在每个边缘共识间隔内, 终端设备都会进行随机梯度下降迭代以对本地模型进行更新, 无人机边缘节点对终端设备的本地模型进行聚合并通过无人机集群内的共识程序定期地同步模型参数。在无人机层引入边缘共识程序有三个主要原因。首先, 边缘共识可以有效地减少模型之间的偏差, 这是在跨终端设备的本地数据非独立同分布场景下联邦学习面临的主要挑战之一。其次, 与全局聚合相比, 边缘共识过程中集群内的 D2D 通信通常在短距离内执行, 因此会产生较低的能耗和延迟, 全局聚合则需要无人机通过上行链路将参数传输到距离较远的云服务器。最后, D2D 正在成为 5G 及以上无线网络的普遍特征[32]。

3.2.1 算法流程

我们使用一组离散的时间指标 $\mathcal{T}=\{1,2,\ldots\}$ 来索引时间。在终端设备进行 τ_1 次本地模型更新后, 每个无人机聚合对应终端设备的本地模型并与相邻的无人机进行边缘共识。在 τ_2 次边缘共识后, 云服务器聚合所有边缘共识模型, 这意味着终端设备与云服务器的通信间隔 $\tau=\tau_1\tau_2$。第 k 次全局聚合发生在时间 $t_k\in\mathcal{T}$ ($t_0=0$), 因此 $\mathcal{T}_k=\{t_{k-1}+1,\ldots,t_k\}$ 表示第 $k-1$ 次全局聚合和第 k 次全局聚合之间的本地模型训练间隔。使用 $w^{(t_k)}\in\mathbb{R}^M$ 表示云服务器在第 k 次全局聚合之后得到的全局模型。整个模型训练过程开始之前, 云服务器通过层次结构将 $w^{(0)}$ 传播到终端设备。完整的训练过程见算法 1。

(1) 本地梯度下降: 每个终端设备拥有一个本地模型, 使用 $w_l^{(t-1)}\in\mathbb{R}^M$ 表示每个终端设备 l 在时间 $t-1$ 时的本地模型。终端设备 l 根据式(1)在本地模型上进行连续的梯度下降, 得到第 t 轮的本地模型。

$$w_l^{(t)}=w_l^{(t-1)}-\eta_{t-1}\nabla F_l\left(w_l^{(t-1)}\right) \tag{1}$$

其中, $\eta_{t-1}>0$ 代表更新步长。若 $t\,|\,\tau_1=0$, 终端设备 l 便将 $w_l^{(t)}$ 上传给对应的无人机以进行边缘聚合和共识。

(2) 边缘聚合: 无人机 c 根据式(2)对接收到的本地模型 $\{w_l^{(t)},l\in\mathcal{S}_c\}$ 进行聚合, 得到边缘聚合模型。

$$\tilde{w}_c^{(t)}=\frac{1}{D_{(c)}}\sum\nolimits_{l\in\mathcal{S}_c}D_l w_l^{(t)} \tag{2}$$

其中, $D_{(c)}=\sum_{l\in\mathcal{S}_c}D_l$, 然后无人机 c 与邻域内的其他无人机节点进行 $d\in\mathbb{N}$ 轮的 D2D 通信以实现边缘共识。

(3) 边缘共识: 边缘共识的目的是获得整个无人机集群内所有边缘聚合模型参数平均值的近似值。一种基本方法是设计消息传递算法, 其中无人机节点与邻居交换参数直到集群中的每个无人机都将所有参数存储在本地。然后每个无人机可以对所有模型参数进行聚合, 且云服务器可以对其中之一进行采样。然而, 随着神经网络的发展, 特别是深度神经网络的出现, 模型参数的数量显著增长。以经典的 ResNet 为例, 它包含了大约 1100 万个参数。因此在每个无人机节点收集所有参数是不可行的。在此, 我们采用一般的线性分布式共识算法[33]。在边缘共识期间, 每个无人机集群内的节点进行 d 轮 D2D, 每轮 D2D 都包含相邻节点之间的参数传输和聚合, 聚合过程见式(3)。在这里, 使用 $t'=0,1,\cdots,d-1$ 对通信轮次进行索引。

$$z_c^{(t'+1)}=\sum\nolimits_{c'\in\mathcal{N}_c\cup\{c\}}\alpha_{cc'}^{(k)}z_{c'}^{(t')} \tag{3}$$

其中, $z_c^{(0)}=\tilde{w}_c^{(t)}$, 在该过程的最后, 无人机 c 使用 $\hat{w}_c^{(t)}=z_c^{(d)}$ 作为自己的边缘共识模型。$\alpha_{cc'}^{(k)}$ 是无人机 c

在第 k 次全局聚合期间分配给无人机 c' 的非负重要性权重。索引 t' 对应于 Col-HFL 中的第二个时间维度, 代表共识过程, 而不是捕获终端设备本地模型更新所经过的时间的索引 t。图 3 描述了这两个时间维度。

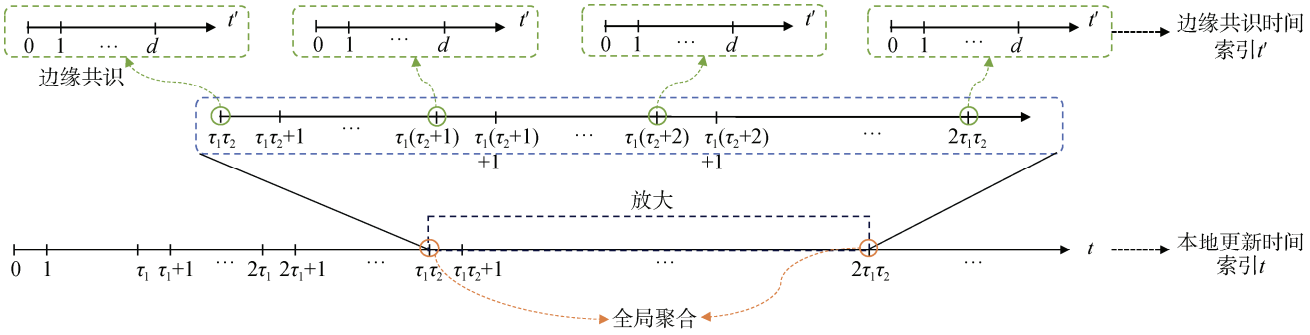


图 3　Col-HFL 中的两个时间维度

Figure 3　Two different time scales in Col-HFL

若 $t \mid \tau_1\tau_2 \neq 0$, 则无人机 c 将 $\hat{w}_c^{(t)}$ 下发给对应的终端设备集群, 集群内的终端设备 $l \in \mathcal{S}_c$ 使用 $w_l^{(t)} = \hat{w}_c^{(t)}$ 更新本地模型。

若 $t \mid \tau_1\tau_2 = 0$, 此时 $t = t_k$, 则云服务器需要对边缘共识模型进行汇总并进行全局聚合。

(4) 全局聚合: 首先, 云服务器从每个无人机集群 $\mathcal{U}_q, q = 1,2,\cdots,Q$ 采样一个无人机 $g_q \in \mathcal{U}_q$ 上传参数, 令 $\mathcal{G}^{(t_k)} = \{g_1, g_2, \cdots, g_Q\}$ 表示被选中的无人机。

在传统的层次联邦学习中, 所有的边缘服务器都需将模型参数上传到云服务器进行全局聚合, 这在大规模联邦学习系统中是难以实现的[32]。首先, 从携带有限资源的无人机边缘节点到云服务器的上行链路传输通常对应于较长的物理距离, 模型参数传输可能耗尽单个设备的电量。此外, 长距离传输会导致高网络流量和长延迟。最后, 大量的模型参数可能会使云服务器过载。引入采样技术能够减少无人机到云服务器上行链路传输的数据量(以无人机集群大小的倍数), 这是通过边缘共识过程实现的, 该过程模仿了集群内的分布式模型聚合。

对于所有 $c \in \mathcal{G}^{(t_k)}$, 无人机 c 将边缘共识模型 $\hat{w}_c^{(t_k)}$ 上传给云服务器以进行全局聚合。云服务器根据式(4)对接收到的所有边缘共识模型 $\{\hat{w}_c^{(t_k)}, c \in \mathcal{G}^{(t_k)}\}$ 进行聚合, 得到更新后的全局模型。

$$w^{(t_k)} = \frac{1}{Q}\sum\nolimits_{c \in \mathcal{G}^{(t_k)}} \hat{w}_c^{(t_k)} \tag{4}$$

然后云服务器将更新后的全局模型通过层次结构传播到终端设备, 终端设备 $l \in \mathcal{L}$ 使用 $w_l^{(t)} = w^{(t_k)}$ 更新本地模型。

算法 1. 基于 D2D 的协作式层次联邦学习算法

输入: 全局聚合轮次 K, 共识轮次 d

输出: 最终的全局模型 $w^{(t_k)}$

(1) 云服务器初始化: 初始化全局模型 $w^{(0)}$ 并通过层次结构传播到终端设备。
(2) FOR $k = 1,2,\cdots,K$ do
(3) FOR $t = t_{k-1}+1,\cdots,t_k$ do
(4) FOR $l = 1,2,\cdots,L$ 并行 do
(5) 设备 l 根据式(1)得到 $w_l^{(t)}$。
(6) END FOR
(7) IF $t \mid \tau_1 = 0$ THEN
(8) FOR $c = 1,2,\cdots,C$ 并行 do
(9) 无人机边缘节点 c 根据式(2)和式(3)得到 $\hat{w}_c^{(t)}$。
(10) IF $t \mid \tau_1\tau_2 \neq 0$ THEN
(11) 设备 $l \in \mathcal{S}_c$ 令 $w_l^{(t)} = \hat{w}_c^{(t)}$。
(12) END IF
(13) END FOR
(14) END IF
(15) IF $t \mid \tau_1\tau_2 = 0$ THEN
(16) 云服务器选择 $\mathcal{G}^{(t_k)} = \{g_1, g_2, \cdots, g_Q\}$。
(17) FOR 所有 $c \in \mathcal{G}^{(t_k)}$ 并行 do
(18) 无人机节点 c 将 $\hat{w}_c^{(t_k)}$ 上传给云服务器。
(19) END FOR
(20) 云服务器根据式(4)计算 $w^{(t_k)}$ 并通过层次结构传播到终端设备。
(21) FOR $l = 1,2,\cdots,L$ 并行 do
(22) 设备 l 令 $w_l^{(t)} = w^{(t_k)}$。
(23) END FOR
(24) END IF

(25) END FOR
(26) END FOR

3.2.2 理论分析

Col-HFL 将无人机层 D2D 通信与多层参数传输相集成, 这为模型训练引入了 D2D 网络维度。接下来, 我们将对无人机集群内基于 D2D 通信的边缘共识过程进行分析[34]。为方便分析, 我们以矩阵形式表达边缘共识过程。令 $\tilde{W}_q^{(t)} \in \mathbb{R}^{u_q \times M}$ 表示无人机集群 $\mathcal{U}_q$ 中 u_q 个无人机节点的边缘聚合模型矩阵, 其中 $\tilde{W}_q^{(t)}$ 的第 i 行对应于无人机 i 的边缘聚合模型 $\tilde{w}_c^{(t)}$。那么, 边缘共识阶段后更新的参数矩阵 $W_q^{(t)}$ 可以写为式(5)。

$$W_q^{(t)} = \left(V_q^k\right)^d \tilde{W}_q^{(t)}, t \in \mathcal{T}_k \tag{5}$$

