李贝贝，董育宁，邱晓晖. 基于客户端选择的不平衡联邦网络流量分类[J]. 智能计算机与应用，2026，16(1)：37-49. DOI：10.20169/j.issn.2095-2163.24032904

基于客户端选择的不平衡联邦网络流量分类

李贝贝，董育宁，邱晓晖
（南京邮电大学 通信与信息工程学院，南京 210003）

摘 要：近年来，网络流量分类的数据安全问题备受关注。联邦学习能够在保障数据隐私的基础上，实现数据共享。目前，联邦学习在流分类上面临客户端数据不平衡的挑战。针对此问题，本文提出了一种基于客户端选择的不平衡联邦网络流量分类方法。针对多样本分布场景，设计了循环传递前置标签集模型获取标签类别指标，结合衡量样本平衡程度的权值散度指标和针对少数目客户端引入的安德鲁·耶奥协议下的客户端样本数目指标，计算指标综合得分，实现客户端选择。实验结果表明，与代表性文献方法相比，本文方法在不增加不平衡网络流量分类时间的情况下，$F1$ 分数可提高 0.5%～10.0%。

关键词：前置标签集；标签类别指标；不平衡流量分类；客户端选择
中图分类号：TP391 **文献标志码**：A **文章编号**：2095-2163(2026)01-0037-13

Imbalanced federated network traffic classification based on client selection

LI Beibei, DONG Yuning, QIU Xiaohui
(School of Communications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: In recent years, the data security issue of network traffic classification has attracted much attention. Federated learning can achieve data sharing on the basis of ensuring data privacy. Currently, federated learning faces the challenge of client data imbalance in flow classification. To address this problem, this paper proposes an imbalanced federated network traffic classification method based on client selection. For multi-sample distribution scenarios, a circular transmission pre-label set model is designed to obtain label category indicators, combined with the weight divergence indicator to measure the degree of sample balance and the number of client samples under the Andrew Yeo protocol introduced for a small number of clients, comprehensive indicator scores are calculated, and client selection is implemented. Experimental results show that compared with representative literature methods, this method can increase the $F1$ score by 0.5% to 10.0% without increasing the time of imbalanced network traffic classification.

Key words: pre-label set; label category indicator; imbalanced traffic classification; client selection

0 引 言

随着互联网的迅猛发展，网络流量分类(Network Traffic Classification, NTC)在网络安全、服务优化等领域变得越来越重要[1]。网络流量分类技术通过识别和分类网络数据包，有助于监测网络中的异常行为、提高服务质量(Quality of Service, QoS)、并加强网络安全[2]。然而，在处理日益庞大和复杂的网络流量数据时，面临着数据隐私保护、通信效率和模型性能等多方面的挑战[3]。

近年来，深度学习(Deep Learning, DL)在图像理解、语音识别、文本生成、流量分类等任务中发挥着重要作用[4]。在面对海量训练数据和庞大模型的挑战时，主流解决方案是分布式机器学习(Machine Learning, ML)。

联邦学习(Federated Learning, FL)本质上是一种分布式机器学习框架，在做到了在保障数据隐私安全及合法合规的基础上，实现数据共享，共同建模[5]。FL 的隐私保护、数据安全、分布式训练、实时性和协作等方面的优势，为流量分类任务提供了一种有效的解决方案，并为流量分类模型的性能和可靠性带来了显著的提升。

作者简介：李贝贝(1998—)，女，硕士研究生，主要研究方向：网络流分类，多媒体通信与无线网络；邱晓晖(1968—)，女，博士，教授，主要研究方向：智能信号处理，图像处理与模式识别。
通信作者：董育宁(1955—)，男，博士，教授，博士生导师，主要研究方向：网络流分类，多媒体通信与无线网络。Email：19900011@njupt.edu.cn。

收稿日期：2024-03-29

0.1 联邦 NTC 面临的问题和研究动机

FL 作为一种分散式机器学习框架,提供了在分布式环境中协同训练模型的有效手段[6]。然而,针对网络流量分类领域,尤其是在面对不平衡的流量分布时,现有的联邦学习方法仍然面临模型性能下降的问题。

目前,联邦学习中一个关键性的挑战是参与方的数据或样本数量上相差悬殊,或在统计上具有异质性、即数据是非独立同分布的(Non-Independent and Identically Distributed,Non-IID),这与传统的分布式 ML 的独立同分布的假设相悖。Non-IID 的特性造成 FL 模型难以优化,与集中存储的方法相比,其模型性能下降明显[7]。

尽管 ML 和深度 FL 方法,在默认独立同分布数据分布下,取得了突出的流量分类效果,但在应对客户端数据不平衡问题方面,性能还有待提高[8]。

0.2 本文贡献

针对上述难点,本文提出一种基于多层感知器(Multi Layer Perceptron,MLP)和客户端选择的联邦不平衡网络流量分类方法。从样本数目不均衡和类别不均衡两个层面出发,提出循环传递标签模型(Loop-Federated Client Select, LFCS),得到标签重要程度指标,避免了以往客户端选择算法可能会造成的'新'标签丢失问题。提出权值散度(Weight Divergence,WD)、客户端样本数和标签重要程度等指标相结合的客户端打分模型来完成客户端选择。实验表明,该方法的分类精度优于代表性文献方法。本文的主要贡献如下:

(1)设计了循环传递前置标签集的模型结构选择客户端。首先,随机打乱客户端顺序。然后,利用计数器的 0 表示还未接收到前置标签集,1 表示已经接收到前置标签集。每个客户端将当前标签集随机传递给计数器为 0 的任意一个客户端,直到所有客户端计数器的值都为 1,则每个客户端都获得了本地客户端和其前置标签集相比的缺失标签数目和新增标签数目。

(2)设计了客户端打分模型。针对较多数目客户端。利用熵权法对 WD、缺失标签数目和新增标签数目三个指标加权获得客户端得分。针对较少数目的客户端。引入客户端样本数目这个指标,利用较多数目客户端方法下的客户端得分,计算出最低和次低的客户端得分差值。如果差值大于阈值则遵循目前的客户端得分进行筛选,否则,再通过比较最低和次低客户端的安德鲁·耶奥协议[9](Andrew Yao's Protocol ,AYP)加密下客户端样本数目进行筛选。

(3)在 2 个真实数据集上验证了方法的有效性。LFCS 在不增加不平衡网络流量分类时间的情况下, $F1$ 分数在仅样本数目不平衡的情况下与代表性文献方法相差不多;在样本数目和类别双重不平衡的情况下,能比文献方法提高 3%~10%。

论文的其余部分安排如下。第 1 节回顾相关的联邦不平衡网络流量分类(Federated Imbalanced Network Traffic Classification,FIBNTC)方法;第 2 节详细叙述 LFCS;第 3 节给出 LFCS 和文献方法的实验结果比较;第 4 节是结论。为了阅读方便,表 1 列出了本文主要使用模型的缩略词含义。

表 1 模型主要缩略词表

Table 1 List of main model abbreviations

词含义	词全称	缩略词
网络流量分类	Network traffic classification	NTC
多层感知器	Multi Layer Perceptron	MLP
循环联邦客户端选择	Loop-Federated Client Select	LFCS
权值散度和客户端训练损失	Weight divergence and Client training Loss	WCL
安德鲁·耶奥协议	Andrew Yao's Protocol	AYP
总标签集	Total_Labels	TLs
当前客户端前置标签集	All_Previous_Labels	APLs
新增标签集	New_Labels	NLs
缺失标签集	Miss_Labels	MLs
新增标签数目	Count_New_Labels	CNLs
缺失标签数目	Count_Miss_Labels	CMLs
得分最低的客户端	client_Lowest score	Lst
得分次低的客户端	client_Second Lowest score	SLst
Lst 与 SLst 得分绝对差值	Absolute difference of Lst and SLst	Adif
得分最低的客户端样本数	Count_Sample Lowest score	CSLst
得分次低的客户端样本数	Count_Sample Second Lowest score	CSSLst

1　相关工作

（1）深度网络流分类方面。Soysal 等学者[10]研究和评估了 3 种监督 ML、贝叶斯网络（Bayesian Network，BN）、决策树（Decision Tree，DT）以及 MLP 等算法对 6 种不同类型的互联网流量进行的流分类实验。结果表明，MLP 算法更适用于高速互联网流量分类。

