中北大学学报(自然科学版)
JOURNAL OF NORTH UNIVERSITY OF CHINA(NATURAL SCIENCE EDITION)
2026年 第47卷 第2期(总第226期)
Vol. 47 No. 2 2026(Sum No. 226)

文章编号：1673-3193(2026)02-0203-12 J. North Univ. China, Nat. Sci. Ed., 2026, 47(2): 203-214.

一种面向网络入侵检测的异构联邦学习双向知识蒸馏方法

王慧强[1]，马晓航[1*]，杨 林[2]

(1. 哈尔滨工程大学 计算机科学与技术学院，黑龙江 哈尔滨 150000；
2. 中国人民解放军军事科学院 系统工程研究院，北京 100000)

摘 要：针对网络入侵检测系统中联邦学习应用场景下模型结构异构与数据类别分布不均衡并存的问题，传统参数聚合或简单知识蒸馏方法难以兼顾全局模型性能与本地模型适应性，开展了异构联邦学习条件下的高效知识融合方法研究，提出一种面向网络入侵检测的联邦类别权重双向知识蒸馏方法FedCKD。在服务器端全局聚合阶段，引入类别级可学习权重矩阵，对不同客户端在各攻击类别上的预测结果进行加权蒸馏，实现异构模型输出的精细化融合；在客户端本地更新阶段，基于各类别$F1$分数构建注意力蒸馏机制，在全局模型指导下强化少数类与难分类别的特征学习能力，从而缓解了类别不均衡对模型性能的影响。该方法在CICIDS2017和CIDDS-001两个公开网络入侵检测数据集上的实验结果表明，与FedAvg、FedMD、FedHe和HFedCKD等方法相比，FedCKD在全局模型层面的宏平均$F1$分数分别提升约0.012～0.045，在客户端本地模型层面对少数类攻击的$F1$分数提升约0.030～0.086，同时保持了稳定的收敛特性与可接受的计算开销。研究结果表明，FedCKD方法能够在无需统一模型结构的条件下，实现异构联邦学习场景中对多类别攻击的有效建模，适用于存在模型异构和类别不均衡特征的网络入侵检测任务。

关键词：网络入侵检测系统；联邦学习；模型异构；知识蒸馏
中图分类号：TN915.08 **文献标识码：**A **doi:** 10.62756/jnuc.issn.1673-3193.2025.08.0009
引用格式：王慧强，马晓航，杨林. 一种面向网络入侵检测的异构联邦学习双向知识蒸馏方法[J]. 中北大学学报(自然科学版)，2026，47(2)：203-214.
WANG Huiqiang, MA Xiaohang, YANG Lin. A bidirectional knowledge distillation method for heterogeneous federated learning in network intrusion detection[J]. Journal of North University of China(Natural Science Edition), 2026, 47(2): 203-214.

A Bidirectional Knowledge Distillation Method for Heterogeneous Federated Learning in Network Intrusion Detection

WANG Huiqiang[1], MA Xiaohang[1*], YANG Lin[2]

(1. College of Computer Science, Harbin Engineering University, Harbin 150000, China;
2. Institute of Systems Engineering, Academy of Military Sciences of the People's Liberation Army, Beijing 100000, China)

Abstract: In federated learning-based network intrusion detection systems, model structural heterogeneity and class-imbalanced data distributions often coexist, making it difficult for conventional parameter aggregation or

收稿日期：2025-08-22
基金项目：国家自然科学基金资助项目(62272126)
作者简介：王慧强(1960—)，男，教授，博士，主要从事网络技术与信息安全、可信计算等方面的研究。
*通信作者：马晓航(1997—)，男，博士生，主要从事网络安全、联邦学习等方面的研究。E-mail：maxiaohang@hrbeu.edu.cn。

simple knowledge distillation methods to simultaneously achieve high global model performance and strong local adaptability. To address this issue, this study investigates efficient knowledge fusion under heterogeneous federated learning settings and proposes a federated class-wise bidirectional knowledge distillation method for network intrusion detection, termed FedCKD. During the global aggregation stage on the server, a class-wise learnable weight matrix is introduced to perform weighted distillation of client model predictions across different attack categories, enabling fine-grained fusion of heterogeneous model outputs. During the local update stage on the clients, an attention-based distillation mechanism constructed from class-wise $F1$-scores is employed to enhance feature learning for minority and hard-to-classify classes under the guidance of the global model, thereby alleviating the impact of class imbalance on model performance. Experimental results on the public CICIDS2017 and CIDDS-001 network intrusion detection datasets show that, compared with FedAvg, FedMD, FedHe, and HFedCKD, FedCKD improves the macro-average $F1$-score of the global model by approximately 0.012 to 0.045, and increases the $F1$-score of minority attack classes in local client models by approximately 0.030 to 0.086, while maintaining stable convergence behavior and acceptable computational overhead. The results indicate that FedCKD enables effective modeling of multi-class attacks in heterogeneous federated learning environments without requiring unified model architectures, and is applicable to network intrusion detection tasks characterized by model heterogeneity and class imbalance.

Key words: network intrusion detection system; federated learning; model heterogeneity; knowledge distillation

0 引 言

随着网络环境日益复杂，网络入侵检测系统(Network Intrusion Detection System，NIDS)作为关键防线，承担着识别DoS、DDoS等攻击行为的重要职责[1]。近年来，深度学习提升了NIDS的检测精度[2]，但其对集中化大规模数据的依赖，使其在数据隐私与合规要求下难以部署[3]。联邦学习(Federated Learning，FL)为此提供了解决路径[4]，通过“数据不出本地”的分布式训练机制，使多个机构能够在数据隔离的前提下实现协同建模，兼顾安全性与模型性能，如图1所示[5]。特别是在跨域、多源场景下，基于联邦学习的网络入侵检测(Federated Learning Network Intrusion Detection System，FL-NIDS)能实现多节点联合，提供更有效的检测模型。不过一般的联邦学习算法要求所有参与训练的模型统一架构，但在实际应用中，各参与方的设备性能、数据类型和业务需求往往存在显著差异，很难统一模型结构[6]，因而传统基于参数平均的联邦方法(如FedAvg)难以投入实际应用。

知识蒸馏是缓解联邦学习模型异构问题的重要途径，但现有框架在聚合阶段大多仅对各客户端的logits进行简单平均，未能充分考虑模型在不同类别上的识别差异。由于客户端数据可能存在类别不平衡现象[7]，尤其是在NIDS中，模型对少数类的识别能力往往不足，若在聚合时仍对所有输出一视同仁，容易放大弱模型的负面影响，导致全局模型在关键攻击类别上的检测性能下降。此外，现有方法在得到全局模型后通常直接替换本地模型，缺乏对结构差异与个性化适配的考虑。

针对上述问题，本文设计了一种基于类别权重知识蒸馏方法的联邦学习架构(Federated Class-Wise Knowledge Distillation，FedCKD)。在全局聚合阶段，FedCKD摒弃传统的logits简单平均策略，设计了一个可优化的类别权重矩阵，为各客户端模型在每一类别的输出分配独立权重，并在训练过程中动态调整，从而实现更精细的知识融合与全局性能提升。本地模型训练阶段，考虑了下发的全局模型需适配类别分布不均的本地数据。

如何在异构模型间实现有效聚合是联邦学习中的关键问题[8]，其中知识蒸馏被广泛认为是重要解决方案[9]。已有研究提出多种基于蒸馏的框架：如Li等[10]的FedMD，首次在异构场景下利用公共数据集对齐客户端logits，实现无需参数共享的协同学习；Qi等[11]的FedBKD引入双向蒸馏并结合生成模型，支持模型结构与数据类别的双重异构；Zheng等[12]通过数据无关蒸馏与逆概率加权实现全局对齐与个性化保持；Chan等[13]提出FedHe，在函数空间聚合logits以降低通信开销并

