面向异构环境的物联网入侵检测方法

刘静[1,2]，慕泽林[1]，赖英旭[1,2]

（1.北京工业大学信息学部，北京 100124；2.智能感知与自主控制教育部工程研究中心，北京 100124）

摘　要：为了解决物联网设备在资源受限和数据非独立同分布（Non-IID）时出现的训练效率低、模型性能差的问题，提出了一种个性化剪枝联邦学习框架用于物联网的入侵检测。首先，提出了一种基于通道重要性评分的结构化剪枝策略，该策略通过平衡模型的准确率与复杂度来生成子模型下发给资源受限客户端。其次，提出了一种异构模型聚合算法，对通道采用相似度加权系数进行加权平均，有效降低了Non-IID数据在模型聚合中的负面影响。最后，网络入侵数据集BoT-IoT上的实验结果表明，相较于现有方法，所提方法能显著降低资源受限客户端的时间开销，处理速度提升20.82%，并且在Non-IID场景下，入侵检测的准确率提高0.86%。

关键词：联邦学习；入侵检测；模型剪枝；非独立同分布

中图分类号：TP302

文献标志码：A

DOI：10.11959/j.issn.1000−436x.2024087

Intrusion detection method for IoT in heterogeneous environment

LIU Jing[1,2], MU Zelin[1], LAI Yingxu[1,2]

1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
2. Engineering Research Center of Intelligent Perception and Autonomous Control, Ministry of Education, Beijing 100124, China

Abstract: In order to address the issue of inadequate training efficiency and subpar model performance encountered by Internet of things (IoT) devices when dealing with resource constraints and non-independent and identically distributed (Non-IID) data, a novel personalized pruning federated learning frame work for IoT intrusion detection was put forth. Initially, a channel importance scoring-based structured pruning strategy was proposed, facilitating the generation of submodels to be disseminated to resource-limited clients, thereby harmonizing model accuracy and complexity. Subsequently, an innovative heterogeneous model aggregation algorithm was introduced, utilizing similarity-weighted coefficients for channel averaging, thereby effectively mitigating the adverse effects of Non-IID data during the model aggregation process. Ultimately, experimental results derived from the network intrusion dataset BoT-IoT substantiate that, relative to existing methods, the proposed method notably curtails the time expenditure of resource-constrained clients, and improves processing speed by 20.82%, while enhancing the accuracy of intrusion detection by 0.86% in Non-IID conditions.

Keywords: federated learning, intrusion detection, model pruning, Non-IID

0　引言

随着5G商用的持续拓展和6G战略性布局拉开帷幕，海量物联网设备正在加速接入互联网[1-2]。因此，物联网设备面临的问题更加复杂，如网络协议多样化、硬件组成碎片化、地理部署分散化等，同时安全机制设计过于简单，导致了面向物联网设备的攻击呈现爆发式增长，频繁出现网络系统瘫痪、用户隐私泄露等严重的网络安全事件[3-4]。为了应对复杂的网络攻击，工业界和学术界对物联网入侵检测技术正在倾注更多的关注和研究，以期成

收稿日期：2023−11−21；**修回日期：**2024−02−05

通信作者：赖英旭，laiyingxu@bjut.edu.cn

基金项目：国家自然科学基金资助项目（No.62372017）

Foundation Item：The National Natural Science Foundation of China (No.62372017)

功应对潜在安全威胁。

研究人员提出了许多基于深度学习的解决方案，用于检测物联网中的攻击和异常[5-7]。但在实际应用中，单一物联网设备只产生少量的流量样本，导致神经网络模型训练难度高，攻击检测准确率低[8-9]。同时，传统的云计算技术需要将本地数据传输到服务器端，这将给用户带来数据的隐私安全问题以及严重的传输时延。为了解决上述问题，Rahman 等[10]研究了基于联邦学习的入侵检测方法，其允许多个设备在不用共享原始数据的情况下，共同训练入侵检测模型，有效改善了数据孤岛和数据隐私问题。

虽然目前基于联邦学习的入侵检测算法具有很高的检测准确率，但相关研究没有考虑到物联网环境中设备计算能力不足、网络带宽低等资源受限的问题[11]，这些问题都会增加联邦学习的训练时间和通信量。尽管有研究提出了基于异构模型的联邦学习方法解决客户端资源受限问题[12]，但其没有考虑到在非独立同分布（Non-IID, non-independent and identically distributed）场景下，异构模型学习能力的差异性导致的联邦聚合模型准确率下降问题。这些问题共同构成了基于联邦学习的物联网入侵检测的挑战。

为了应对上述挑战，本文提出了一种个性化剪枝联邦学习框架，称为 FedTP（federated transcendent pruning）。该框架采用一种个性化剪枝算法为异构环境下的客户端生成剪枝后的子模型，然后发送给客户端进行本地训练。框架还提出了一种聚合算法能够对 Non-IID 场景下的异构的子模型进行联邦聚合。因为资源受限的客户端只训练和传输剪枝后的子模型，从而减少了模型的复杂度和参数量，能够提高异构环境下联邦学习的计算和通信效率。本文主要的研究工作如下。

1) 提出了一种个性化剪枝算法，该算法为每个客户端实施个性化的剪枝策略，以平衡模型准确率与复杂度。资源受限的客户端以剪枝后的子模型参加联邦学习，有效减少了训练时间。

2) 提出了一种异构模型聚合算法，该算法能够聚合网络结构不同的子模型。为了进一步减轻 Non-IID 数据对联邦学习准确率的影响，该算法在通道级别细化了模型聚合过程，引入了基于相似度的权重系数，可以更精确地调节聚合过程中各通道的贡献值。

3) 在资源受限且数据分布呈现 Non-IID 特性的环境下，通过 BoT-IoT 数据集对提出的算法进行了验证。实验结果显示，与现有方法相比，本文方法能显著减少客户端的训练时间，实现了 20.82% 的速度提升，并且在 Non-IID 数据场景下，能将入侵检测的准确率提升 0.86%。

1　相关研究

本节对基于联邦学习的入侵检测相关研究进行了回顾，分析了目前传统的模型剪枝方法在面向异构环境下的联邦学习入侵检测方法中面临的挑战。

1.1　基于联邦学习的物联网入侵检测方法

McMahan 等[13]提出联邦学习用于增强用户隐私保护，随后基于联邦学习的物联网入侵检测方法研究逐渐兴起。联邦学习为物联网入侵检测带来了准确率的显著提高，但作为一种分布式机器学习框架，它可能引入大量的计算和通信负担，成为模型训练的瓶颈[14]。为了应对这些挑战，Truong 等[15]利用自动编码器、Transformer 和傅里叶子层来进行异常检测，通过自动编码器来降低输入数据的维度，然后将 Transformer 中的注意力层替换为离散傅里叶变换来加快 Transformer 的运行时间。Chen 等[16]引入注意力机制来计算客户端的重要性，重要性低于阈值的客户端的参数将不被服务器端接受，以此来减少总的通信开销。Wang 等[17]提出了一个两步特征选择方案来选择入侵检测系统的基本特征，通过去除冗余特征简化模型架构并缩短训练时间。

尽管以上物联网场景下基于联邦学习的入侵检测工作在减少计算开销或通信开销方面都做出了贡献，但都忽略了参与联邦学习的边缘节点具有异构性，即边缘节点的可用资源情况是不同的。一些边缘节点的资源充足，能够满足复杂的计算任务和昂贵的通信开销；而另外一些边缘节点的计算能力和通信带宽非常有限，会极大地影响联邦学习的整体性能，导致模型收敛速率缓慢。

1.2　联邦学习中的模型剪枝方法

深度神经网络通常具有非常高的计算复杂度和庞大的参数数量，这使在资源受限的边缘节点上训练和传输这类模型变得极为困难。为了应对这一挑战，研究者们提出了模型剪枝方法，目前主要有