（2）联邦网络流分类方面。Mun 等学者[11]提出了一种 FL 流量分类协议，可以在不泄露隐私的情况下实现与 DL 相当的准确度。Yang 等学者[6]根据 FL 的应用场景中各个数据集的用户不完全相同、或用户特征不完全相同等数据的不同特点，将 FL 分为 3 类：横向联邦学习、纵向联邦学习和联邦迁移学习。Kim 等学者[12]通过区块链技术对所有的模型更新进行完整的记录，并给予丰厚的奖励来激励用户参与 FL，提出了基于权重的客户端子集选择方案，通过每个客户端局部模型的精度和参与训练的频率来选择用于训练的客户端，实现了较高的稳定性和较快的收敛速度。Zhan 等学者[13]设计了一种基于深度强化学习（Deep Reinforcement Learning，DRL）的激励机制，将传统的资源分配策略应用于 FL 分布式特殊场景中，以达到边缘节点的最佳训练策略和定价策略。

（3）Non-IID 网络流分类方面。Zhou 等学者[14]构建了一个名为 Astraea 的自平衡联邦学习框架，通过基于全局数据分布的数据增强和基于中介器的多客户端重新调度来缓解不平衡。与 FL 算法 FedAvg[15]相比，提高了准确率并降低了通信流量。Mohammed[16]提出了一种在线状态 FL 启发式方法来寻找最佳候选客户，设计了一个物联网客户端警报应用程序，该应用程序利用所提出的启发式方法来训练基于物联网设备类型分类的有状态 FL 全局模型，以提醒客户有关其环境中未经授权的物联网设备。Guo 等学者[17]提出了一种新的 FL 中的基于 WCL 算法中 WD 和客户端训练损失的客户端选择方案。Guo 等学者[18]提出了一种基于卷积神经网络（Convolutional Neural Network，CNN）和联邦分析（Federated Analytics，FA）的 FL 框架 FEAT，用于主动估计客户端偏度并选择 FL 中的低偏度客户端，以减轻异构环境中的流量分类准确性下降，且不会暴露用户隐私，但其收敛速度还有待进一步提高。

上述方法在 FIBNTC 问题上取得了一定的成效，但在类别不平衡情况下还存在不足。本文利用循环前置标签集获取类别不平衡指标，再基于熵权法利用 WD、缺失标签数目、新增标签数目和客户端样本数目几个指标获取客户端得分，从而进行客户端选择，能在一定程度上克服上述问题。

2　本文方法

2.1　模型框架

LFCS 模型框架见图 1。由图 1 可知，模型框架含有一个聚合服务器和多个客户端，包括本地模型训练、客户端指标计算、客户端选择、模型聚合 4 个模块。

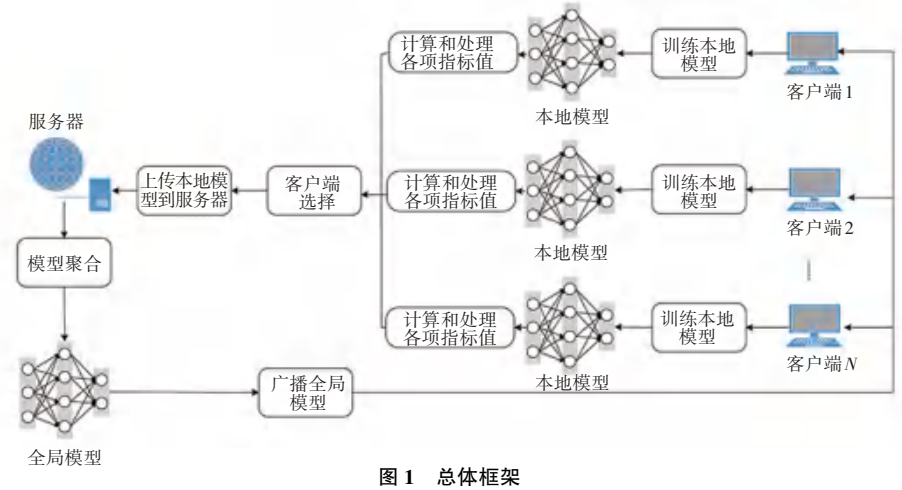

图 1　总体框架

Fig. 1　Overall framework

在本地模型训练阶段，客户端使用本地数据在全局模型的基础上再进行训练构建一个 MLP 流量分类模型，即本地训练模型；在客户端指标计算阶段，利用循环传递前置标签集的模型计算出标签类别相关指标：缺失标签数目和新增标签数目、WD 以及 AYP 加密后的客户端样本数目等 4 个指标，进行数据处理；客户端选择模块是将计算处理后的指标参数，通过熵权法加权获得客户端得分，并筛选掉得

分低的客户端;在模型聚合阶段,是在聚合服务器上将经过筛选后的客户端本地模型进行聚合得到全局模型,再将此全局模型广播到各个客户端,以便进行下一轮的本地模型训练。

2.2 本地模型训练

提取特征中,将 CSE-CIC-IDS2018 数据集[19] pcap 文件输入 CICFlowMeter[20] 模型,提取出 80 个流特征,包括流持续时间、流字节率、流包率、正反向总包数、两条流之间的平均时间、流最小和最大长度、流在空闲之前处于活跃状态的平均时间等等。

客户端使用本地私有数据提取出的流特征在全局模型的基础上再进行训练构建一个 MLP 流量分类模型、即本地训练模型。然后,分别将神经网络权重矩阵参数上传到聚合服务器。

2.3 客户端指标计算

2.3.1 权值散度

权值散度(WD)是 DL 中一个用于衡量神经网络中权重分布变化的指标,在此处用于衡量客户端的样本分布不平衡程度。

在第一轮模型训练完成并聚合以后,得到全局权重矩阵和全局偏置矩阵并下发到各个客户端,从第二轮开始,每个本地客户端都计算当前客户端权重矩阵和偏置矩阵与上一轮的全局权重矩阵和全局偏置矩阵的偏离程度,用 WD 来衡量,计算公式如下:

$$\mathrm{div}_n(k)=\frac{\|x_n(k)-x(k-1)\|}{\|x(k-1)\|} \tag{1}$$

$\mathrm{div}_n(k)$ 表示第 n 个客户端当前训练的本地模型 $x_n(k)$ 和上一轮全局模型 $x(k-1)$ 之间的 WD。

2.3.2 标签指标

分析可知,仅利用 WD 衡量客户端的样本分布不平衡程度会导致标签类别的重要程度降低。例如一个标签类别仅为少数客户端拥有,这些少数客户端与全局相比 WD 会极高,表示这些客户端重要程度很低。但是,由于这个标签类别数据仅存在于这些客户端,显然这些客户端又是极其重要的,从而影响做出较优的客户端选择。

为了解决该问题,设计了循环传递前置标签集(如图 2 所示)的模型来获取客户端的标签指标缺失标签数目和新增标签数目。假设初始有 N 个客户端,每个客户端都有一个计数器(记录该客户端传递标签集的次数)、一个前置标签集(记录从循环传递标签集开始直到当前客户端的前一个客户端所包含的样本标签类别,和总的标签集)。为避免隐私泄露,相邻客户端的选择都是随机的,每个客户端将当前标签集随机传递给计数器为 0 的任意一个客户端,直到所有客户端计数器的值都为 1,则每个客户端也都获得了本地客户端与其前置标签集相比的缺失标签数目和新增标签数目。

经过循环传递模型,获得当前客户端的缺失标签数目和新增标签数目两个指标。研究中假设 C_n 表示第 n 个客户端的标签集,总标签集 $\mathrm{TLs}=\{C_1\cup C_2\cup C_3\cup\cdots\cup C_n\}$,第 n 个客户端的前置标签集 $\mathrm{APLs}_n=\{C_1\cup C_2\cup C_3\cup\cdots\cup C_{n-1}\}$,第 n 个客户端的新增标签集 $\mathrm{NLs}_n=C_n-\mathrm{APLs}_n$,第 n 个客户端的缺失标签集 $\mathrm{MLs}_n=\mathrm{APLs}_n-C_n$,则可得到第 n 个客户端的缺失标签数目公式为:

$$\mathrm{CMLs}_n=\mathrm{card}(\mathrm{MLs}_n)=\mathrm{card}(\mathrm{APLs}_n-C_n) \tag{2}$$

