基于双层联邦学习的高动态车联网业务边缘协作计算机制

徐思雅,郭佳惠
(北京邮电大学网络与交换技术全国重点实验室,北京 100876)

摘 要: 联邦学习作为一种新兴的分布式机器学习架构,允许车联网中多个车辆终端进行本地模型训练,并在兼顾数据隐私保护的条件下实现模型的全局聚合,从而提供可靠的车联网服务. 然而,在联邦学习训练过程中,车辆终端往往因其高移动性在不同区域间切换训练,导致全局模型精度低. 此外,恶意终端频繁上传无效或错误模型数据将导致车联网服务可靠性差. 因此,本文提出了一种基于双层联邦学习的高动态车联网业务边缘协作计算机制. 首先,综合考虑车辆终端的移动性、计算能力和可靠性,构建了终端服务能力模型,并提出了基于深度强化学习的边缘协作计算域构建算法,通过将多个边缘节点覆盖下的车辆终端进行聚簇训练,降低了终端本地模型的切换概率,从而保证联邦学习模型训练的持续性. 进而,构建了包含边缘协作计算域内聚合层和域间聚合层的双层联邦学习框架,分别采用基于自适应聚合因子的本地模型半异步聚合机制和基于数据量的区域模型异步聚合机制,提升了联邦学习系统的聚合效率. 特别地,考虑终端高速移动引起的跨域问题,引入了本地模型部分条件更新机制,避免了高质量模型被低质量模型覆盖的情况,进一步提高了全局模型准确率和系统资源利用率. 仿真结果表明,本文所提机制在模型精度和服务可靠性等方面均优于本地计算、同步联邦学习和异步联邦学习算法.

关键词: 联邦学习;边缘计算;可靠性;高动态;车联网
基金项目: 国家自然科学基金(No.62201074);工业互联网创新发展工程项目
中图分类号: TN92 **文献标识码:** A **文章编号:** 0372-2112(2024)07-2228-14
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20230065

Dual-Layer Federated Learning Based Edge Collaborative Computing Mechanism for High Dynamic Internet of Vehicle Businesses

XU Si-ya, GUO Jia-hui
(*State Key Laboratory of Networking and Switching Technology*, *Beijing University of Posts and Telecommunications*, *Beijing* 100876, *China*)

Abstract: As an emerging distributed machine learning architecture, federated learning (FL) allows multiple users to train local models and achieve global aggregation of models with data privacy protection, thus providing reliable Internet of Vehicle (IoV) services. However, in the training process of FL, many training terminals may switch among domains due to the high mobility, resulting in low accuracy of the global model. Besides, malicious terminals may frequently upload invalid or incorrect model data which leads to low service reliability. Therefore, we build the dual-layer FL based edge collaborative computing mechanism for high dynamic IoV businesses. Firstly, we comprehensively consider the mobility, computing ability and reliability to construct the service capability model for the terminal, and then propose the edge collaborative computing domain (ECCD) construction algorithm based on deep reinforcement learning. By clustering the vehicle terminals covered by multiple edge nodes, the switching probability of the terminal local model will be reduced, and the sustainability of the FL model training can be guaranteed. Furthermore, we design a dual-layer FL framework including the inter-ECCD aggregation layer and cross-ECCD aggregation layer, respectively. It adopts the semi-asynchronous aggregation mechanism for local models based on the adaptive aggregation factor in the inter-ECCD aggregation layer, and the asynchronous aggregation mechanism for domain's regional model based on data volume in the cross-ECCD aggregation layer, which jointly

收稿日期:2023-01-20;修回日期:2023-06-13;责任编辑:孙瑶

improve the aggregation efficiency of the FL system. In particular, considering that the high speed terminals may cause the cross-domain problem, we introduce the partial conditional update mechanism for the local model to avoid the situation that the high-quality models are covered by the low-quality models, which further improves the accuracy of the global model and the utilization of FL system resources. The simulation results verify that the proposed framework outperforms the local computing and asynchronous/synchronous FL algorithms in terms of model accuracy and service reliability.

Key words: federated learning; edge computing; reliability; high dynamic; Internet of Vehicle

Foundation Item(s): National Natural Science Foundation of China (No.62201074); Innovation and Development Project of Industrial Internet

1 引言

随着车联网与电子信息技术的发展,用户对障碍识别、流量预警等高动态智能业务的需求大幅提升,终端与边缘节点之间的数据共享带来了海量的车载数据,如道路环境、车辆轨迹和驾驶员操作等. 并且,车联网终端往往具有高动态性特征,包括具有较高的移动速度以及多变的行驶轨迹. 为提供智能化的交通服务,深度强化学习(Deep Reinforcement Learning,DRL)等人工智能算法被广泛应用于车联网中,但现有方法大多采用集中式模型训练模式[1],会造成中央服务器工作过载或网络流量过大,引起较大时延. 同时,该方法需要频繁地交换梯度和模型参数,易导致带宽消耗过大及车辆终端隐私泄露等问题.