增强兼容性；Yao 等[14]和 Zhang 等[15]则分别通过全局知识蒸馏与原型特征建模实现了无需公共数据的知识融合。Wu 等[16]将知识蒸馏应用在本地客户端，提出了 FedKD 框架，引入 Mentor-Mentee 双模型结构，其中本地大模型负责训练，小模型蒸馏继承其知识，并用于参与联邦聚合。

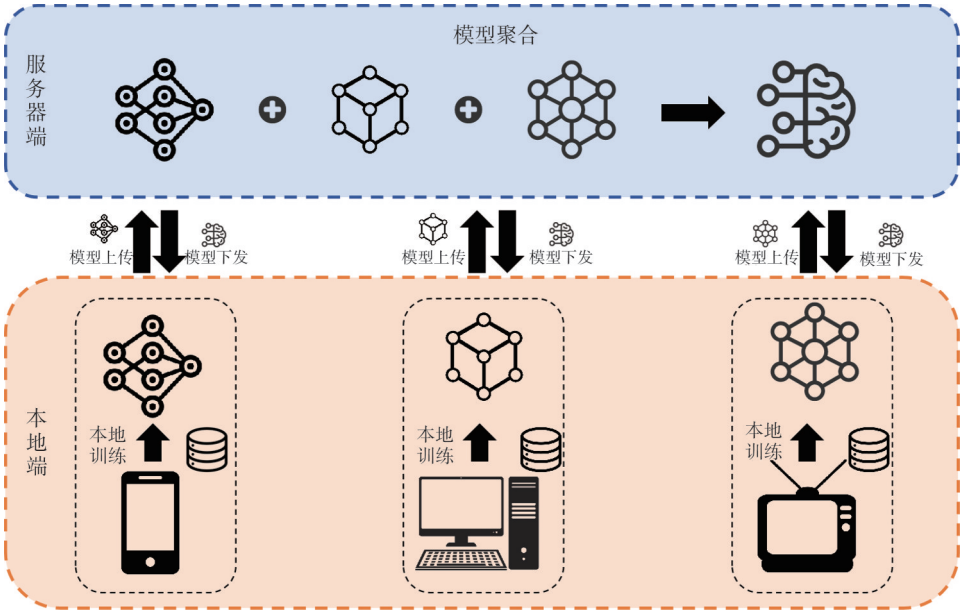


图 1　联邦学习架构图

Fig. 1　Federated learning architecture

尽管现有方法通过知识蒸馏有效缓解了联邦学习中的模型异构问题，但仍存在若干关键限制。首先，大多数方法在聚合阶段采用对各客户端模型输出 logits 的简单平均，未考虑各模型在不同类别上的识别能力差异。这种“平均处理”易受性能较差模型的影响，尤其当某些客户端在特定类别上的识别效果较弱时，可能显著拖累全局模型的整体性能。其次，一些方法未引入双向蒸馏机制，使得全局模型与本地模型在结构或表达能力上的不匹配问题难以缓解。即便如 FedBKD 等工作尝试通过双向蒸馏提升本地适应性，其蒸馏过程仍未充分考虑本地数据分布，尤其在网络入侵检测系统 NIDS 中，常见的类别不均衡问题可能严重影响蒸馏效果和模型收敛质量。

针对上述问题，本文提出了一种联邦类别权重知识蒸馏方法 FedCKD，在全局聚合阶段引入可优化的类别权重矩阵，自适应地为每个客户端在每个类别分配不同的融合权重，有效降低低性能模型在特定类别上的负面影响；在本地蒸馏阶段引入基于 $F1$ 分数的类别注意力机制，动态放大对少数类和弱识别类的梯度更新力度，从而提升模型在类别不均衡条件下的泛化能力，本文的主要贡献如下：

1）提出一种双向知识蒸馏的联邦学习框架，在不统一模型结构的前提下实现异构模型聚合，能够有效提升全局性能与本地适应性。

2）在全局聚合阶段提出类别权重机制，根据客户端在不同类别上的表现自适应分配权重，实现更精细的知识融合；在本地训练阶段引入基于 $F1$ 分数的注意力蒸馏，重点强化少数类学习，提升模型的个性化性能与整体鲁棒性。

3）在两个主流的网络入侵检测数据集 CICIDS2017 和 CIDDS-001 上进行了充分的实验，并与现有的 FedAvg，FedMD，FedHe 和 HFedCKD 进行了对比。实验结果表明，FedCKD 在全局和本地模型性能上均表现出显著的优势，验证了算法的有效性。

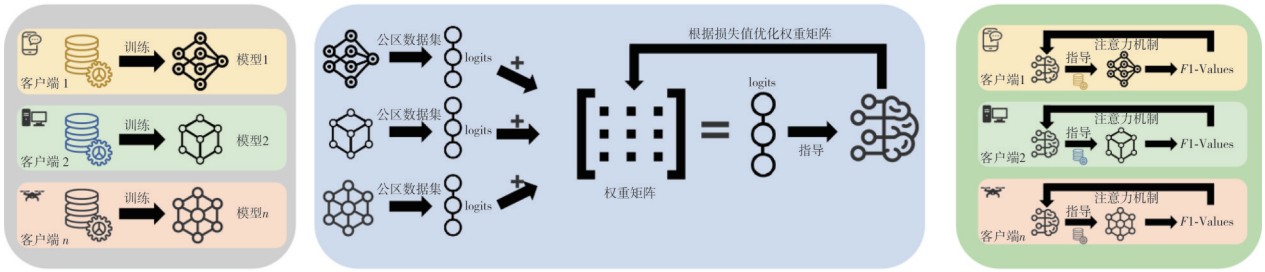
1　联邦类别权重知识蒸馏方法

1.1　方法的提出

1.1.1　FedCKD 算法流程

FedCKD 流程包括 3 个阶段，如图 2 所示。首先，初始阶段各客户端在本地数据上独立训练异构模型，仅上传模型参数以保障数据隐私；其次，在全局聚合阶段，服务器基于公共数据进行类别权重知识蒸馏，通过动态优化权重矩阵自适应融合各客户端在不同类别上的优势，得到高性能全局模型；最后，在本地更新阶段，客户端结合自身类别 $F1$ 分数引入注

意力蒸馏机制，对少数类和难分类别赋予更高权重，在全局模型指导下迭代优化本地模型。FedCKD实现了全局由客户端指导、本地由全局反向优化的双向蒸馏，在异构与非独立同分布场景下同时提升全局性能与本地适应性。



(a) 本地异构模型训练阶段　(b) 服务器端类别加权蒸馏聚合阶段　(c) 客户端注意力蒸馏更新阶段

图2　FedCKD流程图

Fig. 2　Workflow of FedCKD

1.1.2　双向蒸馏机制

1）知识蒸馏：在传统的知识蒸馏聚合中，服务器端的全局教师输出通常通过对各客户端logits取简单平均获得，这种方式假设所有客户端在所有类别上的识别能力相同，但面临数据类别不平衡分布的问题，客户端在特定类别上的预测方差差异显著直接平均会导致高方差类别的劣质预测被同等权重引入，进而增大全局模型的泛化误差。