新增标签数目的公式具体如下:

$$\mathrm{CNLs}_n=\mathrm{card}(\mathrm{NLs}_n)=\mathrm{card}(C_n-\mathrm{APLs}_n) \tag{3}$$

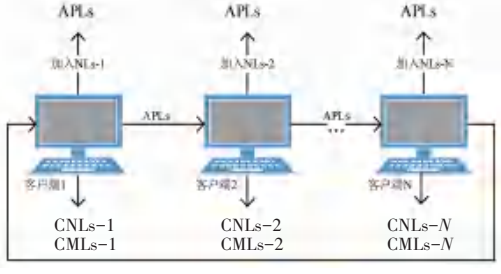

图 2 循环传递前置标签集的模型

Fig. 2 Model of looping forward label set

2.3.3 客户端总样本数目

针对较少数目客户端,在 WD、缺失标签数、新增标签数的基础上,引入第 4 个指标各客户端样本数用于辅助判断客户端选择。在前 3 个指标最低和次低值的差值小于阈值(1/1 000×最小值)时,使用 AYP 的加密方法比较两者样本数的大小,保留样本数大的客户端,剔除样本数小的客户端。

2.3.4 数据归一化

(1)WD(负向指标)。范围为[0,+∞],通过 $e^{-\mathrm{div}_n(k)}$ 对式(1)进行归一化。

(2)缺失标签数(负向指标)。范围为[0,M],公式如下:

$$U_{\mathrm{norm\_miss}}=0.998\cdot\frac{U_{\max}-U}{U_{\max}}+0.002 \tag{4}$$

(3)新增标签数(正向指标)。范围为[0, M),公式如下:

$$U_{\mathrm{norm\_new}}=0.998\cdot\frac{U-U_{\min}}{U_{\max}-U_{\min}}+0.002 \tag{5}$$

(4)客户端样本数(正向指标)。公式如下:

$$U_{\text{norm\_count}} = 0.998 \cdot \frac{U - U_{\min}}{U_{\max} - U_{\min}} + 0.002 \quad (6)$$

其中，U 表示原始缺失标签数/新增标签数/客户端样本数；U_{norm} 表示归一化后的值；$U_{\min}$ 表示数据集中的最小值；$U_{\max}$ 表示数据集中的最大值。

需要提及的是，式(6)仅在对比时使用。参数 0.998 和 0.002 的引入是为了使归一化值 U_{norm} 大于 0，避免后续计算熵值时出现 ln0。

2.4　客户端选择

利用循环传递标签结合式(1)的 WD，从所有客户端中筛选出 $N - m$ 个客户端。首先，每个客户端计算指标 WD、缺失标签数、新增标签数。然后，对客户端指标加权求和计算客户端得分。最后，根据得分筛选掉一定数量的客户端(较少客户端情况下客户端筛选还依据客户端样本数，具体细节在 3.7 节展示)。

2.4.1　利用熵权法获取各个指标的权重 $\boldsymbol{w}$

熵是对不确定信息的度量，熵与信息量成反比。熵值越小，信息量越大，权重越大。熵权法获取权重的一般过程如图 3 所示。

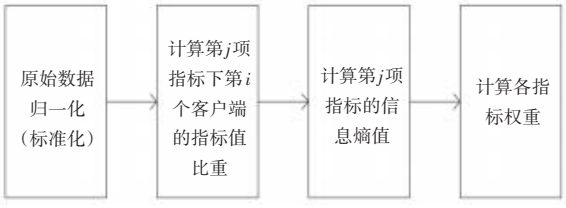

图 3　熵权法获取权重的一般过程

Fig. 3　The general process of obtaining weights using the entropy weight method

计算第 j 项指标下，第 i 个客户端的指标值比重：

$$p_{ij} = \frac{a_{ij}}{\sum_{i=1}^{n} a_{ij}}, \ \sum p_{ij} = 1 \quad (7)$$

计算第 j 项指标的信息熵值：

$$e_j = -b\sum_{i=1}^{N-m} p_{ij} \ln p_{ij}, \ b = \frac{1}{\ln n} \quad (8)$$

计算各指标权重：

$$w_j = \frac{1 - e_j}{\sum_{j=1}^{J} (1 - e_j)} \quad (9)$$

其中，J 表示指标的数目。

在此基础上，进一步推得：

$$\text{clients\_sortScore1}[k] = 1/4 \times \{w_1 \times \text{div}[k] + w_2 \times U_{\text{norm\_miss}} + w_3 \times U_{\text{norm\_new}}\} \quad (10)$$

$$\text{clients\_sortScore}[k] = 1/4 \times \{w_1 \times \text{div}[k] + w_2 \times U_{\text{norm\_miss}} + w_3 \times U_{\text{norm\_new}} + w_4 \times U_{\text{norm\_count}}\} \quad (11)$$

其中，clients_sortScore1 表示 3 个指标的加权和(WD、缺失标签数、新增标签数)，clients_sortScore 表示 4 个指标的加权和(WD、缺失标签数、新增标签数、样本数)。

2.4.2　客户端样本数比较

为了比较出 2 个客户端所拥有的样本数目的大小，引入百万富翁问题。这是一个为了说明隐私保护概念而设计的问题。在这个问题中，2 个富翁想要比较各自的财富多少，但又不愿意直接透露具体的财富数字。本文利用安全多方计算协议(Secure Multiparty Computation Protocol) - AYP[9] 来于解决百万富翁问题。

AYP 是一种基本的协议，通常包含比较和隐私保护的基本操作，而不涉及过于复杂的密码学工具。其中心思想是：彼此把数据 CA，CB 藏在自己的维度上，随后找到一个交点，其中一方根据这个交点的位置来推测出自己所持数据与对方所持数据的大小关系。比较过程详见算法 1。

算法 1　客户端样本数比较过程

Client-a Input：CA 表示客户端 Client-a 的样本数目；s 表示客户端 Client-a 的公钥；p 表示一个素数

Client-b Input：CB 表示客户端 Client-b 的样本数目；s 表示客户端 Client-a 的公钥；sx 表示客户端 Client-b 选定的很大的正整数；p 表示一个素数

Output：Res=1 表示 CA>CB，-1 表示 CA<CB，0 表示 CA=CB

1.　用 3~16 比较 CA 与 CB 的位数，如果不等，则直接获得结果，如果相等再执行 2~26

2.　for $l = 1, 2, \cdots,$ CA do：

3.　$\overline{\text{CA}} \leftarrow \text{int}(\text{str}(\text{CA})[0])$，$\overline{\text{CB}} \leftarrow \text{int}(\text{str}(\text{CB})[0])$

4.　$k = \text{encrypt}(\text{sx}, a) \leftarrow$ Client - b 用公钥 a 对 sx 加密

5.　$g = k - \text{CB} + 1$

6.　Client-a get $g \leftarrow$ Client - b

7.　$\text{Dec}\{g+1, g+2, \cdots, g+10\} \leftarrow$ Client - a 用公钥 a 解密

8.　$z = \text{Dec}\{g+1, g+2, \cdots, g+10\} \pmod p$

9.　for num in z：

```
10.        for i in range(CA, len(z) + 1):
11.            num=num+1
12.        end for
13.      end for
14.      Client-b get z ← Client - a
15.      if zj ==CB(mod p):
16.        Res=1
17.        break
18.      随机数 a1,a2←Client - a, 随机数 b1, b2 ← Client - b
19.      Client-b get a1 → r1 = a2 - b1
20.      Client-a get b1 → r2 = a1 - b2
21.      if r1 + r2 == 0:
22.        CA = int(str(CA)[1:])
23.        CB = int(str(CB)[1:])
24.        continue
25.      Res=-1
26.    end for
```

针对较多数目客户端,根据客户端得分 clients_sortScore1 筛选掉得分最低的 $N-m$ 个客户端。针对较少数目的客户端,先根据客户端得分 clients_sortScore1,找到 client_sortScore1 最低值和次低值的绝对差值如果大于 1/1 000×最小值,则不考虑样本数,否则,比较两者 client_sortScore(散度、缺失标签数、新增标签数、样本数四个指标加权和)的大小,保留 client_sortScore 大的那个,剔除 client_sortScore 小的那个客户端。