其中, $V_q^k = \left[\alpha_{cc'}^{(k)}\right]_{1\le c,c'\le u_q} \in \mathbb{R}^{u_q \times u_q}$ 表示无人机集群 $\mathcal{U}_q$ 在第 $k-1$ 次全局聚合和第 k 次全局聚合之间的共识矩阵, $W_q^{(t)}$ 的第 i 行对应于无人机 i 的边缘共识模型 $\hat{w}_c^{(t)}$。

假设 1。 共识矩阵 V_q^k 满足以下条件: ① $(V_q^k)_{m,n} = 0$, 若 $(m,n) \notin \mathcal{E}_q^k$, 即无人机节点仅接收其邻域内其他无人机发送的参数; ② $V_q^k 1 = 1$, 即行随机性; ③ $V_q^k = (V_q^k)^{\mathrm{T}}$, 即对称性; ④ $\rho(V_q^k - 11^{\mathrm{T}}/u_q) < 1$, 其中 $\rho(A)$ 代表矩阵 A 的谱半径, 即 $V_q^k - 11^{\mathrm{T}}/u_q$ 的最大特征值小于 1。

边缘共识过程可以被视为每个无人机集群中模型的有误聚合。具体来说, 我们可以将无人机 $i \in \mathcal{U}_q$ 处的边缘共识模型写为式(6)。

$$\hat{w}_i^{(t)} = \bar{w}_q^{(t)} + e_i^{(t)} \tag{6}$$

其中, $\bar{w}_q^{(t)}$ 是集群 $\mathcal{U}_q$ 中所有无人机边缘聚合模型的平均值, $e_i^{(t)} \in \mathbb{R}^M$ 表示无人机之间有限的 D2D 轮数(即 $d < \infty$)引起的共识误差, 其上限见引理 1。

引理 1。 无人机 $i \in \mathcal{U}_q$ 使用共识矩阵 V_q^k 在集群 $\mathcal{U}_q$ 中进行 d 轮共识后, 其共识误差 $e_i^{(t)}$ 满足式(7)。

$$e_i^{(t)} \le \left(\lambda_q\right)^d \sqrt{u_q} \max_{j,j' \in \mathcal{U}_q} \left\| \tilde{w}_j^{(t)} - \tilde{w}_{j'}^{(t)} \right\| \tag{7}$$

其中, $\lambda_q = \rho(V_q^k - 11^{\mathrm{T}}/u_q)$。

证明: 设矩阵 $\bar{W}_q^{(t)} = (11^{\mathrm{T}}/u_q)\tilde{W}_q^{(t)}$, 并定义 $E_q^{(t)} = W_q^{(t)} - \bar{W}_q^{(t)} = [(V_q^k)^d - 11^{\mathrm{T}}/u_q][\tilde{W}_q^{(t)} - \bar{W}_q^{(t)}]$, 则 $E_q^{(t)}$ 的第 i 行为 $e_i^{(t)}$。因此, 基于假设 1, 我们可以推理得到共识误差的边界, 推理过程见式(8)。

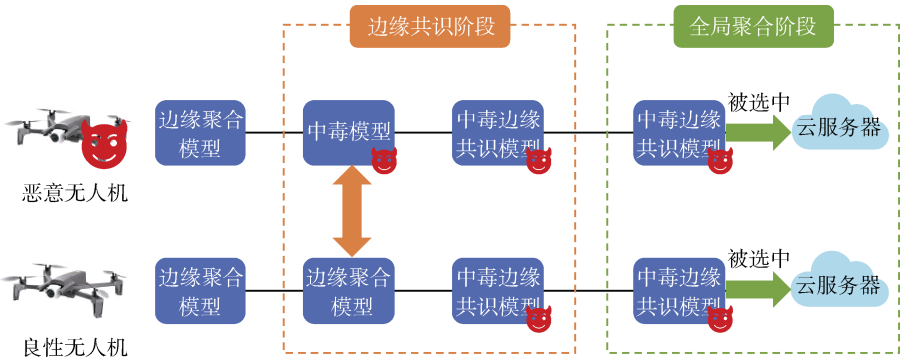
$$\begin{aligned}
\left\|e_i^{(t)}\right\|^2 &\le \operatorname{trace}\left(\left(E_q^{(t)}\right)^{\mathrm{T}} E_q^{(t)}\right) \\
&= \operatorname{trace}\left(\left[\tilde{W}_q^{(t)} - \bar{W}_q^{(t)}\right]^{\mathrm{T}} \left[\left(V_q^k\right)^d - \frac{11^{\mathrm{T}}}{u_q}\right]^2 \left[\tilde{W}_q^{(t)} - \bar{W}_q^{(t)}\right]\right) \\
&\le \left(\lambda_q\right)^{2d} \sum_{j=1}^{u_q} \left\|\tilde{w}_j^{(t)} - \bar{w}_q^{(t)}\right\|^2 \\
&\le \left(\lambda_q\right)^{2d} \frac{1}{u_q} \sum_{j,j'=1}^{u_q} \left\|\tilde{w}_j^{(t)} - \tilde{w}_{j'}^{(t)}\right\|^2 \\
&\le \left(\lambda_q\right)^{2d} u_q \max_{j,j' \in \mathcal{U}_q} \left\|\tilde{w}_j^{(t)} - \tilde{w}_{j'}^{(t)}\right\|^2
\end{aligned} \tag{8}$$

由此可以得到引理 1。引理 1 的界限量化了共识误差如何依赖于多个参数。我们可以观察到, 在其他参数固定时, 增加每个无人机集群的 D2D 轮数会产生更小的边界(因为 d 为 $\lambda_q < 1$ 的指数), 对应着更好的预期模型损失。相比于传统的层次联邦学习(其中 $d = 0$), 引入了 D2D 边缘共识的 Col-HFL 能够提高全局模型的性能。

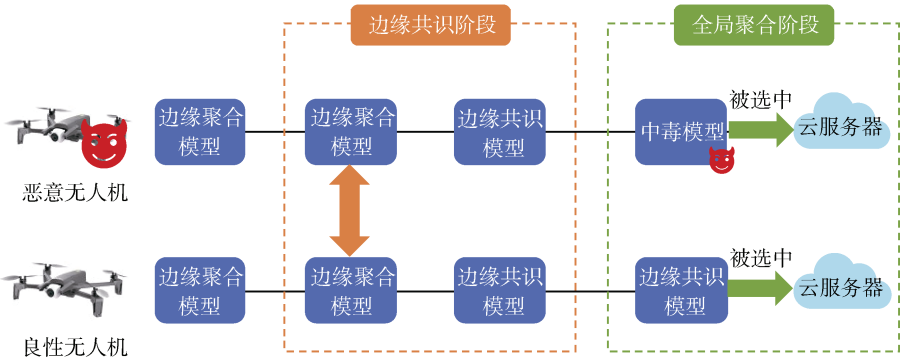
另外, 对于固定数量的 D2D 轮数, 较小的谱半径 λ_q (对应于集群内更好的连接)和 u_q (对应于较小的集群)会产生更小的边界。而较大的模型分歧 $\delta_q^{(t)}$ 会产生更差的边界。$\delta_q^{(t)} = \max_{i,i' \in U_q} \left\|\tilde{w}_i^{(t)} - \tilde{w}_{i'}^{(t)}\right\|$ 捕获了集群 $\mathcal{U}_q$ 在时间 $t \in \mathcal{T}_k$ 时(执行边缘共识之前)的边缘聚合模型的分歧程度。这取决于两个方面: ①集群 $\mathcal{U}_q$ 内各无人机对应的终端设备持有的训练数据的异质性(即非独立同分布的程度); ②每两次边缘共识之间终端设备的本地训练轮次(即 τ_1), 在数据异质时, 随着 τ_1 的增加, 终端设备的本地模型之间的分歧增大, 进而产生更大的 $\delta_q^{(t)}$。在 4.2 节中, 我们在不同程度的非独立同分布场景下对 τ_1 的取值进行了实验探究。

3.3 双阶段鲁棒聚合算法

与在全局聚合阶段恶意无人机发起的投毒攻击相比, 全局模型更容易受到边缘共识阶段的攻击的影响。因为前者只有当恶意无人机被选中上传参数时才能达到攻击目标, 而当恶意无人机在 D2D 通信过程中向其邻居无人机节点发起投毒攻击时, 中毒参数会沿着集群内通信网络拓扑传播, 影响到集群内所有无人机节点的模型参数。在全局聚合时, 该集群内任意一个无人机上传的参数都会破坏全局模型的安全性。在这两个阶段恶意无人机的攻击行为对比见图 4。



(a) 边缘共识阶段的投毒攻击
(a) Poisoning Attack at the Edge Consensus Stage



(b) 全局聚合阶段的投毒攻击
(b) Poisoning Attack at the Global Aggregation Stage

图 4　不同阶段恶意无人机的攻击行为
Figure 4　Attack Behavior of Malicious UAVs at Different Stages

本节对不同阶段投毒攻击的防护关键进行了分析并提出一种双阶段鲁棒聚合算法对全局模型进行保护。该算法由两部分组成: ①无人机侧基于历史参数的鲁棒边缘共识聚合算法; ②服务器侧的基于声誉系统的鲁棒全局聚合算法, 这两种鲁棒聚合算法可以同时应用在 Col-HFL 中, 从而在各个阶段对恶意无人机与良性无人机进行区分, 保护全局模型的安全。

3.3.1　基于历史参数的鲁棒聚合算法

在边缘共识阶段, 当恶意无人机向其邻居无人机节点发起投毒攻击时, 中毒参数会随着 D2D 通信影响到集群内的所有无人机, 使得集群内最终的共识模型参数被破坏。无人机之间的分布式共识过程以及中毒参数的传播路径对于云服务器来说是不可知的, 因此无法在云服务器侧对共识阶段中的恶意无人机进行识别。要成功阻断中毒参数在集群拓扑内的传播, 需要无人机在本地对邻域内的良性参数和中毒参数进行区分。

现有方法包括基于验证数据集的检测和基于参数之间相似度的检测。但是无人机携带的资源有限, 所以在无人机上通过验证数据集进行中毒参数检测是不可行的。当前已有的基于相似度进行检测的工作大都使用当前轮次的参数来计算相似度。在数据异质时, 单轮参数不可避免地带来了随机性, 因此难以对良性参数之间由于数据异质导致的低相似度和恶意参数与良性参数之间的低相似度进行区分。为了摆脱数据异质对检测结果的不利影响, 本文引入历史参数来更加准确地检测出中毒参数。