2）类别加权机制：相较于传统的知识蒸馏聚合，类别加权机制设计的类别权重矩阵 $\boldsymbol{W}\in\mathbb{R}^{N\times C}$，相当于在每个类别上自适应选择最可靠的客户端预测，权重越大表示该客户端在该类别的预测可信度越高，从而减少低性能模型对全局模型的负面影响。

3）注意力机制：在本地蒸馏阶段，FedCKD根据各类别在训练过程中的$F1$分数动态分配注意力权重，$F1$值越低的类别获得越高的权重，从而在损失计算中放大其梯度贡献。

1.2　具体步骤

假定存在N个客户端和1个中心服务器参与联邦学习训练，每个客户端$i\in\{1,\cdots,N\}$拥有本地私有数据集D_i，客户端之间不可共享数据集。在中心服务器存在一个公共数据集D_{public}用于蒸馏和对齐知识表示。该算法的具体步骤如下：

1.2.1　本地客户端的私有模型训练

本阶段每个客户端通过拥有的本地数据集$D_i=\{(x_j^{(i)},y_j^{(i)})\}_{j=1}^{n_i}$训练其私有模型$f_i(\cdot;\theta_i)$，该训练方式为传统神经网络模型训练，目标是最小化标准的交叉熵损失函数，即

$$\min_{\theta_i}L_i^{\text{CE}}=\frac{1}{n_i}\sum_{j=1}^{n_i}L_{\text{CE}}\left(f_i(x_j^{(i)}),y_j^{(i)}\right),\tag{1}$$

式中：L_{CE}为交叉熵损失；f_i为客户端i的私有模型，其模型结构因客户端自身而异，拥有不同的隐藏层数或神经元数。

训练完成之后的模型会被上传到中心服务器中，此过程中只传递模型参数而不共享任何数据。

1.2.2　服务器端类别加权知识蒸馏聚合

中心服务器在收到各个客户端上传的模型f_i后，由所有的客户端模型担任教师模型，共同训练一个高性能的学生模型f_{g}（全局模型）。服务器将公共数据$D_{\text{public}}=\{x_k\}_{k=1}^{M}$输入每个客户端模型，得到模型关于数据集中$C$个类别logits输出

$$z_i(x_k)=f_i(x_k)\in\mathbb{R}^C。\tag{2}$$

一般的知识蒸馏算法是将得到的logits进行平均计算得到教师输出，而本文所提出的类别加权知识蒸馏在这一步引入一个可优化的类别加权矩阵$\boldsymbol{W}\in\mathbb{R}^{N\times C}$，其中第$i$行$W_i\in\mathbb{R}^C$表示客户端$i$对各类别的权重。对每个客户端有

$$\boldsymbol{w}_i=softmax(W_i)。\tag{3}$$

服务器对所有客户端logits按加权矩阵$\boldsymbol{W}$进行计算得到教师输出

$$z_{\text{agg}}(x_k)=\sum_{i=1}^{N}\boldsymbol{w}_i\odot z_i(x_k)。\tag{4}$$

然后对教师输出使用温度T生成softtarget，即

$$p^{\text{teacher}}(x_k)=softmax\left(\frac{z_{\text{agg}}(x_k)}{T}\right)。\tag{5}$$

接着，将该教师分布$p^{\text{teacher}}(x_k)$与f_{g}的输出进行对齐，模型f_{g}采用softtarget蒸馏损失作为优化目标

$$L_{\text{g}}(x_k)=T^2\cdot KL\left(softmax\left(\frac{f_{\text{g}}(x_k)}{T}\right)\Vert\, p^{\text{teacher}}(x_k)\right),\tag{6}$$

式中：T为温度系数；$KL(\cdot\,\Vert\,\cdot)$为Kullback-Leibler散度。由于教师输出z_{agg}为$\boldsymbol{W}$的函数，因此可通

过以下目标对 $\boldsymbol{W}$ 优化，即

$$\min_{\boldsymbol{W}} E_{x_k\sim D_{\text{public}}}[L_{\text{g}}(x_k; f_{\text{g}}, \boldsymbol{W})]。\tag{7}$$

通过不断的循环迭代优化类别加权矩阵 $\boldsymbol{W}$，提升全局模型 f_{g} 的性能。训练过程中，$\boldsymbol{W}$ 和 f_{g} 共享同一损失函数并通过梯度下降更新，训练完成的 f_{g} 将下发到本地客户端上。

1.2.3　本地客户端的注意力蒸馏

由于本地客户端的异构性，下发的全局模型并不能直接部署在本地客户端上使用，而同样是通过知识蒸馏的方法，将全局模型 f_{g} 作为教师模型，指导本地模型 $f_i(\cdot)$ 在本地数据集进行再训练。

考虑到本地数据集中可能存在类别不平衡问题，导致原始本地模型在稀缺类上的识别性能较差，本文在训练过程中引入基于 $F1$ 分数的注意力机制。在每个本地 epoch 结束后，统计各类别的 $F1$ 值并据此更新注意力权重向量 $w_{i\_\text{local}}^{(c)}$，其中，

$$F1=\frac{2\cdot Precision\cdot Recall}{Precision+Recall},\tag{8}$$

$$w_{i\_\text{local}}^{(c)}=\frac{\exp\left(\dfrac{1}{F1_i^{(c)}+\dot{o}}\right)}{\displaystyle\sum_{k=1}^{C}\exp\left(\dfrac{1}{F1_i^{(k)}+\dot{o}}\right)},\tag{9}$$

式中：$\dot{o}$ 是防止除零的微小常数。该权重设计使得 $F1$ 值越低获得越大的关注度，虽然在训练前期可能会出现轻微的震荡，但并不影响最后的收敛。

因此，本地模型在训练过程中结合两个目标：融合了注意力机制的蒸馏损失和模型的交叉熵损失。蒸馏损失 $L^{(i)}$ 通过软标签学习全局模型 f_{g} 的知识，交叉熵损失用于保持对真实标签的监督，分别为

$$L_{\text{KD}}^{(i)}=T^2\cdot E_{(x,y)\sim D_i}\cdot\left[w_{i\_\text{local}}^{(y)}\cdot D_{\text{KL}}\left(softmax\left(\frac{f_i(x)}{T}\right)\middle\|\, softmax\left(\frac{f_{\text{g}}(x)}{T}\right)\right)\right],\tag{10}$$

$$L_{\text{CE}}^{(i)}=E_{(x,y)\sim D_i}[L_{\text{CE}}(f_i(x),y)]。\tag{11}$$

因此，最终损失函数可表示为

$$L^{(i)}=\lambda\cdot L_{\text{KD}}^{(i)}+(1-\lambda)\cdot L_{\text{CE}}^{(i)}。\tag{12}$$

完成训练之后的本地模型将重新上传，继续参与全局模型的聚合训练，直到满足要求或达到指定轮次为止。

1.3　算法分析

1.3.1　性能分析

现有的双向蒸馏方法（如 FedBKD）虽然实现了全局与本地模型的知识交互，但仍存在两方面不足：1）在聚合阶段对客户端预测结果一视同仁，弱模型在关键类别上的劣质输出被等权引入，往往会削弱全局模型的性能；2）在本地蒸馏过程中未考虑类别分布不均衡，当少数类样本稀缺时，模型学习不足，检测效果明显下降。

针对上述问题，FedCKD 在全局端引入类别加权机制，能够根据各客户端在不同类别上的表现自适应分配权重，从而突出优势客户端预测，避免弱模型拖累整体性能；当客户端性能差异较小时，该机制又可自然退化为均值聚合，保证不低于传统方法。在本地端，FedCKD 结合基于 $F1$ 分数的注意力蒸馏，对少数类和难分类别赋予更高关注度，从而缓解类别不均衡带来的性能偏置，提升模型在稀缺类上的检测能力。