2.5 聚合服务器

2.5.1 初始化

聚合服务器基于单隐藏层 MLP 神经网络构建一个初始的未经训练的通用分类模型(称为全局模型)。在这一步中,识别超参数(例如,隐藏层的数量,损失函数 ξ,批大小 β,epoch 数、通信轮次 E 等)。服务器端初始化一个 MLP 全局模型,分为输入层、一个隐藏层、输出层。添加了一个具有 30 个神经元的密集(全连接)层作为输入层,模型编译时采用 ReLU(Rectified Linear Unit)非线性激活函数、交叉熵损失函数、adam 优化器和 $F1$ 分数监测指标,输出层采用 Softmax 激活函数将模型的输出映射为概率分布。交叉熵损失函数公式具体如下:

$$L=\frac{1}{N}\sum_i L_i=\frac{1}{N}\sum_i -[y_i\cdot\log(p_i)+(1-y_i)\cdot\log(1-p_i)] \tag{12}$$

其中,在二分的情况下,模型最后需要预测的结果只有 2 种情况,对于每个类别预测得到的概率为 p 和 $1-p$,y_i 表示样本 i 的 label,正类为 1,负类为 0;p_i 表示样本 i 预测为正类的概率。

2.5.2 模型聚合

聚合服务器使用联邦平均(FedAvg)算法[15](未用加权),即:

$$x(k)=\frac{1}{N-m}\sum_{n=1}^{N-m}x_n(k) \tag{13}$$

通过式(13)在获得所有模型更新后聚合不同客户端的更新,并得到一个新的更新的全局模型。服务器将模型更新信息发送回客户端。客户端将学习新模型参数,对流量进行分类,并检查服务器是否定期更新。整个分类模型算法见如下。

算法 2 基于 MLP 不平衡联邦的客户端选择算法

```
1. 初始化: C = {C_1,…,C_N},n 表示第 n 个客户端;m 表示筛选掉的客户端数量,K 表示通信轮次、初始轮次 k = 0,初始全局模型 x(0)
2. CScore = [],存放客户端得分
3. rem =[],存放筛选掉的客户端
4. for k = 1,2,…,K do
5.    x(k - 1) ← 服务器端广播全局模型
6.    for each client ∈ C
7.      x_n(k) ← 第 n 个客户端第 k 轮本地模型
8.        计算客户端指标
9.        CScore←客户端得分
10.   if   N >= 10:
11.     Lst ← min(enum (CScore))
12.     for _ in range(m):
13.       rem. append(Lst)
14.       Lst = min(enum (CScore))
15.     end for
16.   Lst ← min(enum(CScore))
17.   SLst← np. argsort(CScore)[1]
18.   for _ in range(m):
19.     Adif←Lst 与 SLst 得分绝对差值
20.     if Adif >0. 001×Lst 得分
21.       rem. append(Lst)
22.     AYP 比较 Lst 与 SLst 样本数:
23.     if CSLst >= CSSLst:
24.       rem. append(Lst)
25.     rem. append(SLst)
26.   end for
```

27.　$\boldsymbol{W}_{1\cdots n}(k) \leftarrow$ 上传服务器

28.　$x(k) = 1/(N-m)\sum_{n=1}^{N-m} x_n(k)$ ←全局模型

其中，$C = \{1,\cdots,N\}$ 表示客户端集；参数矩阵 $\boldsymbol{W}$ 包括权值矩阵和偏置矩阵；K 表示通信轮次；$x(k)$ 表示第 k 轮的全局模型；$x_n(k)$ 表示第 k 轮第 n 个客户端的本地训练模型；$\boldsymbol{W}_n(k) = \{w_{1n}(k), w_{2n}(k), w_{3n}(k), w_{4n}(k)\}$ 表示第 k 轮第 n 个客户端的权值矩阵和偏置矩阵。

3　实验结果与讨论

将 LFCS 与文献方法 WCL[17] 和 FEAT[18] 进行性能对比，并分析其内在原因。

3.1　数据集

实验在 CSE－CIC－IDS2018 和 NF－Labeled－75[21] 两个数据集上进行。

3.1.1　CSE-CIC-IDS2018 数据集

具体信息见表 2。

表 2　CSE-CIC-IDS2018

Table 2　CSE-CIC-IDS2018

标签	名称	样本数量
30	Bot	286 191
31	Brute Force-Web	611
32	Brute Force-XSS	230
33	DDoS-HOIC	686 012
34	DDoS-LOIC-UDP	1 730
35	DDoS-LOIC-HTTP	576 191
36	DoS-GoldenEye	41 508
37	DoS-GoldenEye	461 912
38	DoS-SlowHTTPTest	139 890
39	DoS-Slowloris	10 990
40	FTP-BruteForce	193 360
41	Infiltration28	68 881
42	SQL Injection	87
43	SSH-Bruteforce	187 589
44	Infiltration1	93 063

3.1.2　NF-Labeled-75 数据集

NF-Labeled-75 数据集是基于 NetFlow 的物联网网络数据集，数据集共有 3 577 296 个样本，共 75 类数据，包含 87 个特征，流量统计信息是使用 CICFlowmeter 获得的，包括到达间隔时间、上下行的平均吞吐量、上下行包最大和最小长度等等，所使用到的数据具体信息见表 3。

表 3　NF-Labeled-75

Table 3　NF-Labeled-75

标签	应用	样本数量
0	FACEBOOK	10 000
1	CLOUDFLARE	10 000
2	TWITTER	10 000
3	SKYPE	10 000
4	AMAZON	10 000
5	SSL	10 000
6	WINDOWS_UPDATE	10 000
7	MSN	10 000
8	HTTP_CONNECT	10 000

3.2　评价指标

研究中拟对分类性能和时间性能进行评估。

（1）分类性能。采用 2 种评价指标，分别是准确率（Acc）、$F1$ 分数（$F1$_score）；其中，Acc 是分类器对整个样本判断正确的比重，$F1$_score 是 P 和召回率的一种调和平均。数学计算公式具体如下：

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{14}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{15}$$

$$F1\_\mathrm{Score} = \frac{2 \times P \times R}{P + R} \tag{16}$$

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{17}$$

其中，TP 和 TN 分别表示真例和假例被正确分类的样本数；P 表示精确率，是预测为样本的结果中，正确预测的比例；R 表示召回率，是所有正样本中被找出的比例；FP 和 FN 分别表示真例和假例被错误分类的样本数。

（2）时间性能。分为模型训练时间和模型识别时间。

3.3　实验环境

实验在硬件配置为 AMD Ryzen 7 5800H with Radeon Graphics CPU@3.20 GHz，16 G 内存的联想笔记本电脑上完成，操作系统为 Windows 11。分类器均使用具有 3 层神经网络结构（输入层、隐藏层、输出层）的 MLP，输入层到隐藏层和隐藏层到输出层之间都有一个权重矩阵和偏置向量，采用五折交叉验证。未特殊说明情况下采用下述数据集设置。

（1）对于 CSE-CIC-IDS2018 数据集：使用标签为[36,30,44,33,40,43,37,35,39]的样本，每个类别为 10 000 个样本，客户端数目 N 为 20，进行九分

类。

(2)对于 NF-Labeled-75 数据集:使用标签为[0,1,2,3,4,5,6,7,8]的样本,每个类别 10 000 个样本,客户端数目: N 为 20, 进行九分类。

3.4 实验结果

对非联邦 MLP 和联邦 MLP 的批大小、学习率、聚合轮次参数进行研究试验如下:

在非联邦 MLP(批大小 40)的实验中,采用 Adam 优化器针对模型训练损失进行学习率的自动调整,训练和验证损失随训练轮次变化曲线如图 4 所示,在 10~12 轮次范围内 loss 已经基本收敛了,且 $F1$ 分数达到 0.996。

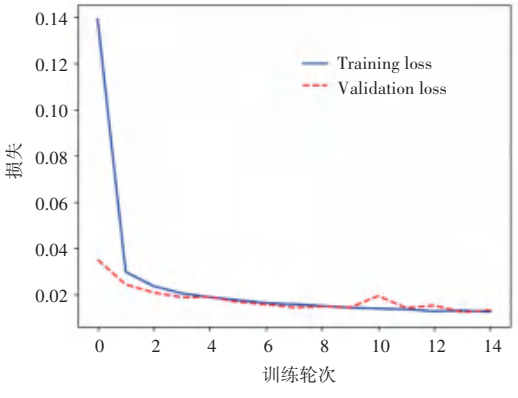

图 4 非联邦 MLP 下 loss 变化曲线

Fig. 4 Loss change curve under non-federated MLP

在联邦 MLP 实验中,批大小 min_samples/2(min_samples 表示客户端样本数的最小值),采用 Adam 优化器针对模型训练损失进行学习率的自动调整,在 10~15 轮次范围内,epochs = $10/(k+1)$ ~ $15/(k+1)$ 时,这里 k 表示第 k 个聚合轮次,识别效果较好。