联邦学习作为一种新兴的分布式机器学习框架[2],可应用于高动态车联网场景中,允许车辆终端从中央云服务器下载模型,并在本地进行模型训练,同时在满足数据隐私保护的条件下将模型参数上传至中央云服务器实现模型的全局聚合. 然而,车辆在高速移动过程中会在域间进行切换,而采用传统的联邦学习方法时,终端进入新的训练域时将丢弃已有的训练模型,可能造成高质量模型被低质量模型覆盖的情况. 此外,模型聚合的滞后性也会引发模型精度和聚合效率低等问题[3~5]. 因此,针对高动态车联网业务场景,研究基于联邦学习的边缘协作计算机制,保障模型训练的稳定性和高效性具有重要的理论和应用价值.

近年来,针对移动网络场景中的联邦学习机制已有较多研究成果[6~8],其中,文献[6]考虑了车辆终端移动性对模型训练造成的不利影响,提出了一种基于隐私保护的车联网联邦学习框架. 在该框架中,将单一边缘节点的覆盖范围设置为计算域,并根据车辆终端的声誉值动态调整计算域内参与训练的终端数量,从而提高了模型训练的准确性. 但是,该框架未考虑车辆终端差异化的服务能力,致使计算域的服务能力不均,造成数据训练周期差异较大,难以保障模型的训练精度. 文献[7]在划分计算域的基础上,考虑了车辆终端在计算资源和模型结构方面的异构性,解决了终端数据异质性引起的训练效率低等问题,但未考虑移动终端在区域间切换对模型可靠训练造成的影响. 文献[8]优化了车联网场景中高速移动终端的无线资源分配与模型选择决策,最小化联邦学习损失函数,提高了模型精度,但上述方法均没有考虑联邦学习系统的模型聚合效率和边缘节点间的协作训练模式,致使边缘节点资源效用较低.

联邦学习根据聚合方式可分为同步联邦学习和异步联邦学习. 在联邦学习同步聚合机制中[9~11],边缘节点需要等待所有车辆终端上传本地模型参数后进行全局模型的聚合,造成较大等待时延. 针对以上问题,文献[12]提出了一种异步小批量模型聚合算法,有效缩短了异构终端空闲等待时间. 文献[13]提出了一种基于成本最优的异步联邦学习机制,在保证用户隐私的同时提高训练效率. 然而,在联邦学习异步聚合机制中[12~15],本地训练终端需频繁上传模型参数,消耗大量带宽资源,并且本地过时模型易对全局模型精度造成不利影响. 文献[16]设计了半异步联邦平均聚合协议、客户端选择与模型聚合算法等方法,利用边缘资源实现了多步骤异步聚合,在一定程度上解决了本地模型时间对齐的问题. 文献[17]也提出了一种半异步聚合机制,允许边缘节点等待一定时间或接收预定义数量的本地训练模型参数后,进行聚合得到全局模型,但上述方法缺乏对节点移动性及可靠性的考虑,不适用于高动态车联网场景. 此外,文献[18]提出了一种终端动态选择算法,考虑了节点的可靠性,通过选择具有更高信任值的终端参与聚合,可保障模型聚合过程中的资源利用率,但上述方法未充分考虑终端的动态行为特征,难以保障车联网服务的稳定性和可靠性.

为解决上述问题,本文提出了一种基于双层联邦学习的高动态车联网业务边缘协作计算机制,构建了终端服务能力模型,设计了基于DRL的边缘协作计算域构建算法,并在域内和域间分别采用基于自适应聚合因子的本地模型半异步聚合机制和基于数据量的区域模型异步聚合机制,进一步提高联邦学习系统的聚合效率以及车联网服务的可靠性. 本文主要的贡献如下:

(1)本文首次将边缘协作计算域的概念引入联邦学习系统中,提出了基于DRL的边缘协作计算域构建算法. 该算法综合考虑移动性、计算能力和可靠性等终端服务能力,将多个边缘节点覆盖的车辆终端聚簇形

成边缘协作计算域,并通过多个边缘节点间的协作聚合,避免了车辆终端高速移动过程中在多边缘节点间频繁切换造成的训练中断,保障了模型训练的连续性和准确性.

(2)本文创新性地设计了模型精度和聚合速率均衡的双层联邦学习系统自适应异步聚合机制.首先,在域内聚合层提出了一种基于自适应聚合因子的本地模型半异步聚合机制,综合考虑了不同终端的等待时间与模型精度,通过动态调整参与每轮域内模型聚合的终端规模,实现均衡高效的域内本地模型半异步聚合;另一方面,考虑到域内车辆终端服务能力和聚合规模不同引起的模型数据量差异,在域间聚合层提出了一种基于数据量的区域模型异步聚合机制,根据各域内有效本地模型数据总量,动态调整模型异步上传的权重,实现精准快速的区域模型异步聚合.该机制通过均衡本地模型聚合频率和提升全局模型聚合速度,提升了联邦学习系统聚合效率.