综合来看，FedCKD 在保留双向蒸馏框架优点的同时，通过类别加权与注意力机制实现了更精细的知识融合，兼顾了全局性能与本地适应性。在异构与不均衡场景下，FedCKD 相比现有方法展现出更优的鲁棒性与泛化能力，验证了其在实际网络入侵检测中的应用潜力。

1.3.2　计算复杂度分析

FedCKD 的计算开销主要由服务器端蒸馏聚合和客户端本地蒸馏训练构成。设公共数据规模为 n_{p}，第 k 个客户端的本地数据集规模为 n_k，服务器端与客户端的迭代轮数分别为 E_{g} 和 E_{c}，则单轮时间复杂度为 $O(E_{\text{g}}n_{\text{p}}+E_{\text{c}}max_{1\leqslant k\leqslant K}n_k)$；若统计总计算量，则为 $O\left(E_{\text{g}}n_{\text{p}}+E_{\text{c}}\sum_{k=1}^{K}n_k\right)$，当 E_{g} 和 E_{c} 固定时，整体复杂度随数据规模近似线性增长，具有可接受的计算开销。

2　实验设计与分析

2.1　实验配置

2.1.1　实验环境

本实验在一台搭载 IntelCorei7-12700KF（3.60 GHz）处理器、16GB 内存及 RTX3070Ti 显卡的 Windows64 位操作系统主机上完成。所使用的编程环境为 Python3.12，深度学习框架采用 PyTorch2.5.1，支持 CUDA12.4。所有模型训练与评估均在本地 GPU 加速下完成。

2.1.2　实验数据集

本实验选取了两个常用的公开网络入侵检测

数据集，包括CICIDS2017和CIDDS-001，涵盖多种攻击类型与流量场景，便于全面评估模型性能。其中，CICIDS2017数据集由加拿大网络安全研究所提供，涵盖DDoS等多种典型攻击；CIDDS-001数据集由德国ERNW团队发布，攻击类型包括BruteForce、PingScan等。

实验开始前对原始数据集进行必要的预处理操作，以确保模型训练的有效性和稳定性。首先，对于CICIDS2017和CIDDS-001数据集中存在缺失值或异常值的样本进行了清洗和删除，以提升数据质量。其次，对原始数据中的非数值型特征(如协议类型、标志字段等)采用标签编码等方式进行量化处理，使其适用于神经网络模型的输入要求。

为了更真实地模拟联邦学习场景中各客户端数据分布不均的实际情况，对两个数据集进行了非独立同分布采样，对两个数据集分别构建了3个类别分布不均衡的本地数据子集，对应参与联邦学习的客户端A、客户端B和客户端C，此外，从原始数据中划分出一部分样本作为公共数据集，用于蒸馏过程中客户端与服务器之间的知识交互；同时构建了一个独立的测试集，用于评估全局模型的性能与泛化能力，具体情况如表1所示。

表1 实验数据集

Tab. 1 Experimental dataset

数据集	数据类型	客户端A	客户端B	客户端C	公共集	测试集
CICIDS2017	Benign	19 983	20 215	19 672	503	9 870
	DoS	103	20 139	20 047	517	9 649
	DDoS	19 854	20 157	115	496	10 201
	PortScan	20 308	97	19 987	483	9 956
	Normal	10 032	9 901	10 115	513	921
CIDDS-001	Portscan	9 756	10 087	306	499	1 078
	DoS	10 248	9 584	10 342	505	1 034
	Pingscan	6 090	312	6 090	516	967
	Bruteforce	299	4 992	4 992	497	1 152

在数据划分中，对CICIDS2017的客户端样本按类别均衡控制在约20 000条，少数类如DoS控制在100条左右，公共集约500条，测试集约10 000条；CIDDS-001中正常类设为10 000条，少数类如Pingscan和Bruteforce受限于原始数据，仅有6 090和4 992条，为避免类别间差距过大，其余类别亦统一为约10 000条。公共集与测试集分别控制在约500和5 000条。

2.2 实验设计

2.2.1 对比算法

在本实验中，为全面评估所提出的FedCKD算法在异构联邦学习场景下的性能，选取了FedAvg、FedMD、FedHe和HFedCKD等4种具有代表性的算法进行对比。

FedAvg是最经典的联邦学习方法，通过参数加权平均实现模型聚合。FedMD是联邦学习中应对模型异构问题的代表方法，基于知识蒸馏机制，允许客户端使用不同模型结构并在公共数据上进行输出对齐。FedHe通过在函数空间对客户端logits进行聚合，而HFedCKD算法则与本文算法有着一定的相似性。

实验将分别在服务器端对比全局模型性能，在客户端对比各算法部署后的本地模型精度，以全面验证FedCKD的有效性。需要指出的是，FedAvg将全局模型直接下发至客户端，本地模型性能与全局模型一致；而FedHe不生成显式的全局模型，因此仅在客户端层面进行性能对比。

2.2.2 模型结构设计

在模型架构设计方面采用了最经典的多层感知机架构，3种算法的架构分别如下：

FedAvg要求各客户端使用统一的模型结构，因此采用一个三层全连接神经网络。该模型包含两层隐藏层，神经元数分别为128和64，最终连接至输出层，适用于所有客户端进行同构参数聚合。

FedCKD与FedMD、FedHe，HFedCKD采用相同的架构。客户端A、B、C分别使用隐藏层结构(128，64)、(256，128，64)和(64)，全局模型隐藏层为(128，64)。不同的是，FedCKD在服务器上采用类别加权知识蒸馏聚合，在本地客户端上采用注意力蒸馏。

2.2.3 本地数据集测试

为验证客户端数据类别不均衡对本地模型性能的影响，在构建的3个客户端子集上分别训练了单个本地模型，并在各自的数据上进行测试。本实验不涉及联邦聚合，仅评估数据分布对模型

本地性能的直接影响。

选用一个简单的多层感知机MLP作为测试模型，模型结构包括两层隐藏层，神经元数分别为128和64，激活函数为ReLU，输出层连接至对应类别数的softmax分类器。每个客户端使用自身数据独立训练该模型，并在本地验证集上测试其性能，具体的每类的精确率、召回率和$F1$值，以及客户端A、B、C的实验结果如表2和表3所示。

表2　CICIDS2017客户端模型精度

Tab. 2　CICIDS2017 client model performance

客户端	Benign			DoS			DDoS			PortScan		
	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$
ClientA	0.84	1	0.91	1	0.79	0.87	0.99	1	1	0.98	1	0.99
ClientB	0.91	0.99	0.96	0.99	1	1	1	1	1	0.94	0.85	0.89
ClientC	0.91	0.99	0.95	0.98	1	0.99	0.91	0.90	0.91	1	1	1

表3　CIDDS-001客户端模型精度

Tab. 3　CIDDS-001 Client model performance

客户端	Normal			Portscan			DoS			Pingscan			Bruteforce		
	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$
ClientA	0.9	1	0.94	0.71	0.96	0.82	1	1	1	0.93	0.98	0.95	1	0.48	0.65
ClientB	0.97	0.98	0.97	0.91	0.96	0.88	1	1	1	0.95	0.75	0.81	0.95	0.99	0.97
ClientC	0.94	0.98	0.96	1	0.62	0.77	1	1	1	0.95	0.98	0.96	0.85	0.95	0.89

根据之前的数据集信息，在CICIDS2017数据集中，客户端A的少数类是DoS，客户端B的少数类是PortScan，客户端C的少数类是DDoS。在CIDDS-001数据集中，客户端A的少数类是BruteForce，客户端B的少数类是Pingscan，客户端C的少数类是Portscan。

