软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software [doi: 10.13328/j.cnki.jos.007186]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于梯度放大的联邦学习激励欺诈攻击与防御*

乐紫莹[1,2,3], 陈 珂[1,2,3], 寿黎但[1,2,3], 骆歆远[1,2,3], 陈 刚[1,2,3]

[1](浙江大学, 浙江 杭州 310027)

[2](区块链与数据安全全国重点实验室 (浙江大学), 浙江 杭州 310027)

[3](浙江省大数据智能计算重点实验室 (浙江大学), 浙江 杭州 310027)

通信作者: 陈珂, E-mail: chenk@zju.edu.cn

摘 要: 在联邦学习领域, 激励机制是吸引高质量数据持有者参与联邦学习并获得更优模型的重要工具. 然而, 现有的联邦学习研究鲜有考虑到参与者可能滥用激励机制的情况, 也就是他们可能会通过操纵上传的本地模型信息来获取更多的奖励. 针对这一问题进行了深入研究. 首先, 明确定义联邦学习中的参与者激励欺诈攻击问题, 并引入激励成本比来评估不同激励欺诈攻击方法的效果以及防御方法的有效性. 其次, 提出一种名为"梯度放大攻击(gradient scale-up attack)"的攻击方法, 专注于对模型梯度进行激励欺诈. 这种攻击方法计算出相应的放大因子, 并利用这些因子来提高本地模型梯度的贡献, 以获取更多奖励. 最后, 提出一种高效的防御方法, 通过检验模型梯度的二范数值来识别欺诈者, 从而有效地防止梯度放大攻击. 通过对 MNIST 等数据集进行详尽地分析和实验验证, 研究结果表明, 所提出的攻击方法能够显著提高奖励, 而相应的防御方法能够有效地抵制欺诈参与者的攻击行为.

关键词: 联邦学习; 激励欺诈攻击; 梯度放大攻击; 恶意参与者检测; 安全保护

中图法分类号: TP309

中文引用格式: 乐紫莹, 陈珂, 寿黎但, 骆歆远, 陈刚. 基于梯度放大的联邦学习激励欺诈攻击与防御. 软件学报. http://www.jos.org.cn/1000-9825/7186.htm

英文引用格式: Yue ZY, Chen K, Shou LD, Luo XY, Chen G. Reward Fraud Attack and Defense for Federated Learning Based on Gradient Scale-up. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7186.htm

Reward Fraud Attack and Defense for Federated Learning Based on Gradient Scale-up

YUE Zi-Ying[1,2,3], CHEN Ke[1,2,3], SHOU Li-Dan[1,2,3], LUO Xin-Yuan[1,2,3], CHEN Gang[1,2,3]

[1](Zhejiang University, Hangzhou 310027, China)

[2](State Key Laboratory of Blockchain and Data Security (Zhejiang University), Hangzhou 310027, China)

[3](Key Laboratory of Big Data Intelligent Computing of Zhejiang Province (Zhejiang University), Hangzhou 310027, China)

Abstract: In the field of federated learning, incentive mechanisms play a crucial role in enticing high-quality data contributors to engage in federated learning and acquire superior models. However, existing research in federated learning often neglects the potential misuse of these incentive mechanisms. Specifically, participants may manipulate their locally trained models to dishonestly maximize their rewards. This issue is thoroughly examined in this study. Firstly, the problem of rewards fraud in federated learning is clearly defined, and the concept of reward-cost ratio is introduced to assess the effectiveness of various rewards fraud techniques and defense mechanisms. Following this, an attack method named the "gradient scale-up attack" is proposed, focusing on manipulating model gradients to exploit the incentive system. This attack method calculates corresponding scaling factors and utilizes them to increase the contribution of the local model to gain more rewards. Finally, an efficient defense mechanism is proposed, which identifies malicious participants by examining the L_2-norms of model updates, effectively thwarting gradient scale-up attacks. Through extensive analysis and experimental validation on datasets such as MNIST, the findings of this research demonstrate that the proposed attack method significantly increases rewards, while

* 基金项目: 浙江省"尖兵"计划 (2024C01021)
收稿时间: 2023-09-28; 修改时间: 2023-11-10, 2024-01-12; 采用时间: 2024-03-26; jos 在线出版时间: 2024-09-14

the corresponding defense method effectively mitigates fraudulent behavior by participants.

Key words: federated learning; reward fraud attack; gradient scale-up attack (GSupA); malicious participant detection; security protection

随着人工智能相关技术的快速发展和广泛应用, 人工智能应用已在制造、交通、互联网等多个领域产生了积极深刻的影响. 人工智能的发展离不开大数据的积累, 而现有的数据孤岛问题使数据信息难以交流整合, 阻碍了人工智能的未来发展. 为了解决人工智能中存在的数据孤岛问题, 谷歌在 2016 年提出了联邦学习框架[1,2]. 在联邦学习框架中, 各参与者使用本地数据集训练模型并将模型梯度上传至服务器, 服务器对各参与者的本地模型梯度进行聚合操作来更新全局模型梯度, 由此各参与者在不转移本地数据的情况下合作训练了一个联合模型, 能够保护数据隐私安全. 联邦学习最终得到的结果很大程度上依赖于各参与者的本地数据和本地模型梯度, 为了吸引更多优质数据持有者参与联邦学习, 提高各方参与训练的积极性和参与水平, 联邦学习框架使用了激励机制[3]对参与者进行奖励, 其中服务器会根据各参与者的本地模型梯度、训练数据量等信息来评估其贡献大小并给予相应的激励. 公平合理的激励机制能够帮助提高参与者的参与水平, 减少低价值数据对模型的影响, 从而得到更好的联邦学习成果.

目前在联邦学习相关的研究工作中, 激励机制的设计已有了大量的研究成果, 同时在其攻击防御领域也开展了广泛的研究工作. 现有的联邦学习攻击方法的研究工作主要分为两大类: 针对数据隐私的推理攻击与损害系统鲁棒性的投毒攻击. 相应地, 防御方法的研究也主要针对这两类攻击, 例如为了防止恶意参与者上传劣质的或是具有破坏性的模型梯度来影响最终模型的质量, 联邦学习通常会采用一定的防御机制来对参与者上传的模型梯度进行检验, 只有通过检验的梯度才会被用于全局模型更新[4,5], 或者调整全局模型更新算法来降低恶意参与者上传的模型梯度对全局模型的影响[6−8]. 然而, 在现有的联邦学习攻击防御的研究工作中, 很少有考虑到参与者针对激励进行的攻击行为. 例如, 恶意参与者在不降低模型梯度质量的基础上适当修改模型相关信息, 可以在通过现有防御机制的情况下提高其所获激励, 一定程度上也对联邦学习系统进行了破坏. 为此, 本文对联邦学习中参与者的激励欺诈攻击行为进行了研究, 定义了激励欺诈攻击问题: 在满足联邦学习系统对本地梯度精度要求的前提下, 参与者为了获得更高的激励在其可接受的攻击计算代价下对要上传的本地模型信息进行处理. 基于此, 使用同时考虑了攻击激励和计算成本的激励成本比可以帮助评估激励欺诈攻击方法的优劣和防御方法的有效性.

联邦学习中, 服务器会对各参与者的贡献进行评估并给予相应的激励, 目前主要基于参与者的本地模型梯度、训练数据量等信息, 通过计算不同联合模型在系统测试集上的准确率、损失函数值等数据来评估各参与者的贡献大小, 其中本地模型梯度是影响参与者所获激励的关键因素. 基于此, 本文提出了一种针对模型梯度进行攻击的激励欺诈攻击方法——梯度放大攻击 (gradient scale-up attack, GSupA). 攻击者在已知联邦学习系统所采用的激励分配机制和防御机制的前提下, 计算获取相应的放大因子, 并使用该放大因子对本地模型的全部或部分网络层的梯度进行放大处理来提高其对全局模型的影响[9], 从而在一定的计算代价下大幅提升所获激励, 满足激励欺诈攻击的要求. 这一欺诈攻击方法与现有模型梯度攻击的攻击方式相似, 都是通过修改模型梯度进行攻击, 但与现有的模型梯度攻击不同的是, 该方法并不关心上传的模型梯度是否会影响聚合后得到的全局模型性能, 其目的为攻击联邦学习的激励机制, 获得更多的激励效益.

由于两种攻击的攻击目标不同, 所以传统模型梯度攻击 (包括后门攻击) 的防御机制无法有效防御激励欺诈攻击. 现有防御机制针对的是影响模型性能的投毒攻击, 在进行梯度检验和削弱梯度影响时针对的是会影响模型性能的梯度, 而激励欺诈攻击可以上传不会影响模型性能的梯度, 也就无法被现有的防御机制检测出来, 因此需要设计针对性的防御机制来有效防御激励欺诈攻击. 针对上述提出的激励欺诈攻击方法, 本文同时设计了相应的防御方法. 梯度放大攻击的核心思想是放大模型梯度来提高贡献, 而放大梯度也会使其二范数值 (L_2-norms) 明显变大, 因此可以将梯度的 L_2 值作为系统的模型检验值来进行欺诈攻击防御. 服务器会在联邦学习的每一轮对各个参与者上传的本地模型梯度计算相应的 L_2 值, 并在全局模型更新阶段忽略 L_2 值过高的梯度. 该防御方法不需要参与者上传其他额外的信息且计算简单, 同时能够有效地识别出放大的模型梯度, 从而防止梯度放大攻击.

