一种面向联邦学习对抗攻击的选择性防御策略

陈 卓[①] 江 辉*[①] 周 杨[②]

[①](重庆理工大学计算机科学与工程学院 重庆 400054)

[②](奥本大学计算机科学与软件工程学院 美国阿拉巴马州 奥本市 36849)

摘 要：联邦学习(FL)基于终端本地的学习以及终端与服务器之间持续地模型参数交互完成模型训练，有效地解决了集中式机器学习模型存在的数据泄露和隐私风险。但由于参与联邦学习的多个恶意终端能够在进行本地学习的过程中通过输入微小扰动即可实现对抗性攻击，并进而导致全局模型输出不正确的结果。该文提出一种有效的联邦防御策略-SelectiveFL，该策略首先建立起一个选择性联邦防御框架，然后通过在终端进行对抗性训练提取攻击特性的基础上，在服务器端对上传的本地模型更新的同时根据攻击特性进行选择性聚合，最终得到多个适应性的防御模型。该文在多个具有代表性的基准数据集上评估了所提出的防御方法。实验结果表明，与已有研究工作相比能够提升模型准确率提高了2%～11%。

关键词：联邦学习；对抗性攻击；防御机制；对抗性训练

中图分类号：TN918; TP181 **文献标识码**：A **文章编号**：1009-5896(2024)03-1119-09

DOI: 10.11999/JEIT230137

A Selective Defense Strategy for Federated Learning Against Attacks

CHEN Zhuo[①] JIANG Hui[①] ZHOU Yang[②]

[①](*College of Computer Science and Engineering, Chongqing University of Technology, Chongqing* 400054, *China*)

[②](*Department of Computer Science and Software Engineering, Auburn University, Auburn* 36849, *United States of America*)

Abstract: Federated Learning (FL) performs model training based on local training on clients and continuous model parameters interaction between terminals and server, which effectively solving data leakage and privacy risks in centralized machine learning models. However, since multiple malicious terminals participating in FL can achieve adversarial attacks by inputting small perturbations in the process of local learning, and then lead to incorrect results output by the global model. An effective federated defense strategy – SelectiveFL is proposed in this paper. This strategy first establishes a selective federated defense framework, and then updates the uploaded local model on the server on the basis of extracting attack characteristics through adversarial training at the terminals. At the same time, selective aggregation is carried out according to the attack characteristics, and finally multiple adaptive defense models can be obtained. Finally, the proposed defense method is evaluated on several representative benchmark datasets. The experimental results show that compared with the existing research work, the accuracy of the model can be improved by 2% to 11%.

Key words: Federated Learning (FL); Adversarial attack; Defense strategy; Adversarial training

1 引言

近年来，以物联网为代表的数据采集技术的快速发展使得数据的广泛多元获取成为可能，这显著地促进了机器学习模型的迭代更新[1-3]。而传统的机器学习模型通常基于集中式的模式设计，即由互联网应用服务商将收集到的用户数据或者感知数据汇聚到数据中心，并利用数据中心的强大算力进行模型训练和结果的输出。但这种需要将数据传输并汇聚于第三方的机器学习模型却存在着潜在的数据和隐私泄露风险，这导致了越来越多的数据持有方不愿意进行数据共享，从而形成了“数据孤岛”并导致无法充分地挖掘数据价值。联邦学习(Feder-

收稿日期：2023-03-07；改回日期：2023-08-03；网络出版：2023-08-18

*通信作者： 江辉 jianghui@stu.cqut.edu.cn

基金项目：国家自然科学基金(61471089, 61401076)

Foundation Items: The National Natural Science Foundation of China (61471089, 61401076)

ated Learning, FL)[4]作为一种新兴机器学习范式，基于“数据不动，模型移动”的思想而设计，即模型分别部署在参与FL的多个终端和服务器之上，而数据无需离开持有数据的终端而进行本地模型训练，终端和服务器之间仅存在模型参数或中间结果的交互，通过这样的持续协作完成全局模型的训练。但是，由于参与联邦学习的终端通常没有事先经过严格的信任度验证，这为恶意终端施行针对FL的对抗攻击提供了可能[5]，例如：模型推理的时候，恶意终端通过向输入添加微小的、人类不可感知的扰动来欺骗神经网络模型，从而使得模型以更大的概率表现出不正确或意外的行为[6]。

机器学习模型的输入形式是一种数值型向量，具有恶意企图的攻击者通过有针对性的数值型向量让机器学习模型做出误判，这即是对抗性攻击。Goodfellow 等人[7]提出一种简单但有效的攻击方法-FGSM，利用损失函数的梯度符号来生成对抗样本。Kurakin等人[8]通过多次应用FGSM小步长引入基本迭代法-BIM。Madry等人[9]从干净输入附近的随机位置开始多次应用FGSM提出了一种更强大的迭代攻击-PGD。Dong等人[10]提出基于动量的迭代算法-MIM，将动量项集成到攻击的迭代过程中，并在迭代过程中摆脱较差的局部最大值，产生更多可转移的对抗性示例。文献[11]提出基于梯度的攻击算法-DeepFool，通过迭代生成最小规范对抗扰动，将位于分类边界内的图像逐步推到边界外，直到出现错误分类。为了在最小化干扰的同时获得更好的攻击效果，Carlini等人[12]提出基于优化函数的攻击方法-C&W。除此之外，文献[13]提出一种通用对抗性后门攻击，以欺骗云边缘协作中的垂直联邦学习。文献[14]基于隐私泄露和成对节点梯度添加噪声的全局节点嵌入生成对抗性扰动提出一种新的针对图神经网络的对抗性攻击方法。上述工作通过不同的扰动添加方式，如基于梯度、动量和优化函数等，使得同时防御上述攻击成为困难。

针对各种新出现的对抗性攻击，学界也不断提出防御机制，Papernot等人[15]使用防御蒸馏平滑训练期间的模型来降低对抗样本对模型的有效性。Guo等人[16]采用未修改的模型来度量对抗样本的可转移性差异。文献[17]通过自适应地校准未归一化的概率来平衡类，解决了在具有标签偏度的非独立同分布(Non-Independent Identically Distributed, Non-IID)数据上的训练不稳定性问题。除此之外，文献[18]提出差分隐私自归一化神经网络，在没有显著增加计算开销的神经网络训练和推理情况下提高敌对的鲁棒性。但是，上述工作主要应用于集中式机器学习，难以胜任FL环境下的防御，而本文主要针对FL环境下的多攻击进行联邦防御。虽然已有工作提出联邦防御方法FDA^3[19]，可以聚合来自不同来源的对抗性实例的防御知识，但该方法通过聚合所有防御模型的思路会导致模型对各类对抗攻击的敏感性明显降低。针对该问题，本文提出了一种新的防御策略，称为SelectiveFL。与现有最新工作不同的是，本文不再是进行单一的模型聚合，而是在聚合阶段根据攻击特性的不同，选择性进行模型聚合，保证模型对各类攻击的高敏感性从而实现更高效的联邦防御。主要贡献概括如下。