客户端的筛选分为无状态和有状态两种。其中,无状态表示从第二轮开始每轮删除一个客户端,此后每一轮都会执行一次客户端选择算法;有状态[22]表示在第二轮一次删除最差的 m 个客户端,候选客户端可以参与在训练全局模型中使用的每个通信和计算轮次。

根据无状态和有状态筛选策略进行实验对比,可得出:在第二轮一次删除最差的 m 个客户端,与从第二轮开始每轮删除一个客户端准确率相比,$F1$ 分数基本一致。但前者明显节省了训练时间,因为后者从第二轮到第 $N-m$ 轮都需要执行客户端选择算法,而前者只需要在第二轮执行选择算法,并且避免了样本分布差的客户端进入到下一轮训练过程中。因此,后续实验均在有状态筛选策略下进行。

本文除了对 WD、缺失标签数目、新增标签数目和客户端样本数的研究之外,还将其与把 $F1$ 分数作为第 5 个指标进行了各项实验对比。得出:在客户端各种不均衡分布情况下,无论是 LFCS、还是文献方法的 $F1$ 分数都没有太大改变,但却导致训练时间大幅度增加。其中,LFCS 考虑 $F1$ 分数指标时,训练时间是不考虑 $F1$ 分数指标时的 9 倍左右。

采用上述设置,进行样本数量不平衡网络流分类,分类混淆矩阵如图 5、图 6 所示。对角线上的元素表示模型正确分类的样本数,即真实标签和预测标签相同的样本数,非对角线上的元素表示模型错误分类的样本数,分类 $F1$ 分数在 2 个数据集上分别能达到 0.994 和 0.968,基本上能达到集中式训练的识别效果。

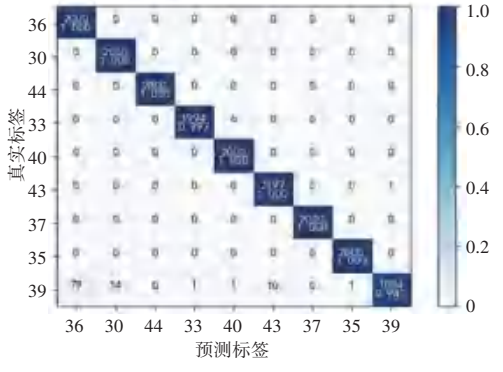

图 5 CSE-CIC-IDS2018 上 LFCS 分类混淆矩阵

Fig. 5 LFCS classification confusion matrix on CSE-CIC-IDS2018

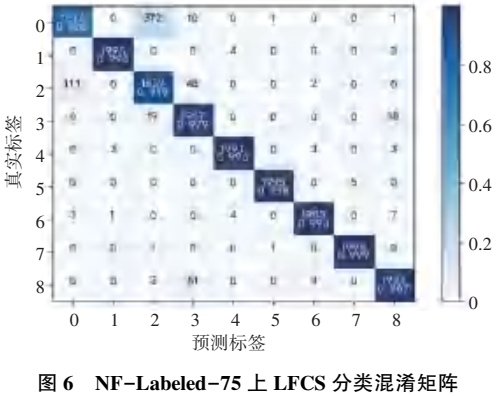

图 6 NF-Labeled-75 上 LFCS 分类混淆矩阵

Fig. 6 LFCS classification confusion matrix on NF-Labeled-75

3.5 客户端筛选数目研究

本文将在 3 种不同的样本分布下进行实验。

(1)样本分布 1:样本的分布仅有数目不平衡,无类别不平衡。即客户端不存在样本数目为 0 的标签,所有客户端都具有全部标签,只是客户端样本数目不同。

(2)样本分布 2:有数目不平衡和类别不平衡。即 1/2 的客户端有[44,33,40,43,37,35]或者[2,

3,4,5,6,7]六个类别以及其他任意类别,其他客户端的标签并集包含剩余类别。

(3)样本分布 3:有数目不平衡和类别不平衡。即 1/5 的客户端有[44,33,40,43,37,35]或者[2,3,4,5,6,7]六个类别以及其他任意类别,其他客户端的标签并集包含剩余类别。

3 种样本分布下 LFCS 准确率、*F*1 分数、模型训练和识别时间随筛选掉客户端数目的变化如图 7~图 12 所示。

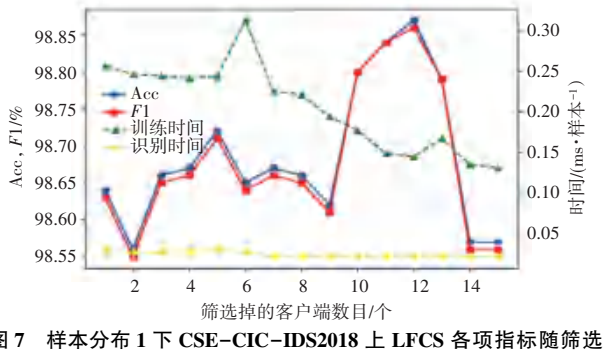

图 7　样本分布 1 下 CSE-CIC-IDS2018 上 LFCS 各项指标随筛选掉客户端数目的变化

Fig. 7　Distribution1: Changes in various LFCS indicators on CSE-CIC-IDS2018 with the number of clients to be filtered out

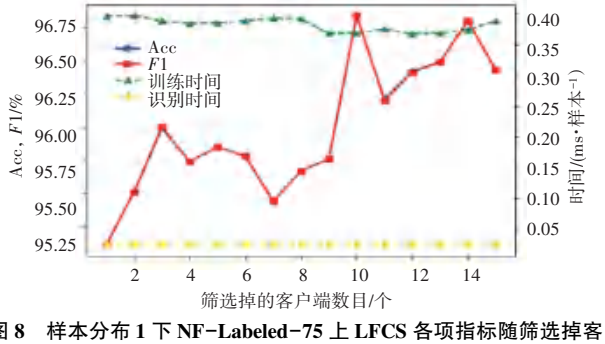

图 8　样本分布 1 下 NF-Labeled-75 上 LFCS 各项指标随筛选掉客户端数目的变化

Fig. 8　Distribution1: Changes in various indicators of LFCS on NF-Labeled-75 with the number of clients to be filtered out

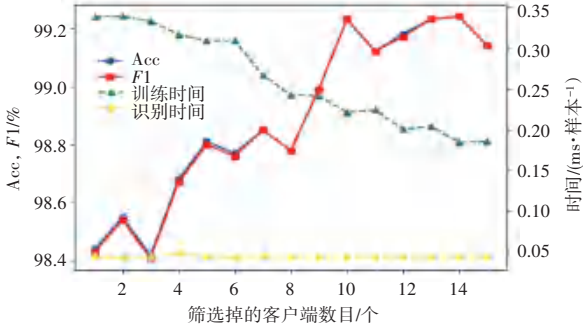

图 9　样本分布 2 下 CSE-CIC-IDS2018 上 LFCS 各项指标随筛选掉客户端数目的变化

Fig. 9　Distribution2: Changes in various LFCS indicators on CSE-CIC-IDS2018 with the number of clients to be filtered out

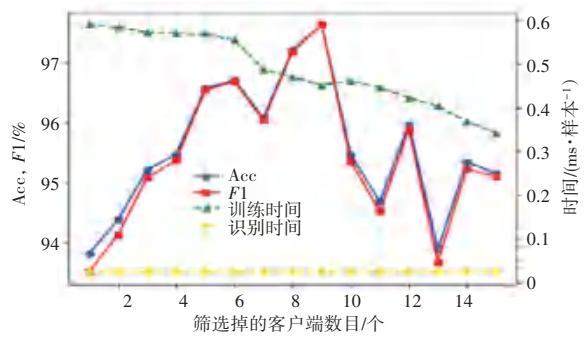

图 10　样本分布 2 下 NF-Labeled-75 上 LFCS 各项指标随筛选掉客户端数目的变化

Fig. 10　Distribution2: Changes in various indicators of LFCS on NF-Labeled-75 with the number of clients to be filtered out