2 种剪枝方法：非结构化剪枝和结构化剪枝。Jiang 等[18]研究了非结构化剪枝在联邦学习中的应用，通过自适应的剪枝方法来调整模型大小，但这种方法显著增加了模型参数的稀疏度[19]，在压缩内存占用方面存在很大的困难，同时需要一些专门的硬件库来加速模型训练[20]。Diao 等[12]和 Wu 等[21]研究了结构化剪枝在联邦学习中的应用，该方法能够移除多余的模型结构而不会导致参数稀疏性的产生。此外，他们改变了传统工作中本地模型必须与全局模型共享相同体系结构的假设。然而，这些研究在执行全局模型的结构化剪枝过程中，未充分考虑到参数的重要性，导致剪枝后的子模型的准确率严重下降。

1.3 异构模型聚合

传统联邦学习假设客户端模型结构一致，而剪枝模型的引入会带来异构模型无法聚合的问题。为了应对这一挑战，Diao 等[12]首次将网络结构不同的本地子模型进行聚合，为不同计算能力的客户端分配对应复杂度的子模型，并采用分层聚合策略，有效地聚合了来自不同设备的异构模型。Jiang 等[22]提出了一种新的参数同步方案，该方案在模型聚合之前将结构不同的本地子模型恢复到同一尺度后，再对恢复后的模型进行联邦模型聚合。尽管上述方法均能处理异构模型的聚合问题，但这些方法没有深入探讨在 Non-IID 数据场景中存在的问题，尤其是异构模型之间学习能力的差异可能会直接影响联邦聚合模型的准确率。

通过对上述相关工作进行提炼与总结，本文提出了一种个性化剪枝算法，并设计了一种能够缓解 Non-IID 影响的异构模型聚合算法。基于以上算法，本文构建了一个高效的联邦学习框架 FedTP。

2 理论基础

本节将介绍本文所需要的理论基础：联邦学习、模型剪枝以及客户端的时间开销定义。表 1 总结了本文使用的关键参数。

2.1 联邦学习

本文所提出的联邦学习框架主要由服务器端和本地客户端（简称客户端）组成。服务器端通过聚合来自客户端的模型参数更新全局模型，本文将第 t 次通信中的全局模型定义为 $\boldsymbol{w}_t^{\mathrm{g}}$，假设有 K 个客户端参与联邦学习，索引号定义为 $K=\{1,2,\cdots,k\}$，则将第 k 个本地模型定义为 $\boldsymbol{w}_t^k$，本地训练集定义为 D_k，更新式如下

$$\boldsymbol{w}_{t+1}^k \leftarrow \mathrm{ClientUpdate}(D_k,\boldsymbol{w}_t^k) \tag{1}$$

表 1 本文使用的关键参数

参数	含义
K	参与联邦学习的客户端总数
R	通信回合的数目
D	本地数据集
D_k	第 k 个客户端的训练集（简称本地训练集）
$\boldsymbol{w}_t^{\mathrm{g}}$	第 t 回合的全局模型参数
$\boldsymbol{w}_t^k$	第 t 回合中第 k 个本地模型参数
$\boldsymbol{w}_t^{\mathrm{g},C}$	全局模型输出通道的参数集合
$\boldsymbol{w}_t^{\mathrm{g},c}$	全局模型中第 c 个通道的参数
α	客户端的剪枝率
C	全局模型总的通道数目
I	参数的损失近似

本地模型训练完成后将模型参数上传到服务器端，每个客户端的加权系数定义为：本地训练集的样本数量 $|D_k|$ 除以 K 个本地数据集的样本数量之和 $|D|$，加权平均式如下

$$\boldsymbol{w}_{t+1}^{\mathrm{g}}=\sum_{k=1}^{K}\frac{|D_k|}{|D|}\boldsymbol{w}_{t+1}^k \tag{2}$$

同时假设客户端都期望通过隐私保护本地数据 D_k 来获取未知样本的知识，以使本地模型 $\boldsymbol{w}_t^k$ 的性能得到提升。

2.2 模型剪枝

为了防止资源受限的客户端增加联邦学习的训练时间，服务器端将给资源受限程度不同的客户端确定不同的剪枝策略。由于本文假设客户端的可用资源在联邦学习过程中保持不变，因此，一旦为每个客户端设定了特定的剪枝策略，服务器便会在全局模型上执行剪枝操作，为各客户端生成相应的剪枝子模型。相较于传统的联邦学习，本文中不同客户端得到的模型结构会有所差异。此外，本文提出的剪枝策略是结构化剪枝，结构化剪枝可以使模型具有更规则的内存布局，从而更容易实现在资源受限的硬件上的加速[20]。

2.3 客户端的时间开销定义

在联邦学习中，客户端的资源受限通常被定义为有限的计算能力和通信带宽[23]，资源受限的客户端往往会产生额外的训练时间和通信量。本文将客户端的资源信息定义为计算能力和通信带宽，将

时间开销定义为T_k，具体又可以分为本地训练时的计算时间$T_{k,\mathrm{comp}}$和参数传输时间$T_{k,\mathrm{comm}}$，第k个客户端的时间开销为

$$T_k = T_{k,\mathrm{comp}} + T_{k,\mathrm{comm}} \tag{3}$$

同时，客户端之间的资源受限情况具有异构性，即不同的客户端训练和传输相同结构的模型的时间开销会有所不同。

3　方案设计

3.1　框架概述

在联邦学习中，全局模型更新需要经过多轮全局通信，每轮全局通信过程包含若干个关键阶段。本文提出的联邦学习框架FedTP如图1所示，全局通信包括以下4个阶段。

1) 初始化阶段

在联邦学习开始之前，服务器端先收集客户端CPU、GPU的处理频率和通信带宽，获取本轮全局通信客户端的资源信息。

2) 模型剪枝阶段

不合适的剪枝策略可能导致模型准确率降低或计算及通信开销增加，本框架采用个性化剪枝算法为资源受限客户端分配剪枝策略，以平衡模型准确率与复杂度。服务器端给资源受限的客户端分配剪枝策略将全局模型$\boldsymbol{w}_t^g$修剪为子模型$\boldsymbol{w}_t^k$，然后将子模型发送给资源受限客户端进行本地训练。同时服务器端给计算能力和通信带宽充足的客户端下发全局模型$\boldsymbol{w}_t^g$进行本地训练。

3) 本地训练阶段

在本地训练中，客户端的本地模型在本地训练集上采用反向传播算法结合随机梯度下降优化算法对模型进行更新，然后将更新的本地模型上传至服务器端。

4) 模型聚合阶段

服务器端接收到所有客户端的本地模型后进行联邦聚合。该阶段利用本文提出的异构模型聚合算法，其不仅能够聚合网络结构不同的子模型，而且能够利用相似度加权系数对通道进行加权平均，以缓解异构联邦聚合受到Non-IID的影响。

在上述4个阶段中，初始阶段仅在首轮通信中执行一次，从第二轮开始FedTP按照模型剪枝、本地训练以及模型聚合的步骤循环，直到全局模型收敛。资源受限的客户端只对剪枝后的子模型进行训练和传输，因此，与传统联邦学习相比，本文提出的方法能够显著地减少设备资源受限带来的影响。