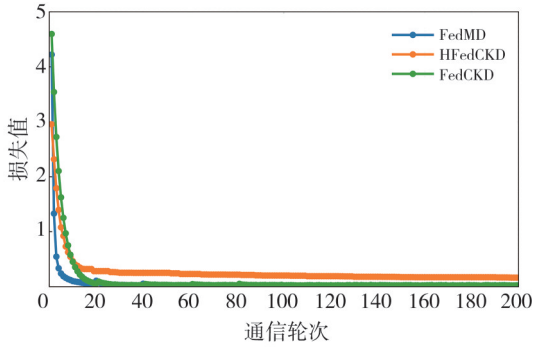
实验结果表明，在CICIDS2017数据集中，虽然各客户端模型的整体性能较好，精确率普遍高于0.9，但少数类的识别效果仍明显弱于多数类。类似地，在CIDDS-001数据集中，少数类在召回率和$F1$值上的表现也相对较差。总体而言，类别不均衡会削弱模型对少数类样本的识别能力。

2.3　实验对比与分析

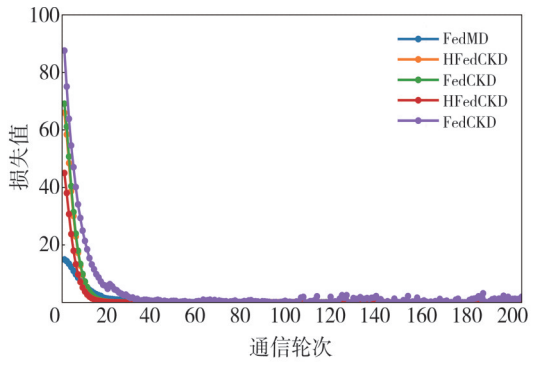
本实验中使用了两个公开的网络入侵检测数据集（CICIDS2017和CIDDS-001）对三种算法进行评估。通过测试集对模型的精度进行评估，具体包括对全局模型和本地客户端模型精度的对比。评估主要关注精确率、召回率和$F1$值，这些指标能够全面反映模型在测试集上的表现，特别是其在各类别间的平衡能力。设置服务器上的全局模型聚合和客户端上的本地模型聚合训练轮次均为20次，整个联邦学习流程迭代10次。

2.3.1　收敛性分析

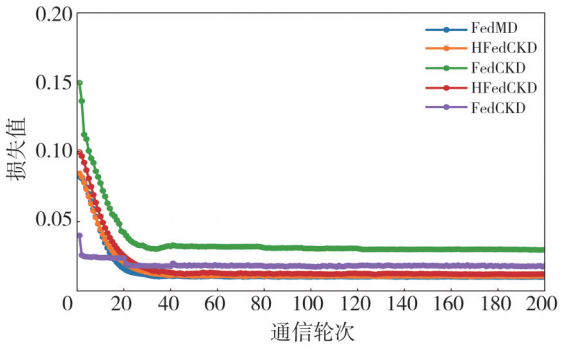
本文基于CICIDS2017数据集对各算法的收敛性进行了对比分析。考虑到CICIDS2017数据集特征维度高、分布复杂，能够更全面地反映算法在真实场景下的收敛特性，因此本节实验基于该数据集展开。由于FedAvg和FedHe不生成统一的全局模型，故在全局收敛上仅对比FedMD、HFedCKD与FedCKD的收敛曲线，如图3所示。



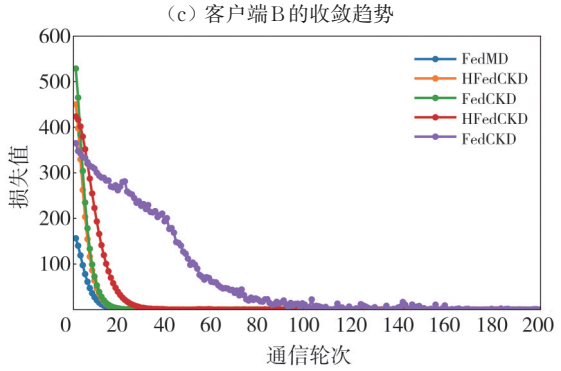
（a）服务器端的收敛趋势



（b）客户端A的收敛趋势



(c) 客户端B的收敛趋势



(d) 客户端C的收敛趋势

图3 FedCKD损失值的收敛趋势

Fig. 3 Convergence trends of loss values in FedCKD

由图3可知，FedCKD的整体收敛速度与其他算法基本保持一致，均能较快实现收敛，仅在客户端C上表现略逊。这主要源于其模型结构较为简单，仅有一层，特征提取能力有限，在早期训练阶段难以充分利用全局蒸馏信息。同时，FedCKD在初期通过$F1$注意力机制调节类别权重，可能导致一定波动，但该差异有限，并未影响其收敛稳定性与最终性能。

2.3.2 计算开销对比

为了评估不同算法在计算效率上的差异，本文对比了各方法在单轮训练中的浮点运算量(FLOPs)。实验设置为客户端模型与全局模型各进行20次迭代训练，并将一次客户端与服务器的交互定义为一个轮次。在该条件下，统计并对比了各算法在数据集CICIDS2017上单轮交互过程中的FLOPs开销，实验结果如图4所示。

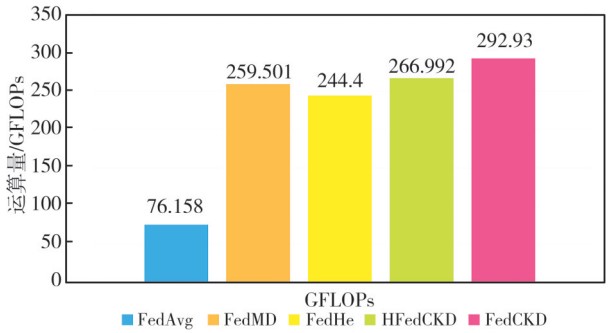


图4 浮点运算量统计

Fig. 4 FLOPs statistics

由图4可知，FedAvg由于只涉及参数聚合与本地训练，计算开销最小，其余算法都是因为涉及到对齐过程，FLOPs明显变高，其中FedHe因为不使用公共数据集且无全局模型训练，所以计算复杂度最小。相比之下，FedCKD的FLOPs最高，但仍然和其他几种处于相近水平，是在计算开销可接受的前提下换取了更优的全局与本地性能的提升。

2.3.3 消融实验

为了验证FedCKD中两个核心组件类别权重聚合与$F1$注意力蒸馏的独立贡献，本文设计了消融实验，将最传统的采用知识蒸馏的FedMD作为基线，然后对比仅使用类别加权，仅使用$F1$注意力，以及完整的FedCKD，实验结果如表4～表7所示。

表4 CICIDS2017上的全局模型消融实验结果

Tab. 4 Global model ablation experiment results in CICIDS2017

算法	Benign			DoS			DDoS			PortScan		
	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$
FedMD	0.88	0.98	0.92	0.99	0.87	0.93	0.99	1	1	0.99	0.99	0.99
类别	0.93	0.99	0.96	0.99	0.94	0.96	1	1	1	0.99	1	1
注意力	0.89	0.98	0.93	0.99	0.90	0.94	0.99	1	1	0.99	0.99	0.99
FedCKD	**0.96**	**0.99**	**0.97**	0.99	**0.97**	**0.98**	**1**	**1**	**1**	**1**	**0.99**	**1**