定义 1　历史参数(historical Parameter, HP) 无人机 c 在前 b 轮 D2D 通信中收到的邻域内每个无人机的参数的平均值。这里 b 可以看作一个滑动窗口, 减少了单轮的随机性, 从而消除了偶然性。使用 $t'=0,1,\cdots,d-1$ 对 D2D 通信进行索引, 对于无人机 $c'\in\mathcal{N}_c$, 它在第 t' 轮的历史参数可以表示为式(9)。

$$w_{\mathrm{HP}}^{c'}=\frac{1}{b}\sum_{H=t'-b}^{t'-1}z_{c'}^{(t')} \tag{9}$$

良性无人机和恶意无人机的历史参数之间存在明显的差距, 然而数据异质会给识别这种差距带来一定程度的干扰。聚类算法作为一种无监督算法, 能够从这些历史参数中发现隐藏的模式和结构, 帮助

我们更好地适应数据异质并找到潜在规律, 因此我们使用聚类算法来进行异常检测。

DBSCAN(Density-Based Spatial Clustering of Applications with Noise)是一种基于密度的空间聚类算法, 特别适用于处理含有噪声(即孤立点或异常值)的数据集。该算法的核心在于通过数据点的密度来判定聚类, 并基于此将密度相连的点集划分为不同的簇。在 DBSCAN 中, 聚类不是基于距离阈值内的点数量来形成的, 而是基于密度可达性。这意味着, 一个点如果可以通过一系列密度相连的点到达另一个点, 则这两个点被视为同一簇的成员。这种密度相连的概念使得 DBSCAN 能够发现任意形状的簇, 而不仅仅是圆形或球形的簇。与 K-means 等聚类算法相比, DBSCAN 不需要预先指定簇的数量。在实际应用中, 无人机集群内的通信拓扑可能会随着它们之间的物理距离和通信状况不断变化, 因此每个无人机邻域内的恶意节点数量也是不固定的, 使用 DBSCAN 能更好地适应这种变化。

算法 2. 基于历史参数的鲁棒聚合算法

输入: D2D 轮数 d, $\mathcal{N}_c$。

输出: $\mathcal{N}_c^{\text{benign}}$, $\mathcal{N}_c^{\text{mal}}$, $\boldsymbol{z}_c^{(d)}$。

(1) FOR $t'=0,1,\cdots,d-1$ do

(2) 接收邻域内所有无人机的参数 $\{\boldsymbol{z}_{c'}^{(t')}, c'\in\mathcal{N}_c\}$。

(3) IF $t'<b$ THEN

(4) 记录 $\{\boldsymbol{z}_{c'}^{(t')}, c'\in\mathcal{N}_c\cup\{c\}\}$。

(5) ELSE THEN

(6) 计算邻域内所有无人机及自己的历史参 $\{w_{\text{HP}}^{c'}, c'\in\mathcal{N}_c\cup\{c\}\}$。

(7) 执行 DBSCAN 算法得到 $\mathcal{N}_c^{\text{benign}}$ 和 $\mathcal{N}_c^{\text{mal}}$。

(8) 令 $\boldsymbol{z}_c^{(t'+1)}=\dfrac{1}{\left|\mathcal{N}_c^{\text{benign}}\right|}\sum_{c'\in\mathcal{N}_c^{\text{benign}}}\boldsymbol{z}_{c'}^{(t')}$。

(9) END IF

(10) END FOR

(11) 令 $\hat{w}_c^{(t)}=\boldsymbol{z}_c^{(d)}$

在 D2D 分布式共识的过程中, 良性无人机 c 从第 b 轮开始中毒参数的检测。考虑到一旦中毒参数被其他良性无人机聚合, 在接下来的几轮 D2D 中, 中毒参数就会被传播到整个集群内的所有无人机节点并对集群内的共识参数造成不可逆转的负面影响, 因此对于 $t'<b$, 无人机 c 对收到的所有邻居的参数仅进行记录, 不进行聚合。

对于 $b\le t'\le d-1$, 无人机 c 首先在邻域内所有无人机及自己的历史参数上应用基于欧几里得距离的 DBSCAN 聚类算法。聚类后会得到两个组: ①由恶意无人机组成的噪声点组 $\mathcal{N}_c^{\text{mal}}$; ②由无人机 c 和其他良性无人机组成的良性组 $\mathcal{N}_c^{\text{benign}}$。对于邻域内不存在恶意节点的无人机 c, DBSCAN 之后得到的噪声点组为空。在对恶意节点和良性节点进行区分后, 无人机 c 仅对良性组内各节点的参数进行平均聚合得到 $\boldsymbol{z}_c^{(t'+1)}$。

最后, 无人机 c 使用 $\boldsymbol{z}_c^{(d)}$ 作为自己的边缘共识模型 $\hat{w}_c^{(t)}$。算法 2 描写了具体过程。

3.3.2 基于声誉系统的鲁棒聚合算法

虽然无人机本地基于历史参数的鲁棒聚合算法能够保证良性无人机之间得到正确的共识模型参数, 但在全局聚合时, 如果恶意无人机被选中上传参数, 全局模型的性能也会因为其上传的中毒参数而下降, 所以我们需要云服务器在全局聚合阶段也能够检测出恶意无人机。对于这一阶段的投毒攻击, 防护的关键在于减小恶意无人机被选择的可能性, 因此, 我们提出了云服务器上的鲁棒全局聚合算法。

具体来说, 云服务器维护一个声誉系统, 每个无人机的声誉取决于无人机上传的参数对全局模型的贡献度。声誉越高的无人机被选择的可能性越高。

算法 3. 基于声誉系统的鲁棒聚合算法

输入: 第 $k-1$ 轮后全局模型的贡献度 $A_{\text{Global}}^{(t_{k-1})}$, 所有无人机的声誉 $A=\{A_1,A_2,\cdots,A_C\}$。

输出: 第 k 轮的全局模型 $w^{(t_k)}$ 及贡献度 $A_{\text{Global}}^{(t_k)}$。

(1) 初始化 $\mathcal{G}^{(t_k)}=\{\}$, $\mathcal{G}_{\text{benign}}^{(t_k)}=\{\}$。

(2) FOR $q=1,2,\cdots,Q$ do

(3) 选择集群 $\mathcal{U}_q$ 内声誉最高的无人机 g_q 加入到 $\mathcal{G}^{(t_k)}$。

(4) END FOR

(5) FOR $c\in\mathcal{G}^{(t_k)}$ do

(6) 计算 $A_c=\mathcal{R}\left(\hat{w}_c^{(t_k)}\right)$ 并更新 A。

(7) IF $A_{\text{Global}}^{(t_{k-1})}-A_c\le \text{MAX}_{\text{DIFF}}$ THEN

(8) 将 c 加入到 $\mathcal{G}_{\text{benign}}^{(t_k)}$ 中。

(9) END IF

(10) END FOR

(11) 令 $w^{(t_k)}=\dfrac{1}{\left|\mathcal{G}_{\text{benign}}^{(t_k)}\right|}\sum_{c\in\mathcal{G}_{\text{benign}}^{(t_k)}}\hat{w}_c^{(t_k)}$, 并计算 $w^{(t_k)}$ 在验证数据集上的贡献度 $A_{\text{Global}}^{(t_k)}$。

整个训练过程开始前, 云服务器计算初始全局模型 $w^{(0)}$ 在验证数据集上的贡献度 $A_{\text{Global}}^{(t_0)}$, 并初始化每个无人机 c 的声誉值 $A_c = A_{\text{Global}}^{(t_0)}$。在第 k 次全局聚合时, 云服务器执行算法 3 来更新无人机的声誉及全局模型。该算法由以下 4 个步骤组成。

(1) 云服务器选择当前每个集群 $\mathcal{U}_q$ 内声誉值最高的无人机 g_q 上传参数, 得到集合 $\mathcal{G}^{(t_k)} = \{g_1, g_2, \cdots, g_Q\}$。如果某一集群内存在多个无人机声誉值相同, 云服务器随机选择一个加入 $\mathcal{G}^{(t_k)}$。

(2) 云服务器收到无人机上传的模型参数 $\{\hat{w}_c^{(t_k)}, c \in \mathcal{G}^{(t_k)}\}$ 后, 根据贡献度度量函数 $\mathcal{R}$ 得到每个参数在验证数据集上的贡献度 $\{A_c, c \in \mathcal{G}^{(t_k)}\}$, 并使用贡献度结果对相应无人机的声誉进行更新。

实验中, 我们使用准确率来度量贡献度, 这是因为准确率是最能直观衡量参数对全局模型贡献度的指标, 因此能够准确地区分良性参数和中毒参数。

(3) 考虑到在数据异质时, 良性无人机上传的参数也可能会使贡献度降低, 抛弃这类参数会使全局模型无法学习到代表性不足的样本, 影响全局模型的收敛速度。因此, 我们设计了一种原谅机制, 其中使用参数 MAX_{DIFF} 作为良性参数贡献度下降的上限。云服务器会将不满足 $A_{\text{Global}}^{(t_{k-1})} - A_c \le \text{MAX}_{\text{DIFF}}$ 的参数判定为中毒参数, 并在本轮全局聚合中抛弃中毒参数, 仅使用剩余的良性参数来得到更新后的全局模型和全局模型贡献度 $A_{\text{Global}}^{(t_k)}$。

(4) 云服务器将更新后的全局模型 $w^{(t_k)}$ 通过层次结构传播到所有终端设备以更新其本地模型。

4 实验结果与分析

本节在真实数据集上进行实验评估, 主要讨论以下几方面的问题: ①探究 Col-HFL 算法中关键参数对性能的影响; ②验证 Col-HFL 算法能够降低通信开销, 且与传统的层次联邦学习算法性能相当; ③验证算法 2 能够有效抵御恶意无人机在边缘共识阶段发起的投毒攻击; ④验证算法 3 能够有效抵御恶意无人机在全局聚合阶段发起的投毒攻击, 并探究参数 MAX_{DIFF} 对结果的影响; ⑤验证 TSRA 算法(同时部署算法 2 和算法 3)能够有效抵御恶意无人机在各个阶段发起的投毒攻击。4.1 节介绍了本文的实验设置, 包括实验环境、数据集及投毒攻击等设置, 4.2 节给出了实验结果, 并对实验结果进行分析。

4.1 实验设置

4.1.1 实验环境设置

实验中, 我们使用 Python 的 NetworkX 库来生成无人机集群内的拓扑图, 通过调整图生成器的半径参数, 使图的平均度数与图的所需度数相差在 0.2 的公差范围内。

在实际应用中, 无人机边缘节点的位置会随着时间发生变化。我们假设它们的移动是缓慢的, 因此无人机集群内的通信拓扑图在每个全局聚合期间保持一致, 但在连续的全局聚合之间会发生变化。实验环境设置见表 1。

表 1 实验环境设置

Table 1 Experimental settings

实验环境	版本
操作系统	Linux 5.19.0-45-genetic
CPU	Intel(R) Xeon(R) Gold 6330N CPU @ 2.20GHz
GPU	NVIDIA GeForce RTX 4090
Python	3.8.15
Pytorch	1.9.0
NetworkX	3.1