本文的主要贡献如下.

(1) 定义了激励欺诈攻击问题, 提出了激励成本比来评估激励欺诈攻击方法的优劣和防御方法的有效性.

(2) 提出了一种新的针对模型梯度进行攻击的激励欺诈攻击方法 GSupA, 通过放大模型梯度来提高激励, 能够成功进行激励欺诈攻击.

(3) 设计了针对梯度放大攻击的防御方法, 计算和比较参与者上传梯度的二范数值来进行模型检验, 在无需多余信息的情况下高效地进行防御.

(4) 通过在 MNIST 等多个数据集上进行图像识别分类模型训练实验来对梯度放大攻击和基于 L_2-norms 的防御方法进行评估, 结果显示使用梯度放大攻击能大幅提升所获激励, 得到满足要求的激励成本比, 而基于 L_2-norms 的防御方法能有效防止梯度放大攻击, 验证了本文方法的有效性.

本文第 1 节综述本文提及方法所需的背景知识和主要概念. 第 2 节定义激励欺诈攻击问题. 第 3 节对本文提出的梯度放大的激励欺诈攻击方法进行详细的描述. 第 4 节对本文提出的基于 L_2-norms 的防御方法进行详细说明. 第 5 节通过实验验证本文方法的有效性. 最后总结全文.

1 背景知识

本文探讨的是联邦学习中的激励欺诈攻击与防御, 涉及联邦学习及其激励机制和防御机制, 下面对这些相关概念和基本知识进行介绍.

1.1 联邦学习

联邦学习这一概念由谷歌在 2016 年最先提出[1], 最开始是为了解决手机终端用户数据的模型训练问题, 目前该技术主要用于解决数据在不共享情况下的联合建模问题. 联邦学习系统通常由一个联邦服务器和多个客户端 (参与者) 组成, 这些客户端都持有一定的本地数据集, 在联邦学习中使用本地数据集训练本地模型并上传至服务器, 服务器接收到各客户端的本地模型梯度后进行聚合操作来更新全局模型. 联邦学习的具体训练过程如下: 在第 t 轮, 联邦服务器将上一轮的全局模型参数 w^{t-1} 广播发送给各个客户端, 每个客户端在获取初始模型参数后会在本地使用其持有的私有数据集训练模型并得到相应的本地模型参数 w_i^t, 接着将本地模型参数而不是原始训练数据发送给联邦服务器. 在从各个客户端接收到本地模型参数后, 联邦服务器对这些参数进行聚合操作得到 w^t 来更新全局模型, 再将更新后的全局模型参数广播发送给各个客户端. 上述这一过程会多轮次重复进行直到满足要求即停止, 如达到期望的模型测试准确率或达到规定的训练轮次等. 在联邦学习中, 服务器不需要将所有本地数据收集起来合为一个数据集来进行模型训练, 而是数据持有者作为联邦学习的参与者联合训练一个机器学习模型, 和传统的模型训练方法对比, 充分保护了数据的隐私安全.

联邦服务器在更新全局模型时使用的聚合算法在联邦学习中起到了至关重要的作用, 采用不同的聚合算法得到的全局模型不同, 目前主流使用的聚合算法为联邦平均算法 FedAvg[2]. FedAvg 的思想是基于各客户端的训练数据量对其本地模型参数进行加权计算, 如公式 (1):

$$w^t = \sum_{i=1}^{N} \frac{n_i}{n} w_i^t \tag{1}$$

其中, n_i 为客户端 i 的本地训练数据量, n 为所有客户端的本地训练数据量之和, N 为联邦学习中参与者的数量.

1.2 联邦学习中的激励机制

联邦学习系统最终得到的结果很大程度上依赖于各参与者的本地数据和本地模型梯度的质量, 然而数据持有者可能不愿意在没有足够报酬的情况下参与联邦学习并分享他们的本地模型梯度, 同时联邦学习中的参与者是独立自主的, 他们可以自行决定如何参与联邦学习, 其行为是服务器无法控制的. 激励在一定程度上可以影响参与者的决定, 因此需要在联邦学习系统中引入激励机制. 采用不同的激励机制, 参与者也会实行不同的训练策略, 从而影响最终得到的联合模型的表现. 如何有效评估每个参与者的贡献是联邦学习实际应用与长远发展的关键问题[10], 也是激励分配机制设计中最具挑战性的.

个体法 (individual)[10]是基于参与者自身数据价值或其在特定任务上的表现来评估贡献, 这类贡献评估方法

侧重于参与者个体表现, 而非其在整体联邦学习中的表现. CROWDFL[11]中采用的贡献评估方法为对比参与者的本地模型梯度与全局模型梯度之间的相似度来衡量参与者的贡献, 相似度通过计算本地模型梯度与全局模型梯度之间的 L_2-norms 距离得到, 距离越小, 与全局模型的相似度越高, 则参与者的贡献越大. Lv 等人[12]提出了 PCA 贡献评估方法, 该方法基于同伴预测的思想, 通过计算各参与者本地模型之间的互信息来评估参与者的贡献. 个体法简单高效, 但未考虑到参与者对联邦学习整体的边际价值贡献, 在很多情况下无法得到公平合理的贡献评估结果.

留一法 (leave one out)[10]是机器学习中的一种交叉验证法, Refiner[4]中采用的激励分配机制就是基于留一法的思想, 将联邦学习移除掉某个参与者后所造成的损失函数边际损失值作为该参与者的贡献, 并基于该贡献分配激励, 如公式 (2) 和公式 (3):

$$\Phi_i = \mathcal{L}(w, D_U) - \mathcal{L}(w^i, D_U) \tag{2}$$

$$\mathcal{R}_i = \frac{\Phi_i}{\sum \Phi} \mathcal{R} \tag{3}$$

其中, Φ_i 代表参与者 i 在联邦中的贡献, w 为所有参与者的本地模型聚合后得到的全局模型参数, w^i 为不包含参与者 i 外其他参与者本地模型聚合后得到的模型参数, D_U 为系统使用的测试数据集, $\mathcal{L}$ 为使用的损失函数, $\mathcal{R}_i$ 代表参与者 i 基于贡献得到的激励, $\sum \Phi$ 为联邦学习中所有参与者的贡献, $\mathcal{R}$ 为服务器设定的每一轮的总激励.

夏普利值 (Shapley value)[13]起源于合作博弈理论, 是一种基于贡献的分配方式, 目前被广泛用于评估联邦学习中各参与者的贡献, 计算所得的 Shapley value 即为各参与者的贡献. 使用 Shapley value 对联邦学习中各参与者的贡献进行评估的计算方法如公式 (4):

$$\Phi_i = \mathbb{E}_{\pi \in \Pi}[V(S_\pi^i \cup \{i\}) - V(S_\pi^i)] = \frac{1}{N!} \sum\nolimits_{\pi \in \Pi} [V(S_\pi^i \cup \{i\}) - V(S_\pi^i)] \tag{4}$$

其中, Φ_i 代表参与者 i 在联邦中的贡献, Π 为所有参与者组合成的所有排列可能, π 为 Π 中的一种排列, V 为价值函数, S_π^i 为排列 π 中在参与者 i 之前的组合, N 为联邦中参与者的数量. 由计算公式可知, Shapley value 的思想是枚举所有不包含参与者 i 的参与组合, 在每个组合下计算引入参与者 i 所带来的边际价值贡献, 得到的边际价值贡献均值即为参与者 i 对联邦的贡献. 使用 Shapley value 能够满足联邦中参与者贡献评估的公平性和合理性, 但相应地需要极高的计算成本, 直接计算 Shapley value 的计算代价随参与者的数量而指数级增长, 因此实际上不能直接使用 Shapley value 来评估联邦学习中各参与者的贡献.

为了让 Shapley value 更好地应用于联邦学习中的贡献评估, 目前 Shapley value 的主要研究方向为在提高近似 Shapley value 计算效率的同时保持其准确率. 相关研究工作主要使用了抽样、组测试、截断等方法, 如 Wang 等人[14]提出的 Fed-SV 采用了组测试和随机排列抽样的方法来估计每一轮各参与者的 Shapley value, Ghorbani 等人[15]提出的截断蒙特卡洛夏普利值 (truncated Monte Carlo Shapley, TMC-Shapley) 采用了蒙特卡洛 (Monte-Carlo) 抽样方法, 同时截断每一轮排列抽样中不必要的评估计算来计算 shapley value 的近似值. Liu 等人[16]提出的引导截断梯度夏普利值 (guided truncation gradient Shapley, GTG-Shapley) 采用了引导蒙特卡洛抽样、轮间截断和轮内截断的方法, 做到了在未引入明显近似误差的同时极大地减少计算开销, 提高计算效率, 在近似计算效率和准确性之间实现了良好的平衡.

1.3 联邦学习中的攻击方法

由于联邦学习的原理和运行机制, 其很容易受到恶意攻击, 尤其是由联邦学习框架内部成员发起的攻击, 往往破坏性大且较为隐蔽. 目前联邦学习攻击方法的研究工作主要分为两大类: 针对数据隐私的推理攻击[17,18]与损害系统鲁棒性的投毒攻击[9,19].