3.2　剪枝决策

FedTP的模型剪枝阶段提出了一种个性化剪枝算法，该算法降低卷积神经网络（CNN, convolu-

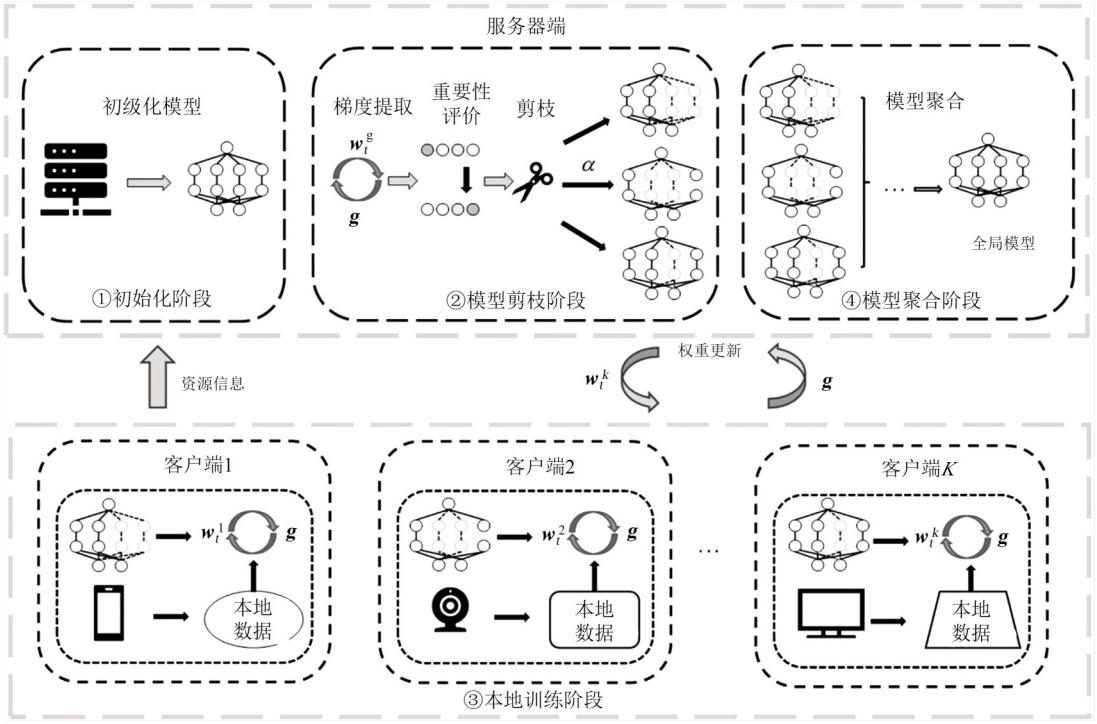


图1　联邦学习框架FedTP

tional neural network）的计算复杂度和去除冗余信息，在保证本地模型准确率的同时，减少客户端的训练时间。该算法对CNN进行剪枝，为了明确哪些通道应被剪除，下面将对通道的重要性进行定义。

1) 通道重要性的定义

通道指的是CNN中一个卷积层的输出通道，输出通道的数量决定了卷积操作后生成的特征图的数量。通过定义模型通道的重要性可以反映出哪些通道是冗余的，并据此进行剪枝，从而生成准确率更高的子模型。为了对卷积通道的重要性进行全局评价，本文采用Molchanov等[24]提出的重要性评价方法，即利用梯度和权重的乘积来近似卷积通道对损失的影响，重要性评价表示为

$$I_c(\boldsymbol{w}) = \sum_{s \in c} (\boldsymbol{g}_s \boldsymbol{w}_s)^2 \tag{4}$$

其中，$I_c(\boldsymbol{w})$表示对模型参数的第c个通道的重要性评价，$\boldsymbol{g}_s$表示通道内第s个神经元的梯度，$\boldsymbol{w}_s$表示第s个神经元的权重值，$I_c(\boldsymbol{w})$的值越小代表该通道对模型的输出和损失函数的影响越小。为了修剪对模型影响最小的部分通道，本文对CNN的C个通道的重要性评价按照从高到低的顺序进行排序，得到排序集合$\boldsymbol{I}(\boldsymbol{w})$为

$$\boldsymbol{I}(\boldsymbol{w}) = \text{sort}(I_1(\boldsymbol{w}), I_2(\boldsymbol{w}), \cdots, I_C(\boldsymbol{w})) \tag{5}$$

为了选出综合效率较好的排序算法，本文通过分析式(6)来进行衡量。

$$F = aT + bS \tag{6}$$

其中，T和S分别为算法的时间复杂度和空间复杂度，a和b为对应的权重。本文应用场景中，因为服务器的排序时间会影响整体联邦学习效率，而服务器通常具备足够的内存资源来处理空间需求，所以本文设置a=0.8和b=0.2。假设需要进行排序的卷积通道的数量是100，不同排序算法的综合效率如表2所示。从表2可以看出，堆排序具有最低的F值，代表其在本文场景下综合效率较好，因此本文选择堆排序作为重要性排序的排序算法。

表2 不同排序算法的综合效率

排序算法	时间复杂度	空间复杂度	F值
堆排序	$O(n\log n)$	$O(1)$	368.61
快速排序	$O(n\log n)$	$O(\log n)$	369.33
归并排序	$O(n\log n)$	$O(n)$	388.41
冒泡排序	$O(n^2)$	$O(1)$	8 000.2

2) 剪枝率的决策

为了从排序集合中确定可以被剪枝的通道数量，本节提出了一种剪枝评分准则，该准则能够通过重要性评价与客户端的资源信息来平衡准确率和模型复杂度。在本文中假设第k个客户端的资源信息为f_k，$f_{\max}$被定义为资源充足的客户端的资源信息，当$f_k < f_{\max}$时需要进行剪枝。评分准则下

$$R_k(\alpha_i) = \frac{\sum(I(\boldsymbol{w}, \alpha_i))}{\left| T_k(\alpha_i) - T_{\min}(\alpha_i) \right|} \tag{7}$$

其中，α_i是集合$\boldsymbol{P}$内第i个剪枝率，$R_k(\alpha_i)$是第k个客户端剪枝率为α_i时的评分；分子表示当剪枝率为α_i时子模型的重要性之和，$T_k(\alpha_i)$表示当剪枝率为α_i时第k个客户端的理论时间，$T_{\min}(\alpha_i)$表示当剪枝率为α_i时资源信息为$f_{\max}$的理论时间。同时，本文引入一个超参数$\theta(0 \leqslant \theta < 1)$，将区间$[0,1)$划分成大小为$\frac{1}{\theta}$的剪枝率集合$\boldsymbol{P}$。

第k个客户端的理论时间为

$$T_k(\alpha_i) = \frac{\text{param}(\alpha_i)}{f_k^{\text{comp}}} + \frac{\text{param}(\alpha_i)}{f_k^{\text{comm}}} \tag{8}$$

其中，$\text{param}(\alpha_i)$表示剪枝率为α_i时子模型的参数数量，f_k^{comp}表示第k个客户端硬件的浮点计算速度，f_k^{comm}表示第k个客户端的通信带宽。对于第k个客户端，在剪枝率集合P中选取$R_k(\alpha_i)$值最大时的α_i作为第k个客户端的剪枝率。个性化剪枝算法如算法1所示。

算法1 个性化剪枝算法

初始化 迭代次数t，客户端数量$K = 5$

输入 剪枝率集合$\boldsymbol{P}$，资源信息f

输出 剪枝率集合$\boldsymbol{P}_k$

1) for each client $k \in \{1, 2, \cdots, K\}$ do

2) for each Pruning Ratio $\alpha \in \boldsymbol{P}$

3) 通过式(5)进行重要性排序

4) 通过式(6)和式(7)评分

5) 选择剪枝率$\alpha = \arg\max(R(\alpha))$

6) end for

7) end for

重要性评价在剪枝前需要对模型的梯度进行提取，导致全局模型聚合后的参数发生改变。为了保证模型参数在评价前后保持不变，本文在重要性评价之前对全局参数进行复制，通道的重要性评价将在复制模型上进行，服务器端获取需要被剪枝的通

道索引号后在全局模型上进行剪枝，以保证全局模型在重要性评价后从各个客户端上获取到的知识不被破坏，并且全局模型的准确率不会因为重要性评价而受到影响。剪枝过程如图2所示。

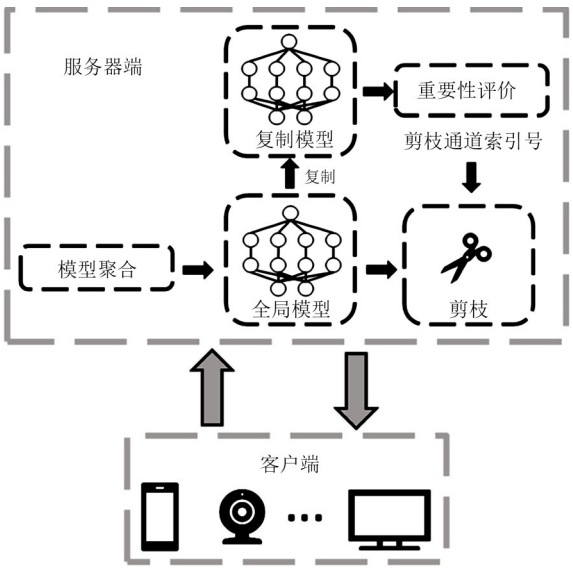


图2　剪枝过程

3.3　异构模型聚合

剪枝后的子模型被分发至客户端进行本地训练。完成训练后，客户端会将模型参数回传给服务器进行聚合。为了实现结构不同的子模型参与联邦聚合，本文提出了一种异构模型聚合算法。在聚合过程中引入了一种基于相似度的加权系数算法，解决了Non-IID数据背景下异构模型之间学习能力的差异可能导致全局聚合模型准确率降低的问题。