(3)特别地,本文针对终端高速移动引起的跨域问题,创新性地引入了本地模型部分条件更新机制,首先根据用途将模型分为通用模型块和业务模型块,进而在跨域过程中实时评估业务模型块质量并设置本地模型块动态更新策略,进一步提高了全局模型准确率和系统资源效用.

本文中符号定义如表1所示.

表1 符号定义

符号	定义	符号	定义
$\boldsymbol{w}$	划域决策	h_m	边缘节点的高度
n	车辆终端	loc_n	车辆n的初始位置
t	时隙	$D_{m,n}^{\mathrm{initmod}}$	车辆终端n从边缘节点m下载的初始模型数据量
$V_n(t)$	车辆终端n在时隙t的移动速度	X_{model}	单位源数据上训练模型所需的CPU周期数
$C_n(t)$	车辆终端n在时隙t的计算能力	$T_{\mathrm{train},n}$	本地训练模型的时延
$K_n(t)$	车辆终端n在时隙t的可靠性	$D_{m,n}^{\mathrm{local}}$	终端n本地存储的模型数据量
$\boldsymbol{S}_n(t)$	终端服务能力模型	$D_{n,m}^{\mathrm{mod}}$	终端n上传的本地模型数据量
v_n	某一时刻车辆终端n的速度	$T_{\mathrm{upload},n,m}$	上传模型所花费的时间
μ	车辆终端n的速度的标准差	$T_{\mathrm{tot},n}(\mathrm{Domain}_i)$	车辆终端n本地训练总时延
r	终端n当前所连边缘节点的覆盖半径	$p_{i,n}$	车辆终端n在域Domain_i内高速移动引起跨域的概率
(x_n,y_n)	车辆终端n的坐标	$T_{\mathrm{ada}}(\mathrm{Domain}_i)$	域Domain_i内半异步机制中每轮自适应聚合的等待时间
(x_m,y_m)	边缘节点m的坐标	$T(\mathrm{Domain}_i)$	域Domain_i内模型训练时延
$l_{n,m}$	车辆终端n到该边缘节点m的水平距离	$K_{i,n}$	域$T_{\mathrm{ada}}(\mathrm{Domain}_i)$内车辆终端的可靠性
$\mathrm{Euclid}_{n,m}$	车辆终端n与边缘节点m之间的欧式距离	$P_{n,m}$	车辆终端n向边缘节点m上传本地模型的无线传输功率
τ_n	车辆n在当前边缘节点覆盖区域内的停留时间	$M_n^m(q)$	车辆终端n上传到边缘节点m的训练任务
$\mathrm{PCI}(\Delta v_n,\theta)$	车辆终端n的潜在的碰撞指数	$E_n(\mathrm{Domain}_i)$	域Domain_i内车辆终端n的本地模型训练能耗
$P_{\mathrm{exit}}^{n,m}$	车辆终端n的中途退出概率	$E_m(\mathrm{Domain}_i)$	域Domain_i内边缘节点m的能耗
ψ_n	车辆终端本地模型信任值的评级	$E(\mathrm{Domain}_i)$	域Domain_i内的总能耗
x_n	车辆终端n的信任值	q_n	车辆终端n的训练任务
ε	信任因子中潜在碰撞指数的权重	q_m	边缘节点m的训练任务
ϖ	信任因子中中途退出概率的权重	$\theta_{i,m}$	边缘节点m覆盖的车辆终端数量占域内所有终端的比例
ζ	信任因子中本地模型信任值评级的权重	$\left\|H_{m,n}^i\right\|$	域Domain_i内被边缘节点m覆盖的终端n的源数据量总量
Domain_i	边缘协作计算域i	$F_i(w)$	训练损失函数
m	边缘节点	MAE	车辆终端的本地模型精度
$\omega_{n,m}$	信道带宽	$J(\mathrm{Domain}_i)$	域Domain_i内的自适应聚合因子
$g_{n,m}$	信道增益	$\boldsymbol{N}_{\mathrm{Domain}_i}$	域Domain_i内的所有车辆终端集合
$p_{n,m}$	传输功率	$\mathrm{VDC}_i(t)$	训练中所产生的有效模型数据总量
σ^2	背景噪声功率	$\mathrm{weight}_{\mathrm{VDC}_i}$	边缘节点进行本地模型区域性聚合的权重

2 高动态车联网业务边缘协作计算框架

本文提出的高动态车联网业务边缘协作计算框架可分为边缘协作计算域构建阶段(阶段1)和联邦学习模型训练阶段(阶段2),如图1所示.