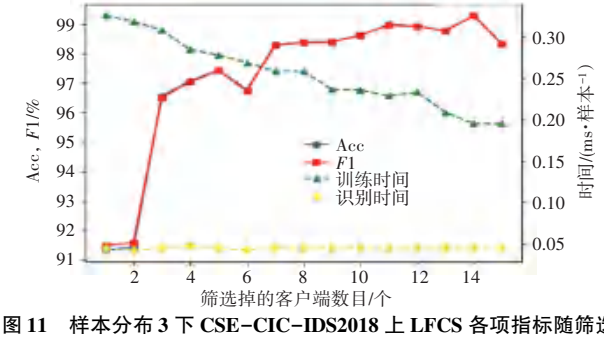

图 11　样本分布 3 下 CSE-CIC-IDS2018 上 LFCS 各项指标随筛选掉客户端数目的变化

Fig. 11　Distribution3: Changes in various LFCS indicators on CSE-CIC-IDS2018 with the number of clients to be filtered out

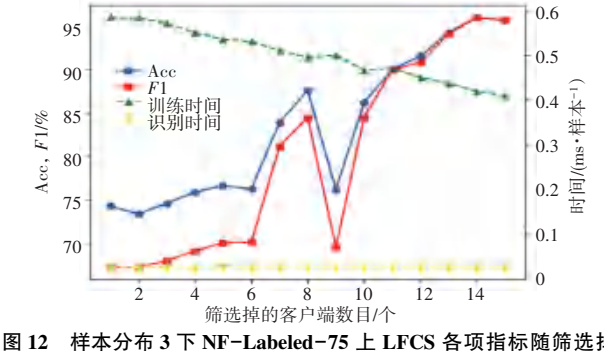

图 12　样本分布 3 下 NF-Labeled-75 上 LFCS 各项指标随筛选掉客户端数目的变化

Fig. 12　Distribution3: Changes in various indicators of LFCS on NF-Labeled-75 with the number of clients to be filtered out

由图 7~图 12 可看出,在各种客户端分布情况下,在筛选掉客户端数目的 1/2 附近时,识别的 *F*1 分数能达到最高,且训练时间也相对较少。在筛选掉数目较少时,由于较差客户端模型参与聚合,使得总体模型性能较差,删除过多数目的客户端也会导致训练样本少,模型过拟合。

实验发现在 10 个客户端及以上大多符合此规律。

3.6　不同方法对比实验

LFCS 与 WCL 和 FEAT 方法在 3 种不同样本分布下的 Acc、*F*1、训练时间以及识别时间对比见表 4~表 6。

表 4 样本分布 1 下 2 个数据集上不同方法的 Acc、*F*1、训练时间以及识别时间对比数据

Table 4 Distribution1: Comparison data of Acc, *F*1, training time and recognition time of different methods on the two data sets

方法	CSE-CIC-IDS2018				NF-Labeled-75			
	Acc/%	*F*1/%	训练时间/ms	分类时间/(ms·样本$^{-1}$)	Acc/%	*F*1/%	训练时间/ms	分类时间/(ms·样本$^{-1}$)
LFCS	99.4	99.4	17.170	0.025	96.8	96.8	26.570	0.025
WCL	99.3	99.3	18.842	0.026	96.3	96.3	19.506	0.025
FEAT	**99.5**	**99.6**	51.914	0.037	**97.0**	**97.0**	159.569	0.039
无筛选	99.1	99.1	17.277	0.025	96.0	96.0	29.641	0.025

表 5 样本分布 2 下 2 个数据集上不同方法的 Acc、*F*1、训练时间以及识别时间对比数据

Table 5 Distribution2: Comparison data of Acc, *F*1, training time and recognition time of different methods on the two data sets

方法	CSE-CIC-IDS2018				NF-Labeled-75			
	Acc/%	*F*1/%	训练时间/ms	分类时间/(ms·样本$^{-1}$)	Acc/%	*F*1/%	训练时间/ms	分类时间/(ms·样本$^{-1}$)
LFCS	98.7	98.7	16.461	0.025	**97.6**	**97.6**	32.539	0.025
WCL	**99.2**	**99.2**	23.522	0.026	77.6	72.3	39.743	0.025
FEAT	98.7	98.6	61.505	0.038	94.0	94.0	160.467	0.039
无筛选	93.2	93.0	20.984	0.025	93.5	93.1	43.599	0.025

表 6 样本分布 3 下 2 个数据集上不同方法的 Acc、*F*1、训练时间以及识别时间对比数据

Table 6 Distribution3: Comparison data of Acc, *F*1, training time and recognition time of different methods on the two data sets

方法	CSE-CIC-IDS2018				NF-Labeled-75			
	Acc/%	*F*1/%	训练时间/ms	分类时间/(ms·样本$^{-1}$)	Acc/%	*F*1/%	训练时间/ms	分类时间/(ms·样本$^{-1}$)
LFCS	**99.1**	**99.1**	15.839	0.024	**96.1**	**96.1**	30.171	0.025
WCL	92.2	92.2	16.579	0.024	57.1	48.9	38.803	0.025
FEAT	97.1	97.1	98.483	0.037	85.4	85.7	162.564	0.039
无筛选	89.6	89.2	19.236	0.025	70.4	63.9	40.920	0.025

由以上几种分布及实验结果可得出如下结论：

(1)LFCS 在客户端样本分布相对平衡时，*F*1 分数与 WCL 方法相差不多，FEAT 方法识别效果稍好，但此时样本的不平衡度是比较低的(仅有各个类别样本数目的不平衡)，此时的识别准确率优势可能在于 FEAT 复杂的 CNN 模型，是以训练时间为代价来提高模型的准确率。

(2)LFCS 在客户端样本分布不平衡度比较高时，*F*1 分数有明显提高，在样本 2 和样本 3 分布情况下，LFCS 的识别准确率和 F1 分数几乎都优于文献方法，且样本不平衡度越高，优势越明显。

(3)LFCS 的训练时间和识别时间与 WCL 方法相差不多，且比较稳定，但 FEAT 方法的训练时间是 LFCS 的 3~6 倍左右。

为了进一步研究具体类别的分别效果，在 2 个真实数据集上，以样本分布 3 为例，得出了 LFCS 与文献方法 WCL 和 FEAT 的分类混淆矩阵如图 13~图 18 所示。