隐私推理攻击为攻击者尝试获取联邦学习中的全局模型或各参与者上传的本地模型梯度, 并试图由此推断获得敏感隐私信息的攻击行为[20–25]. 典型的隐私推理攻击主要有成员推断攻击、属性推断攻击和训练数据推断攻击. 成员推断攻击是指攻击者从模型参数或梯度中推断某个特定的数据是否包含在训练数据集中的攻击手段,

Nasr 等人[17]设计了一种主动成员推断攻击方法, 对一组目标数据进行梯度上升, 并根据其增加损失值的变化来判断该目标数据是否存在训练数据集中; 属性推断攻击是指攻击者通过获取联邦学习模型参数或梯度来推断模型训练数据的属性, 还可以推断出某个属性何时在训练数据中出现和消失; 训练数据推断攻击也称模型逆向攻击, 是指攻击者从模型参数或梯度中提取和训练数据有关的隐私信息, 恢复用于模型训练的原始输入和标签. Hitaj 等人[18]提出了使用生成对抗网络 (generative adversarial network, GAN) 来攻击联邦学习, 其中攻击者可以主动攻击其他参与者, 通过训练 GAN 来生成其他参与者本地训练数据的原型样本. 隐私推理攻击的攻击目的和攻击方式与激励欺诈攻击截然不同.

模型投毒攻击为参与者通过上传恶意的本地模型信息来阻碍联合模型的训练, 影响最终得到的联合模型性能的攻击行为[20,21,25]. 根据攻击目标, 投毒攻击可以分为两类: 非定向投毒攻击和定向投毒攻击. 非定向投毒攻击旨在破坏联合模型的可用性, 通过注入大量的恶意数据或上传恶意梯度, 影响联合模型的性能, Cao 等人[19]提出了 MPAF, 将虚假的参与者注入联邦学习系统, 并将放大后的虚假本地模型梯度上传至服务器来降低全局模型的测试准确率; 定向投毒攻击的目标是破坏联合模型的完整性, 攻击者为具有特定特征的输入指定特定的标签, 以使联合模型无法正确识别该类输入的标签, 而其他输入的标签识别不受影响, 即影响联合模型在目标子任务上的性能, 使其存在缺陷. 后门攻击是典型的定向投毒攻击, Bagdasaryan 等人[9]提出了使用模型替代来构造后门攻击模型, 防止聚合模型遗忘后门数据, 从而能够在联邦学习中对联合模型进行有效的后门攻击. 模型投毒攻击的攻击方式虽然与本文提出的激励欺诈攻击相似, 都是通过修改上传梯度进行攻击, 但两者的攻击目的不同, 模型投毒攻击并未考虑联邦学习的激励机制, 而激励欺诈攻击也并不关心上传的模型梯度是否会影响聚合后得到的全局模型性能.

1.4 联邦学习中的防御机制

目前联邦学习中防御机制的研究工作主要针对现有的两大类攻击方法.

针对联邦学习中的隐私推理攻击行为, 保护联邦学习中各参与者的隐私数据, 现有防御工作主要基于机器学习中常见的隐私保护技术[20], 包括同态加密、安全多方计算和差分隐私. 同态加密技术允许对加密后的信息直接进行运算, 运算后解密得到的结果与明文运算的结果相同; 安全多方计算使各参与者能够在不泄露隐私数据的情况下共同进行数据运算; 差分隐私通过为数据增加噪声来保护隐私. 上述这些隐私保护技术只能防御联邦学习中的推理攻击, 不仅无法防止激励欺诈攻击行为, 而且出于对数据隐私的保护, 这些防御方法阻止对各参与者的本地模型梯度进行检验审查, 反而为投毒攻击和激励欺诈攻击提供了方便.

为了防止恶意参与者的投毒攻击行为, 排除低质量的模型梯度, 联邦学习中的安全防御机制研究工作主要可以分为两类: 模型梯度检验[4,5]和调整全局模型更新算法[6−8].

模型梯度检验即对参与者上传的模型梯度进行评估检验, 未通过检验的劣质梯度无法用于全局模型更新, 相应的, 该梯度对全局模型也就无法产生影响. 如 Zhang 等人[4]提出的 Refiner 使用了一种模型评估机制, 通过计算每一个参与者上传的本地模型在联邦服务器持有的验证集上的损失函数值来检验梯度, 只有满足服务器预设定阈值的梯度才能通过检验被服务器所接受, 同样只有通过检验的模型梯度才能够参与后续的模型聚合和贡献评估流程. Zhang 等人[5]提出的 FLDetector 通过检查参与者上传的模型梯度的一致性来检测恶意参与者, 服务器会基于参与者之前上传的梯度值预测其该轮的梯度, 并计算预测梯度值与实际梯度值之间的欧几里得 (Euclidean) 距离作为恶意性评分来检验模型梯度的一致性, 被检测出的恶意参与者无法继续参加后续的联邦学习.

调整全局模型更新算法指采用合适的模型更新算法来尽可能减少具有破坏性的模型梯度对全局模型的影响, 在存在部分恶意梯度的情况下仍然可以训练得到一个高质量的联合模型. Krum[6]、Trimmed-Mean[7]和 FLTrust[8]是拜占庭稳健 (Byzantine-robust) 的全局模型更新算法. 其中 Krum 是在所有参与者的本地模型中选择单个模型作为新的全局模型, 计算每个模型梯度与其最近的 $N-k-2$ 个模型梯度之间的 Euclidean 距离, 被选中的模型计算得到的梯度距离值最小; Trimmed-Mean 在计算新的全局模型时会分别聚合模型参数的各个维度, 针对任一维度, Trimmed-Mean 先将所有模型参数对应维度的值进行排序, 在排除最大和最小的 k 个值后, 将剩下的 $N-2k$ 个值的

平均值作为新的全局模型参数中该维度的值; FLTrust 要求联邦服务器持有一个额外的验证数据集, 基于参与者上传的梯度与使用验证数据集训练得到的梯度之间的余弦相似性给予信任评分, 并对参与者上传梯度的幅度进行归一化处理, 最终使用信任评分对其归一化后的本地模型梯度进行加权计算来得到新的全局模型.

上述这些防御机制能够防止目前存在的大部分影响模型性能的攻击, 但激励欺诈攻击可以通过上传满足系统对模型性能要求的梯度来获取额外激励, 因此, 现有的防御机制很难防御激励欺诈攻击, 需要设计针对性的防御机制来有效防御激励欺诈攻击.

2 激励欺诈攻击问题

本节形式化定义激励欺诈攻击. 对联邦学习中的任意一轮训练 t, 记 (w,n) 为一个攻击者在第 t 轮通过诚实训练 (即严格遵守学习算法进行的训练) 产生的本地模型信息, 其中 w 为模型参数, n 为本地训练数据集大小; 记 $(\tilde{w},\tilde{n})$ 为该攻击者在第 t 轮产生的欺诈本地模型信息, 其中 $\tilde{w}$ 为欺诈模型参数, $\tilde{n}$ 为欺诈的数据集大小. 类似地, 记 $\mathcal{R}(w,n)$ 为攻击者通过提交诚实本地模型信息 (w,n) 所获得的激励, $C(w,n)$ 为产生 (w,n) 所付出的训练代价; 记 $\mathcal{R}(\tilde{w},\tilde{n})$ 为攻击者提交欺诈本地模型信息 $(\tilde{w},\tilde{n})$ 所获得的欺诈激励, $C(\tilde{w},\tilde{n})$ 为所付出的欺诈训练代价. 记 $\Delta\mathcal{R}=\mathcal{R}(\tilde{w},\tilde{n})-\mathcal{R}(w,n)$, 我们有如下定义.

定义 1. 激励欺诈攻击. 对联邦学习中的任意一轮训练, 给定阈值 $\mu>0$, 激励欺诈攻击是一个攻击者通过提交欺诈本地模型信息 $(\tilde{w},\tilde{n})$ 获得激励成本比 $R\geqslant\mu$ 的行为, 其中 R 的计算方式如公式 (5) 所示:

$$R=\frac{\Delta\mathcal{R}}{C(\tilde{w},\tilde{n})/C(w,n)} \tag{5}$$

由公式 (5) 可得, R 为参与者进行欺诈攻击后使用攻击计算代价相较于诚实计算代价的提升比对参与者欺诈攻击后的激励提升值进行缩放所得的激励值, 只有当欺诈攻击后获得的激励成本比 R 不小于阈值 μ 时, 参与者才愿意实施激励欺诈攻击.

激励欺诈攻击虽然不会直接影响联合模型的性能, 但是会影响到激励机制的运行, 由此导致的不合理的激励分配会降低优质数据持有者参与联邦学习的积极性, 从而间接影响到最终模型的性能.

恶意参与者实施激励欺诈攻击时必须在通过联邦学习系统防御机制的前提下提高激励, 因此本文假定恶意参与者在实施激励欺诈攻击行为时有如下背景知识.

(1) 了解联邦学习系统采用的激励分配机制与防御机制, 如系统使用 Refiner 中的模型评估机制, 恶意参与者也能够知道系统预设定的模型检验阈值.

(2) 拥有一个与服务器使用的测试数据集 D_U 分布近似的验证数据集 D_V.

(3) 知道前 e 轮其他参与者上传的本地模型信息, e 的取值与系统采用的防御机制相关.