为确保服务器在计算相似度时能准确识别客户端子模型中保留的通道信息，本文引入了一种机制用于记录每个本地子模型的通道索引。通过使用二进制向量来标记通道索引，若全局模型的第c个通道存在于本地子模型中，则标记为1；否则标记为0。鉴于服务器通常具备充足的存储空间和强大的计算能力，这种二进制的存储与计算方法所需的存储空间和计算资源消耗是可以忽略不计的。

1) 模型恢复策略

由于客户端上传到服务器端的子模型和全局模型的结构不同，本文采用了一种模型恢复策略以保证聚合的顺利进行。在参与联邦聚合前，将剪枝后的客户端本地子模型恢复到和全局模型网络结构一致，再与其余上传到服务器端的客户端模型进行加权平均。

模型恢复过程如图3所示。首先，初始化一个恢复模型，其模型结构和全局模型一致，但参数全部为0。其次，由于服务器端保存了客户端子模型$\boldsymbol{w}_t^k$的通道索引号信息，因此当本地模型上传到服务器端时，可以按照未被剪枝的索引号将本地子模型的参数复制到恢复模型的相应通道上。最后，根据已被剪枝的通道索引号，将全局模型对应通道的参数复制到恢复模型上。值得注意的是，每轮恢复阶段将采用全局测试准确率最高的全局模型参数对缺失的通道参数进行恢复，而不一定是当前通信轮次聚合完成的全局模型参数。

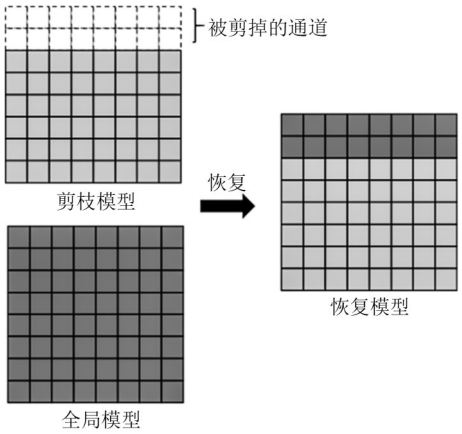


图3　模型恢复过程

2) 相似度加权系数算法

在子模型完成恢复之后，为了缓解Non-IID下各个子模型学习能力的差异对全局模型更新带来的影响，服务器端需要控制本地模型每个通道的加权系数来控制其对联邦聚合的贡献。本文假设参与聚合的客户端数目为K，则K个客户端的本地模型的第c个通道的参数分别表示为$\boldsymbol{w}_t^{1,c},\boldsymbol{w}_t^{2,c},\cdots,\boldsymbol{w}_t^{k,c}$。需要计算$K$个客户端在该通道的算术平均向量为

$$\boldsymbol{w}_t^c = \frac{1}{K}\sum_{k}^{K}\boldsymbol{w}_t^{k,c} \tag{9}$$

K个参数分别与$\boldsymbol{w}_t^c$计算余弦相似度得到相似度集合为

$$e_t^c = \{\, e_{t,k}^c | e_{t,k}^c = \frac{\boldsymbol{w}_t^{k,c}\boldsymbol{w}_t^c}{\left\|\boldsymbol{w}_t^{k,c}\right\|\left\|\boldsymbol{w}_t^c\right\|}, k = 1,2\cdots,K \,\} \tag{10}$$

其中，$e_{t,\,k}^c$代表第k个客户端在通道c上与算术平均向量$\boldsymbol{w}_t^c$的余弦相似度。通道相似度计算过程如图4所示。

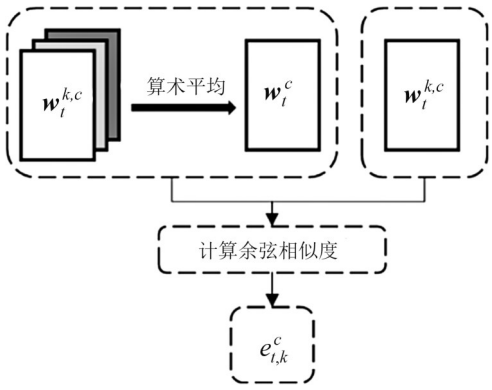


图4　通道相似度计算过程

在每一轮交互都会计算相似度集合e_t^c。全局更新式如下

$$\boldsymbol{w}_t^{g,c}=\frac{|D_n|}{|D|}\frac{e_{t,k}^c}{\sum_i^n e_{t,k}^c}\boldsymbol{w}_t^{k,c} \tag{11}$$

其中，$\boldsymbol{w}_t^{g,c}$表示在t轮联邦聚合后全局模型的第c个通道的参数，$\frac{|D_n|}{|D|}$表示样本数量的加权系数，$\frac{e_{t,\,k}^c}{\sum_i^n e_{t,\,k}^c}$表示第$k$个客户端第$c$个通道的相似度加权系数。式(11)通过相似度加权系数来调整通道的权重，通道参数的相似度加权系数越大，说明该通道参数在训练后产生的偏移越小，在聚合过程中具有越高的贡献度，而相似度加权系数较低的通道的贡献度被减弱。这样可以平衡不同通道之间的贡献度，避免降低聚合后的全局模型的准确率。

4　实验分析

4.1　实验设置

1) 环境设置

本文实验将CNN作为联邦学习的全局模型。CNN模型参数如表3所示，一共包含2个卷积层、2个BatchNorm层和2个线性层。CNN模型和联邦学习框架基于PyTorch实现。实验环境配置为Windows 11操作系统，搭配NVIDIA RTX 3070 Ti GPU进行计算。

为了反映客户端异构的计算能力，本文设置了3种计算模式，其参数如表4所示。从计算模式1到计算模式3，客户端的计算能力逐渐下降，其中，计算模式1配备了CPU和GPU双重计算资源，而计算模式2与计算模式3则仅依赖CPU进行计算，“×8”表示该CPU配置为8核16线程。此外客户端的通信能力在实际应用场景中也可能不同，本文通过设置设备的网络传输速率的差异来模拟客户端通信能力的异构性。具体地，设置计算模式1的传输速率为250 kbit/s，计算模式2和计算模式3传输速率分别为100 kbit/s和50 kbit/s。在实验中，总共有5个客户端参与联邦学习，其中，3个客户端设置为计算模式1，以模拟资源充足的环境；2个客户端分别设置为计算模式2和计算模式3，旨在模拟资源受限的客户端情况。

表3　CNN模型参数

层级	层名称	输入尺寸	输出尺寸
1	Conv2d	(1,5,6)	(8,4,5)
2	BtachNorm	(8,4,5)	(8,4,5)
3	Conv2d	(8,4,5)	(16,4,5)
4	BtachNorm	(16,4,5)	(16,4,5)
5	Linear	320	64
6	ReLU	64	64
7	Linear	64	11

表4　3种计算模式的参数

计算模式	CPU/GHz	GPU/GHz	传输速率/(kbit·s^{-1})
计算模式1	2.80×8	1.80	250
计算模式2	1.57×8	—	100
计算模式3	1.89×8	—	50

2) 数据集描述

BoT-IoT数据集[25]是由UNSW Canberra的Cyber Range实验室设计和创建的，旨在模拟真实物联网环境中的攻击和正常行为，数据集包含了10种僵尸网络流量和1种正常流量。本实验在该数据集上进行了训练和测试，以证明本文提出方法的有效性。