表5 CIDDS-001上的全局模型消融实验结果

Tab. 5 Global model ablation experiment results in CIDDS-001

算法	Normal			Portscan			DoS			Pingscan			Bruteforce		
	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$
FedMD	0.91	0.98	0.95	0.91	0.91	0.91	1	1	1	0.94	0.94	0.94	0.95	0.82	0.88
类别	0.96	0.95	0.95	0.97	0.93	0.95	1	1	1	0.97	0.95	0.96	0.94	0.97	0.96
注意力	0.91	0.97	0.95	0.92	0.93	0.93	1	1	1	0.95	0.96	0.95	0.95	0.90	0.93
FedCKD	**0.99**	0.94	**0.98**	**1**	**0.95**	**0.97**	**1**	**1**	**1**	**0.96**	**0.98**	**0.97**	0.93	**0.99**	**0.96**

表 6　CICIDS2017 上的客户端模型消融实验结果

Tab. 6　Client model ablation experiment results in CICIDS2017

客户端	算法	Benign			DoS			DDoS			PortScan		
		Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1
A	FedMD	0.87	0.99	0.93	0.99	0.91	0.95	1	1	1	0.99	0.99	0.99
	类别	0.88	0.99	0.93	0.98	0.8	0.88	1	1	1	0.99	0.99	0.99
	注意力	**0.90**	0.99	**0.95**	0.99	**0.91**	0.94	1	1	1	1	0.99	1
	FedCKD	0.88	**0.99**	0.93	**1**	0.85	0.91	**1**	**1**	**1**	**1**	**1**	**1**
B	FedMD	0.90	0.99	0.94	0.99	0.95	0.95	1	1	1	0.98	0.90	0.94
	类别	0.95	0.99	0.97	1	0.91	0.95	1	1	1	0.95	0.90	0.93
	注意力	0.97	0.99	0.98	**1**	0.93	0.96	1	1	1	0.99	0.96	0.97
	FedCKD	**0.98**	**0.99**	**0.99**	0.99	**0.99**	**0.99**	**1**	**1**	**1**	**1**	**0.99**	**0.99**
C	FedMD	0.89	0.99	0.94	0.99	0.96	0.97	0.95	0.94	0.95	0.99	1	0.99
	类别	0.93	0.99	0.96	0.99	1	1	0.99	0.90	0.95	1	0.99	1
	注意力	**0.92**	**0.99**	**0.96**	0.99	1	1	0.99	0.96	0.98	1	1	1
	FedCKD	0.92	0.99	0.96	**0.99**	**1**	**0.99**	**1**	**0.96**	**0.98**	**1**	**1**	**1**

表 7　CIDDS-001 上的客户端模型消融实验结果

Tab. 7　Client model ablation experiment results in CIDDS-001

客户端	算法	Normal			Portscan			DoS			Pingscan			Bruteforce		
		Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1
A	FedMD	0.91	0.99	0.95	0.88	0.93	0.90	1	1	1	0.94	0.97	0.95	0.92	0.85	0.88
	类别	0.93	0.99	0.95	0.94	0.95	0.95	1	1	1	0.95	0.96	0.96	0.90	0.82	0.86
	注意力	**0.96**	0.99	**0.97**	0.95	**0.97**	0.96	1	1	1	**0.97**	0.98	**0.98**	0.97	**0.91**	0.94
	FedCKD	0.93	**1**	0.96	**0.95**	0.96	**0.96**	**1**	**1**	**1**	0.96	**0.98**	0.97	**1**	0.9	**0.95**
B	FedMD	0.96	0.98	0.97	**0.88**	0.94	0.91	1	1	1	0.94	0.80	0.88	0.95	0.95	0.95
	类别	0.97	0.98	0.98	0.90	0.92	0.91	1	0.99	1	0.91	0.79	0.85	0.96	0.97	0.97
	注意力	0.98	0.97	0.97	0.92	0.92	**0.92**	0.99	0.99	1	0.94	**0.87**	0.91	**0.97**	0.96	0.96
	FedCKD	**0.98**	**0.98**	**0.98**	0.85	**0.97**	0.91	**1**	**1**	**1**	**0.98**	0.85	0.92	0.95	**0.99**	**0.97**
C	FedMD	0.94	0.98	0.96	0.91	0.90	0.90	1	1	1	0.95	0.97	0.96	0.92	0.91	0.91
	类别	0.97	0.98	0.98	0.93	0.89	0.91	1	1	1	0.97	0.98	0.98	0.96	0.97	0.97
	注意力	0.98	0.96	0.97	0.95	0.92	0.94	1	0.99	1	**0.97**	0.98	**0.98**	**0.97**	0.97	**0.97**
	FedCKD	**0.98**	**0.98**	**0.98**	**1**	**0.93**	**0.96**	**1**	**1**	**1**	0.96	**0.98**	0.97	0.93	**0.98**	0.96

消融实验结果表明，两项核心组件均能带来性能提升。类别权重聚合在服务器端有效提升了全局模型的精度与鲁棒性，而基于*F*1的注意力机制主要改善本地训练，对少数类识别效果尤为显著。进一步对比发现，仅用类别权重时全局性能更优，但少数类表现不足；仅用注意力机制时客户端少数类表现提升，但全局聚合效果不如前者。FedCKD 则结合两者，实现了全局与本地的同步优化。

2.3.4　全局模型对比

在全局模型对比中，本文通过比较 FedCKD 与 FedAvg，FedMD，FedHe，HFedCKD 在测试集上的表现，验证了类别加权知识蒸馏聚合对模型性能的提升，具体实验结果如图 5 所示，具体每一类的测试精度如表 8 和表 9 所示。在全局模型对比实验中，各方法均采用相同的模型结构，从而保证了对比的公平性。

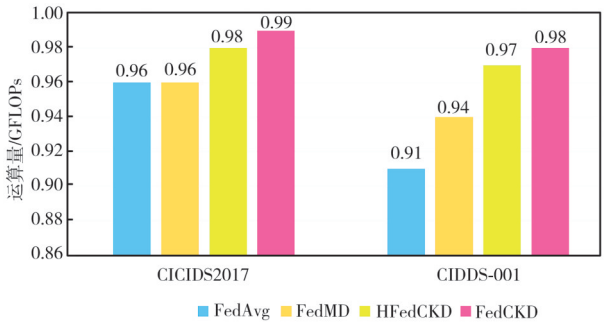


图 5　服务器端的全局模型准确率对比

Fig. 5　Comparison of global model accuracy on server

从实验结果可以看出：在CICIDS2017数据集中，FedAvg 和 FedMD 在常见类别上表现良好，但在少数类上精度不足；HFedCKD 整体性能优于 FedAvg 和 FedMD，但在 DoS 等少数类上仍存在欠缺。相比之下，FedCKD 在各类别均表现出更高的精度和*F*1值，尤其在少数类上优势明显。在 CIDDS-001 数据集中，FedAvg 在 Bruteforce、FedMD 在 Portscan 上表现较差，HFedCKD 虽整体稳定，但在 Pingscan 等

类别上仍存在短板。FedCKD在所有类别上表现更为均衡，展现出更强的鲁棒性与稳定性。综上所述，FedCKD在CICIDS2017和CIDDS-001上整体精度均优于其他方法。

表8 CICIDS2017上的联邦学习全局模型性能对比

Tab. 8 Comparison of global model performance of federated learning in CICIDS2017