在边缘协作计算域构建阶段,首先依据移动性、计算能力和可靠性等建立了终端服务能力模型,进而采用基于DRL的边缘协作计算域构建算法制定划域决策$\boldsymbol{w}$,并为每个边缘协作计算域选择域首节点和域成员

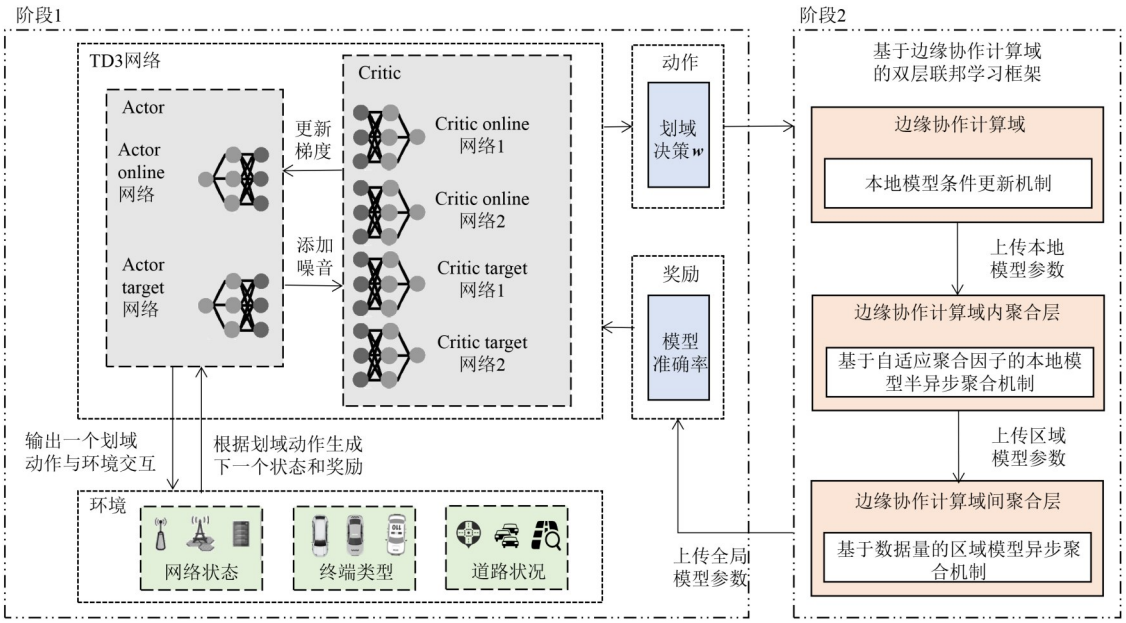

图1　基于双层联邦学习的高动态车联网业务边缘协作计算框架

节点.

域首节点:选择计算能力最强、覆盖范围最广的边缘节点作为域首节点,连接该边缘协作计算域内的域成员节点.

域成员节点:协作域中剩余的边缘节点为域成员节点. 车辆终端将本地模型参数上传至域成员节点,域成员节点将接收到的本地模型进行聚合,并将聚合后的模型参数上传至域首节点.

如图2所示,边缘协作计算域构建阶段的具体工作流程如下:

步骤1.1:中央云服务器选择计算能力最强、覆盖范围最广的边缘节点作为域首节点;

步骤1.2:车辆终端建立终端服务能力模型,并将该模型发送给域首节点;

步骤1.3:域首节点在时延与能耗的约束条件下,根据终端服务能力制定划域决策$\boldsymbol{w}$,确定域成员节点,并将多个边缘节点覆盖范围内的车辆终端聚簇形成边缘协作计算域;

步骤1.4:域首节点将划域决策$\boldsymbol{w}$经由域成员节点下发至车辆终端,完成协作域的构建.

在联邦学习模型训练阶段,构建了包含边缘协作计算域内聚合层和域间聚合层的双层联邦学习框架,分别采用基于自适应聚合因子的本地模型半异步聚合机制和基于数据量的区域模型异步聚合机制,进一步提高联邦学习系统的聚合效率. 如图2所示,该阶段可划分为域内聚合阶段与域间聚合阶段,其中域内聚合阶段的详细步骤如下:

步骤2.1:车辆终端将联邦学习本地模型经由域成员节点上传至域首节点;

步骤2.2:域首节点在接收到本地模型后,综合考虑模型精度与等待时延,设置自适应聚合因子、均衡阈值和时间窗口,将满足聚合要求的终端纳入本轮聚合范围;

步骤2.3:域首节点将更新后的联邦学习区域模型经由域成员节点下发至车辆终端;

步骤2.4:车辆终端和域成员节点进行本地模型的更新.