(1) 本文面向FL中对抗性攻击的多样性，设计了一种新的损失函数并用于对抗性训练。

(2) 本文提出一种新的联邦防御策略(SelectiveFL)，基于对抗性训练提取出不同终端的攻击特性，进而使服务器通过攻击特性进行选择性参数聚合，并实现适应性的对抗防御。

(3) 本文在具有代表性的基准数据集上进行了深入的实验，验证了所提策略的有效性和扩展性。

本文其余部分安排如下。第2节讨论系统模型和问题描述。第3节详细描述本文的联邦防御策略。第4节介绍实验细节和结果。最后，第5节对本文进行总结。

2 问题描述

传统联邦学习通过FedAvg算法聚集本地终端的模型参数来学习联邦模型参数

$$\boldsymbol{\omega}_{\mathrm{g}}^{t}=\sum_{k=1}^{N}\frac{|\boldsymbol{D}_k|}{\sum_{j=1}^{N}|\boldsymbol{D}_j|}\boldsymbol{\omega}_{k}^{t} \tag{1}$$

如图1所示，本文考虑在数据集($\boldsymbol{X}$，$\boldsymbol{Y}$)的C类上进行联邦模型训练，其中$\boldsymbol{X}$是特征空间，$\boldsymbol{Y}$={1,2,⋯,C}是所有类标签的集合。传统的联邦学习目标是获得优化的全局模型参数$\boldsymbol{\omega}_{\mathrm{g}}^{t}$，该参数通过最小化下式中的损失函数，如式(2)所示

$$L(\boldsymbol{\omega}_{\mathrm{g}}^{t})=\sum_{k=1}^{N}\frac{|\boldsymbol{D}_k|}{\sum_{j=1}^{N}|\boldsymbol{D}_j|}L(\boldsymbol{\omega}_{k}^{t}) \tag{2}$$

由于不同终端可能受到不同类型的对抗性攻击的干扰，即每个终端的损失函数由自然样本损失和对抗样本损失组成，每个终端k通过优化其本地模型参数$\boldsymbol{\omega}_{k}^{t}$来最小化本地损失函数$L(\boldsymbol{\omega}_{k}^{t})$，即

$$\min_{\boldsymbol{\omega}_{k}^{t}}L\left(\boldsymbol{\omega}_{k}^{t}\right)=\min_{\boldsymbol{\omega}_{k}^{t}}-\sum_{i=1}^{C}p^{k}(y=i)E_{x|y=i}\cdot\left[\alpha\cdot\ln F_i(x,\boldsymbol{\omega}_{k}^{t})+(1-\alpha)\cdot\ln F_i(\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{\omega}_{k}^{t})\right] \tag{3}$$

其中F_i表示数据点属于$\boldsymbol{Y}$的第i类的概率，α是一个超参数，用于调整自然样本和对抗样本属于$\boldsymbol{Y}$的第i类的概率对数值，$E_{x|y=i}$表示属于第i类的数据点的自然样本和对抗样本的概率对数期望，本文主要符号含义如表1所示。

为了获得最优参数$\boldsymbol{\omega}_k^{t+1}$，使用基于梯度下降的方法并通过式(4)迭代地求解优化问题

$$\begin{aligned}\boldsymbol{\omega}_k^{t+1} =&\boldsymbol{\omega}_k^t-\eta\cdot\nabla_{\boldsymbol{\omega}}L(\boldsymbol{\omega}_k^t)\\ =&\boldsymbol{\omega}_k^t+\eta\cdot\sum_{i=1}^{C}p^k(y=i)\nabla_{\boldsymbol{\omega}}\mathrm{E}_{x|y=i}\\ &\cdot\left[\alpha\cdot\ln F_i(\boldsymbol{x},\boldsymbol{\omega}_k^t)+(1-\alpha)\cdot\ln F_i(\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{\omega}_k^t)\right]\end{aligned}\tag{4}$$

3　适应性对抗性防御策略

3.1　防御框架描述

图2详细描述了SelectiveFL的框架及其工作流程。此框架由两部分组成，FL终端和聚合服务器。FL终端负责模型训练，但由于终端未经过严格的身份认证且FL环境存在各种对抗性攻击，这导致现有联邦防御部署无法实现有效的防御。聚合服务器包括攻击管理模块和防御模型生成模块。攻击管理模块负责管理攻击方案及其防御模型，并根据终端攻击方案进行模型分配。防御模型生成模块负责新防御模型的生成和同步，通过其内部聚合器进行周期性的数据收集，选择性聚合并更新旧模型生成新模型，最终将新模型同步到攻击管理模块中。

每轮训练中各终端需要经历3个步骤。首先，基于攻击管理模块分配防御模型，在本地生成对抗性样本形成再训练集。其次，本地对抗训练周期性上传本地模型更新到聚合器。最后，聚合器选择性聚合得到全局模型更新，并实现新模型的生成和同步。

由于终端多样性和数据异构性，新模型具有更强的鲁棒性，可实现更有力的防御，当新终端加入时应为其分配最新防御模型。并且因为聚合服务器和FL终端之间仅有权重信息交互，并没有相关数据传递，可以保证终端间的数据隐私。除此之外，一些隐私保护方法，例如同态加密[20]和差分隐私[21]等，也可以用来保护敏感数据的隐私。

表 1　主要符号汇总

符号	含义
$\boldsymbol{\omega}_{\mathrm{g}}^t$	迭代轮次t的全局防御模型
$\boldsymbol{\omega}_k^t$	迭代轮次t的终端k的本地防御模型
$\lvert\boldsymbol{D}_k\rvert$	终端k数据集的大小
N	终端数量
$L(\boldsymbol{\omega}_g^t)$	迭代轮次t的全局损失
$L(\boldsymbol{\omega}_k^t)$	迭代轮次t的终端k的本地损失
η	学习率
$\boldsymbol{x}$	自然样本示例
$\boldsymbol{y}$	分类标签
x_{adv}	对抗样本示例
$\boldsymbol{\omega}$	全局防御模型
$\boldsymbol{\omega}^k$	终端k的本地防御模型
$L_{\mathrm{att}}(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}\lvert\boldsymbol{\omega})$	模型在att攻击下的损失函数
$\lvert\boldsymbol{D}_k\rvert_{\mathrm{attack=att}}$	遭受att攻击的终端k所拥有的数据量
$\boldsymbol{\omega}_{\mathrm{g}}^{\mathrm{att},t}$	迭代轮次t中攻击att对应的全局防御模型
$\boldsymbol{\omega}_k^{\mathrm{att},t}$	迭代轮次t中受到att攻击的终端k对应的本地防御模型
δ	对抗攻击扰动