算法	Benign			DoS			DDoS			PortScan		
	Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1
FedAvg	0.89	0.99	0.94	0.99	0.93	0.96	0.98	0.98	0.98	0.98	0.98	0.98
FedMD	0.88	0.98	0.92	0.99	0.87	0.93	0.99	1	1	0.99	0.99	0.99
HFedCKD	0.92	0.99	0.96	**1**	0.91	0.95	1	1	1	1	0.99	0.99
FedCKD	**0.96**	**0.99**	**0.97**	0.99	**0.97**	**0.98**	**1**	**1**	**1**	**1**	**0.99**	**1**

表9 CIDDS-001上的联邦学习全局模型性能对比

Tab. 9 Comparison of global model performance of federated learning in CIDDS-001

算法	Normal			Portscan			DoS			Pingscan			Bruteforce		
	Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1
FedAvg	0.94	0.98	0.96	0.88	0.85	0.84	1	1	1	0.93	0.92	0.92	0.88	0.81	0.81
FedMD	0.91	0.98	0.95	0.91	0.91	0.91	1	1	1	0.94	0.94	0.94	0.95	0.82	0.88
HFedCKD	0.93	**0.99**	0.96	0.98	0.95	0.97	1	1	1	0.96	0.98	0.97	**0.98**	0.92	0.95
FedCKD	**0.99**	0.94	**0.98**	**1**	**0.95**	**0.97**	**1**	**1**	**1**	**0.96**	**0.98**	**0.97**	0.93	**0.99**	**0.96**

在CIDDS-001数据集上，FedAvg在Bruteforce明显性能不足，FedMD在Portscan类别上表现较差。相比之下，FedCKD在所有类别上表现更加稳定，展现出其更强的识别能力和稳定性。

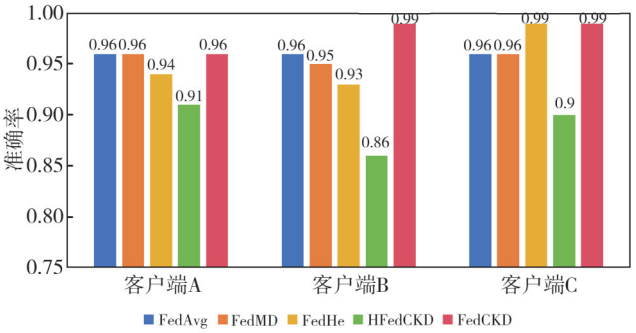
上述实验证明，FedCKD通过类别加权知识蒸馏机制有效避免了低性能模型对全局模型的负面影响。而FedAvg及FedMD则不同，在聚合时很容易受到某些客户端模型的拖累，从而导致全局模型精度下降。

2.3.5 客户端精度对比

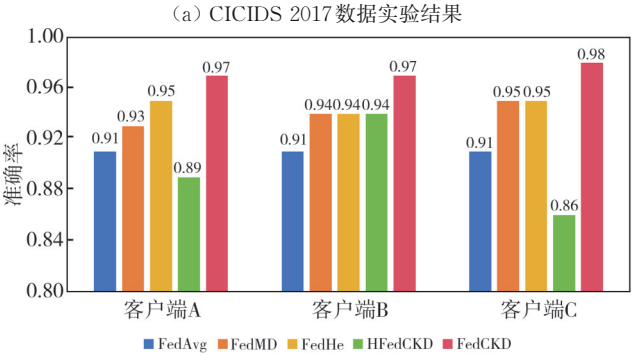
在客户端的实验中，FedAvg直接将全局模型部署到本地，代替原有的本地模型进行训练。FedHe通过聚合客户端的logits结合本地数据更新本地模型。FedMD和FedCKD则使用全局模型来指导本地模型，通过本地数据集计算logits。不同之处在于，FedCKD引入了注意力机制，根据训练过程中各类别的识别性能调节相应的权重，从而优化知识蒸馏过程，整体模型的精度对比如图6所示，具体的每一类指标情况如表10和表11所示。

在本地部署精度的比较中，FedCKD在两个数据集的大多数客户端与攻击类别上均展现出最佳性能，整体效果明显优于其他方法。FedCKD通过引入类别注意力机制，能够根据不同类别的重要性动态调整蒸馏权重，有效缓解了本地数据中存在的类别不均衡问题，提升了少数类的识别能力，从而实现了更稳定、更全面的模型训练效果。

相比之下，FedAvg直接部署全局模型，缺乏对本地数据特征的适应能力；FedMD虽采用蒸馏方式优化本地模型，但在类别分布不平衡的数据场景下容易出现性能波动；FedHe在不依赖公共数据的前提下完成知识共享，整体性能也优于FedAvg与FedMD，如果面临复杂的数据集(如CICIDS2017)会略显不稳定，而HFedCKD虽然也引入了双向蒸馏，但是由于其在蒸馏时没有考虑到数据类别不平衡的问题，因此都出现了少数类性能不佳的情况。



(a) CICIDS 2017数据实验结果



(b) CIDDS-001数据实验结果

图6 客户端的本地模型精确率对比

Fig. 6 Comparison of accuracy of client-side local model

表 10　CICIDS2017 上的联邦学习模型本地部署性能对比

Tab. 10　Performance comparison of local deployment of federated learning models in CICIDS2017

客户端	算法	Benign			DoS			DDoS			PortScan		
		Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1
A	FedAvg	**0.89**	0.99	**0.94**	0.99	**0.93**	**0.96**	0.98	0.98	0.98	0.98	0.98	0.98
	FedMD	0.87	0.99	0.93	0.99	0.91	0.95	1	1	1	0.99	0.99	0.99
	FedHe	0.82	**1**	0.90	1	0.78	0.87	0.99	1	1	0.99	1	1
	HFedCKD	0.77	0.99	0.87	1	0.65	0.79	0.96	1	0.98	0.99	0.99	0.99
	FedCKD	0.88	0.99	0.93	**1**	0.85	0.91	**1**	**1**	**1**	**1**	**1**	**1**
B	FedAvg	0.89	0.99	0.94	0.99	0.93	0.96	0.98	0.98	0.98	0.98	0.98	0.98
	FedMD	0.90	0.99	0.94	0.99	0.95	0.95	1	1	1	0.98	0.90	0.94
	FedHe	0.79	**1**	0.88	**1**	**1**	**1**	0.99	1	1	1	0.73	0.84
	HFedCKD	0.64	0.99	0.78	1	1	1	1	1	1	1	0.44	0.61
	FedCKD	**0.98**	0.99	**0.99**	0.99	0.99	0.99	**1**	**1**	**1**	**1**	**0.99**	**0.99**
C	FedAvg	0.89	0.99	0.94	0.99	0.93	0.96	0.98	0.98	0.98	0.98	0.98	0.98
	FedMD	0.89	0.99	0.94	0.99	0.96	0.97	0.95	0.94	0.95	0.99	1	0.99
	FedHe	**0.99**	**0.99**	**0.99**	0.99	0.99	0.99	1	**0.99**	**0.99**	0.99	1	0.99
	HFedCKD	0.73	0.98	0.84	0.97	0.99	0.99	1	0.96	0.77	0.99	1	0.99
	FedCKD	0.92	0.99	0.96	**0.99**	**1**	**0.99**	**1**	0.96	0.98	**1**	**1**	**1**

表 11　CIDDS-001 上的联邦学习模型本地部署性能对比

Tab. 11　Performance comparison of local deployment of federated learning models in CIDDS-001