3 梯度放大攻击 GSupA

3.1 攻击总览

Bagdasaryan 等人[9]提出的构造适用于联邦学习的后门攻击模型的方法中, 为了让训练得到的后门攻击模型在多次模型聚合后仍然能够保持一定的性能, 而不让聚合后的模型遗忘后门数据, 消去后门攻击模型的贡献, 攻击者对攻击模型梯度进行了放大, 来增加模型聚合时攻击模型的权重, 提高其对聚合模型的影响. 模型梯度放大后的聚合算法如公式 (6):

$$\begin{aligned} w^t&=\sum_{i=1}^{N}\frac{n_i}{n}w_i^t=w^{t-1}+\sum_{i=1}^{N}\frac{n_i}{n}(w_i^t-w^{t-1})\\ &=w^{t-1}+\sum_{i=1}^{N-1}\frac{n_i}{n}(w_i^t-w^{t-1})+\frac{n_j}{n}(\tilde{w}_j^t-w^{t-1})\\ &=w^{t-1}+\sum_{i=1}^{N-1}\frac{n_i}{n}(w_i^t-w^{t-1})+\gamma\frac{n_j}{n}(w_j^t-w^{t-1}) \end{aligned} \tag{6}$$

其中, w_j^t 为攻击者未放大时的模型参数, $\tilde{w}_j^t$ 为放大后的模型参数, $\tilde{w}_j^t-w^{t-1}$ 为放大后的模型梯度, 可以改写为

$\gamma(w_j^t - w^{t-1})$, γ 为将模型梯度放大的倍数. 从公式 (6) 可以明显看出模型梯度放大后其在模型聚合时的权重提高了, 使聚合得到的全局模型受到了该模型更大的影响.

采用模型梯度放大的方法可以提高本地模型对全局模型的影响和贡献, 而对全局模型的影响和贡献与其所获激励有着重要的关联, 意味着放大模型梯度一定程度上可以提高所获激励. 基于此, 本文提出了梯度放大攻击方法 GSupA 来进行激励欺诈攻击, 恶意参与者 j 先在本地训练数据集上进行正常的模型训练, 训练得到本地模型梯度后对该模型的全部或局部梯度进行放大处理, 并将放大后的梯度上传至联邦服务器. 梯度放大的公式如下:

$$\tilde{w}_{j,q^t}^t = \gamma(w_{j,q^t}^t - w_{q^t}^{t-1}) + w_{q^t}^{t-1} \tag{7}$$

$$\tilde{w}_j^t \leftarrow (\tilde{w}_{j,q^t}^t, w_{j,p^t}^t) \tag{8}$$

其中, q^t 是第 t 轮放大梯度的网络层, p^t 为第 t 轮模型中不需要放大的网络层, $q^t \cap p^t = \varnothing$, $q^t \cup p^t$ 为本地模型的所有网络层, $\tilde{w}_{j,q^t}^t$ 是第 t 轮参与者 j 本地模型中被放大的网络层 q^t 的参数, w_{j,q^t}^t 是第 t 轮参与者 j 未放大模型时网络层 q^t 的参数, $w_{q^t}^{t-1}$ 是第 $t-1$ 轮全局模型中网络层 q^t 的参数, w_{j,p^t}^t 是第 t 轮参与者 j 本地模型中网络层 p^t 的参数, $\tilde{w}_j^t$ 为第 t 轮参与者 j 对本地模型梯度进行放大处理后得到的模型参数, 也是要上传至联邦服务器的模型参数, γ 为放大模型梯度时使用的放大因子, 当 γ 为 1 时计算得到的结果与未放大时相同, γ 大于 1 时梯度被放大, γ 小于 1 时梯度被缩小.

由公式 (7) 和公式 (8) 可得, 恶意参与者在进行梯度放大攻击时可以选择放大梯度的网络层, 选择不同的网络层梯度进行放大, 得到的欺诈效果不同. 在大多数情况下, 对全部网络层梯度进行放大得到的激励提升效果是最佳的, 但是在部分防御机制下会存在放大全部网络层梯度无法通过防御检验或是激励提高效果不明显等情况. 例如, 联邦学习系统使用 FLTrust[8]聚合算法时, 会对参与者梯度的幅度进行归一化处理, 攻击者放大全部网络层梯度得到的模型梯度被归一化处理后会失去欺诈效果, 无法提高所获激励, 因此在 FLTrust 机制下实施梯度放大攻击应当选择部分网络层梯度进行放大.

对模型梯度进行放大的计算代价与放大梯度网络层的参数量相关, 参数量越多, 所需的计算代价越大. 不过, 梯度放大的计算代价极低, 即使放大所有网络层的梯度, 其所需的计算代价与诚实训练所需的计算代价相比也微不足道.

3.2 放大因子的选择

梯度放大公式中的放大因子 γ 决定了梯度放大的幅度, 很大程度上决定了攻击模型能否通过系统的防御机制以及最终欺诈攻击的效果, 因此, 放大因子的选择十分重要.

当参与者在联邦学习中实施激励欺诈攻击时, 尽管获得的激励成本比已经满足了阈值要求, 但是他们通常希望能够获得更高的激励成本比, 所以在计算放大因子时应选择其中欺诈效果最佳的放大因子. 反过来说, 如果使用欺诈效果最佳的放大因子所获的激励成本比无法满足阈值要求, 那么, 使用其他的放大因子同样也无法满足要求. 因此, 参与者在实施梯度放大攻击时计算放大因子的目标为选择能够最大化所获激励成本比的值.

放大因子的计算方法可以分为以下两大类.

(1) 如算法 1 所示, 攻击者使用自己的本地训练数据提前模拟联邦学习, 在模拟联邦学习中将训练数据分为多个子数据集, 每个子数据集作为模拟联邦学习中各个“参与者”的数据集用于训练模型, 其中随机选择“参与者”实施欺诈攻击, 多次尝试不同的放大因子对选择的“参与者”的梯度进行放大, 选择梯度放大后可以通过防御机制并且欺诈效果最佳的放大因子应用于实际的联邦学习激励欺诈攻击中. 由于在不同的放大因子下模拟联邦学习欺诈攻击所需的计算代价相同, 因此在比较欺诈效果时只需要比较欺诈后所获的激励值即可. 这类方法比较适用于使用调整模型聚合算法而无模型检验环节的防御机制的联邦学习系统.

该方法通过模拟联邦学习, 使用不同的放大因子进行欺诈攻击实验, 从而在可选的放大因子集合中选取能够在模拟联邦学习中产生最大激励成本比的放大因子, 相应地在实际的联邦学习中使用该放大因子的欺诈效果是最佳的.

算法 1. 模拟联邦学习计算最佳放大因子 (以恶意参与者 j 为例).

输入: 本地训练数据 D_j, 验证数据集 D_V, 多个不同的放大因子值集合 γ_list;
输出: 最佳放大因子 γ_{res}.

1. 将本地训练数据集 D_j 分为 N 个子数据集 $D=\{D_j^1, D_j^2, \ldots, D_j^N\}$
2. 随机选择子数据集 D_j^k 作为攻击者
3. 最高激励值 $\mathcal{R}_{\max}=0$
4. **for** γ in γ_list
5. //模拟联邦学习
6. 总激励值 $\mathcal{R}_k=0$
7. **for** 每一轮 $t=1,2,\ldots$
8. 使用子数据集 D 分别训练模型得到 $w^t_{\{i\in\{N\}\}}=\{w_1^t, w_2^t, \ldots, w_N^t\}$
9. 使用 γ 放大 w_k^t 得到 $\tilde{w}_k^t$
10. $\tilde{w}^t_{\{i\in\{N\}\}}=\{w_1^t, \ldots, \tilde{w}_k^t, \ldots, w_N^t\}$
11. **if** $\tilde{w}_k^t$ 能通过防御机制
12. 计算参与者 k 的激励 $\mathcal{R}_k^t=incentive(\tilde{w}^t_{\{i\in\{N\}\}}, D_V)$
13. **else**
14. $\mathcal{R}_k^t=0$
15. **end if**
16. $\mathcal{R}_k=\mathcal{R}_k+\mathcal{R}_k^t$
17. **end for**
18. **if** $\mathcal{R}_k>\mathcal{R}_{\max}$
19. $\gamma_{\text{res}}=\gamma$
20. $\mathcal{R}_{\max}=\mathcal{R}_k$
21. **end if**
22. **end for**
23. **return** γ_{res}

(2) 在联邦学习中的每一轮计算该轮适用的放大因子, 使用上一轮所有参与者的本地模型信息模拟这一轮的欺诈攻击, 多次尝试不同的放大因子对上一轮攻击者持有的诚实模型梯度进行放大, 选择梯度放大后可以通过防御机制并且欺诈效果最佳的放大因子应用于本轮的激励欺诈攻击. 这类方法比较适用于通过对模型梯度进行检验来排查恶意参与者的防御机制下的联邦学习.

这类计算放大因子的方法的思想是基于联邦学习相邻两轮之间各参与者训练所得的模型梯度较为相近, 如果上一轮的诚实模型梯度使用放大因子放大后能够得到最大的激励成本比, 那么使用该放大因子对这一轮的本地模型梯度进行放大也能够得到最佳的欺诈效果.