4.1.2 场景与数据集划分

我们考虑了以下两种场景, 以对 Col-HFL 算法在不同网络规模下的性能进行验证。

(1) 场景 1: 终端设备的数量为 125, 每 5 个设备组成一个集群, 每个集群与 25 个无人机中的一个进行通信, 每 5 个无人机组成一个集群(125-25-5), 集群内通信拓扑图的平均度数为 3。

(2) 场景 2: 终端设备的数量为 250, 每 5 个设备组成一个集群, 每个集群与 50 个无人机中的一个进行通信, 每 10 个无人机组成一个集群(250-50-5), 集群内通信拓扑图的平均度数为 6。

我们分别在数据集 MNIST 和 Fashion-MNIST 上评估上述两种架构, 并使用全连接神经网络(NN)分类器作为全局模型。

MNIST 是一个手写体数字的图片数据集。数据集一共有 10 个类别(0～9), 由 60000 个训练样本和 10000 个测试样本组成, 每个样本都是一张 28×28 像素的灰度手写数字图片。Fashion-MNIST 中涵盖了来自 10 个类别的共 7 万个不同商品的正面图片。它的大小、格式以及训练集和测试集的划分与 MNIST 完全一致。

为了模拟设备之间不同程度的统计数据异质性, 我们考虑以下三种数据集划分方式。

(1) 极端非独立同分布($p=1$), 其中每个本地数据集仅具有来自 1 个标签的数据点;

(2) 中等非独立同分布($p=3$), 其中每个本地数据集包含来自 10 个标签中的 3 个标签的数据点;

(3) 独立同分布($p=10$), 其中每个本地数据集都有涵盖所有 10 个标签的数据点。在上述三种数据划分方式下, 每个设备上的数据量都相同。

4.1.3 投毒攻击设置

实验中, 使用参数 m 表示每个集群内恶意无人机的占比。本文假设 $m<50\%$, 这是投毒攻击防护中的一般前提, 因为当恶意参与者占比超过一半时, 任何基于相似度进行中毒参数检测的方案都不能成功抵御攻击。恶意无人机节点可以通过以下两种方式生成无针对性的中毒模型参数。

(1) 附加噪声(Additive Noise, AN), 在每个训练轮次, 恶意无人机从 $\mu=0$ 和 $\sigma=0.01$ 的高斯分布中采样并添加到真实参数中。

(2) 符号翻转(Sign Flipping, SF), 恶意无人机改变真实参数的符号, 比如把正号改成负号, 但是参数具体数值不会改变。在实际应用中, 硬件故障也可能会导致这种情况, 因此这种攻击更接近现实。

实验中, 我们考虑了以下三种攻击方式。

(1) AN/SF, 所有恶意无人机随机选择 AN 或 SF 来生成中毒参数;

(2) AN, 所有恶意无人机都使用 AN 来生成中毒参数;

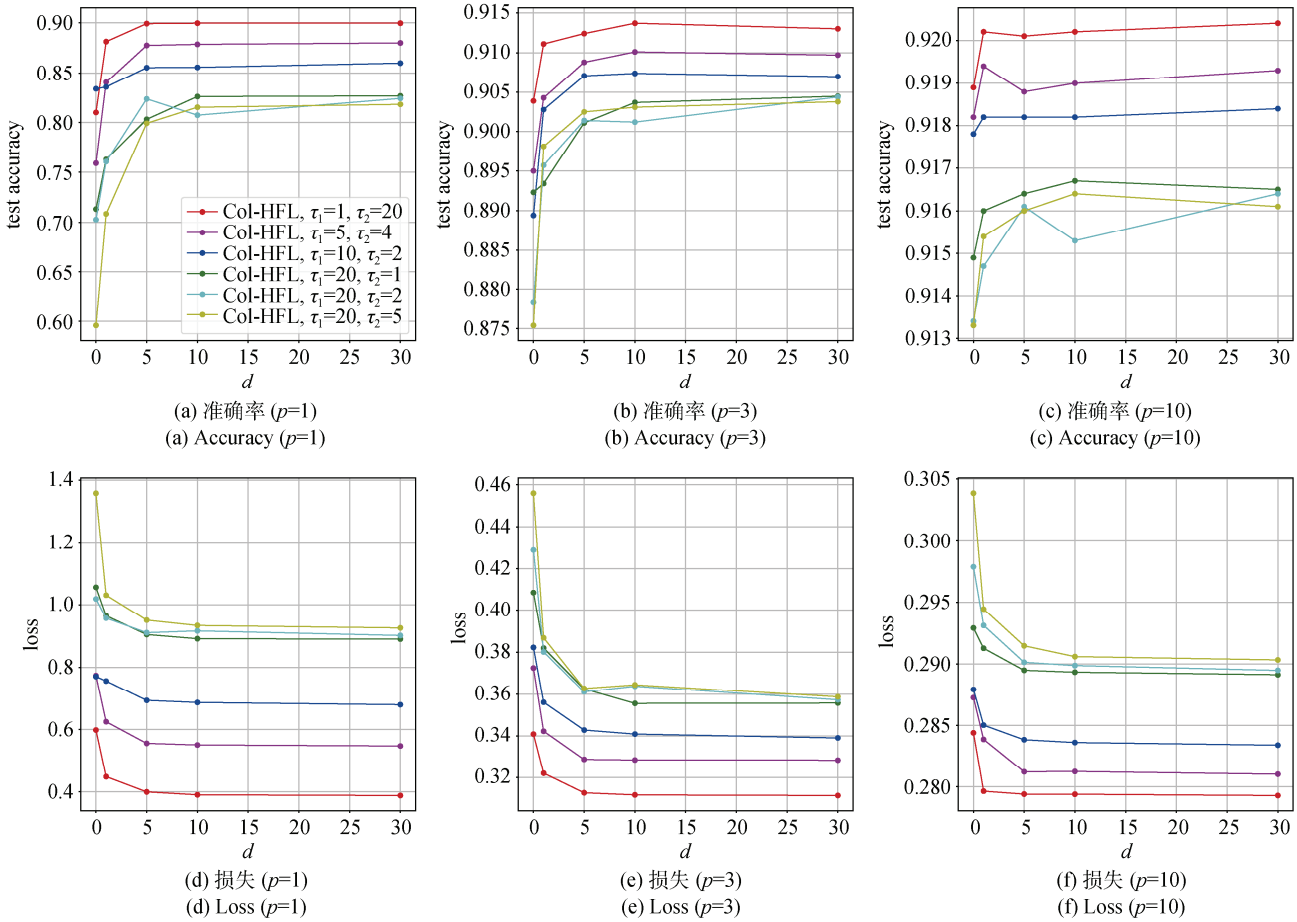
(3) SF, 所有恶意无人机都使用 SF 来生成中毒参数。

4.2 结果分析

4.2.1 参数 τ_1 和 τ_2 对 Col-HFL 算法性能的影响

τ_1(终端设备与无人机边缘节点的通信间隔)和 τ_2(无人机边缘节点与云服务器的通信间隔)是 Col-HFL 中两个重要的参数。图 5 和图 6 衡量了不同 τ_1 和 τ_2 对全局模型性能的影响。

首先, 固定终端设备与云服务器的通信间隔为 20, 即 $\tau=\tau_1\tau_2=20$。可以观察到, 对于三种不同的非独立同分布数据划分方式, 与无人机边缘节点更频繁的通信(即更小的 τ_1)可以提高全局模型的性能。这



(a) 准确率 (p=1) (a) Accuracy (p=1)
(b) 准确率 (p=3) (b) Accuracy (p=3)
(c) 准确率 (p=10) (c) Accuracy (p=10)
(d) 损失 (p=1) (d) Loss (p=1)
(e) 损失 (p=3) (e) Loss (p=3)
(f) 损失 (p=10) (f) Loss (p=10)

图 5 不同 τ_1 与 τ_2 时 Col-HFL 的性能比较(125-25-5)

Figure 5 Performance of Col-HFL under different τ_1 and τ_2 (125-25-5)

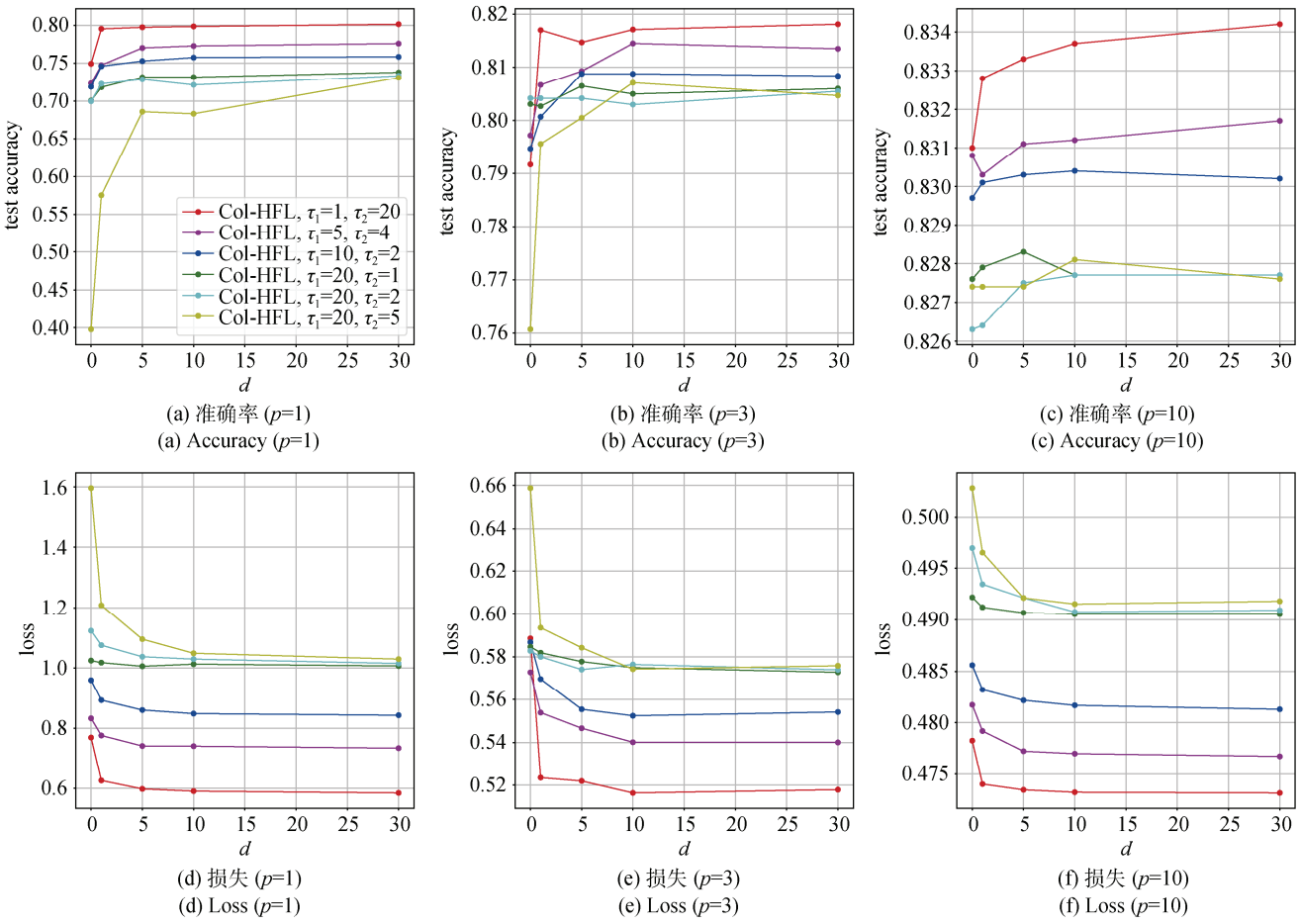


图 6　不同 τ_1 与 τ_2 时 Col-HFL 的性能比较(250-50-5)