如图2所示,域间聚合阶段的详细步骤如下:

步骤3.1:在边缘协作计算域的域间聚合层,针对跨域切换的车辆终端,进行本地模型的部分条件更新;

步骤3.2:车辆终端将更新后的本地模型经由域成员节点上传至域首节点;

步骤3.3:域首节点将区域模型上传至中央云服务器;

步骤3.4:中央云服务器采用基于数据量的联邦模型异步聚合机制,对收集到的各区域模型参数进行异步加权聚合;

步骤3.5:中央云服务器将更新后的联邦学习全局模型下发至域首节点;

步骤3.6:域首节点在接收到新的全局模型后进行区域模型更新;

步骤3.7:域首节点将更新后的区域模型经由域成员节点下发至车辆终端;

步骤3.8:域成员节点和车辆终端在接收到区域模型后,进行本地模型的更新.

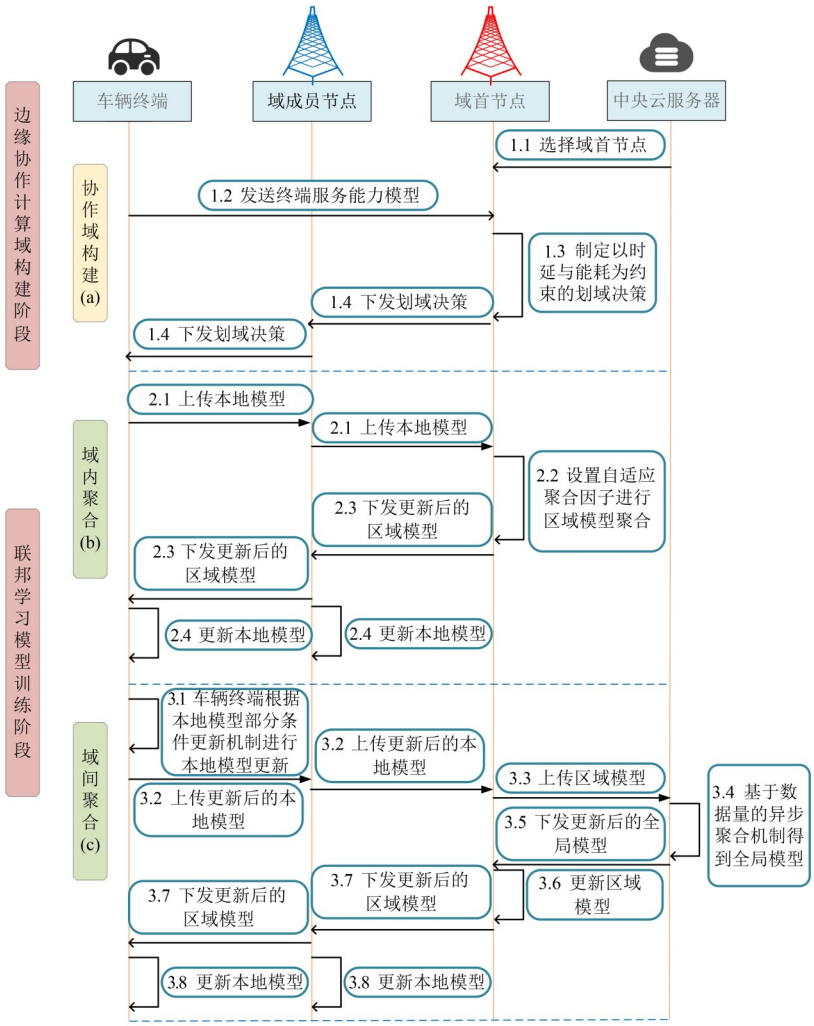

图2 面向高动态车联网业务的边缘协作计算服务流程

3 基于DRL的边缘协作计算域构建算法

在高动态车联网中,车辆终端往往具有高动态性,包括较高的移动速度以及复杂的行驶轨迹. 在高速移动过程中,终端会在多边缘节点间切换造成训练中断. 因此,本文采用划分边缘协作计算域的方式,根据终端服务能力将多个边缘节点覆盖范围内的车辆终端聚簇形成边缘协作计算域,提高模型训练的连续性和准确性. 面向高速移动车联网业务的边缘协作计算域构建方法如图3所示.