该方法是在联邦学习运行的过程中计算放大因子, 由于联邦学习运行时服务器一般对参与者训练并上传模型梯度有一定的时间限制, 不会无期限的等待参与者上传梯度, 同时, 尝试的放大因子数量越多, 攻击所需的计算代价越大, 因此攻击者需要尽可能减少实验的放大因子数量并能够从中找到合适的放大因子.

在 Refiner 的模型评估机制下计算放大因子的伪代码如算法 2 所示, 从中等大小的 γ 值开始实验, 根据梯度放大后能否通过防御 (即通过 Refiner 的模型梯度检验) 并提高激励来决定增加或是减小 γ 值, 如果梯度放大后能够通过防御并提高激励, 则意味着目前使用的 γ 值是满足攻击要求的, 可以尝试更大的 γ 值来获得更高的激励; 否则, 说明目前使用的 γ 值过大, 导致梯度放大后偏差过大, 需要减小 γ 值继续实验. 修改 γ 值时以 $step/2$ 为步长, 每次

修改都会将步长减半来使计算更加精细, 当 γ 值与当前得到的最佳放大因子 γ_{res} 之间的距离不大于预设定的阈值 τ 时即停止计算, 并将当前得到的 γ_{res} 作为该轮的放大因子. 使用算法 2 计算最佳放大因子时, 尝试的放大因子数量与设定的初始放大因子和阈值 τ 相关, 由此算法中使用不同的放大因子所需的计算代价相同, 可以通过比较欺诈激励来比较欺诈效果.

算法 2. Refiner 第 t 轮计算最佳放大因子 (以恶意参与者 j 为例).

输入: 第 $t-1$ 轮各参与者的本地模型信息 $w_{\{i\in\{N\}\}}^{t-1}=\{w_1^{t-1},w_2^{t-1},\ldots,w_N^{t-1}\}$, 验证数据集 D_V, 初始放大因子 γ_{init}, 放大因子阈值 τ;
输出: 最佳放大因子 γ_{res}.

1. $\gamma_{\text{res}}=1;\ \mathcal{R}_{\max}=0;\ \gamma=\gamma_{\text{init}};\ step=\gamma_{\text{init}}/2$
2. **while** $|\gamma_{\text{res}}-\gamma|>\tau$
3. 使用 γ 放大 w_j^{t-1} 得到 $\tilde{w}_j^{t-1}$
4. $\tilde{w}_{\{i\in\{N\}\}}^{t-1}=\{w_1^{t-1},\ldots,\tilde{w}_j^{t-1},\ldots,w_N^{t-1}\}$
5. **if** $\tilde{w}_j^{t-1}$ 能通过防御机制且所得激励 $\mathcal{R}_j^t=incentive(\tilde{w}_{\{i\in\{N\}\}}^{t-1},D_V)>\mathcal{R}_{\max}$
6. $\mathcal{R}_{\max}=\mathcal{R}_j^t$
7. $\gamma_{\text{res}}=\gamma$
8. $\gamma=(\gamma+step/2)$
9. **else**
10. $\gamma=(\gamma-step/2)$
11. **end if**
12 $step=step/2$
13. **end while**
14. **return** γ_{res}

FLDetector 通常在联邦学习运行了一定轮数后开始检测恶意参与者, 检测时会基于之前上传的模型梯度对该轮模型梯度一致性进行检验, 因此使用的放大因子不能与前几轮使用的放大因子差距过大. 在联邦学习前期 FLDetector 未运行时可以使用算法 2 计算放大因子, 也可以选用固定的放大因子, FLDetector 运行时计算放大因子的具体伪代码如算法 3 所示, 主要围绕上一轮的放大因子 γ_{pre} 进行实验, 从较小的 γ 值开始, 如梯度放大后的恶意性评分满足要求, 即采用当前 γ 值作为最佳放大因子, 不再继续实验计算, 尽可能减少计算代价, 当实验的所有 γ 值都无法满足要求时则继续使用上一轮的放大因子作为本轮的放大因子.

算法 3. FLDetector 第 t 轮计算最佳放大因子 (以恶意参与者 j 为例).

输入: 第 $t-1$ 轮各参与者的本地模型信息 $w_{\{i\in\{N\}\}}^{t-1}=\{w_1^{t-1},w_2^{t-1},\ldots,w_N^{t-1}\}$, 第 $t-2$ 轮各参与者上传的模型信息 $\tilde{w}_{\{i\in\{N\}\}}^{t-2}=\{w_1^{t-2},\ldots,\tilde{w}_j^{t-2},\ldots,w_N^{t-2}\}$, 上一轮使用的放大因子 γ_{pre}, 放大因子范围限制 τ, 合格区间扩大范围 ε;
输出: 最佳放大因子 γ_{res}.

1. $\gamma_{\text{res}}=\gamma_{\text{pre}};\ \gamma=\gamma_{\text{pre}}-\tau;\ step=\tau/5$
2. **while** $\gamma\leqslant\gamma_{\text{pre}}+\tau$
3. 使用 γ 放大 w_j^{t-1} 得到 $\tilde{w}_j^{t-1}$
4. $\tilde{w}_{\{i\in\{N\}\}}^{t-1}=\{w_1^{t-1},\ldots,\tilde{w}_j^{t-1},\ldots,w_N^{t-1}\}$
5. 计算各参与者的恶意性评分 $S_{\{i\in\{N\}\}}=\{s_1,s_2,\ldots,s_N\},s=mal\_score(\tilde{w}_{\{i\in\{N\}\}}^{t-1},\tilde{w}_{\{i\in\{N\}\}}^{t-2})$
6. 根据其他参与者的恶意性评分计算合格的评分区间 $[\min(S_{\{i\in\{N\},i\neq j\}})-\varepsilon,\max(S_{\{i\in\{N\},i\neq j\}})+\varepsilon]$

7. **if** 梯度放大后的恶意性评分 s_j 位于合格区间内
8. $\gamma_{\text{res}} = \gamma$
9. **break**
10. **end if**
11. $\gamma = \gamma + step$
12. **end while**
13. **return** γ_{res}

4 基于 L_2-norms 的模型评估机制

梯度放大攻击的本质是对本地模型梯度值进行放大来提高本地模型对全局模型的影响, 由此提高贡献和所获激励. 模型梯度的大小与其 L_2 值直接相关, 放大梯度, 其 L_2 值明显会增大, 因此可以将模型梯度的 L_2 值作为检验值来检测恶意参与者.

本文提出的模型评估机制基于参与者上传模型梯度的 L_2 值来检测恶意参与者, 防止梯度放大攻击. 在获取各个参与者上传的本地模型梯度后, 服务器计算各梯度的 L_2 值, 并根据该轮所得的 L_2 值集合来计算相应的阈值, 只有低于阈值的梯度才能够通过检验, 参与之后的模型聚合和激励分配, 未通过检验的参与者无法获得激励.

系统检验 L_2 值的阈值是动态的, 每一轮的阈值各不相同, 根据本轮各参与者梯度的 L_2 值来决定. 服务器在接收到该轮各参与者的模型梯度后计算相应的 L_2 值集合 $L_2(N)$, 对 $L_2(N)$ 进行排序, 在排除最大和最小的 $k(k < N/2)$ 个 L_2 值后, 剩余 L_2 值的集合为 $L_2(N-2k)$, 计算其中最大值与最小值之间的差 $\delta = \max[L_2(N-2k)] - \min[L_2(N-2k)]$, $\max[L_2(N-2k)] + \delta$ 即为所得阈值. 计算时的 k 值越大, 得到的阈值越小, 系统的检验标准也就越严格. 如服务器已知恶意参与者的数量, 在进行防御时可将 k 值设置为恶意参与者的数量, 可以得到较为精细准确的防御结果; 如无法确定攻击者的数量, 服务器可通过计算比较大小相邻的 L_2 值之间的差来估测 k 值, 即计算差值后从中选取较其他差值明显偏高的差 φ_i (排序后 L_2 集合中第 i 个和第 $i+1$ 个值之间的差), 可将 k 值设为 $N-i$.

从参与者的角度, 基于 L_2-norms 的模型评估机制不需要上传额外的模型信息, 不会给参与者带去额外的计算和通信开销. 从服务器的角度, 检验 L_2 值时使用的数据都是本轮获取计算得到的, 不需要存储额外的信息, 无存储开销, 此外, 每轮计算 L_2 值的时间复杂度为 $O(Nlen)$, 其中 N 为参与者数量, len 为训练模型的参数量, 由此可得检验 L_2 值的存储和计算开销对服务器来说都是可以接受的.

5 实验分析

5.1 实验设置

(1) 数据集

本文使用了 3 个数据集 MNIST、FashionMNIST[26]、CIFAR10[27]来对提出的梯度放大攻击和基于 L_2-norms 的模型检验机制进行评估. MNIST 是一个 10 种数字图像分类的数据集, 由不同人手写的数字构成, 是 28×28 的灰度图片, 其中包含了 60 000 个训练集和 10 000 个测试集. FashionMNIST 包含了 10 种衣物图像, 图像为 28×28 的灰度图片, 具有预定义的 60 000 个训练集和 10 000 个测试集. CIFAR10 是一个彩色图像数据集, 有 10 个类别, 图片尺寸为 32×32, 共有 50 000 个训练集和 10 000 个测试集.