Figure 6　Performance of Col-HFL under different τ_1 and τ_2　(250-50-5)

与前面的理论分析结果一致，τ_1 越小，无人机集群边缘共识的误差就越小，全局聚合时云服务器采样得到的边缘共识模型更接近整个集群内所有边缘聚合模型的平均值，因此对应着更小的全局模型误差。

另外，在固定 $\tau_1=20$ 的同时逐渐增大 τ_2，当 D2D 轮数 $d=30$ 时，不同 τ_2 得到的全局模型测试准确率几乎一致。在三种不同的非独立同分布数据划分方式下，$\tau_2=1$ 和 $\tau_2=5$ 的准确率差距分别接近 0.01、0.001 和 0.0005(125-25-5)，以及 0.008、0.001 和 0.0001(250-50-5)。当 d 逐渐减小时，$\tau_2=1$ 和 $\tau_2=5$ 的准确率差距则逐渐增大。该结果表明当设备之间的数据集独立同分布(意味着无人机之间的数据也是独立同分布的)且与无人机边缘服务器的通信频率固定时，如果无人机在边缘共识过程中的 D2D 通信轮数足够大(对应着较小的边缘共识误差)，那么降低无人机与云服务器的通信频率不会对 Col-HFL 的性能产生较大影响。这说明在独立同分布场景下，我们能够进一步减少与云服务器的高成本通信，而性能的损失可以忽略不计。

基于以上实验结果，在后续的实验中，我们固定 $\tau_1=1$。

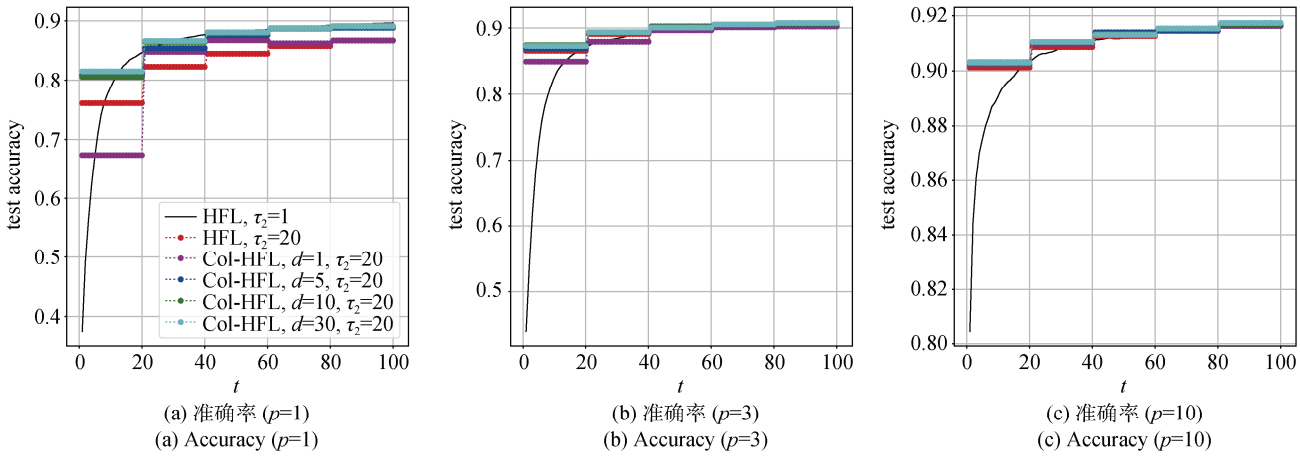
4.2.2　参数 d 对 Col-HFL 算法性能的影响

在图 7 和图 8 中，我们固定 Col-HFL 的 $\tau_2=20$，并不断增加边缘共识过程中的 D2D 轮数 d。HFL[18](基线)假设所有无人机完全参与(即所有无人机在每次全局聚合时都将参数上传到云服务器)。基线 1 在每次边缘聚合后进行全局聚合，即 $\tau_2=1$，基线 2 的 $\tau_2=20$。实验结果验证了无人机集群内的 D2D 通信可以显著提高机器学习模型的性能。具体来说，当数据分布是中等非独立同分布和极端非独立同分布时，可以看到，在全局聚合间隔不变的情况下，增加 d 可以提高训练模型的性能。另外，随着 Col-HFL 的性能接近基线 1 时，增加 d 带来的模型性能增益会递减。最后，可以观察到，在图(c)和图(f)中，模型准确率与损失不会随着 d 的增加而产生明显变化。也就是说，仅当终端设备间的数据分布为非独立

同分布($p=1,3$)时, 通过 D2D 通信获得的增益才会出现。该部分的实验结果与上面的理论分析一致, 即增加无人机集群的 D2D 轮数会产生更小的边缘共识误差, 从而提高全局模型的性能。

4.2.3 Col-HFL 与 HFL 的对比

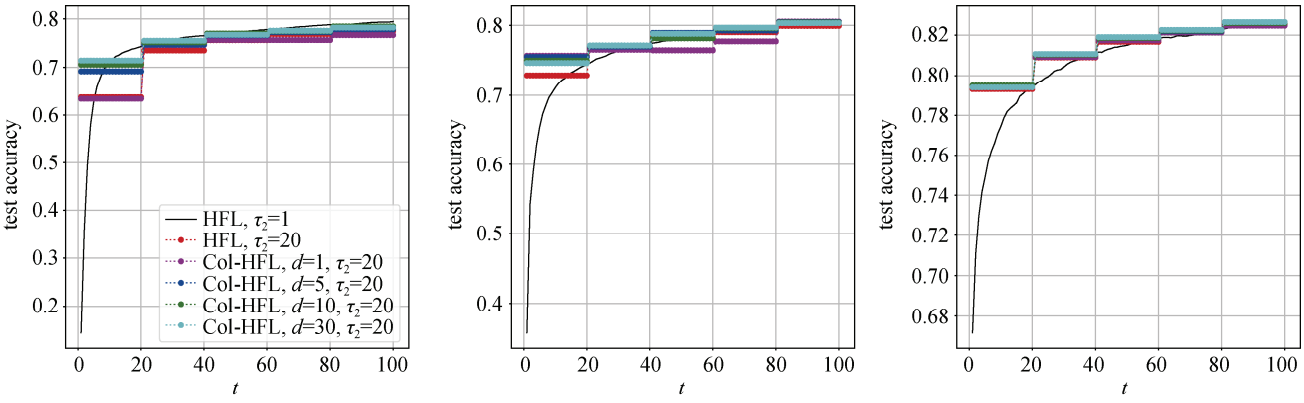
在图 9 和图 10 中, 我们将增加全局聚合间隔的 Col-HFL 性能与不利用 D2D 边缘共识过程的基线进行比较。由于基线中所有无人机都需要上传参数到云服务器, 因此每次全局聚合时的上行链路资源密集度是 Col-HFL 的 5 倍(125-25-5)和 10 倍(250-50-5)。我们随着 τ_2 的增加而增加 Col-HFL 中 D2D 的轮数 d。实验结果证实, 当全局聚合频率降低时(即更大的 τ_2), Col-HFL 仍然可以优于基线 2, 并非常接近基线 1。例如在图 9(a)中, 在终端设备本地训练 100 轮后, τ_2



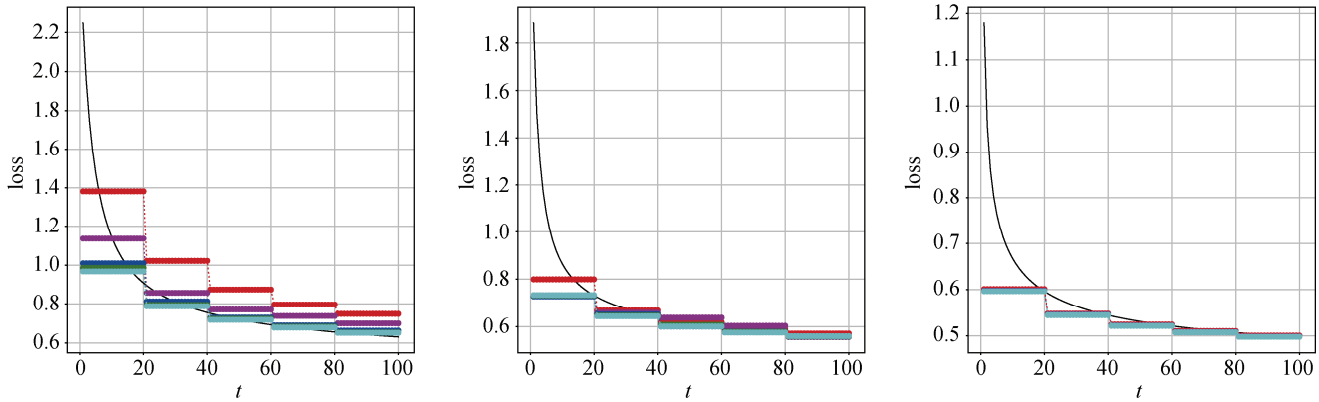
(a) 准确率 (p=1) (a) Accuracy (p=1)
(b) 准确率 (p=3) (b) Accuracy (p=3)
(c) 准确率 (p=10) (c) Accuracy (p=10)
(d) 损失 (p=1) (d) Loss (p=1)
(e) 损失 (p=3) (e) Loss (p=3)
(f) 损失 (p=10) (f) Loss (p=10)

图 7 不同 d 时 Col-HFL 的性能比较(125-25-5)

Figure 7 Performance of Col-HFL under different d (125-25-5)



(a) 准确率 (p=1) (a) Accuracy (p=1)
(b) 准确率 (p=3) (b) Accuracy (p=3)
(c) 准确率 (p=10) (c) Accuracy (p=10)