3.1 终端服务能力模型

根据车辆终端n在时隙t的移动速度$V_n(t)$、计算能力$C_n(t)$和可靠性$K_n(t)$,定义终端服务能力模型$\boldsymbol{S}_n(t)$,表示为

$$\boldsymbol{S}_n(t)=\{V_n(t),C_n(t),K_n(t)\} \tag{1}$$

假设车辆终端n的移动速度$V_n(t)$遵循高斯分布且满足独立同分布,可通过终端速度与停留时间来衡量终端的移动性. 假设某一时刻车辆终端n的速度为v_n,$v_n^{\max}$与$v_n^{\min}$分别表示v_n的最大值与最小值,则v_n的概率分布函数可定义为

$$f(v_n)=\begin{cases}\dfrac{2\mathrm{e}^{\frac{-(v_n-\bar{v})^2}{2\mu^2}}}{\mu\sqrt{2\pi}\left(\mathrm{erf}\left(\dfrac{v^{\max}-\bar{v}}{\mu\sqrt{2}}\right)-\mathrm{erf}\left(\dfrac{v^{\min}-\bar{v}}{\mu\sqrt{2}}\right)\right)}, & v^{\min}\leqslant v_n\leqslant v^{\max}\\ 0, & \text{otherwise}\end{cases} \tag{2}$$

其中,$\bar{v}$为车辆终端移动的平均速度;μ为车辆速度的标准差. 假设终端n当前所连边缘节点的覆盖半径为r,车辆终端n的坐标为(x_n,y_n),边缘节点m的坐标为(x_m,y_m),车辆终端n到该边缘节点m的水平距离为$l_{n,m}=\sqrt{(x_n-x_m)^2+(y_n-y_m)^2}$,则车辆$n$在该边缘节点

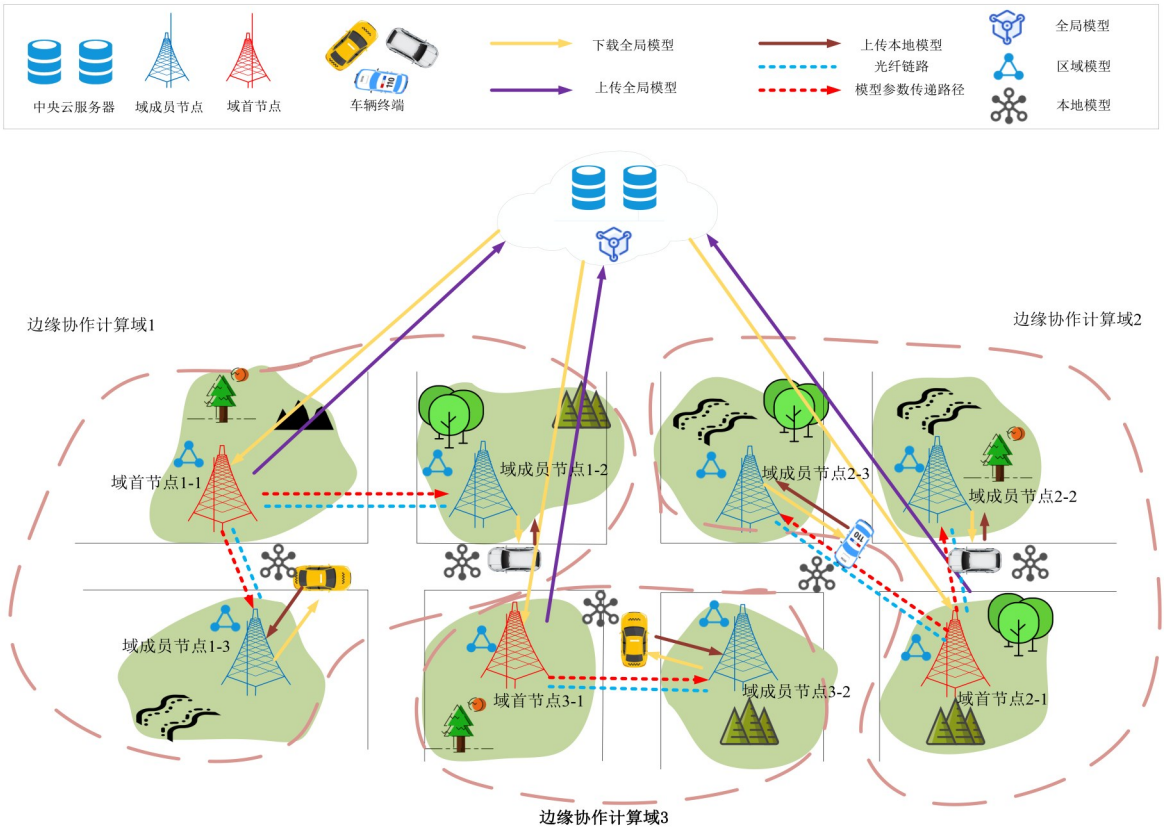

图 3　面向高速移动车联网业务的边缘协作计算域构建方法

覆盖区域内的停留时间为

$$\tau_n=\left(2r-l_{n,m}\right)/v_n \tag{3}$$

3.2　可靠性模型

综合利用碰撞指数、退出概率以及信任评级等参数[19],设计信任因子$K_n(t)$用以评估车辆终端n的可靠性,$K_n(t)$定义为

$$K_n(t)=\varepsilon\left(\mathrm{PCI}\left(\Delta v_n,\theta\right)\right)+\varpi P_{\mathrm{exit}}^{n,m}+\zeta\psi_n \tag{4}$$