本文按照文献 [28] 中的方法将训练数据集分配给联邦学习中的各个参与者, 假设在数据集中有 c 类数据, 将参与者随机分成 c 组, 一个标签为 l 的训练数据会以 $q > 0$ 的概率被分到 l 组, 会有 $\frac{1-q}{c-1}$ 的概率被分到别的组, 在同一组中, 数据会被均匀地分给各个参与者. q 控制了参与者持有本地训练数据集的分布差异, 当 $q = 1/c$ 时, 各个参与者的数据是独立同分布的 (IID), 除此之外, 分配的数据都是非独立同分布的 (Non-IID), q 越大, 意味着数据 Non-IID 的程度就越高.

(2) 联邦学习设置

默认情况下, 对于所有的数据集, 本文设置的联邦学习参与者数量为 10, 每个参与者持有 500 个训练数据, 数据分配时采用的 q 为 0.1, 这 10 个参与者都参与了联邦学习的每个运行轮次. 对于 MNIST 和 FashionMNIST 数据集, 本文使用的训练模型为一个 4 层 CNN 网络, 如表 1 所示, 训练时使用的学习率为 1×10^{-4}, batch_size 为 32, 本地训练轮数为 2, 分别运行 300 和 400 轮. 训练 CIFAR10 数据集时, 本文使用的网络是 VGG16[29]的微调版本, 移除了原有的全连接层, 添加了一个 Softmax 层来进行分类, 共有 14 层网络, 其中模型的卷积层使用了在 ImageNet[30]数据集上预训练好的权重, 设置的 batch_size 为 64, 本地训练轮数为 5, 以 3×10^{-7} 的学习率共训练 500 轮.

表 1 MNIST 和 FashionMNIST 使用的训练模型网络结构

网络层	大小
Input	28×28×1
Convolution + ReLU	3×3×10
Max Pooling	2×2
Convolution + ReLU	3×3×20
Max Pooling	2×2
Fully connected + ReLU	100
Softmax	10

实验使用了 Refiner[4]和 GTG-Shapley[12]两种激励分配机制, 这两种激励分配方法都具有可行性和可用性, 能够满足激励分配的公平合理性需求, 应用较为广泛, 具有代表性, 且两种方法进行贡献评估和激励分配的思想完全不同, 能够得到激励欺诈攻击与防御在不同激励机制下的效率, 体现攻击与防御方法的广泛适用性. 其中在 Refiner 下每轮设置的总激励为 1, GTG-Shapley 下计算所得的 Shapley value 即参与者所获的激励.

本文实验时使用了 5 种现有的防御机制: Refiner[4]中基于 loss 的模型评估机制、FLDetector[5]、Krum[6]、Trimmed-Mean[7]和 FLTrust[8], 其中在 Refiner 和 FLDetector 机制下使用了 FedAvg 聚合算法. 在这些防御机制下检验了梯度放大攻击的效果, 并与本文提出的基于 L_2-norms 的模型评估机制作对比.

(3) 攻击和防御设置

默认情况下, 本文随机选取 10% 的参与者作为恶意参与者来实施梯度放大攻击, 即 10 个参与者中存在 1 个恶意参与者. 在 Refiner 的防御机制下, 参与者使用算法 2 计算放大因子, 其中 γ_{init} 为 2, τ 为 0.02, 得到的放大因子范围为 [1, 3). 在 FLDetector 下, 参与者在前 10 轮使用算法 2 计算放大因子, 之后使用算法 3 计算放大因子, 设置 τ 为 0.05. 在 Krum、Trimmed-Mean 和 FLTrust 下, 参与者每一轮都使用一样的放大因子, 分别为 10、2 和 3. 针对梯度放大的网络层, 参与者在除 FLTrust 外的所有防御机制下放大所有网络层的梯度, 而在 FLTrust 下, 参与者只放大部分网络层的梯度, 如在 4 层 CNN 网络中只放大前 2 层梯度, 在 14 层 VGG16 微调模型中只放大前 5 层梯度. 恶意参与者会在联邦学习的每一个轮次都实施攻击.

服务器在运行基于 L_2-norms 的模型评估机制时将 k 值设置为恶意参与者的数量, 即 $k=1$.

(4) 评估指标

本文使用的评估指标为总体激励成本比 (R) 和总体激励增长率 (growth rate, GR). 总体激励成本比 (R) 是参与者在联邦学习中进行激励欺诈攻击后得到的每一轮的激励成本比之和, 是其参与联邦学习全过程获得的激励成本比, 与单轮的激励成本比相比, 能够简单直接地显示出在联邦学习中进行激励欺诈的全局攻击效率, 且通过计算各轮激励成本比之和来得到总体激励成本比的方式符合对于激励欺诈攻击的定义, 具有合理性. 总体激励增长率 (GR) 是参与者进行激励欺诈攻击后得到的总体激励成本比与使用诚实梯度所获总体激励之比, 能够显示出进行激励欺诈攻击后参与者所获激励的提升率, 同时考虑了实施攻击所需的计算成本, 从另一个角度对激励欺诈攻击的全局效率进行合理的评估, 帮助比较不同激励机制下的攻击效率.

5.2 实验结果与分析

表 2 给出了在不同的激励机制和防御机制下, 使用不同的数据集训练模型时参与者进行梯度放大攻击得到的

总体激励成本比和总体激励增长率 (%) 的实验结果. 根据实验结果, 可以有如下结论. 首先, 无论是使用 Refiner 的激励分配机制, 还是使用 GTG-Shapley 进行贡献评估, 在多种现有的防御机制下, 本文所提出的梯度放大攻击能够在合理的攻击计算代价下明显提高参与者的所获激励, 获得相应的激励成本比. 第二, 梯度放大攻击在仅有一位参与者实施攻击的情况下能够帮助提高激励, 获得一定的激励成本比. 第三, 梯度放大攻击在 Krum 下得到的总体激励增长率比其他聚合算法的普遍要高. 第四, 在 Refiner 和 GTG-Shapley 两种激励机制下获得的总体激励成本比相差极大, 这是由两个激励机制采用的激励分配方式的不同导致的, Refiner 中每轮的总激励是固定的, 参与者根据其贡献占所有参与者贡献之和的比例瓜分总激励, 其所获激励与贡献值和设置的总激励相关, 而 GTG-Shapley 下参与者的所获激励等同于其贡献值, 即仅与贡献值相关. 第五, 梯度放大攻击无法在本文提出的基于 L_2-norms 的模型评估机制下攻击成功, 由于梯度放大后的 L_2 值无法通过检验, 参与者无法分配到激励, 因此攻击后的激励反而下降, 得到的总体激励成本比小于 0, 无法满足攻击要求.

表 2　在不同的激励机制、防御机制下训练不同的数据集进行梯度放大攻击的实验结果 (IID)

防御机制	数据集	Refiner		GTG-Shapley	
		R	GR (%)	R	GR (%)
Refiner	MNIST	27.42	38.12	0.60	83.2
	FashionMNIST	14.35	51.45	0.43	110.3
	CIFAR10	7.23	10.6	0.33	122.1
FLDetector	MNIST	21.63	29.8	0.63	92.5
	FashionMNIST	17.87	63.3	0.54	135.8
	CIFAR10	29.06	42.4	0.35	132.2
Krum	MNIST	16.41	2491.0	0.53	193.0
	FashionMNIST	19.42	425.6	0.10	42.9
	CIFAR10	53.89	425.9	0.54	166.6
Trimmed-Mean	MNIST	27.73	40.6	0.20	29.0
	FashionMNIST	8.61	30.9	0.08	20.3
	CIFAR10	25.35	35.6	0.18	67.3
FLTrust	MNIST	59.94	159.3	0.16	55.7
	FashionMNIST	41.43	208.2	0.15	86.7
	CIFAR10	53.36	79.0	0.01	6.9
L_2-norms	MNIST	−40.41	−55.9	−0.38	−54.9
	FashionMNIST	−6.64	−23.7	−0.19	−46.5
	CIFAR10	−62.86	−92.1	−0.24	−90.0

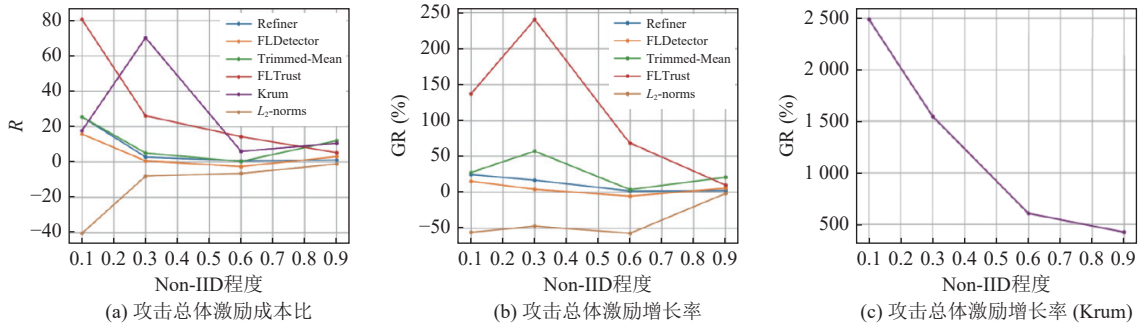
5.3 参数影响分析

为了充分探讨本文提出的方法受各参数的影响状况, 分别在不同条件下进行实验来检验分析不同参数对攻击方法和防御方法的影响.