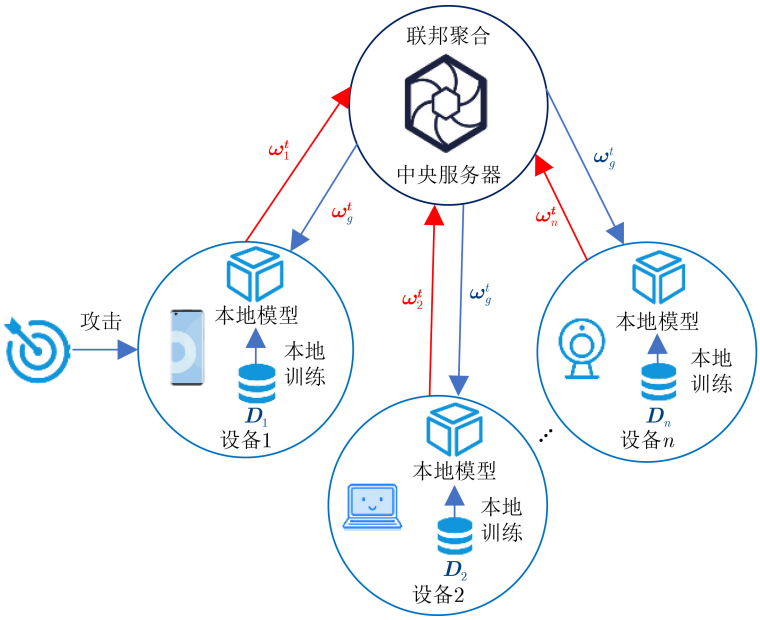

图 1 攻击下的联邦学习框架

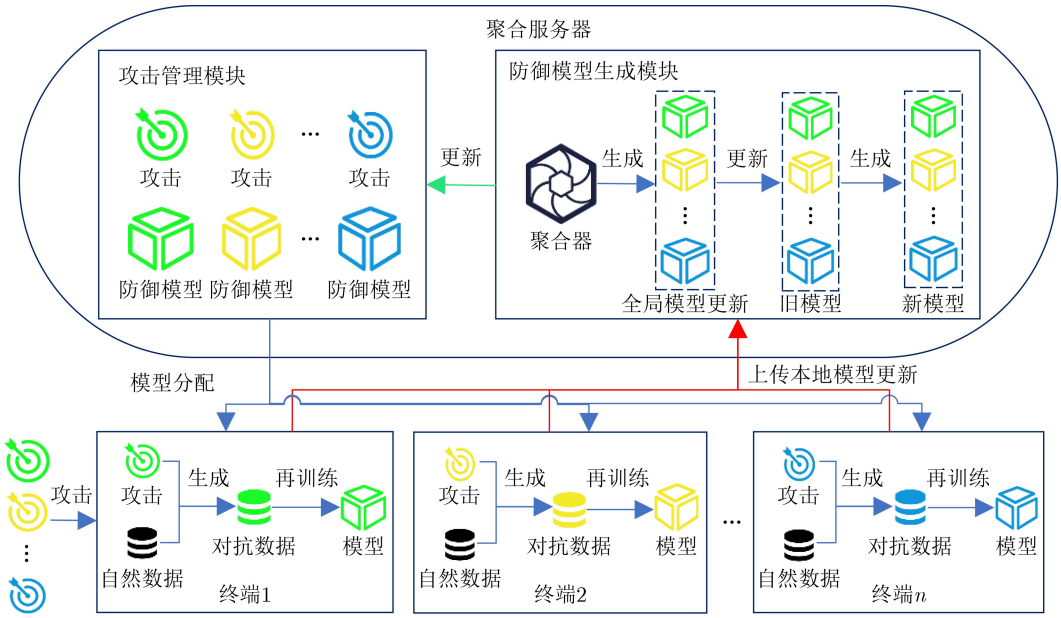

图 2 SelectiveFL的框架和工作流程

3.2 损失函数

机器学习中，损失函数用来度量模型的预测值与真实值的差异程度，损失函数越小，模型的鲁棒性越好。在考虑对抗性攻击的时候，损失函数的定义不同于传统的定义，而是由自然损失函数和对抗损失函数两部分组成，可以表示为

$$L(\boldsymbol{x}^i,\boldsymbol{x}^i_{\rm adv},\boldsymbol{y}^i|\boldsymbol{\omega}^i)=\alpha\cdot L_{\rm nat}(\boldsymbol{x}^i,\boldsymbol{y}^i|\boldsymbol{\omega}^i)+(1-\alpha)\cdot L_{\rm adv}(\boldsymbol{x}^i_{\rm adv},\boldsymbol{y}^i|\boldsymbol{\omega}^i) \tag{5}$$

其中α是超参数，用于根据自然损失函数$L_{\rm nat}(\boldsymbol{x}^i,\boldsymbol{y}^i|\boldsymbol{\omega}^i)$和对抗损失函数$L_{\rm adv}(\boldsymbol{x}^i_{\rm adv},\boldsymbol{y}^i|\boldsymbol{\omega}^i)$来调整损失值的比值，$\alpha$值越高，表明自然损失函数对总体损失函数的贡献权重越高。

在FL环境中存在多种不同类型的对抗攻击，这些对抗攻击由于添加扰动方式的不同，并且同一种攻击还具有多种范数下的攻击(例如：C&W攻击有L_0, L_2和L_∞等3种范数)。为了涵盖N个终端所有可能遭受的攻击，本文根据攻击类型的不同重新定义了如式(6)所示的模型损失函数

$$L_{\rm att}(\boldsymbol{x},\boldsymbol{x}_{\rm adv},\boldsymbol{y}|\boldsymbol{\omega})=\sum_{k=1}^{N}\frac{|\boldsymbol{D}_k|_{\rm attack=att}}{\sum_{j=1}^{N}|\boldsymbol{D}_j|_{\rm attack=att}}L(\boldsymbol{x}^k,\boldsymbol{x}^k_{\rm adv},\boldsymbol{y}^k|\boldsymbol{\omega}^k) \tag{6}$$