在数据划分中，80%的数据作为本地数据集，20%的数据作为全局测试集。本地数据集进一步划分为本地训练集和本地测试集，且没有样本重叠。客户端之间的攻击类别互不相同，以模拟Non-IID场景。流量类别一共包含11类，用数字0~10表示，其中，数字6代表正常流量，其余10个数字代表僵尸网络流量，客户端的数据分布情况和计算模式如表5所示。为了精确评估不同计算模式下客户端时间开销的差异，本文设置每个客户端的本地数据集的样本数量为20 000条样本。值得注意的是，全局场景是在全局测试集上进行测试的，而本地场景是

在分发给每个客户端的测试集上进行测试的。后文实验中的客户端设置都与表5一致。

表5　客户端的数据分布情况和计算模式

客户端	正常:攻击	样本类别	计算模式
客户端1	9:1	3,4,6	1
客户端2	9:1	0,5,6	1
客户端3	13:1	1,6,8	1
客户端4	17:1	6,9,10	2
客户端5	10:1	2,5,6	3

3) 对比方法

为了验证本文提出的联邦学习框架的有效性，本文采用了以下2种方法作为对比方法。① FedAvg算法[12]，它通过上传本地训练的模型参数而非训练数据来减少网络传输数据量，从而节约了带宽和通信成本。通过加权平均的方式，FedAvg能够确保全局模型均衡地受益于各个客户端的贡献，进而提升全局模型的性能和泛化能力。② FedProx[26]，该方法引入本地迭代轮数因子来调整客户端本地训练的迭代次数，旨在解决资源受限的设备无法承担过多的计算任务的问题，此外，FedProx在模型更新的目标函数中引入了正则化项，有效地缓解Non-IID数据给本地模型更新带来的影响。

4) 相关参数

实验参数设置如表6所示。

表6　实验参数设置

参数名称	参数值
客户端数目	5
全局总通信轮数	150
客户端本地训练轮数	5
Batch Size	256
优化算法	SGD
优化器参数(学习率)	0.001

5) 安全性假设

本文实验分析重点研究和验证个性化剪枝算法及联邦学习框架的性能表现。因此，实验设计中默认服务器和客户端均为安全可信的。

6) 评估指标

为了衡量模型的入侵检测能力，本文定义入侵攻击检测准确率为

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{12}$$

其中，TP为真阳性样本数，TN为真阴性样本数，FP为假阳性样本数，FN为假阴性样本数。

4.2　个性化剪枝算法评估

1) 个性化剪枝算法的准确率评估

本节实验旨在评估所提出的个性化剪枝算法在FedTP中能够保证准确率的同时减少时间开销。首先，利用所有客户端的本地训练数据集训练一个集中式模型，然后在集中式模型上利用本文提出的个性化剪枝算法与基于1范数的剪枝策略和基于2范数的剪枝策略进行对比。本文对剪枝后的模型不进行微调，直接利用全局测试集对剪枝后的模型进行测试，这样可以直接观察不同剪枝策略对剪枝后模型准确率带来的影响。由于本文所提出的剪枝算法是基于梯度的提取，为了反映剪枝效果的严谨，实验将对每一目标剪枝率进行5次剪枝，并计算5次剪枝模型准确率的平均值。当剪枝率为0.1时意味着被剪通道的数量为10%。不同剪枝算法准确率对比如图5所示。

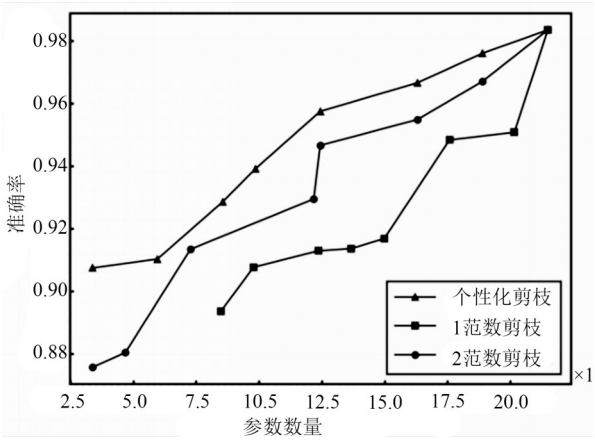


图5　不同剪枝算法准确率对比

从图5可以看出，随着模型参数减少，模型的准确率呈现下降趋势。相较于其他2种方法，本文的个性化剪枝算法准确率下降的速度相对平缓，这意味着本文的个性化剪枝算法保留的通道对于CNN来说更加重要。而1范数剪枝的准确率迅速下降，且保留的模型参数数量相较其余2种方法更多，说明重要性更高的通道被剪枝了。这是因为第一层卷积层每个通道的模型参数量小于第二层卷积层每个通道的模型参数量，这意味着1范数剪枝剪掉了更低层的卷积通道，然而在CNN当中，低层卷积网络往往提取的是更基础同时也更重要的特征，剪掉基

础特征会让CNN的性能损失得更加严重。相比之下，2范数剪枝的剪枝效果明显好于1范数剪枝。

为了验证资源受限客户端采用个性化剪枝算法对准确率和时间开销的影响，本文比较了应用剪枝算法的FedTP和未采用剪枝算法的FedTP-p的准确率和时间开销。本文采用全局测试集测试每个客户端的本地模型准确率。FedTP和FedTP-p的准确率对比如表7所示。

表7 FedTP和FedTP-p的准确率对比

客户端	FedTP	FedTP-p
客户端1	96.52%	96.59%
客户端2	97.16%	96.87%
客户端3	97.76%	97.81%
客户端4	96.77%	96.81%
客户端5	96.84%	96.79%

从表7可以看出，虽然客户端4、客户端5的模型经过了剪枝，但是准确率和未剪枝的准确率相当，这意味着本文提出的个性化剪枝算法使资源有限的客户端能够使用更简化的模型而不牺牲性能。不同方法进行150轮全局通信的客户端时间开销对比如图6所示。从图6可以看出，采用个性化剪枝策略后，资源受限客户端的时间开销相比于未进行剪枝处理的客户端有明显减少，其中客户端4实现了16.82%的加速效果，而客户端5实现了19.26%的加速效果。

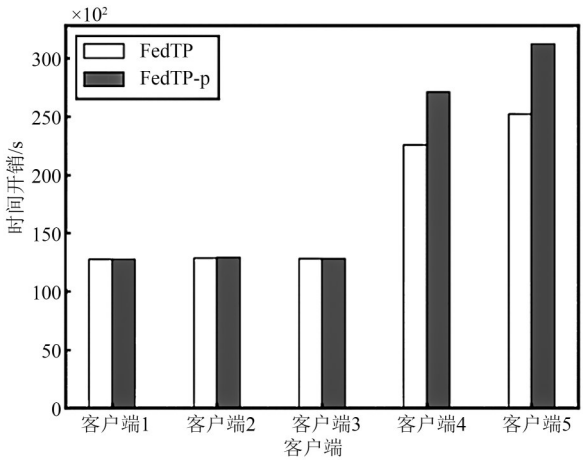


图6 不同方法进行150轮全局通信的客户端时间开销对比

2) 剪枝算法的内存开销评估

本节实验旨在评估剪枝过程中复制全局模型参数对服务器内存开销的影响。为了准确检测内存使用情况，本文从联邦学习开始以每0.1 s一次的频率监测服务器内存，通过对比有剪枝和未剪枝场景下的内存开销可以直观地了解剪枝算法给内存带来的额外开销，如图7所示。本节实验在持续150轮全局通信的条件下进行，旨在模拟长期联邦学习过程的内存开销的变化。

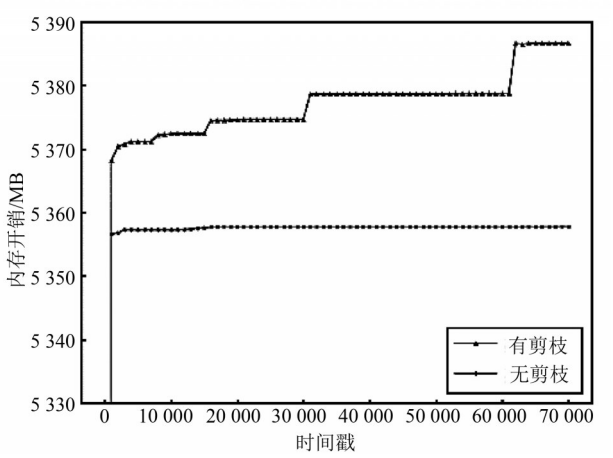


图7 有剪枝和无剪枝场景下的内存开销

从图7可以看出，有剪枝场景的联邦学习过程中，服务器的内存开销出现显著波动，最高达5 386 MB。相较之下，无剪枝场景的内存开销波动微小，主要维持在5 357 MB。由于有剪枝场景内存开销波动大，选用最大内存开销进行评估，旨在衡量极端条件下的资源需求。无剪枝场景下，稳定的内存开销反映了较小的性能影响。通过比较2种情况下的内存使用差异，增加的内存开销百分比为