其中,$\mathrm{PCI}\left(\Delta v_n,\theta\right)$表示车辆终端$n$的潜在碰撞指数,$\mathrm{PCI}\left(\Delta v_n,\theta\right)=\mathrm{PCI}\left(\Delta v_n\right)\mathrm{PCI}(\theta),\mathrm{PCI}\left(\Delta v_n\right)=\dfrac{\Delta v_n}{D}$,其中,$\Delta v_n$是碰撞前后的车速变化,$D$表示障碍车辆与当前车辆$n$之间的距离,$\mathrm{PCI}(\theta)$表示车辆终端的航向角值[20];$P_{\mathrm{exit}}^{n,m}$表示车辆终端$n$的中途退出概率,但由于退出原因与退出概率之间的相关性非常复杂,超出了本文研究的范围,因此假设车辆终端n的退出概率遵循正态分布$P_{\mathrm{exit}}^{n,m}\sim\mathrm{N}\left(\mu,\sigma^2\right)$,$\psi_n$表示车辆终端本地模型信任值的评级,定义为

$$\psi_n=\frac{\sum\limits_{n\in\mathrm{Domain}_i}x_n}{\sum\limits_{n\in\mathrm{Domain}_i}x_n^m} \tag{5}$$

其中,x_n是车辆终端n的信任值,可通过车辆终端历史训练数据获得;ε,ϖ和ζ分别表示3个参数所对应的权重,且$\varepsilon+\varpi+\zeta=1$.

3.3　时延模型

(1)终端本地训练时延

对于边缘协作计算域Domain_i,车辆终端n完成一轮联邦学习模型训练过程的总时延包括初始模型下载时延、本地模型训练时延和本地模型上传时延. 当车辆终端n以速度v_n行驶时,可通过香农公式计算车辆终端n与边缘节点m之间的数据传输速率:

$$R_{n,m}=\omega_{n,m}\log_2\left\{1+\frac{p_{n,m}g_{n,m}\left(\mathrm{Euclid}_{n,m}\right)^{-\theta}}{\sigma^2}\right\} \tag{6}$$

$$\mathrm{Euclid}_{n,m}=\sqrt{{h_m}^2+\left(r-\mathrm{loc}_n-v_nt\right)^2} \tag{7}$$

其中,$\omega_{n,m}$和$g_{n,m}$分别为车辆终端n与边缘节点m之间的信道带宽和信道增益;$p_{n,m}$表示传输功率;σ^2表示背景噪声功率[8];$\mathrm{Euclid}_{n,m}$为车辆终端n与边缘节点m之间的欧式距离;h_m为边缘节点m的高度;loc_n为车辆n的初始位置.

假设车辆终端n从边缘节点m下载的初始模型数据量大小为$D_{m,n}^{\mathrm{initmod}}$,则下载初始模型所花费的时间开销为

$$T_{\mathrm{download},m,n}=\frac{D_{m,n}^{\mathrm{initmod}}}{R_{n,m}} \tag{8}$$

假设X_{model}为在单位源数据上训练模型所需的CPU周期数，$D_{m,n}^{\text{local}}$为终端本地存储的模型数据量，$C_n(t)$为车辆终端n的计算能力，因此，本地训练模型的时延为

$$T_{\text{train},n}=\frac{D_{m,n}^{\text{local}}X_{\text{model}}}{C_n(t)} \tag{9}$$

假设车辆终端n上传的本地模型数据量为$D_{n,m}^{\text{mod}}$，则上传模型所花费的时间为

$$T_{\text{upload},n,m}=\frac{D_{n,m}^{\text{mod}}}{R_{n,m}} \tag{10}$$

综上分析，车辆终端n本地训练总时延$T_{\text{tot},n}\left(\text{Domain}_i\right)$可表示为

$$T_{\text{tot},n}\left(\text{Domain}_i\right)=\frac{D_{m,n}^{\text{initmod}}}{R_{n,m}}+\frac{D_{m,n}^{\text{initmod}}X_{\text{model}}}{C_n(t)}+\frac{D_{n,m}^{\text{mod}}}{R_{n,m}} \tag{11}$$

（2）域内模型训练时延

对于一个边缘协作计算域Domain_i，其域内模型训练时延包括车辆终端本地模型训练时延、边缘协作计算域间切换时延与域内聚合层半异步聚合时延，因此域内模型训练时延可表示为

$$\begin{aligned}T\left(\text{Domain}_i\right)=&\sum_{i=1}^{\text{Domain}}K_{i,n}T_{\text{tot},n}\left(\text{Domain}_i\right)\\&+\sum_i p_{i,n}T_{\text{tot},n}\left(\text{Domain}_i\right)+T_{\text{ada}}\left(\text{Domain}_i\right)\end{aligned} \tag{12}$$

其中，$T_{\text{tot},n}\left(\text{Domain}_i\right)$为车辆终端$n$的本地模型训练时延；$K_{i,n}$代表域$\text{Domain}_i$内车辆终端的可靠性；$p_{i,n}$为车辆终端$n$在域$\text{Domain}_i$内高速移动引起跨域的概率；$T_{\text{ada}}\left(\text{Domain}_i\right)$为半异步机制中每轮自适应聚合的等待时间.