因此，本文的联邦防御优化目标就是找出使得$L_{\rm att}(\boldsymbol{x},\boldsymbol{x}_{\rm adv},\boldsymbol{y}|\boldsymbol{\omega})$最小的$\boldsymbol{\omega}_{\rm att}$，可用公式表示为

$$\boldsymbol{\omega}^*_{\rm att}=\arg\min_{\boldsymbol{\omega}}\mathop{\rm E}_{\boldsymbol{x}\subseteq\boldsymbol{D}}\left[\max_{\|\boldsymbol{x}_{\rm adv}-\boldsymbol{x}\|_\infty<\varepsilon}L_{\rm att}(\boldsymbol{x},\boldsymbol{x}_{\rm adv},\boldsymbol{y}|\boldsymbol{\omega})\right] \tag{7}$$

其中$\boldsymbol{D}$表示训练集，$\|\boldsymbol{x}_{\rm adv}-\boldsymbol{x}\|_\infty<\varepsilon$规定了自然样本示例和对抗样本示例之间允许的差异，$\boldsymbol{\omega}^*_{\rm att}$表示在本文的联邦防御下针对att攻击的联邦防御模型。从式(7)可以看出，本文提出的联邦防御方法试图寻找多个针对特定攻击的统一防御模型，它可以实现对多个对抗攻击的精准防御。

3.3 对抗性防御算法

本节将分别介绍SelectiveFL防御策略在聚合服务器和FL终端的算法。算法1详细描述了SelectiveFL防御方法在聚合服务器的执行部分。假设训练开始前攻击管理模块为空。步骤1中，根据攻击方案对模型进行初始化，所有攻击的初始化模型都是使用训练数据集预训练后的模型。步骤2～步骤10

算法1 聚合服务器部署的算法伪码

输入：数据集$\boldsymbol{D}_i$，终端数量N，全局迭代数量T

(1) 初始化各攻击的全局防御模型$\boldsymbol{\omega}_{\rm g}^{\rm att,0}$;
(2) for t=0,1,$\cdots$,T–1 do
(3) 随机抽取一组终端集合S_t;
(4) for $i\in\boldsymbol{S}_t$ in parallel do
(5) 事件：接收终端的攻击方案att;
(6) 发送att攻击的全局防御模型$\boldsymbol{\omega}_{\rm g}^{\rm att,t}$给终端;
(7) 事件：接收终端的本地模型更新$\Delta\boldsymbol{\omega}_i^{\rm att,t+1}$;
(8) end
(9) $\boldsymbol{\omega}_{\rm g}^{\rm att,t+1}=\boldsymbol{\omega}_{\rm g}^{\rm att,t}+\sum_{j\in S_t}\frac{|\boldsymbol{D}_j|_{\rm attack=att}}{\sum_{i\in S_t}|\boldsymbol{D}_i|_{\rm attack=att}}\Delta\boldsymbol{\omega}_j^{\rm att,t+1}$
(10) end

中，服务器需要与所有终端进行T轮交互，以形成多个针对攻击的联合防御模型。步骤3中，根据一定的比率从所有终端中选择一部分进行联合防御。在步骤5接收到终端发送的攻击方案后，步骤6服务器为其分配相应的防御模型。在步骤7接收所有终端的本地模型更新后，步骤9根据攻击特性进行选择性聚合，实现防御模型的更新，如式(8)所示。注意，步骤5和步骤7是一个阻塞事件，需要等待事件完成后才能继续执行。

$$\boldsymbol{\omega}_{\mathrm{g}}^{\mathrm{att},t+1}=\boldsymbol{\omega}_{\mathrm{g}}^{\mathrm{att},t}+\sum_{j\in S_t}\frac{|\boldsymbol{D}_j|_{\mathrm{attack=att}}}{\sum\limits_{i\in S_t}|\boldsymbol{D}_i|_{\mathrm{attack=att}}}\Delta\boldsymbol{\omega}_j^{\mathrm{att},t+1} \tag{8}$$

其中S_t是迭代轮次t中参与者的集合，$\Delta\boldsymbol{\omega}_j^{\mathrm{att},t+1}$是迭代轮次$t$+1中受到att攻击的终端$j$对应的本地模型更新。$\Delta\boldsymbol{\omega}_j^{\mathrm{att},t+1}$定义为

$$\Delta\boldsymbol{\omega}_j^{\mathrm{att},t+1}=\boldsymbol{\omega}_j^{\mathrm{att},t+1}-\boldsymbol{\omega}_j^{\mathrm{att},t} \tag{9}$$

算法2详细描述了SelectiveFL防御方法在FL终端的执行部分。假设索引为i的FL终端已加入联邦防御。步骤1中，终端上传自身攻击方案到聚合服务器。步骤2中，终端接收聚合服务器发送的全局防御模型。注意，步骤2是一个阻塞事件，需要等待接收到对应全局防御模型后才能继续执行。步骤3将全局防御模型转换为本地防御模型。步骤4中，终端基于本地防御模型和攻击方案通过对抗工具箱将本地数据生成为对抗样本。步骤5～步骤14中，终端进行E轮的本地训练，以获得更高防御能力的本地防御模型。在步骤7和步骤8中分别得到自然样本和对抗样本通过本地防御模型后的预测标签，在步骤9和步骤10中分别计算自然样本和对抗样本的交叉熵后，步骤11将自然损失和对抗损失通过式(5)中定义的损失函数计算总损失，步骤12通过梯度下降进行权重更新，如式(10)所示。在步骤15计算本地模型更新后，步骤17将本地模型更新上传到服务器。