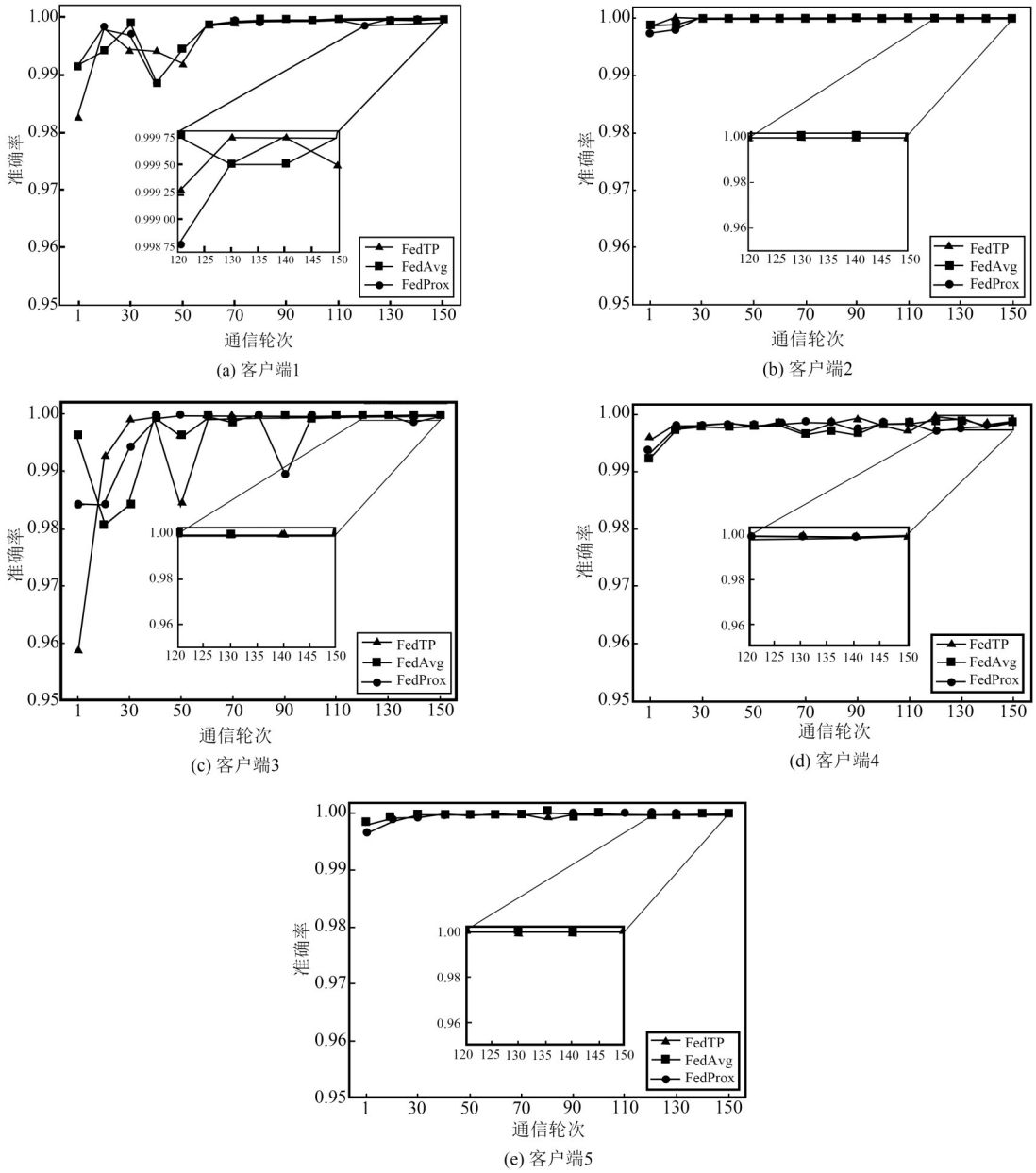
$$\Delta M = \frac{M_{\text{pruning}} - M_{\text{nopruning}}}{M_{\text{nopruning}}} \times 100\% \tag{13}$$

可以得出，剪枝算法增加的内存开销为0.54%。这表明虽然复制操作增加了额外的内存开销，但这种开销相对于服务器整体可用内存来说是非常有限的。

4.3 联邦学习框架的性能评估

1) 本地场景测试

本节实验旨在评估FedTP在本地场景下的测试能力，5个客户端的本地模型准确率如图8所示。因为本地测试集的攻击流量的类别和本地训练集攻击流量的类别一致，所以3种联邦学习框架的入侵检测准确率都有较高的水平。与其他2个方法相比，FedTP在客户端1、客户端3和客户端4都体现出更高的准确率。证明了FedTP框架的本地模型性能更加优异。



(a) 客户端1　(b) 客户端2

(c) 客户端3　(d) 客户端4

(e) 客户端5

图8　本地场景测试5个客户端的本地模型准确率

2) 全局场景测试

本节实验旨在评估FedTP在全局场景下的测试能力，先后在全局模型和本地模型上进行了全局场景测试。全局测试集一共有11类流量，相较于本地测试集，这增加了分类任务的难度。实验首先观察了每轮通信完成联邦聚合后的全局模型在全局测试集上的攻击检测准确率，如图9所示。

从图9可以看出，FedTP的最高准确率为98.19%，FedAvg的最高准确率为97.94%，FedProx的最高准确率为98.17%。由此可见，本文提出的方法的全局模型的性能高于对比方法。