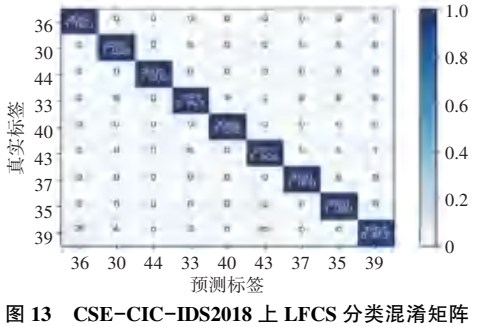

图 13 CSE-CIC-IDS2018 上 LFCS 分类混淆矩阵

Fig. 13 LFCS classification confusion matrix on CSE-CIC-IDS2018

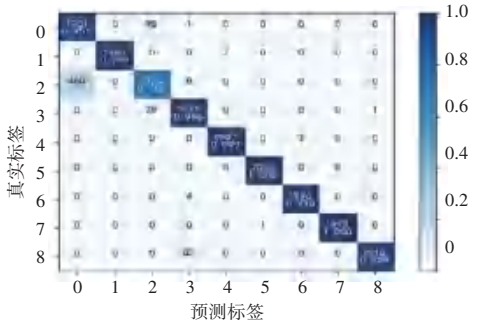

图 14 NF-Labeled-75 上 LFCS 分类混淆矩阵

Fig. 14 LFCS classification confusion matrix on NF-Labeled-75

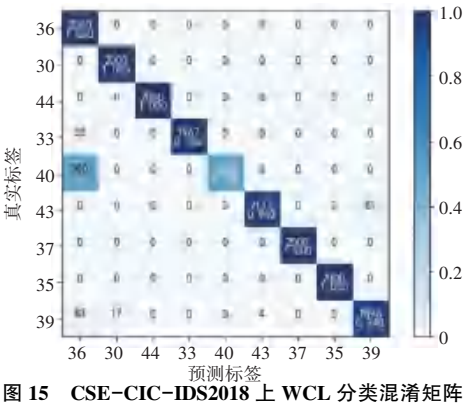

图 15　CSE-CIC-IDS2018 上 WCL 分类混淆矩阵

Fig. 15　WCL classification confusion matrix on CSE-CIC-IDS2018

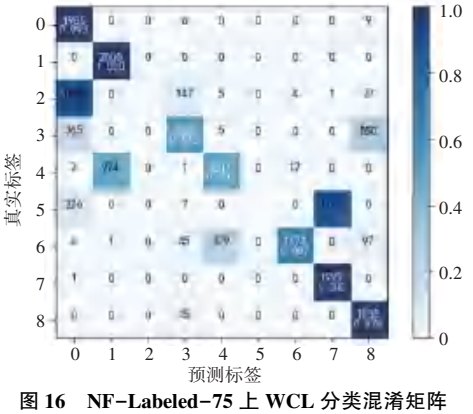

图 16　NF-Labeled-75 上 WCL 分类混淆矩阵

Fig. 16　WCL classification confusion matrix on NF-Labeled-75

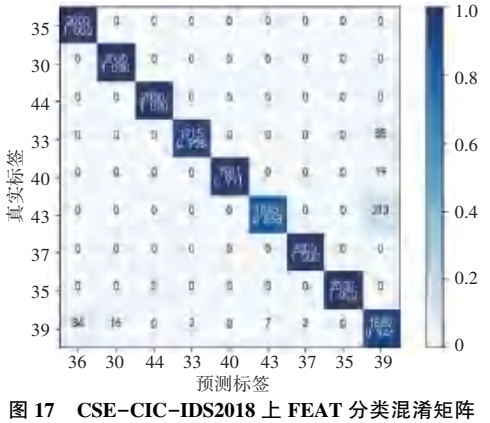

图 17　CSE-CIC-IDS2018 上 FEAT 分类混淆矩阵

Fig. 17　FEAT classification confusion matrix on CSE-CIC-IDS2018

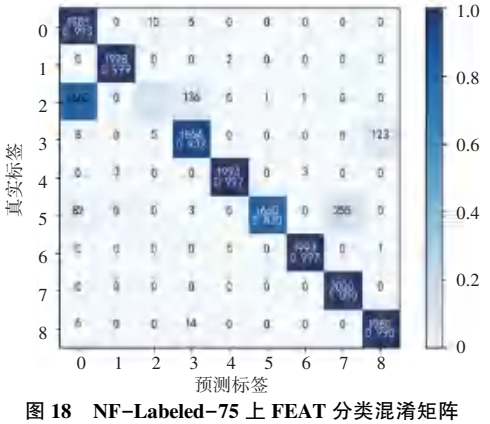

图 18　NF-Labeled-75 上 FEAT 分类混淆矩阵

Fig. 18　FEAT classification confusion matrix on NF-Labeled-75

由图 13～图 18 可以看出：

(1) LFCS 在 CSE-CIC-IDS2018 上在 9 个类别上的分类效果都不错；在 NF-Labeled-75 上 0 和 2 两个类别有少量错分，分别代表 FACEBOOK 和 TWITTER；与图 6 对比可以看出，尽管是只有样本数目不平衡的情况，仍然存在此类错分。所以，LFCS 在样本分布 3 下没有因为样本分布问题明显影响分类效果，是因为 NF-Labeled-75 数据集中 FACEBOOK 和 TWITTER 属于比较难分的同类社交软件。

(2) WCL 在 CSE-CIC-IDS2018 上，标签 40 容易错分成标签 36；在 NF-Labeled-75 上[2,3,4,5,6]标签的分类效果都较差。这些分类效果差的标签类别属于类别分布不平衡的类别。所以，WCL 不能适应类别分布不平衡的场景。

(3) FEAT 在 CSE-CIC-IDS2018 上，标签 33 和 43 少量错分成标签 39；在 NF-Labeled-75 上标签 2 和 5 容易错分成其他类别。而 2 和 5 都是类别分布不均衡的标签。FEAT 在样本分布 3 下能改善分布不平衡的问题，但提升效果有限，且在不同数据集上的改善效果不稳定。

3.7　针对较少客户端数目

在较少数目客户端情况下，客户端样本数不能忽视，当 2 个客户端 client_sortScore1 区分度不足够大时，直接加权客户端样本数指标，获取 client_sortScore，会放大客户端样本数指标的作用。研究中对客户端样本数指标应用策略进行阐释分述如下。

(1) 策略 1：直接计算 4 个指标的加权和计算客户端的得分。

(2) 策略 2：先计算出每个客户端的 3 个指标加权和的得分 client_sortScore1(WD、缺失标签数、新增标签数三个指标和)，如果 client_sortScore1 最小值和第二小的值的差值大于 1/1 000×最小值，则不考虑样本数，否则，比较两者样本数的大小(AYP 的加密方法)，保留样本数大的那个，剔除样本数小的那个。具体过程如图 19 所示。

(3) 策略 3：先计算出每个客户端的 3 个指标加权和的得分 client_sortScore1(WD、缺失标签数、新增标签数三个指标和)，如果 client_sortScore1 最小值和第二小的值的差值大于 1/1 000×最小值，则不考虑样本数，否则，比较两者 client_sortScore(WD、缺失标签数、新增标签数、样本数四个指标加权和)的大小，保留 client_sortScore 大的那个，剔除 client_sortScore 小的那个客户端。具体过程如图 20 所示。

在 9 个类别中，每个类别 2 500 个样本，客户端

数目：$N=5$ 设置下，2 个数据集上，筛选掉客户端数目分别为 1~3 时，策略 1 到策略 3 与不考虑客户端样本数指标（只有 3 个指标）情况下的 Acc、$F1$、训练时间以及识别时间对比数据见表 7~表 9。

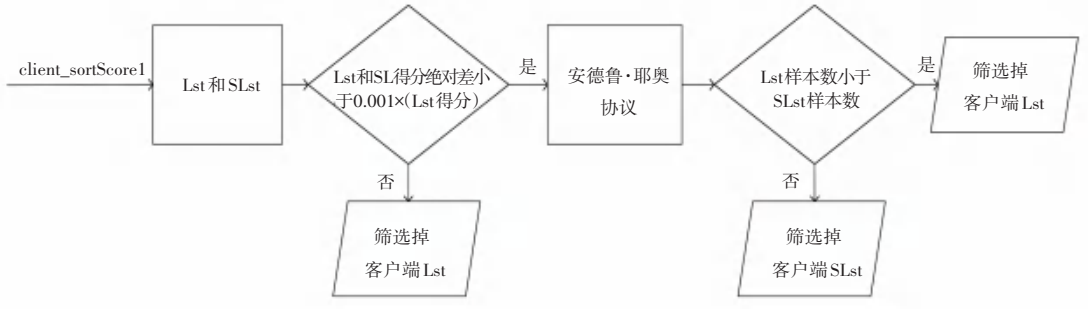

图 19 客户端样本数指标利用策略 2 过程

Fig. 19 Process of the client sample number indicator utilization strategy 2

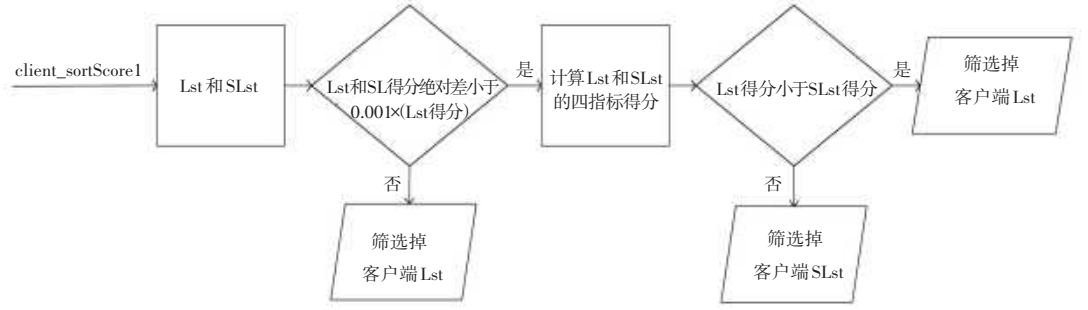

图 20 客户端样本数指标利用策略 3 过程

Fig. 20 Process of the client sample number indicator utilization strategy 3

表 7 筛选掉 1 个客户端时，2 个数据集上 3 个策略数据对比

Table 7 Comparison of three policy data on two data sets when one client is filtered out