算法2　FL终端部署的算法伪码

输入：数据集$\boldsymbol{D}_i$，终端模型$\boldsymbol{F}$，本地mini-batch大小B，本地迭代数量E，学习率η，超参数α

(1) 上传攻击方案att到聚合服务器;
(2) 事件：接收att攻击的全局防御模型$\boldsymbol{\omega}_{\mathrm{g}}^{\mathrm{att},t}$;
(3) 转换为本地防御模型：$\boldsymbol{\omega}_i^{\mathrm{att},t}\leftarrow\boldsymbol{\omega}_{\mathrm{g}}^{\mathrm{att},t}$;
(4) $\boldsymbol{x}_{i,\mathrm{adv}}=\mathrm{AdvGen}(x_i,\mathrm{att},F,\boldsymbol{\omega}_i^{\mathrm{att},t})$ //对抗工具箱
(5) for e=0,1,⋯,E–1 do
(6)　　for b=0,1,...,$\left\lceil\frac{|\boldsymbol{D}_i|}{B}\right\rceil-1$ do
(7)　　$\boldsymbol{y}_i^*=\boldsymbol{F}(\boldsymbol{x}_i,\boldsymbol{\omega}_i^{\mathrm{att},t})$
(8)　　$\boldsymbol{y}_{i,\mathrm{adv}}^*=\boldsymbol{F}(\boldsymbol{x}_{i,\mathrm{adv}},\boldsymbol{\omega}_i^{\mathrm{att},t})$
(9)　　$\mathrm{nat\_loss}=\mathrm{CrossEntropy}(\boldsymbol{y}_i^*,\boldsymbol{y}_i)$
(10)　　$\mathrm{adv\_loss}=\mathrm{CrossEntropy}(\boldsymbol{y}_{i,\mathrm{adv}}^*,\boldsymbol{y}_i)$
(11)　　$L=\alpha\cdot\mathrm{nat\_loss}+(1-\alpha)\cdot\mathrm{adv\_loss}$
(12)　　$\boldsymbol{\omega}_i^{\mathrm{att},t+1}=\boldsymbol{\omega}_i^{\mathrm{att},t}-\eta\cdot\nabla L(\boldsymbol{\omega}_i^{\mathrm{att},t},\boldsymbol{x}_i,\boldsymbol{x}_{i,\mathrm{adv}},\boldsymbol{y}_i)$
(13)　end
(14) end
(15) 计算本地模型更新;
(16) $\Delta\boldsymbol{\omega}_i^{\mathrm{att},t+1}=\boldsymbol{\omega}_i^{\mathrm{att},t+1}-\boldsymbol{\omega}_i^{\mathrm{att},t}$
(17) 上传本地模型更新到聚合服务器;

$$\boldsymbol{\omega}_i^{\mathrm{att},t+1}=\boldsymbol{\omega}_i^{\mathrm{att},t}-\eta\cdot\nabla L(\boldsymbol{\omega}_i^{\mathrm{att},t},\boldsymbol{x}_i,\boldsymbol{x}_{i,\mathrm{adv}},\boldsymbol{y}_i) \tag{10}$$

其中$\boldsymbol{\omega}_i^{\mathrm{att},t+1}$通过Adam优化器来降低随机梯度$\nabla L(\boldsymbol{\omega}_i^{\mathrm{att},t},\boldsymbol{x}_i,\boldsymbol{x}_{i,\mathrm{adv}},\boldsymbol{y}_i)$实现模型更新。注意，Adam优化器绝不是定义对抗性攻击的黄金优化器，其他类型的优化器也可以集成到本文算法中。

3.4　可行性分析

本节通过定理1和定理2对所提方法Selective-FL进行可行性分析。

定理1　对抗训练作为一种正则化方法使模型更加鲁棒和稳健。

证明　由式(5)知，对抗训练损失函数表示为$L(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega})=\alpha\cdot L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega})+(1-\alpha)\cdot L(\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega})$，其中，$\boldsymbol{x}_{\mathrm{adv}}=\boldsymbol{x}+\delta$，如果扰动较小，可以使用一阶泰勒展开来近似，则损失函数变为

$$\begin{aligned}L(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega})\approx&\alpha\cdot L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega})+(1-\alpha)\cdot L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega})\\&+(1-\alpha)\cdot\delta\cdot\partial_x L(x,y|\omega)\\=&L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega})+(1-\alpha)\cdot\delta\cdot\partial_x L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega})\end{aligned} \tag{11}$$

其中第2项为扰动给损失函数带来的影响，即

$$\begin{aligned}\delta\cdot\partial_{\boldsymbol{x}}L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega})=&\max_{\|\delta\|_\infty<\varepsilon}|L(\boldsymbol{x}+\delta,\boldsymbol{y}|\boldsymbol{\omega})-L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega})|\\\approx&\max_{\|\delta\|_\infty<\varepsilon}|\partial_{\boldsymbol{x}}L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega})\cdot\delta|\\=&\varepsilon\|\partial_{\boldsymbol{x}}L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega})\|_1\end{aligned} \tag{12}$$

其中$\|\cdot\|_1$是$\|\cdot\|_\infty$的对偶范数，定义为

$$\|\boldsymbol{x}\|_1=\sup_{\|\boldsymbol{z}\|_\infty\le1}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{z} \tag{13}$$

将式(12)中的结果代入式(11)得到

$$L(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega})\approx L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega})+(1-\alpha)\cdot\varepsilon\cdot\|\partial_{\boldsymbol{x}}L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega})\|_1 \tag{14}$$

即加入一个特殊的针对梯度的正则化。　证毕

定理2　通过对抗攻击A得到的防御模型针对A攻击的防御力最强。

证明　由式(7)可知$\boldsymbol{\omega}_{\mathrm{A}}^*=\arg\min\limits_{\boldsymbol{\omega}}\mathrm{E}\limits_{\boldsymbol{x}\subseteq\boldsymbol{D}}\left[\max\limits_{\|\boldsymbol{x}_{\mathrm{adv}}-\boldsymbol{x}\|_\infty<\varepsilon}L_{\mathrm{A}}(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega})\right]$，如果将这个最优模型用于防御A攻击，由式(6)可知$L_{\mathrm{att}}(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega})\propto L(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega})$，可以得到防御A攻击的损失函数

$$L_{\mathrm{A}}(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega}_{\mathrm{A}}^{*}) \propto L(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega}_{\mathrm{A}}^{*}) = \alpha\cdot L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega}_{\mathrm{A}}^{*}) + (1-\alpha)\cdot L(\boldsymbol{x}+\delta_{\mathrm{A}},\boldsymbol{y}|\boldsymbol{\omega}_{\mathrm{A}}^{*}) \tag{15}$$

同理我们可以得到防御非A攻击的损失函数

$$L_{\bar{\mathrm{A}}}(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega}_{\mathrm{A}}^{*}) \propto L(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega}_{\mathrm{A}}^{*}) = \alpha\cdot L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega}_{\mathrm{A}}^{*}) + (1-\alpha)\cdot L(\boldsymbol{x}+\delta_{\bar{\mathrm{A}}},\boldsymbol{y}|\boldsymbol{\omega}_{\mathrm{A}}^{*}) \tag{16}$$

由于模型$\boldsymbol{\omega}_{\mathrm{A}}^{*}$是通过最大最小化$L_{\mathrm{A}}(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega})$得到的，由此可以得出

$$\begin{aligned}\boldsymbol{\omega}_{\mathrm{A}}^{*} &= \arg\min_{\boldsymbol{\omega}} \mathop{E}_{\boldsymbol{x}\subseteq\boldsymbol{D}}\left[\max_{\|\boldsymbol{x}_{\mathrm{adv}}-\boldsymbol{x}\|_{\infty}<\varepsilon} L_{\mathrm{A}}(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega})\right]\\ &\propto \arg\min_{\omega} \mathop{\mathrm{E}}_{\boldsymbol{x}\subseteq\boldsymbol{D}}\left[\max_{\delta_{\mathrm{A}}<\varepsilon}(\alpha\cdot L(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\omega}) + (1-\alpha)\cdot L(\boldsymbol{x}+\delta_{\mathrm{A}},\boldsymbol{y}|\boldsymbol{\omega}))\right]\end{aligned} \tag{17}$$