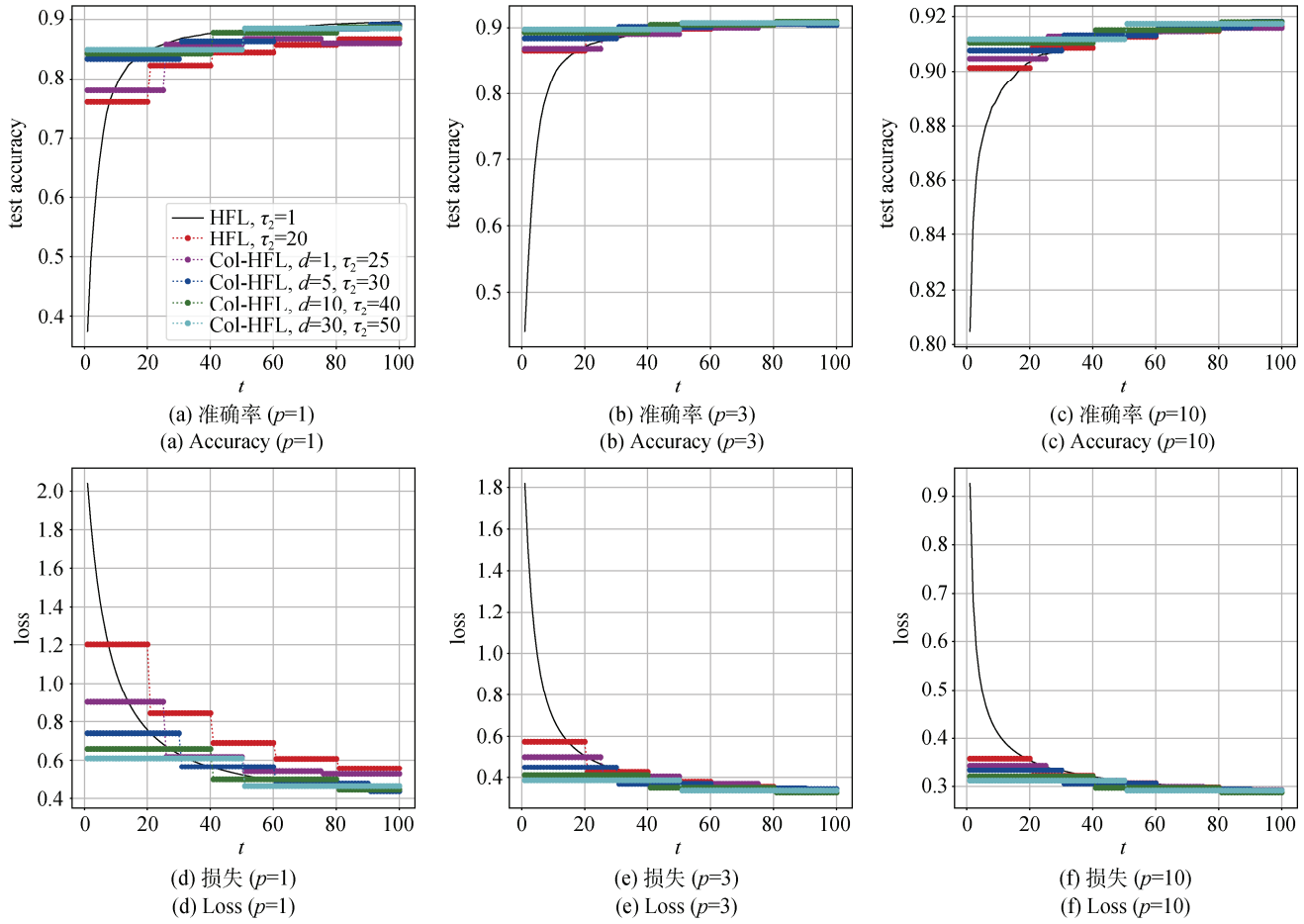
(d) 损失 (p=1)
(d) Loss (p=1)

(e) 损失 (p=3)
(e) Loss (p=3)

(f) 损失 (p=10)
(f) Loss (p=10)

图 8　不同 d 时 Col-HFL 的性能比较(250-50-5)

Figure 8　Performance of Col-HFL under different d (250-50-5)



(a) 准确率 (p=1)
(a) Accuracy (p=1)

(b) 准确率 (p=3)
(b) Accuracy (p=3)

(c) 准确率 (p=10)
(c) Accuracy (p=10)

(d) 损失 (p=1)
(d) Loss (p=1)

(e) 损失 (p=3)
(e) Loss (p=3)

(f) 损失 (p=10)
(f) Loss (p=10)

图 9　Col-HFL 与 HFL 的性能比较(125-25-5)

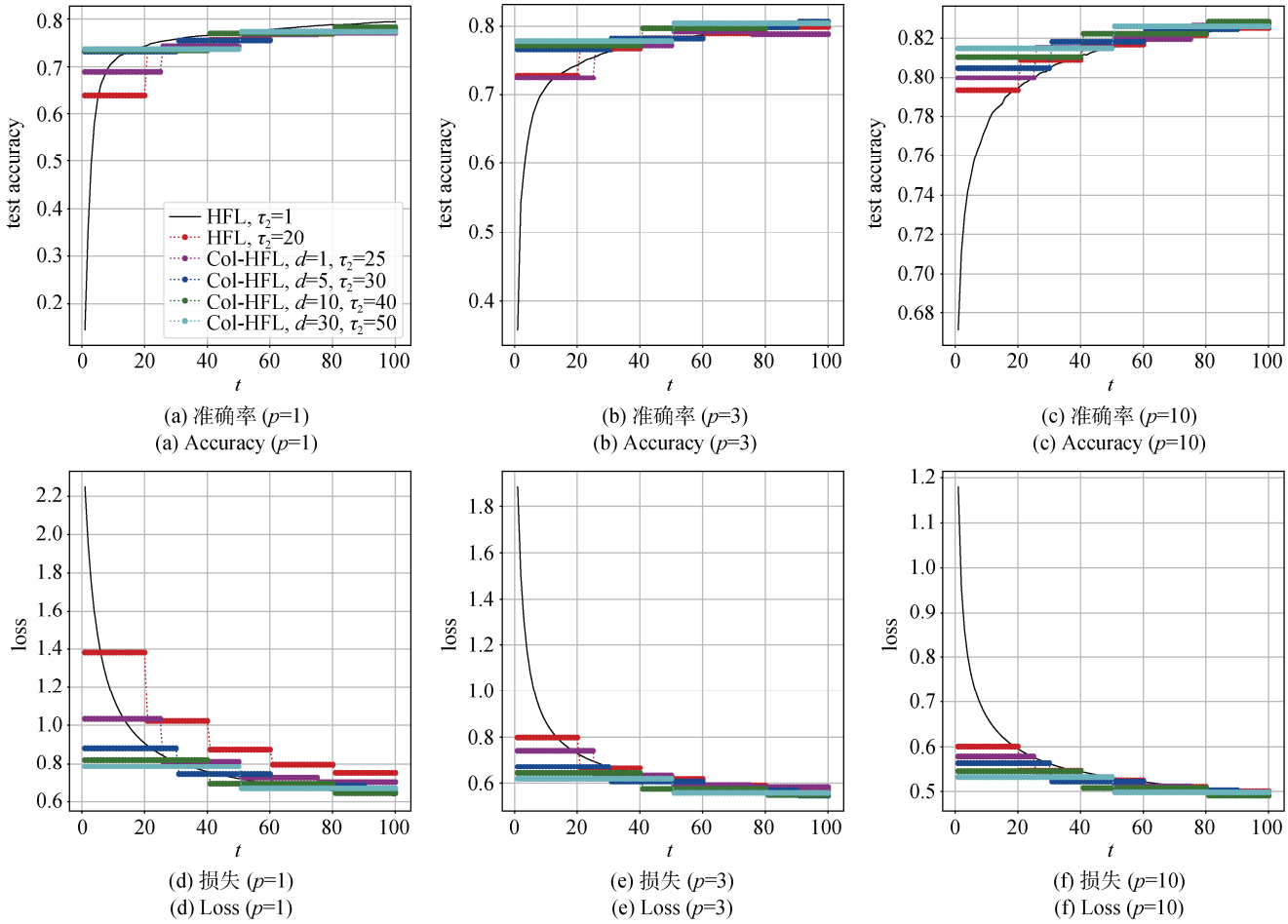
Figure 9　Performance comparison of Col-HFL and HFL (125-25-5)

为 50 的 Col-HFL 的全局模型准确率在 90%左右, 而基线 2 的全局模型准确率仅为 86%。由此可以得出结论: 可以通过更高程度的局部共识过程(更大的 d)来抵消全局聚合间隔增加对全局模型性能的影响。

在终端设备本地训练 200 轮后, 我们对比了不同算法在极端非独立同分布场景下(p = 1)得到的全局模型的准确率、训练时间以及无人机到云服务器的上行链路中传输的模型参数数量。

由表 2 中可以观察到, Col-HFL 能够在保持与 HFL 相近的模型准确率的同时减少训练时长和通信

开销。例如当 $d=1$, $\tau_2=25$ 时, Col-HFL 与基线 1 的模型准确率只相差了 1%左右, 而训练时间却减少了约 41%。在整个训练过程中, Col-HFL 只需进行 8 次全局聚合, 每次全局聚合时每个集群内选择一个无人机上传模型参数, 所以在整个训练过程中架构 125-25-5 和架构 250-50-5 中无人机到云服务器的上行链路中传输的模型参数数量都为 40。基线 1 则需要 200 次全局聚合, 每次全局聚合时所有无人机都需上传模型参数, 所以在整个训练过程中架构 125-25-5 和架构 250-50-5 需要上传的参数数量分别为 5000 和 10000。由此可见, Col-HFL 能够显著降低大规模联邦学习系统的时间开销及通信开销。



(a) 准确率 (p=1) (a) Accuracy (p=1)
(b) 准确率 (p=3) (b) Accuracy (p=3)
(c) 准确率 (p=10) (c) Accuracy (p=10)
(d) 损失 (p=1) (d) Loss (p=1)
(e) 损失 (p=3) (e) Loss (p=3)
(f) 损失 (p=10) (f) Loss (p=10)

图 10 Col-HFL 与 HFL 的性能比较(250-50-5)

Figure 10 Performance comparison of Col-HFL and HFL (250-50-5)

表 2 Col-HFL 与 HFL 的模型准确率与时间开销比较

Table 2 Comparison of Col-HFL and HFL in terms of their model accuracy and time cost

算法	模型准确率		时间开销(s)		上行链路传输的模型参数数量	
	125-25-5	250-50-5	125-25-5	250-50-5	125-25-5	250-50-5
基线 1: HFL($\tau_2=1$)	90.31%	81.14%	574	595	5000	10000
基线 2: HFL($\tau_2=20$)	88.23%	79.05%	335	343	250	500
Col-HFL($d=1$, $\tau_2=25$)	**89%**	**79.33%**	**334**	**353**	**40**	**40**
Col-HFL($d=5$, $\tau_2=30$)	89.69%	79.51%	345	385	35	35
Col-HFL($d=10$, $\tau_2=40$)	89.79%	79.83%	361	418	25	25
Col-HFL($d=30$, $\tau_2=50$)	89.82%	79.74%	420	572	20	20

4.2.4 算法 2 针对边缘共识阶段的投毒攻击的防御效果

为验证 TSRA 算法对于数据异质的鲁棒性, 我们在极端非独立同分布($p=1$)的情况下对本文提出的投毒攻击防御算法进行验证, 并固定 $\tau_1=1$, $\tau_2=20$, $d=30$, 终端设备本地训练的轮数为 200。因此在整个训练过程中, 全局模型会更新 10 次。