客户端	算法	Normal			Portscan			DoS			Pingscan			Bruteforce		
		Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1	Pre	Rec	*F*1
A	FedAvg	**0.94**	0.98	0.96	0.88	0.85	0.84	1	1	1	0.93	0.92	0.92	0.88	0.81	0.81
	FedMD	0.91	0.99	0.95	0.88	0.93	0.90	1	1	1	0.94	0.97	0.95	0.92	0.85	0.88
	FedHe	0.92	0.99	0.96	0.90	0.96	0.93	1	0.99	1	0.95	0.97	0.96	0.99	0.92	0.90
	HFedCKD	0.89	0.99	0.94	0.72	0.96	0.82	1	1	1	0.93	0.98	0.95	0.99	0.5	0.68
	FedCKD	0.93	**1**	**0.96**	**0.95**	**0.96**	**0.96**	**1**	**1**	**1**	**0.96**	**0.98**	**0.97**	**1**	**0.9**	**0.95**
B	FedAvg	0.94	0.98	0.96	0.88	0.85	0.84	1	1	1	0.93	**0.92**	**0.92**	0.88	0.81	0.81
	FedMD	0.96	0.98	0.97	**0.88**	0.94	0.91	1	1	1	0.94	0.80	0.88	0.95	0.95	0.95
	FedHe	0.97	0.98	0.97	0.84	0.96	0.90	1	1	1	0.97	0.78	0.87	0.94	0.99	0.96
	HFedCKD	0.98	0.98	0.98	0.82	0.97	0.89	1	1	1	0.97	0.77	0.86	0.95	0.98	0.97
	FedCKD	**0.98**	**0.98**	**0.98**	0.85	**0.97**	**0.91**	**1**	**1**	**1**	**0.98**	0.85	0.92	**0.95**	**0.99**	**0.97**
C	FedAvg	0.94	0.98	0.96	0.88	0.85	0.84	1	1	1	0.93	0.92	0.92	0.88	0.81	0.81
	FedMD	0.94	0.98	0.96	0.91	0.90	0.90	1	1	1	0.95	0.97	0.96	0.92	0.91	0.91
	FedHe	0.98	0.98	0.98	1	0.83	0.90	1	0.99	1	0.95	0.98	0.96	0.85	0.98	0.91
	HFedCKD	0.99	0.96	0.98	0.99	0.37	0.53	1	1	1	0.95	0.98	0.96	0.60	0.99	0.75
	FedCKD	**0.98**	**0.98**	**0.98**	**1**	**0.93**	**0.96**	**1**	**1**	**1**	**0.96**	**0.98**	**0.97**	**0.93**	**0.98**	**0.96**

综上，FedCKD 在多客户端、多类别任务中展现出更强的精度表现与模型稳定性，具有更好的本地适应性和推广能力。

3　结　论

本文针对联邦学习在网络入侵检测系统中面临的模型异构与类别不均衡问题，提出了一种联邦类别权重知识蒸馏算法 FedCKD。该方法在无需统一模型架构的前提下，通过构建类别级别的可学习权重矩阵实现了更精细的 logits 蒸馏聚合，有效减弱了个别模型在某类数据上性能不佳对全局性能的影响，并进行了简要的算法分析。在两个主流入侵检测数据集上，与 FedAvg、FedMD、FedHe 和 HFedCKD 进行了系统对比，结果表明 FedCKD 在全局与本地模型精度上均取得更优性能，验证了其在异构联邦学习中的有效性与鲁棒性。

本文 FedCKD 尚未考虑联邦学习中的异步通信问题[17]，而在真实的 NIDS 应用中，客户端常因网络延迟、计算能力差异等因素导致训练进度不同，难以满足同步更新的假设。同时，如 FedHe 算法，虽然放弃公共数据可能会影响模型性能，但是更符合现实场景。此外，本文实验仅验证了 3 个客户端的场景。在实际部署中，往往存在 50 个以上的客户端及高度异构的硬件环境。FedCKD 的类别加权与注意力机制在理论上具备一定适应性，但其在大规模和强异构条件下的通信效率与收敛性能仍需进一步研究，这也将作为

后续工作的重点方向。

利益冲突声明/Conflict of Interests

所有作者声明不存在利益冲突。

All authors declare no relevant conflict of interests.

参考文献:

[1] KHRAISAT A, ALAZAB A, SINGH S, et al. Survey on federated learning for intrusion detection system: Concept, architectures, aggregation strategies, challenges, and future directions[J]. ACM Computing Surveys, 2025, 57(1): 1-38.

[2] THAKKAR A, LOHIYA R. A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions [J]. Artificial Intelligence Review, 2022, 55(1): 453-563.

[3] LANSKY J, ALI S, MOHAMMADI M, et al. Deep learning-based intrusion detection systems: A systematic review [J]. IEEE Access, 2021(9): 101574-101599.

[4] GUAN H, YAP P T, BOZOKI A, et al. Federated learning for medical image analysis: A survey[J]. Pattern Recognition, 2024, 151: 110424.

[5] 徐久强, 周洋洋, 王进法, 等. 基于流时间影响域的网络流量异常检测[J]. 东北大学学报(自然科学版), 2019, 40(1): 26-31.(in Chinese)
XU Jiuqiang, ZHOU Yangyang, WANG Jinfa, et al. Anomaly detection of network traffic based on flow time influence domain[J]. Journal of Northeastern University(Natural Science), 2019, 40(1): 26-31.(in Chinese)

[6] YE M, FANG X W, DU B, et al. Heterogeneous federated learning: State-of-the-art and research challenges[J]. ACM Computing Surveys, 2024, 56(3): 1-44.

[7] ZHANG L, SHEN L, DING L, et al. Fine-tuning Global Model Via Data-Free Knowledge Distillation For Non-Iid Federated Learning [C]//2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022: 10164-10173.

[8] MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[C]//20th International Conference on Artificial Intelligence and Statistics, 2017: 1273-1282.

[9] BAGUI S, LI K Q. Resampling imbalanced data for network intrusion detection datasets[J]. Journal of Big Data, 2021, 8(1): 6.

[10] LI D L, WANG J P. FedMD: Heterogeneous federated learning via model distillation[DB/OL]. (2019-10-08) [2025-08-22]. https://arxiv.org/abs/1910.03581.

[11] QI P H, ZHOU X Y, DING Y L, et al. FedBKD: Heterogeneous federated learning via bidirectional knowledge distillation for modulation classification in IoT-edge system[J]. IEEE Journal of Selected Topics in Signal Processing, 2023, 17(1): 189-204.

[12] ZHENG Y T, LIN B H, CHEN J Q, et al. HFedCKD: Toward robust heterogeneous federated learning via data-free knowledge distillation and two-way contrast[DB/OL]. (2025-03-09)[2025-08-22]. https://arxiv.org/abs/2503.06511.

[13] CHAN Y H, NGAI E C H. FedHe: Heterogeneous Models and Communication Efficient Federated Learning [C]//17th International Conference on Mobility, Sensing and Networking (MSN), 2021: 207-214.

[14] YAO D Z, PAN W N, DAI Y T, et al. FedGKD: Toward heterogeneous federated learning via global knowledge distillation[J]. IEEE Transactions on Computers, 2024, 73(1): 3-17.

[15] HAN J G, SHAN C F, ZHANG J Q. FedGMKD: An efficient prototype federated learning framework through knowledge distillation and discrepancy-aware aggregation[J]. Advances in Neural Information Processing Systems, 2024, 37: 118326-118356.

[16] WU C H, WU F Z, LÜ L J, et al. Communication-efficient federated learning via knowledge distillation [J]. Nature Communications, 2022, 13(1): 2032.

[17] HU C H, CHEN Z, LARSSON E G. Scheduling and aggregation design for asynchronous federated learning over wireless networks[J]. IEEE Journal on Selected Areas in Communications, 2023, 41(4): 874-886.