从上式知，$\boldsymbol{\omega}_{\mathrm{A}}^{*}$使得$L(\boldsymbol{x}+\delta_{\mathrm{A}},\boldsymbol{y}|\boldsymbol{\omega})$最小，即

$$L(\boldsymbol{x}+\delta_{\mathrm{A}},\boldsymbol{y}|\boldsymbol{\omega}_{\mathrm{A}}^{*}) \le L(\boldsymbol{x}+\delta_{\bar{\mathrm{A}}},\boldsymbol{y}|\boldsymbol{\omega}_{\mathrm{A}}^{*}) \tag{18}$$

综上所述，可以得到

$$L_{\mathrm{A}}(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega}_{\mathrm{A}}^{*}) \le L_{\bar{\mathrm{A}}}(\boldsymbol{x},\boldsymbol{x}_{\mathrm{adv}},\boldsymbol{y}|\boldsymbol{\omega}_{\mathrm{A}}^{*}) \tag{19}$$

这意味着通过对抗攻击A得到的防御模型针对A攻击的拟合效果更好。因此，模型对于A攻击的防御力是最强的。 证毕

通过定理1和定理2说明，SelectiveFL在终端进行对抗训练得到本地最优防御模型，并在服务器端对上传的本地模型更新根据攻击特性进行选择性聚合可得到对各个攻击的全局最优防御模型。

4 实验及结果分析

4.1 实验设置

数据集：本文在3个常用数据集MNIST, F-MNIST和SVHN上评估了所提方法的性能。MNIST和F-MNIST分别是灰度手写数字和衣服的集合，它们都由60 000个训练样本和10 000测试样本组成，每个样本为28×28的灰度图像。SVHN数据集摘自Google街景图像中的门牌号，由73257个训练样本和26 032个测试样本组成，每个样本为32×32的彩色图像。

攻击方案：本文采用了6种基准对抗攻击，包括：FGSM[7], BIM[8], PGD[9], MIM[10], DeepFool[11], C&W[12]。其中FGSM, BIM, PGD, MIM的实现由Advertorch提供，DeepFool和C&W的实现由Foolbox提供。为了实验公平，除FGSM外的所有攻击的迭代次数设为10。针对MNIST和F-MNIST数据集，步长设为0.01，ε设为0.3。针对SVHN数据集，步长设为0.007，ε设为0.031。请注意，本文主要关注L_{∞}距离。

对比算法：为了评估我们防御算法的性能，本文使用文献[19]的FDA^{3}算法作为本实验的基准(base)算法，为了实验公平，本文方法与基准方法采用相同的训练参数。

架构：本文分别使用LeNet模型对MNIST 和F-MNIST数据集，ResNet-18模型对SVHN数据集进行实验。使用来自MNIST和F-MNIST数据集的训练示例分别预训练初始LeNet模型，使用来自SVHN数据集的训练示例预训练初始ResNet-18模型。请注意，本文将以上3个数据的测试集分为两半，其中一半测试集用于重新训练，另一半测试集用于测试。预训练阶段将超参数α设为1，这表明只使用自然样本进行预训练；再训练阶段将超参数α设为0，这表明只使用对抗样本进行再训练。

4.2 单一攻击防御性能

本文首先评估在数据集符合IID分布情况下，SelectiveFL对白盒攻击的防御能力。特别的，在白盒攻击中，对手对分类器足够了解。这里主要评估6种具有代表性的对抗性攻击的防御能力，包括FGSM, BIM, PGD, MIM, DeepFool, C&W，考虑在FL环境下10个终端进行联合防御。图3给出在单一攻击下的自然和鲁棒准确性的结果。

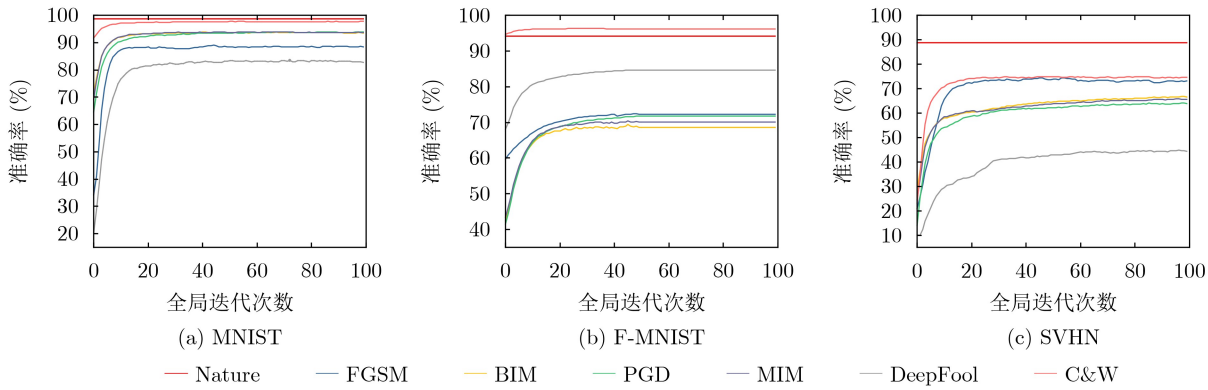

图 3 单一攻击下的自然和鲁棒准确性

从图3中，可以发现各数据集在干净图像都达到较高的精度，而在各攻击下的鲁棒准确性都有一定程度的下降，并且在各数据集上都很快实现了收敛。例如：在SVHN数据集的干净图像上精度达到94.22%，而在PGD攻击下的鲁棒精度只达到了71.54%。对于这种下降，我们分析主要由攻击本身特性和数据集复杂程度两个原因造成：从攻击本身特性分析，相较于单一迭代的FGSM，多次迭代的BIM, MIM, PGD以及DeepFool对FL的影响更大，而基于优化的C&W攻击添加的扰动人眼几乎无法察觉，但是攻击时间长，通过调整攻击参数可以人为控制攻击力度；从数据集复杂程度分析，MNIST数据集最简单，而SVHN数据集最复杂，由于复杂数据集所拥有的数据特征更复杂且数据量更大，通过对数据集添加微小扰动的对抗性攻击会使得复杂数据集更难以学习，从而出现更加明显的下降。该实验表明在典型的FL环境下进行联邦学习，终端容易受到对抗攻击的干扰，通过SelectiveFL能够实现不同数据集下对单一对抗攻击的防御。