当恶意无人机在 D2D 边缘共识过程中投毒时, 我们在无人机本地使用基于历史参数的鲁棒聚合算法($b=3$, 即无人机从第三轮 D2D 通信开始恶意邻居节点的检测), 并设置平均聚合(mean)、中位数聚合(median)、几何中位数聚合(geometry median, gm)、krum 和 multi-krum 这 5 种典型的投毒攻击防御方法作为对比方法。图 11 和图 12 表明, 我们的方法能够在边缘共识阶段有效地剔除掉中毒参数, 且防护效果优于其他对比方法。此时, 因为没有在云服务器上部署鲁棒全局聚合算法, 所以随着 m 的增加, 恶意无人机在全局聚合阶段被选中上传参数的可能性也增加, 全局模型的性能会有一定程度的下降。

另外, 观察不同攻击方式下的结果, 可以发现通过 SF 生成的中毒参数的攻击效果远强于 AN 生成的中毒参数。随着 m 的增加, 我们的鲁棒聚合算法在 AN 攻击下准确率会下降 1%～2%, 而在 SF 攻击下准确率会下降 20%左右, 在 AN/SF 攻击下的准确率会下降 10%左右, 介于前两者之间。这是因为在实验中, 当通过 AN 攻击生成中毒参数时, 仅有少量的随机噪声(均值为 0, 方差为 0.01)被添加到真实的模型参数中, 因此恶意无人机的中毒参数与真实的模型参数差距较小, 对全局模型的影响也较小。与 AN 攻击不同, 通过 SF 攻击生成的中毒参数的符号与真实模型参数都是相反的, 因此一旦被聚合, 就会导致全局模型的性能大幅度下降。接下来, 我们更进一步讨论在全局聚合阶段的投毒攻击防护。

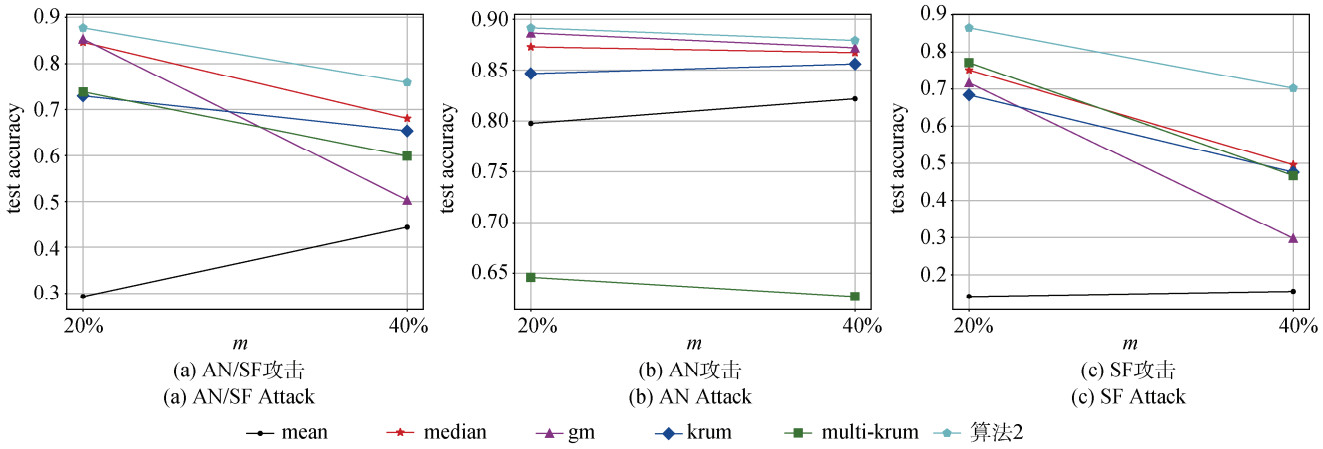


图 11 算法 2 针对边缘共识阶段的投毒攻击的防御效果(125-25-5)

Figure 11 Defense effect of algorithm 2 at the edge consensus process (125-25-5)

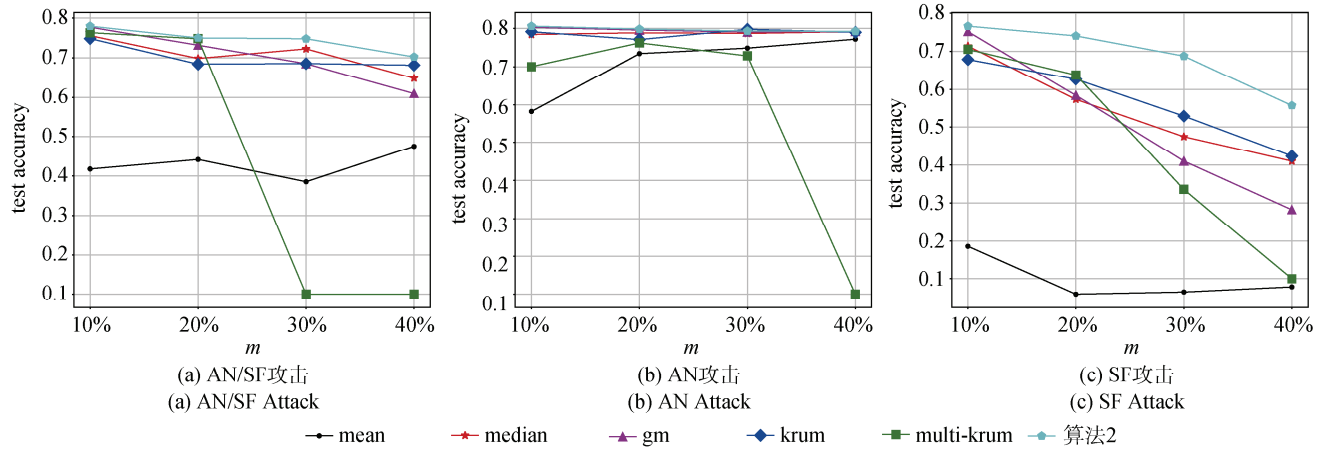


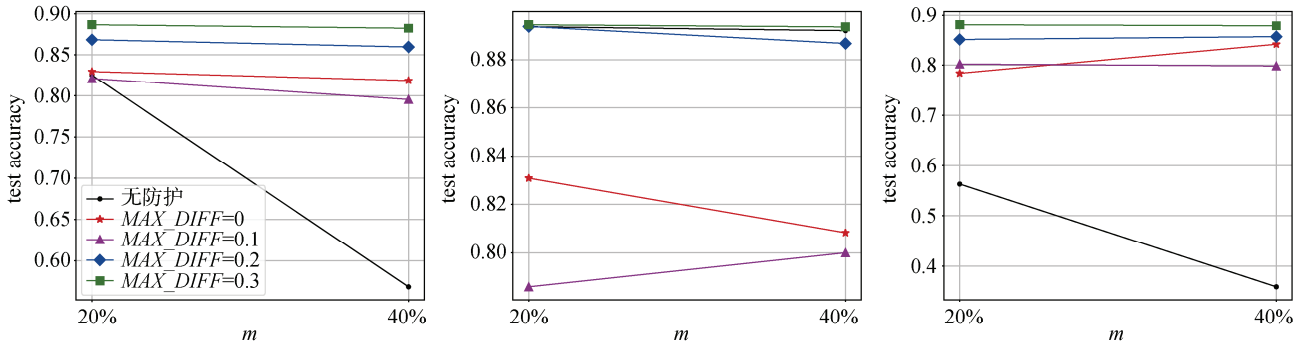
图 12 算法 2 针对边缘共识阶段的投毒攻击的防御效果(250-50-5)

Figure 12 Defense effect of algorithm 2 at the edge consensus process (250-50-5)

4.2.5 算法 3 针对全局聚合阶段的投毒攻击的防御效果

我们在恶意无人机对云服务器投毒的同时在云服务器上部署基于声誉系统的聚合算法。依次设置 MAX_{DIFF} 为 0、0.1、0.2 和 0.3，并与没有防御算法的 Col-HFL(云服务器从每个无人机集群中随机选择一个无人机上传参数，并以平均聚合所有边缘共识模型的方式更新全局模型)进行对比。

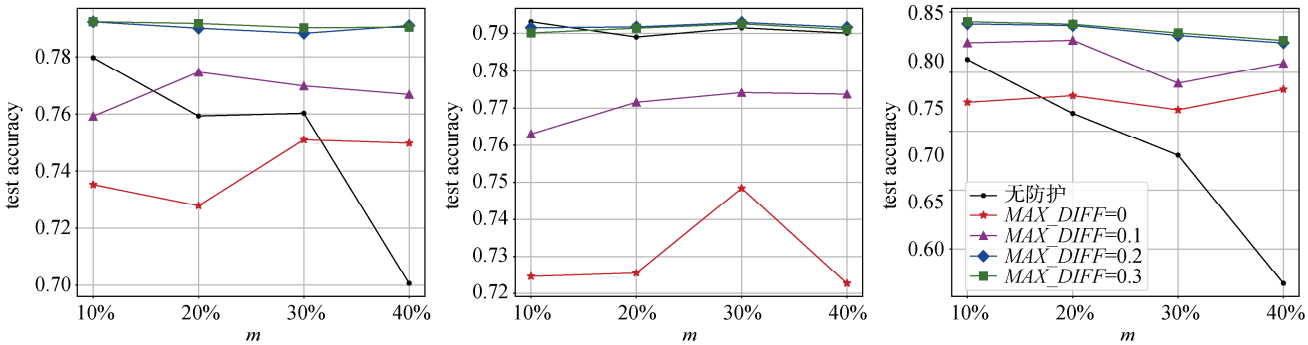
从图 13 和图 14 可以看出，我们的鲁棒全局聚合方法能够在全局聚合阶段有效地检测出恶意无人机上传的中毒参数。另外，当 MAX_{DIFF} 设置得过小时，添加了防御的 Col-HFL 性能会比没有防御的 Col-HFL 性能还要差。这是由于在数据异质时，良性参数在验证数据集上的准确率也可能出现正常的下降现象，过小的 MAX_{DIFF} 会使得云服务器将此类参数错误地判定为中毒参数并抛弃，因此全局模型性能下降。根据实验结果，我们在接下来的实验中设置 $MAX_{DIFF}=0.3$。



(a) AN/SF攻击 (a) AN/SF Attack　(b) AN攻击 (b) AN Attack　(c) SF攻击 (c) SF Attack

图 13　算法 3 针对全局聚合阶段的投毒攻击的防御效果(125-25-5)

Figure 13　Defense effect of algorithm 3 at the global aggregation process (125-25-5)



(a) AN/SF攻击 (a) AN/SF Attack　(b) AN攻击 (b) AN Attack　(c) SF攻击 (c) SF Attack

图 14　算法 3 针对全局聚合阶段的投毒攻击的防御效果(250-50-5)

Figure 14　Defense effect of algorithm 3 at the global aggregation process (250-50-5)