图 1 展示了在不同防御机制下参与者持有数据的 Non-IID 程度对梯度放大攻击的影响. 首先, 除 Krum 外在其他现有防御机制下, Non-IID 程度增大, 梯度放大攻击所获的总体激励成本比减少, 直到总体激励成本比减少至接近 0 后会在 0 左右波动. 其次, 由于使用不同 Non-IID 程度的数据在无欺诈攻击的情况下进行联邦学习得到的诚实激励不同, 存在 Non-IID 程度变化后总体激励成本比变动与总体激励增长率变动完全相反的情况, 例如, 在 FLTrust 机制下, Non-IID 为 0.3 与 0.1 的相比较, 攻击总体激励成本比降低, 但是总体激励增长率反而有所提高. 最后, 在基于 L_2-norms 的防御机制下, 使用不同 Non-IID 程度的数据进行欺诈攻击得到的总体激励成本比和总体激励增长率都低于 0, 无法满足攻击要求.

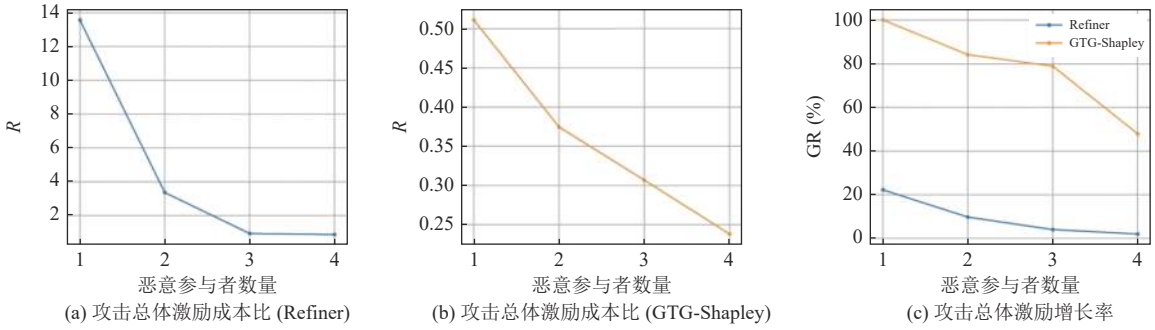
图 2 展示了在不同激励机制下实施攻击的恶意参与者的数量对梯度放大攻击的影响, 图中的攻击总体激励成本比和攻击总体激励增长率皆为所有恶意参与者获得的相应值的均值. 从图 2 中可以得到, 在两种激励分配机制下, 恶意参与者的数量越多, 每个攻击者平均得到的攻击总体激励成本比和总体激励增长率都越低, 这是由于在进

行激励分配时, 各个参与者之间的梯度存在牵制影响, 尤其是在 Refiner 激励机制下, 每一轮服务器提供的激励固定, 各参与者根据贡献瓜分激励, 当其他参与者的贡献提高时, 该参与者的贡献占比就会降低, 能分到的激励也就因此减少.



(a) 攻击总体激励成本比 (b) 攻击总体激励增长率 (c) 攻击总体激励增长率 (Krum)

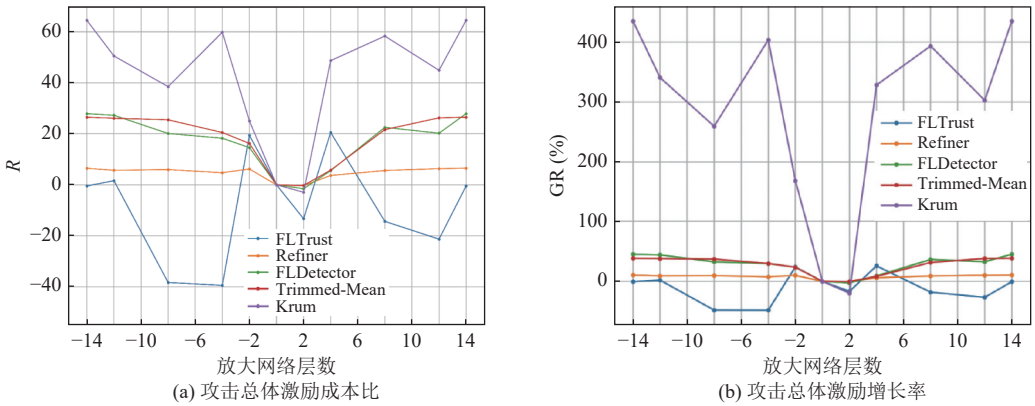
图 1 使用 MNIST 在 Refiner 激励机制和不同防御机制的情况下, 数据 Non-IID 程度对梯度放大攻击的影响



(a) 攻击总体激励成本比 (Refiner) (b) 攻击总体激励成本比 (GTG-Shapley) (c) 攻击总体激励增长率

图 2 使用 FashionMNIST 在 Refiner 防御机制和不同激励机制的情况下, 恶意参与者数量对梯度放大攻击的影响

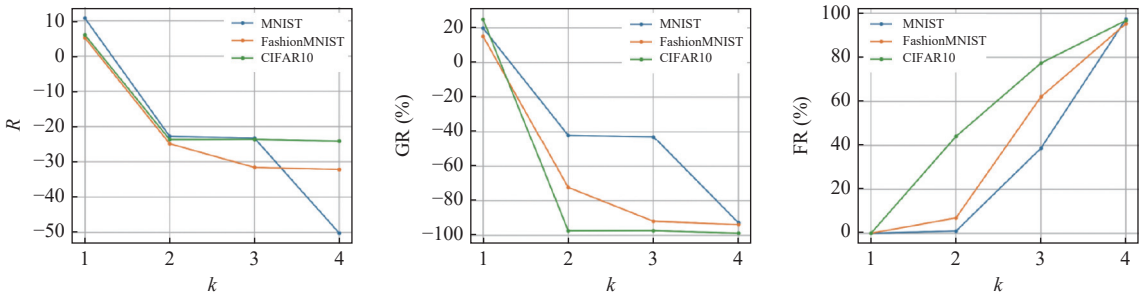
图 3 展示了在不同防御机制下放大不同的网络层梯度对梯度放大攻击的影响, 图中横坐标的正数代表着从第一层开始放大的网络层数, 负数代表着从最后一层开始放大的网络层数, 例如 4 指的是只放大前 4 层网络层梯度. 从图中可以得出如下观察结果: 首先, 除 FLTrust 外在其他防御机制下, 放大 VGG16 微调模型所有的网络层梯度得到的总体激励成本比和总体激励增长率都是最高的, 即欺诈攻击效果最佳. 其次, 在 FLTrust 防御机制下, 放大所有网络层梯度无法提高激励, 需要放大部分网络层梯度才能够提高所获激励, 攻击成功.



(a) 攻击总体激励成本比 (b) 攻击总体激励增长率

图 3 使用 CIFAR10 在 Refiner 激励机制和不同防御机制的情况下, 放大网络层数对梯度放大攻击的影响

图 4 展示了基于 L_2-norms 的模型评估机制中超参数 k 对防御机制的影响. 为了更好地表现过小的 k 与过大的 k 对防御的影响, 实验中随机选取 20% 的参与者作为恶意参与者来实施梯度放大攻击, 即存在 2 个恶意参与者. 通过对比恶意参与者获得的平均总体激励成本比的变化和将诚实者误识别为攻击者的概率 (false rate, FR) 来评估防御效果. 从图 4 中可以得到, 随着 k 的增加, 每个恶意参与者平均得到的攻击总体激励成本比和攻击总体激励增长率都在降低, 与此同时, 系统将诚实者误识别为攻击者的概率也在提高. 其中当 k 为 2 时, 恶意参与者进行梯度放大攻击后得到的总体激励成本比为负值, 且误识攻击者的概率偏低, 得到的防御效果较好.



(a) 攻击总体激励成本比 (b) 攻击总体激励增长率 (c) 误识攻击者的概率

图 4 使用不同数据集在 Refiner 激励机制和基于 L_2-norms 的模型评估机制的情况下, 超参数 k 对防御机制的影响

6 总 结

激励欺诈攻击问题在现有的联邦学习研究中很少被提及, 但在实际应用中会对联邦学习系统的生态造成破坏. 本文对联邦学习中的参与者激励欺诈攻击问题进行了研究, 对该问题进行了定义, 根据参与者欺诈后所获的激励成本比来评估激励欺诈攻击方法和防御方法的优劣. 接着, 本文提出了梯度放大攻击, 一种通过放大模型梯度来提高激励的欺诈攻击方法, 并针对该方法提出了基于 L_2-norms 的模型评估机制, 通过检验模型梯度的二范数值大小来识别欺诈者. 本文在 3 个流行数据集上使用两种不同的激励机制和 6 种不同的防御机制进行了实验, 验证了梯度放大攻击能够在现有的防御机制下攻击成功, 但无法通过基于 L_2-norms 模型评估机制的检验. 未来可以对激励欺诈攻击进行更加深入的研究, 探寻其他攻击方法和防御方法, 帮助完善整个联邦学习系统.

References:

[1] Konečný J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D. Federated learning: Strategies for improving communication efficiency. arXiv:1610.05492, 2016.