4.3 多攻击性能比较

本文全面评估了SelectiveFL在多种攻击下的FL终端的鲁棒准确性。实验模拟了10个终端进行联合防御，终端受到来自FGSM, BIM, PGD, MIM等4种攻击的随机性攻击，即在各个轮次中，终端所受到的攻击都是随机的。图4显示了本文算法与基线算法在多攻击下的预测精度。

如图4所示，本文算法在MNIST, F-MNIST和SVHN数据集的预测精度分别达到91.39%, 66.62%和70.33%，而Base算法只达到88.47%, 56.60%和65.28%。这是因为在多攻击下将众多模型聚合在一起会降低模型对单一攻击的敏感性，而我们的算法通过攻击方式进行选择性聚合，保证了每个模型对攻击的敏感性。换句话说，我们的算法相较于Base算法能够达到更高的鲁棒准确性，实现更优的联合防御。

4.4 Non-IID性能分析

为评估SelectiveFL在Non-IID下的性能，利用参数α=0.5的狄利克雷分布生成各终端的数据分布，在10个终端下进行联合防御，终端受到来自FGSM, BIM, PGD, MIM等4种攻击的随机性攻击，并与IID分布进行性能比较。图5显示本文算法与Base算法在IID和Non-IID下的预测精度对比。

从图5中，可以看到IID和Non-IID分布下的准确性存在一定的差异，在MNIST和F-MNIST数据集上，差异不太明显，可能是测试数据集相对较小的原因；而在SVHN数据集上，Non-IID分布下的准确性明显低于IID分布下的准确性，这说明数据分布不均的Non-IID分布同样会对防御性能产生一定的影响。其次，本文算法在Non-IID分布下的MNIST, F-MNIST和SVHN数据集上的准确性分别达到90.17%, 67.23%和66.12%，而Base算法只达到88.72%, 56.37%和62.02%，但是都在较少轮次内实现收敛，具有较优的收敛性能。可以看到，无论数据集是符合IID分布还是符合Non-IID分布的情况下，SelectiveFL算法都优于Base算法。

4.5 攻击扩展性分析

之前的实验只考虑了4种攻击的情况，但是FL环境复杂且面临的潜在攻击种类较多，为了评估SelectiveFL在更为复杂的FL环境中的表现，本

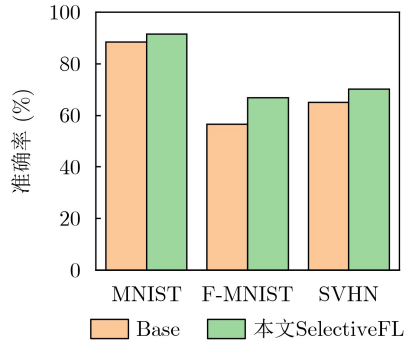

图 4 多攻击下的鲁棒准确性

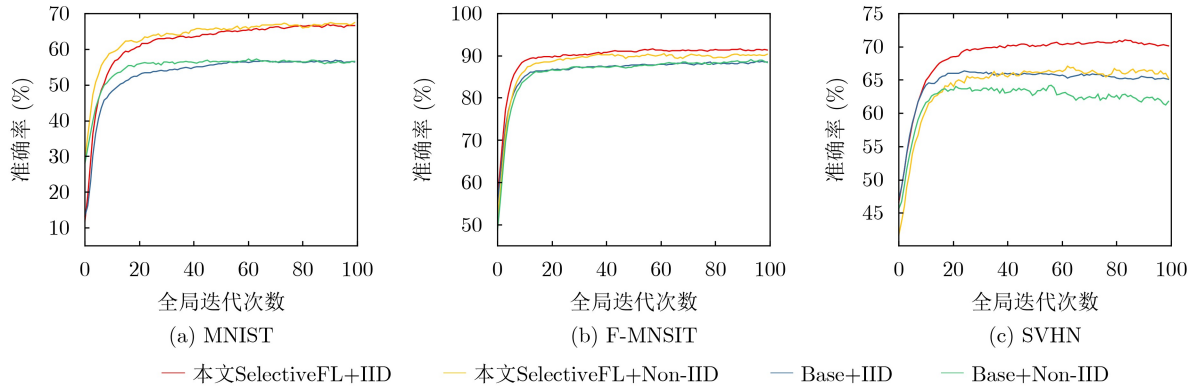

图 5 IID和Non-IID下的鲁棒准确性

文继续评估了SelectiveFL对攻击的可伸缩性，表2展示了不同攻击类型下的鲁棒准确性。该实验使用10个终端进行联合防御，攻击从{FGSM, BIM, PGD, MIM, DeepFool, C&W}中依次抽取攻击类型数量$A\in\{2,3,4,5,6\}$进行实验，即当A=2时，终端受到来自FGSM和BIM两种攻击类型的混合攻击，随着A的增大，依次加入攻击类型。表中数据为收敛后的最后10个轮次的平均值。从表2中，可以发现，随着攻击类型数量的增加，准确性发生了一些不规则变化，这是由于不同攻击类型对不同复杂度的数据集影响不同。但是无论攻击类型数量如何变化，SelectiveFL所达到的准确性都是高于Base算法的，即使在复杂的FL环境下依然能达到一个较优的防御能力。

4.6 终端扩展性分析

前面的实验只考虑了10个终端联合防御的情况，而传统FL环境下可能同时存在规模更大的终端参与FL协同学习。为了验证SelectivFL在更大节点规模下的表现，本文继续评估了SelectiveFL对终端的可伸缩性，表3展示了不同终端数目下的鲁棒准确性。该实验考虑4种攻击类型{FGSM, BIM, PGD, MIM}，终端数目$C\in\{10,20,30,40,50\}$。从表3可以发现，当更多的终端参与联邦防御时，准确性仍然有一定程度的提升。当终端从10增加到50，本文算法在MNIST, F-MNIST和SVHN的准确性分别提高了0.91%, 1.25%和0.55%。除此之外，在3个数据集上，终端数从10到50，本文算法准确性都是优于Base算法的。由此可以看出，SelectivFL在大规模FL环境下也能实现更优的防御性能。

表 2 不同攻击类型数量的鲁棒准确性(%)

数据集	算法	攻击类型数量				
		2	3	4	5	6
MNIST	Base	86.50	90.77	88.47	88.24	88.32
	本文SelectiveFL	90.89	90.92	91.39	89.95	89.77
F-MNIST	Base	60.71	53.64	56.60	53.33	54.26
	本文SelectiveFL	67.81	67.87	66.62	60.42	61.27
SVHN	Base	63.81	63.94	65.28	69.35	73.39
	本文SelectiveFL	70.40	70.93	70.33	73.31	77.15

表 3 不同终端数目的鲁棒准确性(%)