4.2.6 双阶段鲁棒聚合算法的防御效果

当恶意无人机在 D2D 边缘共识过程中进行投毒攻击时，我们在无人机本地部署算法 2，并对比在云服务器上部署/不部署算法 3 时 Col-HFL 的性能。从表 3 和表 4 可以看出，与仅在边缘共识时使用基于历史参数的鲁棒聚合算法相比，在全局聚合时使用基于声誉系统的鲁棒聚合后，对于任何 m，全局模型的性能都会得到一定程度的提升。在集群内恶意无人机占比为 40%时，全局模型准确率仍然可以达到未受到攻击时的水平(仅下降 1%～3%)。

由以上结果可知，我们提出的双阶段鲁棒聚合算法能够有效地防御各个阶段存在的投毒攻击，并且对数据异质也具有鲁棒性。

5 总结

本文提出了一种基于 D2D 的协作式层次联邦学习算法(Col-HFL)，它通过集群内的分布式共识来缓解通信状态不稳定的无人机对全局模型的影响，同时减小了全局聚合频率和无人机上行链路的数据传输量。其次，我们针对架构在不同环节存在的投毒攻

表 3　部署/不部署全局鲁棒聚合算法时的防御效果 (125-25-5)

Figure 3　Defense effect with/without global robust aggregation algorithm (125-25-5)

m	AN/SF		AN		SF	
	w/o	*w*	*w/o*	*w*	*w/o*	*w*
m = 20%	80.01%	90.1%	89.15%	90.1%	86.45%	89.84%
m = 40%	75.86%	87.75%	87.92%	88.74%	70.22%	86.43%

表 4　部署/不部署全局鲁棒聚合算法时的防御效果 (250-50-5)

Figure 4　Defense effect with/without global robust aggregation algorithm (250-50-5)

m	AN/SF		AN		SF	
	w/o	*w*	*w/o*	*w*	*w/o*	*w*
m = 10%	77.99%	80.74%	80.62%	80.69%	76.59%	80.86%
m = 20%	75.04%	80.79%	79.87%	80.91%	74.05%	80.93%
m = 30%	74.84%	80.46%	79.35%	80.41%	68.78%	80.57%
m = 40%	70.27%	80.82%	79.28%	80.21%	55.72%	80.64%

击提出了一种双阶段鲁棒聚合算法(TSRA), 在无人机本地和云服务器上分别部署基于历史参数和声誉系统的鲁棒聚合算法。最后, 在真实数据集上的实验结果证明了 Col-HFL 在不同场景下对训练模型性能和通信开销的改进, 以及针对终端设备之间数据异质的鲁棒性。同时, TSRA 能够有效地在边缘共识和全局聚合期间剔除掉中毒模型参数, 保障全局模型的安全性。

参考文献

[1] McMahan H B, Moore E, Ramage D, et al. Communication-Efficient Learning of Deep Networks from Decentralized Data[C]. *International Conference on Artificial Intelligence and Statistics*, 2016.

[2] Wu X Y, Huang H G, Ding Y L, et al. FedNP: Towards Non-IID Federated Learning *via* Federated Neural Propagation[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023, 37(9): 10399-10407.

[3] Li Q B, Diao Y Q, Chen Q, et al. Federated Learning on Non-IID Data Silos: An Experimental Study[C]. *2022 IEEE 38th International Conference on Data Engineering*, 2022: 965-978.

[4] Hsieh K, Phanishayee A, Mutlu O, et al. The Non-IID Data Quagmire of Decentralized Machine Learning[EB/OL]. 2019: arXiv: 1910.00189. https://arxiv.org/abs/1910.00189.

[5] Hsu T H, Qi H, Brown M. Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification[EB/OL]. 2019: arXiv: 1909.06335. https://arxiv.org/abs/1909.06335.

[6] Yemini M, Saha R, Ozfatura E, et al. Semi-Decentralized Federated Learning with Collaborative Relaying[C]. *2022 IEEE International Symposium on Information Theory*, 2022: 1471-1476.

[7] Rodríguez-Barroso N, Jiménez-López D, Luzón M V, et al. Survey on Federated Learning Threats: Concepts, Taxonomy on Attacks and Defences, Experimental Study and Challenges[J]. *Information Fusion*, 2023, 90: 148-173.

[8] Kairouz P, McMahan H B, Avent B, et al. Advances and Open Problems in Federated Learning[J]. *Foundations and Trends® in Machine Learning*, 2021, 14(1/2): 1-210.

[9] Cao X Y, Jia J Y, Zhang Z X, et al. FedRecover: Recovering from Poisoning Attacks in Federated Learning Using Historical Information[C]. *2023 IEEE Symposium on Security and Privacy*, 2023: 1366-1383.

[10] Shejwalkar V, Houmansadr A, Kairouz P, et al. Back to the Drawing Board: A Critical Evaluation of Poisoning Attacks on Production Federated Learning[C]. *2022 IEEE Symposium on Security and Privacy*, 2022: 1354-1371.

[11] Chen Z Y, Liao W X, Hua K, et al. Towards Asynchronous Federated Learning for Heterogeneous Edge-Powered Internet of Things[J]. *Digital Communications and Networks*, 2021, 7(3): 317-326.

[12] Hao J S, Zhao Y C, Zhang J L. Time Efficient Federated Learning with Semi-Asynchronous Communication[C]. *2020 IEEE 26th International Conference on Parallel and Distributed Systems*, 2021: 156-163.

[13] Imteaj A, Hadi Amini M. FedAR: Activity and Resource-Aware Federated Learning Model for Distributed Mobile Robots[C]. *2020 19th IEEE International Conference on Machine Learning and Applications*, 2021: 1153-1160.

[14] Wu W T, He L G, Lin W W, et al. SAFA: A Semi-Asynchronous Protocol for Fast Federated Learning with Low Overhead[J]. *IEEE Transactions on Computers*, 2021, 70(5): 655-668.

[15] Fung C, Yoon C J M, Beschastnikh I. Mitigating Sybils in Federated Learning Poisoning[EB/OL]. 2018: arXiv: 1808.04866. https://arxiv.org/abs/1808.04866.

[16] Blanchard P, El Mhamdi E M, Guerraoui R, et al. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent[C]. *Neural Information Processing Systems*, 2017

[17] Cao X Y, Fang M H, Liu J, et al. FLTrust: Byzantine-Robust Federated Learning *via* Trust Bootstrapping[C]. *Proceedings 2021 Network and Distributed System Security Symposium*, 2021: 1-18.

[18] Liu L M, Zhang J, Song S H, et al. Client-Edge-Cloud Hierarchical Federated Learning[C]. *ICC 2020 - 2020 IEEE International Conference on Communications*, 2020: 1-6.

[19] Bonawitz K, Eichner H, Grieskamp W, et al. Towards Federated Learning at Scale: System Design[EB/OL]. 2019: arXiv: 1902.01046. https://arxiv.org/abs/1902.01046.

[20] Sun Z T, Kairouz P, Suresh A T, et al. Can You Really Backdoor Federated Learning? [EB/OL]. 2019: arXiv: 1911.07963. https://arxiv.org/abs/1911.07963.

[21] Nguyen T D, Nguyen T A, Tran A, et al. IBA: Towards Irreversible Backdoor Attacks in Federated Learning[C]. *37th Conference on Neural Information Processing Systems*, 2023, 1-13.

[22] Wang H Y, Sreenivasan K, Rajput S, et al. Attack of the Tails: Yes, You Really Can Backdoor Federated Learning[EB/OL]. 2020: arXiv: 2007.05084. https://arxiv.org/abs/2007.05084.

[23] Fang P, Chen J H. On the Vulnerability of Backdoor Defenses for Federated Learning[C]. *The Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, 2023: 11800-11808.

[24] Bhagoji A N, Chakraborty S, Mittal P, et al. Analyzing Federated Learning through an Adversarial Lens[EB/OL]. 2018: arXiv: 1811.12470. https://arxiv.org/abs/1811.12470.

[25] Tomsett R, Chan K S, Chakraborty S. Model Poisoning Attacks Against Distributed Machine Learning Systems[C]. *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 2019: 46.

[26] El Mhamdi E M, Guerraoui R, Rouault S. The Hidden Vulnerability of Distributed Learning in Byzantium[EB/OL]. 2018: arXiv: 1802.07927. https://arxiv.org/abs/1802.07927.

[27] Yu Y, Liu Q, Wu L K, et al. Untargeted Attack Against Federated Recommendation Systems *via* Poisonous Item Embeddings and the Defense[C]. *The Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, 2023: 4854-4863.

[28] Baruch M, Baruch G, Goldberg Y. A Little Is Enough: Circumventing Defenses for Distributed Learning[EB/OL]. 2019: arXiv: 1902.06156. https://arxiv.org/abs/1902.06156.

[29] Wei K, Li J, Ding M, et al. Covert Model Poisoning Against Federated Learning: Algorithm Design and Optimization[J]. *IEEE Transactions on Dependable and Secure Computing*, 2024, 21(3): 1196-1209.

[30] Yin D, Chen Y D, Ramchandran K, et al. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates[EB/OL]. 2018: arXiv: 1803.01498. https://arxiv.org/abs/1803.01498.

[31] Chen Y D, Su L L, Xu J M. Distributed Statistical Machine Learning in Adversarial Settings: Byzantine Gradient Descent[J]. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2017, 1(2): 1-25.

[32] Hosseinalipour S, Azam S S, Brinton C G, et al. Multi-Stage Hybrid Federated Learning over Large-Scale D2D-Enabled Fog Networks[J]. *IEEE/ACM Transactions on Networking*, 2022, 30(4): 1569-1584.

[33] Xiao L, Boyd S. Fast Linear Iterations for Distributed Averaging[J]. *Systems & Control Letters*, 2004, 53(1): 65-78.

[34] Lin F P, Hosseinalipour S, Azam S S, et al. Semi-Decentralized Federated Learning with Cooperative D2D Local Model Aggregations[J]. *IEEE Journal on Selected Areas in Communications*, 2021, 39(12): 3851-3869.

梁梦晴 于 2021 年在北京交通大学信息安全(保密技术)专业获得学士学位。现在北京交通大学网络空间安全专业攻读硕士学位。研究领域为联邦学习、隐私计算。Email: 21120475@bjtu.edu.cn

王健 于 2008 年在北京邮电大学密码学专业获得博士学位。现任北京交通大学副教授、博士生导师。研究领域为数据安全及隐私计算、网络安全、密码应用及区块链。Email: wangjian@bjtu.edu.cn

江文彬 于 2020 年在北京交通大学信息安全专业获得本科学位。现在北京交通大学网络空间安全专业攻读博士学位。研究领域为联邦学习、数据估值、机器遗忘。Email: wenbin.jiang@bjtu.edu.cn

王雪微 于 2021 年在北京交通大学信息安全专业获得工学学士学位。现在北京交通大学网络空间安全专业攻读硕士学位。研究领域为联邦学习、区块链。Email: 21120487@bjtu.edu.cn

刘吉强 于 1999 年在北京师范大学数学所获得博士学位。现任北京交通大学软件学院院长。CCF 会员，研究领域为隐私保护、可信计算、物联网安全。Email: jqliu@bjtu.edu.cn