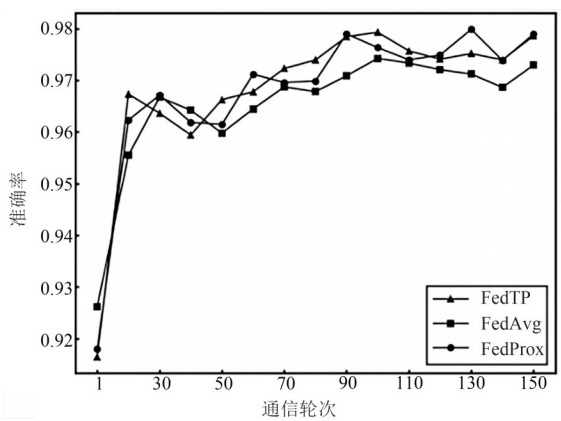
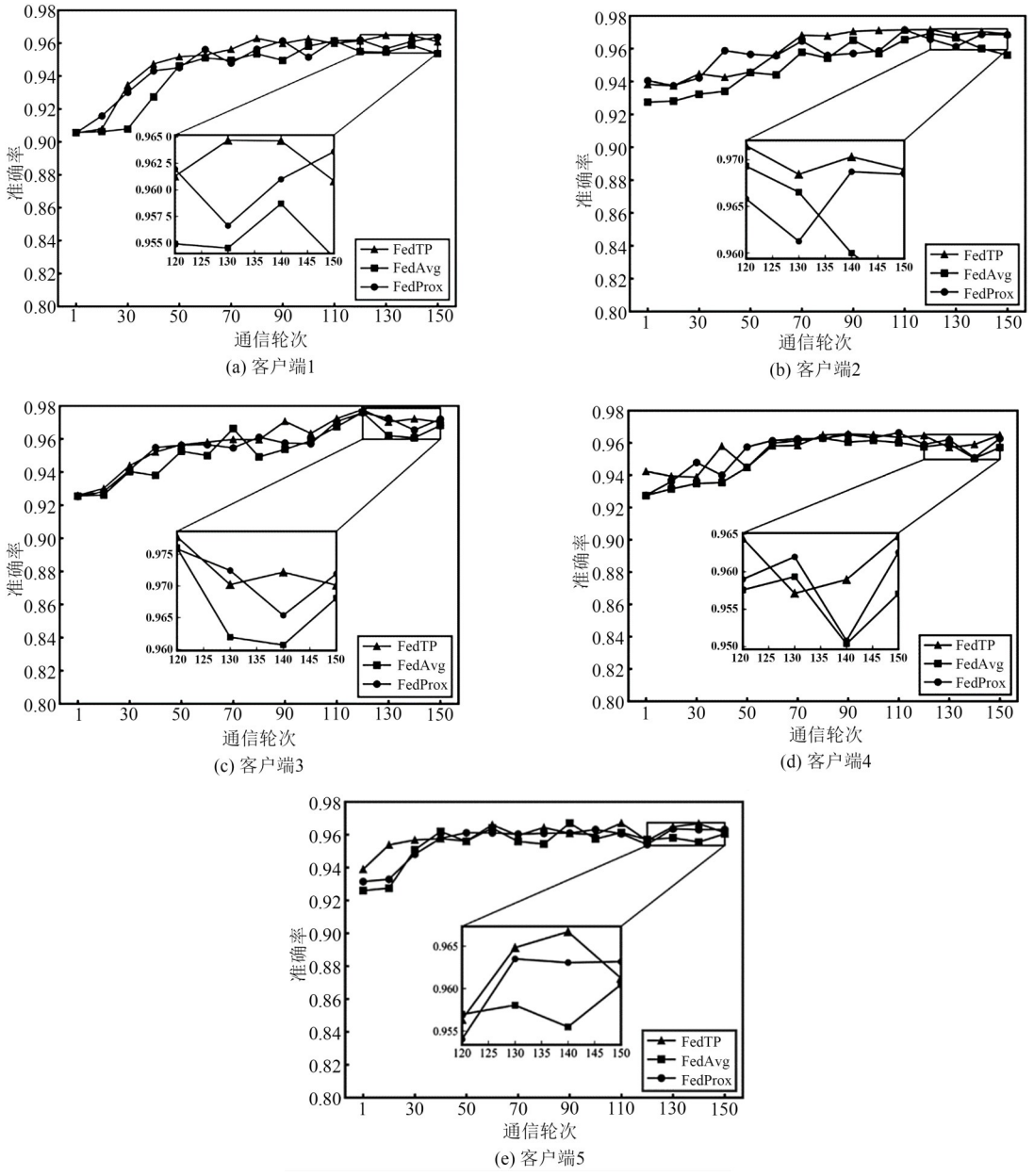


图9　全局模型在全局测试集上的攻击检测准确率

实验观察了 5 个客户端的本地模型在全局测试集上的性能。5 个客户端的本地模型准确率如图 10 所示。结果表明，在面对复杂的全局场景下，尽管 FedTP 有剪枝算法的参与，本地模型的全局测试相较对比方法具有更高的准确率。特别是，FedTP 的客户端展现出更高的平均准确率，客户端 1 的平均准确率为 95.02%，比 FedAvg 高 0.76%，客户端 2 的平均准确率为 95.95%，比 FedAvg 高 0.86%，客户端 3 的平均准确率为 95.88%，比 FedAvg 高 0.68%，客户端 4 的平均准确率为 95.62%，比 FedAvg 高 0.52%，客户端 5 的平均准确率为 95.93%，比 FedAvg 的平均准确率高 0.51%。值得注意的是，FedTP 在客户端 4 和客户端 5 上的优势小于其余 3 个客户端，这是因为在本文的实验设置中，客户端 4 和客户端 5 的本地模型是因资源受限而剪枝的子模型，导致学习能力略低于其余资源不受限的本地模型。

3) 时间开销测试

在验证了 FedTP 分别在本地场景和全局场景都具有优秀的检测效果后，本节实验进一步地对本地模型训练过程中产生的时间开销进行测试。具体地，实验记录了参与联邦学习的 5 个客户端在



(a) 客户端1　(b) 客户端2

(c) 客户端3　(d) 客户端4

(e) 客户端5

图 10　全局场景测试 5 个客户端的本地模型准确率

150 轮通信过程中本地训练和传输模型的时间开销。实验结果如图 11 所示，结果表明在资源受限的客户端 4 和客户端 5 上，FedTP 相较 FedAvg 能够分别减少 19.28% 和 20.82% 的时间开销，这得益于 FedTP 对资源不足的客户端的本地模型进行了剪枝，有效降低了本地训练时的计算复杂度。与 FedProx 相比，FedTP 在客户端 4 和客户端 5 上的时间开销分别少了 18.13% 和 12.11%，尽管 FedProx 对减少时间开销做出了一定的贡献，但因为 FedProx 在客户端本地训练时增加了约束项的计算，这增加了 FedProx 额外的本地运算时间。

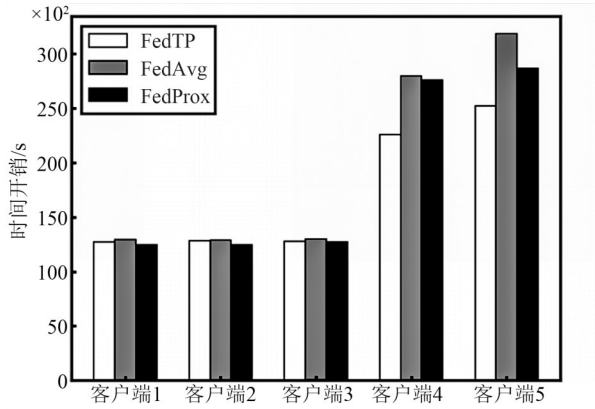


图 11　联邦学习 150 轮客户端时间开销

综上所述，本文提出的联邦学习框架 FedTP 在本地和复杂全局场景中都表现出了优越的性能，而且满足了资源受限的客户端使用联邦学习的需求，同时确保了较高的攻击检测准确率。

4.4　联邦学习框架的安全性评估

1) 恶意服务器对框架的安全威胁评估

为了全面评估服务器恶意剪枝行为对 FedTP 全局模型准确率的潜在影响，本节实验设置了恶意服务器对不同数量的客户端进行恶意剪枝的场景，具体而言，恶意服务器分别抽取比例为 20%、40%、60%、80% 以及 100% 的客户端进行恶意剪枝。不同客户端数量场景下，实验均设置进行 150 轮全局通信，恶意服务器在每一轮全局通信中，向受影响的客户端分发子模型时随机剪除 90% 的卷积通道。这种极端恶意剪枝行为会影响客户端本地模型的学习能力，进而影响全局模型的聚合。

实验结果如表 8 所示，随着受到恶意剪枝影响的客户端比例的增加，全局模型的准确率降低，受影响的客户端比例为 100% 时，全局模型准确率仅下降 5.04%。这表明了本文提出的联邦学习框架在面对恶意服务器的恶意剪枝行为时具有一定的鲁棒性。

表 8　　恶意服务器对 FedTP 的影响

受影响的客户端比例	全局模型准确率
0	98.19%
20%	97.67%
40%	96.96%
60%	96.26%
80%	93.66%
100%	93.15%

2) 恶意客户端对框架的安全威胁评估

为了深入探讨和评估恶意客户端对 FedTP 整体训练过程的潜在影响，本节设计并执行了 2 个含有恶意客户端的实验场景：数据中毒攻击实验和模型中毒攻击实验。在这 2 种攻击场景下，将恶意客户端比例分别设置为 20%、40%、60%，以评估不同比例的恶意客户端对全局模型准确率的影响。在数据中毒攻击实验中，恶意客户端采取了数据标签随机翻转的策略，以模拟对训练数据的直接干预。具体地，实验设置随机翻转率分别为 50%、75%、100%，代表 3 种攻击强度，以评估不同攻击强度对全局模型准确率的影响。数据中毒攻击实验全局模型准确率如表 9 所示。

表 9　　数据中毒攻击实验全局模型准确率

恶意客户端比例	翻转率为 50%	翻转率为 75%	翻转率为 100%
20%	98.04%	96.91%	96.50%
40%	97.20%	96.90%	96.34%
60%	93.16%	93.74%	92.64%