数据集	算法	终端数目				
		10	20	30	40	50
MNIST	Base	88.47	89.23	89.37	89.53	89.53
	本文SelectiveFL	91.39	91.95	92.12	92.31	92.30
F-MNIST	Base	56.60	55.56	57.00	57.26	56.95
	本文SelectiveFL	66.62	66.11	67.64	67.81	67.87
SVHN	Base	65.28	65.68	65.11	65.46	65.66
	本文SelectiveFL	70.33	70.73	70.80	70.89	70.88

5 结束语

本文提出了一种面对由FL环境中终端发起对抗性攻击的防御策略。该策略通过在多个参与协同学习的终端上进行对抗性训练以提取攻击特性，而在服务器端根据攻击特性进行选择性聚合，提升了模型对攻击的敏感性，进而使得FL模型能够有效抵抗不同来源的多种对抗攻击。本文在3个著名的基准数据集上进行了大量实验，验证了该方法的有效性和可扩展性，显著提升了模型在遭受对抗攻击情况下的模型精确度。本文为在FL环境下改进模型的安全性提供了借鉴。

参 考 文 献

[1] WU Yulei, DAI Hongning, and WANG Hao. Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0[J]. *IEEE Internet of Things Journal*, 2021, 8(4): 2300–2317. doi: 10.1109/JIOT.2020.3025916.

[2] LIU Yi, YU J J Q, KANG Jiawen, *et al*. Privacy-preserving traffic flow prediction: A federated learning approach[J]. *IEEE Internet of Things Journal*, 2020, 7(8): 7751–7763. doi: 10.1109/JIOT.2020.2991401.

[3] KHAN L U, YAQOOB I, TRAN N H, *et al*. Edge-computing-enabled smart cities: A comprehensive survey[J]. *IEEE Internet of Things Journal*, 2020, 7(10): 10200–10232. doi: 10.1109/JIOT.2020.2987070.

[4] WAN C P and CHEN Qifeng. Robust federated learning with attack-adaptive aggregation[EB/OL]. https://doi.org/10.48550/arXiv.2102.05257, 2021.

[5] HONG Junyuan, WANG Haotao, WANG Zhangyang, *et al*. Federated robustness propagation: Sharing adversarial robustness in federated learning[C/OL]. The Tenth International Conference on Learning Representations, 2022.

[6] REN Huali, HUANG Teng, and YAN Hongyang. Adversarial examples: Attacks and defenses in the physical world[J]. *International Journal of Machine Learning and Cybernetics*, 2021, 12(11): 3325–3336. doi: 10.1007/s13042-020-01242-z.

[7] GOODFELLOW I J, SHLENS J, and SZEGEDY C. Explaining and harnessing adversarial examples[C]. The 3rd International Conference on Learning Representations, San Diego, USA, 2015: 1–11.

[8] KURAKIN A, GOODFELLOW I J, and BENGIO S. Adversarial examples in the physical world[C]. The 5th

International Conference on Learning Representations, Toulon, France, 2017: 1–14.

[9] MADRY A, MAKELOV A, SCHMIDT L, *et al*. Towards deep learning models resistant to adversarial attacks[C]. The 6th International Conference on Learning Representations, Vancouver, Canada, 2018.

[10] DONG Yinpeng, LIAO Fangzhou, PANG Tianyu, *et al*. Boosting adversarial attacks with momentum[C]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, 2018: 9185–9193. doi: 10.1109/CVPR.2018.00957.

[11] MOOSAVI-DEZFOOLI S M, FAWZI A, and FROSSARD P. DeepFool: A simple and accurate method to fool deep neural networks[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, USA, 2016: 2574–2582. doi: 10.1109/CVPR.2016.282.

[12] CARLINI N and WAGNER D. Towards evaluating the robustness of neural networks[C]. 2017 IEEE Symposium on Security and Privacy (SP), San Jose, USA, 2017: 39–57. doi: 10.1109/SP.2017.49.

[13] CHEN Peng, DU Xin, LU Zhihui, *et al*. Universal adversarial backdoor attacks to fool vertical federated learning in cloud-edge collaboration[EB/OL]. https://doi.org/10.48550/arXiv.2304.11432, 2023.

[14] CHEN Jinyin, HUANG Guohan, ZHENG Haibin, *et al*. Graph-fraudster: Adversarial attacks on graph neural network-based vertical federated learning[J]. *IEEE Transactions on Computational Social Systems*, 2023, 10(2): 492–506. doi: 10.1109/TCSS.2022.3161016.

[15] PAPERNOT N, MCDANIEL P, WU Xi, *et al*. Distillation as a defense to adversarial perturbations against deep neural networks[C]. 2016 IEEE Symposium on Security and Privacy (SP), San Jose, USA, 2016: 582–597. doi: 10.1109/SP.2016.41.

[16] GUO Feng, ZHAO Qingjie, LI Xuan, *et al*. Detecting adversarial examples via prediction difference for deep neural networks[J]. *Information Sciences*, 2019, 501: 182–192. doi: 10.1016/j.ins.2019.05.084.

[17] CHEN Chen, LIU Yuchen, MA Xingjun, *et al*. CalFAT: Calibrated federated adversarial training with label skewness[EB/OL]. https://doi.org/10.48550/arXiv.2205.14926, 2022.

[18] IBITOYE O, SHAFIQ M O, and MATRAWY A. Differentially private self-normalizing neural networks for adversarial robustness in federated learning[J]. *Computers & Security*, 2022, 116: 102631. doi: 10.1016/j.cose.2022.102631.

[19] SONG Yunfei, LIU Tian, WEI Tongquan, *et al*. FDA^3: Federated defense against adversarial attacks for cloud-based IIoT applications[J]. *IEEE Transactions on Industrial Informatics*, 2021, 17(11): 7830–7838. doi: 10.1109/TII.2020.3005969.

[20] FENG Jun, YANG L T, ZHU Qing, *et al*. Privacy-preserving tensor decomposition over encrypted data in a federated cloud environment[J]. *IEEE Transactions on Dependable and Secure Computing*, 2020, 17(4): 857–868. doi: 10.1109/TDSC.2018.2881452.

[21] FENG Jun, YANG L T, REN Bocheng, *et al*. Tensor recurrent neural network with differential privacy[J]. *IEEE Transactions on Computers*, 2023: 1–11. doi: 10.1109/TC.2023.3236868.

陈　卓：男，副教授，博士，硕士生导师，研究方向为联邦学习与物联网技术.

江　辉：男，硕士生，研究方向为联邦学习与分布.

周　杨：男，助理教授，博士，博士生导师，研究方向为机器学习及隐私保护.

责任编辑：马秀强