[2] McMahan B, Moore E, Ramage D, Hampson S, Arcas BAY. Communication-efficient learning of deep networks from decentralized data. In: Proc. of the 20th Int'l Conf. on Artificial Intelligence and Statistics. Fort Lauderdale: AISTATS, 2017. 1273−1282.

[3] Zhan YF, Zhang J, Hong ZC, Wu LJ, Li P, Guo S. A survey of incentive mechanism design for federated learning. IEEE Trans. on Emerging Topics in Computing, 2022, 10(2): 1035–1044. [doi: 10.1109/TETC.2021.3063517]

[4] Zhang ZB, Dong DJ, Ma YH, Ying YL, Jiang DW, Chen K, Shou LD, Chen G. Refiner: A reliable incentive-driven federated learning system powered by blockchain. Proc. of the VLDB Endowment, 2021, 14(12): 2659–2662. [doi: 10.14778/3476311.3476313]

[5] Zhang ZX, Cao XY, Jia JY, Gong NZ. FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In: Proc. of the 28th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining. Washington: ACM, 2022. 2545–2555. [doi: 10.1145/3534678.3539231]

[6] Blanchard P, El Mhamdi EM, Guerraoui R, Stainer J. Machine learning with adversaries: Byzantine tolerant gradient descent. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: ACM, 2017. 118–128. [doi: 10.5555/3294771.3294783]

[7] Yin D, Chen YD, Ramchandran K, Bartlett PL. Byzantine-robust distributed learning: Towards optimal statistical rates. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: ICML, 2018. 5636–5645.

[8] Cao XY, Fang MH, Liu J, Gong NZ. FLTrust: Byzantine-robust federated learning via trust bootstrapping. arXiv:2012.13995v1, 2020.

[9] Bagdasaryan E, Veit A, Hua YQ, Estrin D, Shmatikov V. How to backdoor federated learning. In: Proc. of the 23rd Int'l Conf. on Artificial Intelligence and Statistics. Palermo: AISTATS, 2020. 2938–2948.

[10] Wang Y, Li GL, Li KY. Survey on contribution evaluation for federated learning. Ruan Jian Xue Bao/Journal of Software, 2023, 34(3): 1168–1192 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6786.htm [doi: 10.13328/j.cnki.jos.006786]

[11] Zhao BW, Liu XM, Chen WN. When crowdsensing meets federated learning: Privacy-preserving mobile crowdsensing system. arXiv:2102.10109, 2021.

[12] Lv HT, Zheng ZZ, Luo T, Wu F, Tang SJ, Hua LF, Jia RF, Lv CF. Data-free evaluation of user contributions in federated learning. In: Proc. of the 19th Int'l Symp. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt). Philadelphia: IEEE, 2021. 1–8. [doi: 10.23919/WiOpt52861.2021.9589136]

[13] Shapley LS. A value for n-person games. In: Kuhn HW, Tucker AW, eds. Contributions to the Theory of Games. Princeton: Princeton University Press, 1953. 307–318. [doi: 10.1515/9781400881970-018]

[14] Wang TH, Rausch J, Zhang C, Jia RX, Song D. A principled approach to data valuation for federated learning. In: Yang Q, Fan LX, Yu H, eds. Federated Learning: Privacy and Incentive. Cham: Springer, 2020. 153–167. [doi: 10.1007/978-3-030-63076-8_11]

[15] Ghorbani A, Zou JY. Data shapley: Equitable valuation of data for machine learning. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: ICML, 2019. 2242–2251.

[16] Liu ZL, Chen YY, Yu H, Liu Y, Cui LZ. GTG-Shapley: Efficient and accurate participant contribution evaluation in federated learning. ACM Trans. on Intelligent Systems and Technology, 2022, 13(4): 60. [doi: 10.1145/3501811]

[17] Nasr M, Shokri R, Houmansadr A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In: Proc. of the 2019 IEEE Symp. on Security and Privacy (SP). San Francisco: IEEE, 2019. 739–753. [doi: 10.1109/SP.2019.00065]

[18] Hitaj B, Ateniese G, Perez-Cruz F. Deep models under the GAN: Information leakage from collaborative deep learning. In: Proc. of the 2017 ACM SIGSAC Conf. on Computer and Communications Security. Dallas: ACM, 2017. 603–618. [doi: 10.1145/3133956.3134012]

[19] Cao XY, Gong NZ. MPAF: Model poisoning attacks to federated learning based on fake clients. In: Proc. of the 2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops. New Orleans: IEEE, 2022. 3395–3403. [doi: 10.1109/CVPRW56347.2022.00383]

[20] Lyu LJ, Yu H, Ma XJ, Chen C, Sun LC, Zhao J, Yang Q, Yu PS. Privacy and robustness in federated learning: Attacks and defenses. IEEE Trans. on Neural Networks and Learning Systems, 2024, 35(7): 8726–8746. [doi: 10.1109/TNNLS.2022.3216981]

[21] Gu YH, Bai YB. Survey on security and privacy of federated learning models. Ruan Jian Xue Bao/Journal of Software, 2023, 34(6): 2833–2864 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6658.htm [doi: 10.13328/j.cnki.jos.006658]

[22] Tang LT, Chen ZN, Zhang LF, Wu D. Research progress of privacy issues in federated learning. Ruan Jian Xue Bao/Journal of Software, 2023, 34(1): 197–229 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6411.htm [doi: 10.13328/j.cnki.jos.006411]

[23] Liu YX, Chen H, Liu YH, Li CP. Privacy-preserving techniques in federated learning. Ruan Jian Xue Bao/Journal of Software, 2022, 33(3): 1057–1092 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6446.htm [doi: 10.13328/j.cnki.jos.006446]

[24] Tan ZW, Zhang LF. Survey on privacy preserving techniques for machine learning. Ruan Jian Xue Bao/Journal of Software, 2020, 31(7): 2127–2156 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6052.htm [doi: 10.13328/j.cnki.jos.006052]

[25] Wei LF, Chen CC, Zhang L, Li MS, Chen YJ, Wang Q. Security issues and privacy preserving in machine learning. Journal of Computer Research and Development, 2020, 57(10): 2066–2085 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2020.20200426]

[26] Xiao H, Rasul K, Vollgraf R. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747, 2017.

[27] Krizhevsky A. Learning multiple layers of features from tiny images [MS. Thesis]. Toronto: University of Toronto, 2009. [doi: 10.1.1.222.9220]

[28] Fang MH, Cao XY, Jia JY, Gong NZ. Local model poisoning attacks to Byzantine-robust federated learning. In: Proc. of the 29th USENIX Security Symp. USENIX, 2020. 1605–1622.

[29] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proc. of the 3rd Int'l Conf. on Learning Representations. San Diego: ICLR, 2015.

[30] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. In: Proc. of the 2009 IEEE Conf. on Computer Vision and Pattern Recognition. Miami: IEEE, 2009. 248–255. [doi: 10.1109/CVPR.2009.5206848]

附中文参考文献:

[10] 王勇, 李国良, 李开宇. 联邦学习贡献评估综述. 软件学报, 2023, 34(3): 1168–1192. http://www.jos.org.cn/1000-9825/6786.htm [doi: 10.13328/j.cnki.jos.006786]

[21] 顾育豪, 白跃彬. 联邦学习模型安全与隐私研究进展. 软件学报, 2023, 34(6): 2833–2864. http://www.jos.org.cn/1000-9825/6658.htm [doi: 10.13328/j.cnki.jos.006658]

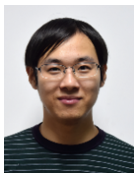
[22] 汤凌韬, 陈左宁, 张鲁飞, 吴东. 联邦学习中的隐私问题研究进展. 软件学报, 2023, 34(1): 197–229. http://www.jos.org.cn/1000-9825/6411.htm [doi: 10.13328/j.cnki.jos.006411]

[23] 刘艺璇, 陈红, 刘宇涵, 李翠平. 联邦学习中的隐私保护技术. 软件学报, 2022, 33(3): 1057–1092. http://www.jos.org.cn/1000-9825/6446.htm [doi: 10.13328/j.cnki.jos.006446]

[24] 谭作文, 张连福. 机器学习隐私保护研究综述. 软件学报, 2020, 31(7): 2127–2156. http://www.jos.org.cn/1000-9825/6052.htm [doi: 10.13328/j.cnki.jos.006052]

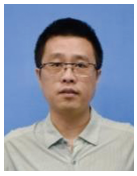
[25] 魏立斐, 陈聪聪, 张蕾, 李梦思, 陈玉娇, 王勤. 机器学习的安全问题及隐私保护. 计算机研究与发展, 2020, 57(10): 2066–2085. [doi: 10.7544/issn1000-1239.2020.20200426]

乐紫莹(1999－), 女, 硕士生, 主要研究领域为联邦学习中的安全保护技术.



骆歆远(1988－), 男, 博士, 助理研究员, 主要研究领域为大数据管理, 大数据智能计算, 信息检索.

陈珂(1977－), 女, 博士, 副研究员, CCF 专业会员, 主要研究领域为非结构化数据管理, 数据挖掘, 隐私保护.



陈刚(1973－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为数据库, 大数据管理系统, 大数据智能计算.

寿黎但(1976－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为非结构化数据管理, 移动社会媒体数据管理, 多媒体挖掘.