模型中毒攻击实验采用在本地训练完成的模型中添加均匀分布噪声的策略。噪声水平分别设定为 0.01、0.03、0.05，代表不同的攻击强度，以评估加入噪声影响模型的全局更新过程及全局模型的准确率，从而揭示模型中毒攻击在不同强度下的破坏性。模型中毒攻击实验全局模型准确率如表 10 所示。

从表 9 和表 10 可以看出，在数据中毒攻击实验中，即使在恶意客户端比例达到 60%，数据标签随机翻转率为 100% 的情况下，全局模型准确率也能

保持在92.64%以上。同样，模型中毒攻击实验中可以观察到类似的趋势，即在恶意客户端比例为60%，噪声强度为0.05的情况下，全局模型准确率依然可以达到94.44%。这进一步表明，即使面对较强的恶意客户端攻击，全局模型的性能仅轻度下降。本文采用的基于相似度加权的聚合方法一定程度上削弱了来自恶意客户端的异常更新，从而保证了全局模型的鲁棒性。

表10 模型中毒攻击实验全局模型准确率

恶意客户端比例	噪声强度为0.01	噪声强度为0.03	噪声强度为0.05
20%	96.40%	96.30%	95.92%
40%	96.16%	95.84%	95.25%
60%	96.04%	95.58%	94.44%

5 结束语

考虑到物联网场景中客户端面临着数据Non-IID、设备资源受限的双重挑战，本文提出了一种面向异构环境的物联网入侵检测方法。为了减少资源受限的客户端的训练时间，本文提出了一种个性化剪枝算法，该算法能够平衡客户端的模型准确率和模型复杂度，为资源受限的客户端分配一个子模型参与联邦学习。进一步地，针对Non-IID环境下异构子模型学习能力的差异性导致全局聚合模型准确率降低的问题，本文在异构的子模型聚合过程中提出了相似度加权系数算法。通过实验验证，本文提出的FedTP框架在本地场景和全局场景下相较于对比方法具有更高入侵检测准确率，同时有更少的时间开销。然而，基于相似度加权的方法对于缓解异构模型下的Non-IID问题是有限的，未来的研究将进一步探索针对该问题的解决方案。

参考文献：

[1] GUNTURU V, BANSAL V, SATHE M, et al. Wireless communications implementation using blockchain as well as distributed type of IoT[C]// Proceedings of the 2023 International Conference on Artificial Intelligence and Smart Communication (AISC). Piscataway: IEEE Press, 2023: 979-982.

[2] 赵羽, 杨洁, 刘淼, 等. 面向视频监控基于联邦学习的智能边缘计算技术[J]. 通信学报, 2020, 41(10): 109-115.
ZHAO Y, YANG J, LIU M, et al. Federated learning based intelligent edge computing technique for video surveillance[J]. Journal on Communications, 2020, 41(10): 109-115.

[3] 孙佳佳, 李承礼, 常德显, 等. 基于生成对抗网络的入侵检测类别不平衡问题数据增强方法[J]. 科学技术与工程, 2022, 22(18): 7965-7971.
SUN J J, LI C L, CHANG D X, et al. Data augmentation method for intrusion detection imbalance problem using generative adversarial networks[J]. Science Technology and Engineering, 2022, 22(18): 7965-7971.

[4] MANGALA N, VENUGOPAL K R. Short paper: current challenges in IoT cloud smart applications[C]// Proceedings of the 2021 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM). Piscataway: IEEE Press, 2021: 36-40.

[5] SAHARKHIZAN M, AZMOODEH A, DEHGHANTANHA A, et al. An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic[J]. IEEE Internet of Things Journal, 2020, 7 (9): 8852-8859.

[6] SAHU N K, MUKHERJEE I. Machine learning based anomaly detection for IoT Network: (anomaly detection in IoT network)[C]//Proceedings of the 2020 4th International Conference on Trends in Electronics and Informatics. Piscataway: IEEE Press, 2020: 787-794.

[7] 范伟, 彭诚, 朱大立, 等. 移动边缘计算网络下基于静态贝叶斯博弈的入侵响应策略研究[J]. 通信学报, 2023, 44(2): 70-81.
FAN W, PENG C, ZHU D L, et al. Research on intrusion response strategy based on static Bayesian game in mobile edge computing network[J]. Journal on Communications, 2023, 44(2): 70-81.

[8] NGUYEN T D, MARCHAL S, MIETTINEN M, et al. DIoT: a federated self-learning anomaly detection system for IoT[C]//Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). Piscataway: IEEE Press, 2019: 756-767.

[9] ZHAO Z J, LAI Y X, WANG Y P, et al. A few-shot learning based approach to IoT traffic classification[J]. IEEE Communications Letters, 2022, 26(3): 537-541.

[10] RAHMAN S A, TOUT H, TALHI C, et al. Internet of things intrusion detection: centralized, on-device, or federated learning?[J]. IEEE Network, 2020, 34(6): 310-317.

[11] IMTEAJ A, THAKKER U, WANG S Q, et al. A survey on federated learning for resource-constrained IoT devices[J]. IEEE Internet of Things Journal, 2022, 9(1): 1-24.

[12] DIAO E, DING J, TAROKH V. HeteroFL: computation and communication efficient federated learning for heterogeneous clients[J]. arXiv Preprint, arXiv: 2010.01264, 2020.

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

[14] AGRAWAL S, SARKAR S, AOUEDI O, et al. Federated Learning for intrusion detection system: concepts, challenges and future directions[J]. Computer Communications, 2022, 195(C): 346-361.

[15] TRUONG H T, TA B P, LE Q A, et al. Light-weight federated learning-based anomaly detection for time-series data in industrial control systems[J]. Computers in Industry, 2022, 140: 103692.

[16] CHEN Z, LV N, LIU P F, et al. Intrusion detection for wireless edge networks based on federated learning[J]. IEEE Access, 2020, 8:

217463-217472.

[17] WANG N, CHEN Y M, HU Y, et al. FeCo: boosting intrusion detection capability in IoT networks via contrastive learning[C]//Proceedings of the IEEE INFOCOM 2022 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2022: 1409-1418.

[18] JIANG Y A, WANG S Q, VALLS V, et al. Model pruning enables efficient federated learning on edge devices[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(12): 10374-10386.

[19] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural networks[C]//Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1. Massachusetts: MIT Press, 2015: 1135-1143.

[20] HAN S, LIU X Y, MAO H Z, et al. EIE: efficient inference engine on compressed deep neural network[C]//Proceedings of the 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA). Piscataway: IEEE Press, 2016: 243-254.

[21] WU X Y, YAO X, WANG C L. FedSCR: structure-based communication reduction for federated learning[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(7): 1565-1577.

[22] JIANG Z D, XU Y, XU H L, et al. Computation and communication efficient federated learning with adaptive model pruning[J]. IEEE Transactions on Mobile Computing, 2024, 23(3): 2003-2021.

[23] WANG S Q, TUOR T, SALONIDIS T, et al. Adaptive federated learning in resource constrained edge computing systems[J]. IEEE Journal on Selected Areas in Communications, 2019, 37(6): 1205-1221.

[24] MOLCHANOV P, MALLYA A, TYREE S, et al. Importance estimation for neural network pruning[C]//Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2019: 11256-11264.

[25] KORONIOTIS N, MOUSTAFA N, SITNIKOVA E, et al. Towards the development of realistic botnet dataset in the Internet of things for network forensic analytics: BoT-IoT dataset[J]. Future Generation Computer Systems, 2019, 100(C): 779-796.

[26] LI T, SAHU A K, ZAHEER M, et al. Federated optimization in heterogeneous networks[J]. Proceedings of Machine Learning and Systems, 2020, 2: 429-450.

[作者简介]

刘静（1978– ），女，北京人，博士，北京工业大学助理研究员，主要研究方向为工业互联网安全、可信计算等。

慕泽林（1997– ），男，重庆人，北京工业大学硕士生，主要研究方向为联邦学习等。

赖英旭（1973– ），女，辽宁抚顺人，博士，北京工业大学教授，主要研究方向为工业控制网络和软件定义网络安全等。