$m=1$	CSE-CIC-IDS2018				NF-Labeled-75			
	Acc/%	$F1$/%	训练时间/ms	分类时间/(ms·样本$^{-1}$)	Acc/%	$F1$/%	训练时间/ms	分类时间/(ms·样本$^{-1}$)
策略 1	99.0	99.0	4.791	0.097	83.2	81.2	4.031	0.036
策略 2	**99.4**	**99.4**	6.816	0.098	**88.0**	**87.2**	5.549	0.035
策略 3	99.3	99.3	6.782	0.097	88.0	87.1	5.543	0.036
3 个指标	99.3	99.3	4.796	0.097	87.1	86.4	5.413	0.037

表 8 筛选掉 2 个客户端时，2 个数据集上 3 个策略数据对比

Table 8 Comparison of three policy data on two data sets when two clients are filtered out

$m=2$	CSE-CIC-IDS2018				NF-Labeled-75			
	Acc/%	$F1$/%	训练时间/ms	分类时间/(ms·样本$^{-1}$)	Acc/%	$F1$/%	训练时间/ms	分类时间/(ms·样本$^{-1}$)
策略 1	99.3	99.3	4.443	0.097	87.6	86.5	3.718	0.036
策略 2	**99.4**	**99.4**	5.868	0.098	90.9	90.5	4.610	0.036
策略 3	**99.4**	**99.4**	6.260	0.100	**92.1**	**91.6**	4.623	0.035
3 个指标	97.5	97.5	4.472	0.098	89.0	89.7	4.976	0.037

表 9 筛选掉 3 个客户端时，2 个数据集上 3 个策略数据对比

Table 9 Comparison of three policy data on two data sets when three clients are filtered out

$m=3$	CSE-CIC-IDS2018				NF-Labeled-75			
	Acc/%	$F1$/%	训练时间/ms	分类时间/(ms·样本$^{-1}$)	Acc/%	$F1$/%	训练时间/ms	分类时间/(ms·样本$^{-1}$)
策略 1	**98.9**	**98.9**	3.686	0.099	76.7	75.0	3.084	0.036
策略 2	**98.9**	**98.9**	4.994	0.100	87.7	87.3	4.020	0.035
策略 3	98.7	98.7	4.691	0.099	**88.6**	**88.5**	4.010	0.034
3 个指标	98.5	98.5	3.642	0.096	86.8	86.8	3.999	0.035

综上分析可知：

（1）将 3 个指标与策略 1 到策略 3 对比发现，3 个指标时由于客户端样本数分布的不确定性（样本数差距可能很大、也可能不明显），识别效果不稳定；策略 1 在 $m=1$ 时的效果不如 3 个指标和其他策略，可能原因是直接加权计算得分的方式放大了

客户端样本数这个指标；策略 2 和策略 3 的识别效果相差不多，都能比较好地在较少客户端数目情况下进行样本识别；策略 2 的识别准确率和 $F1$ 分数总优于 3 个指标和策略 1。

(2)在策略 2 的情况下，通过 AYP 的加密，不会泄露客户端样本数这个隐私。

(3)在客户端数目比较少的情况下，采用策略 2 的方法仍基本满足筛选掉 1/2 左右的客户端数目可以达到比较好的效果。

4　结束语

本文提出了一种基于客户端选择的不平衡联邦网络流量分类方法。首先，由聚合服务器初始化一个全局模型，本地客户端在全局模型的基础上，利用本地的数据集进行再训练；接着，客户端计算 WD 不平衡度指标和循环传递前置标签集获得的标签类别指标(结合 AYP 的客户端样本数指标)；由熵权法计算客户端最终得分从而进行客户端筛选，最终通过收敛的聚合后全局模型进行样本识别。在 2 个真实网络数据集上进行方法验证，结果表明，在 3 种样本分布情况下，LFCS 的 $F1$ 分数均能达到 0.9 以上。与文献方法相比，LFCS 在分类准确率上具有明显优势，时间性能相差不多。

本文方法存在的局限性：未解决在线客户端选择问题，即客户端只能在训练阶段离线选择。

参考文献

[1] WANG Wei, ZHU Ming, WANG Jinlin, et al. End-to-end encrypted traffic classification with one-dimensional convolution neural networks[C]//Proceedings of 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). Piscataway, NJ: IEEE, 2017: 43-48.

[2] JAIN A V. Network traffic identification with convolutional neural networks[C]//Proceedings of 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech). Piscataway, NJ: IEEE, 2018: 1001-1007.

[3] LIU Yang, WANG Jiabo, LIU Qinbo, et al. FedTC: A personalized federated learning method with two classifiers[J]. Computers Materials & Continua, 2023, 76(9): 3013-3027.

[4] SHONE N, NGOC T N, PHAI V D, et al. A deep learning approach to network intrusion detection[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2018, 2(1): 41-50.

[5] LI Li, FAN Yuxi, TSE M, et al. A review of applications in federated learning[J]. Computers & Industrial Engineering, 2020, 149: 106854.

[6] YANG Qiang. AI and data privacy protection: the way to federated learning[J]. Journal of Information Security Research, 2019, 5(11): 961-965.

[7] XIAO Wenjie, TANG Xuehai, ZHOU Biyu, et al. Fed-Tra: Improving accuracy of deep learning model on Non-IID in federated learning[C]//Proceedings of International Conference on Algorithms and Architectures for Parallel Processing. Cham: Springer, 2021: 790-803.

[8] ZHANG Lin, LUO Yong, BAI Yan, et al. Federated learning for non-IID data via unified feature learning and optimization objective alignment[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, NJ: IEEE, 2021: 4420-4428.

[9] SNYDER P. Yao′s garbled circuits: Recent directions and implementations[EB/OL]. (2014-01-01). https://api.semanticscholar.org/CorpusID:14833041.

[10] SOYSAL M, SCHMIDT E G. Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison[J]. Performance Evaluation, 2010, 67(6): 451-467.

[11] MUN H, LEE Y. Internet traffic classification with federated learning[J]. Electronics, 2020, 10(1): 27.

[12] KIM Y J, HONG C S. Blockchain-based node-aware dynamic weighting methods for improving federated learning performance[C]//Proceedings of the 20th Asia-Pacific Network Operations and Management Symposium (APNOMS). Piscataway, NJ: IEEE, 2019: 1-4.

[13] ZHAN Yufeng, LI Peng, QU Zhihao, et al. A learning-based incentive mechanism for federated learning[J]. IEEE Internet of Things Journal, 2020, 7(7): 6360-6368.

[14] ZHOU Yuhao, SHI Minjia, TIAN Yuxin, et al. DeFTA: A plug-and-play decentralized replacement for FedAvg[J]. arXiv preprint arXiv, 2204.02632, 2022.

[15] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[J]. arXiv preprint arXiv, 1602.05629, 2023.

[16] MOHAMMED I, TABATABAI S, AL-FUQAHA A, et al. Budgeted online selection of candidate IoT clients to participate in federated learning[J]. IEEE Internet of Things Journal, 2020, 8(7): 5938-5952.

[17] GUO Yingya, HUANG Kai, CHEN Jianshan. WCL: Client selection in federated learning with a combination of model weight divergence and client training loss for Internet traffic classification[EB/OL]. (2021-12-01). https://doi.org/10.1155/2021/3381998.

[18] GUO Y, WANG D. Feat: A federated approach for privacy-preserving network traffic classification in heterogeneous environments[J]. IEEE Internet of Things Journal, 2022, 10(2): 1274-1285.

[19] UNB. CSE-CIC-IDS2018 on AWS[EB/OL]. (2018-01). https://www.unb.ca/cic/datasets/ids-2018.html.

[20] UNB. CICFlowMeter (formerly ISCXFlowMeter)[EB/OL]. (2017-04-01). https://www.unb.ca/cic/research/applications.html#CICFlowMeter.

[21] Kaggle. IP network traffic flows labeled with 75 apps[EB/OL]. (2018-01-01). https://www.kaggle.com/datasets/jsrojas/ip-network-traffic-flows-labeled-with-87-apps/data.

[22] MOHAMMED I, TABATABAI S, AL-FUQAHA A, et al. Budgeted online selection of candidate IoT clients to participate in federated learning[J]. IEEE Internet of Things Journal, 2020, 8(7): 5938-5952.