3.4 能耗模型

（1）车辆终端本地模型训练能耗

对于一个边缘协作计算域Domain_i，其域内所有车辆终端完成训练的能耗包括车辆终端n的本地模型训练能耗、边缘节点m的模型聚合能耗以及车辆终端在边缘协作域间切换的能耗.

定义$P_{n,m}$为车辆终端n向边缘节点m上传本地模型的无线传输功率，q_n代表车辆终端n的训练任务，$M_n^m\left(q_n\right)$为车辆终端n上传到边缘节点m的训练任务，$R_{n,m}$为车辆终端n与边缘节点m之间的数据传输速率.因此，车辆终端n的本地模型训练能耗可表示为

$$E_n\left(\text{Domain}_i\right)=\frac{P_{n,m}M_n^m\left(q_n\right)}{R_{n,m}} \tag{13}$$

（2）边缘节点能耗

边缘节点的能耗为该节点处理所有计算任务q_m的总能耗，可由边缘节点m的计算能力以及所有计算任务q_m所需的CPU周期$f_m\left(q_m\right)$进行衡量，表示为

$$E_m\left(\text{Domain}_i\right)=\pi\left(f_m\left(q_m\right)\right)^3 \tag{14}$$

其中，π为与芯片结构相关的有效开关电容[21].

由于边缘协作计算域的能耗是域内所有边缘节点的能耗之和，因此边缘协作计算域Domain_i的能耗$E\left(\text{Domain}_i\right)$可表示为

$$\begin{aligned}E\left(\text{Domain}_i\right)=&\sum_{i=1}^{\text{Domain}}K_{i,n}E_n\left(\text{Domain}_i\right)\\&+\sum_{i=1}^{m}\theta_{i,m}\pi\left(f_m(q_m)\right)^3\\&+\sum_i p_{i,n}E_n\left(\text{Domain}_i\right)\end{aligned} \tag{15}$$

其中，$E_n\left(\text{Domain}_i\right)$为车辆终端$n$本地模型的训练能耗；$K_{i,n}$代表域$\text{Domain}_i$内车辆终端的可靠性；$\theta_{i,m}$表示边缘节点$m$覆盖的车辆终端数量占域内所有终端的比例；$p_{i,n}$为车辆终端$n$在域$\text{Domain}_i$内高速移动引起跨域的概率.

3.5 基于DRL的联邦学习节点选择算法

面向高速移动车联网业务，根据所提出的终端服务能力模型，设计一种基于DRL的联邦学习节点选择算法，以实现边缘协作计算域的精准划分. 该算法利用测试数据集的损失函数来表示车辆终端在一轮联邦学习中的训练效果，并通过制定最优的车辆终端选择决策进而提高联邦学习模型的准确性. 在时隙t中引入$\boldsymbol{\varphi}^t=\left[\varphi_n^t\right]$作为车辆终端状态的指示向量，$\varphi_n^t=1$代表被选择，反之，$\varphi_n^t=0$代表未被选择. 车辆终端选择的最优化问题模型可表示为

$$\min_{\varphi_n^t} L\left(\boldsymbol{x}_{\text{test}},\boldsymbol{y}_{\text{train}};\boldsymbol{w}\right) \tag{16}$$

s. t.

$$\text{C1: }\varphi_n^t=\{0,1\},\forall n\in\text{Domain}_i \tag{17}$$

$$\text{C2: }\left(L_n^x(t)-L_m^x(t)\right)^2+\left(L_n^y(t)-L_m^y(t)\right)^2\leqslant {r_m}^2 \tag{18}$$

$$\text{C3: }0\leqslant k_n\leqslant 1,\forall n\in\text{Domain}_i \tag{19}$$

$$\text{C4: }E\left(\text{Domain}_i\right)\leqslant E_{\text{req}} \tag{20}$$

$$\text{C5: }T\left(\text{Domain}_i\right)\leqslant T_{\text{req}} \tag{21}$$

$$\text{C6: }\sum_{m=1}^{\text{Domain}}\text{CR}_m(t)\geqslant\text{Com}_{\text{tot}},\forall m\in\mathbf{EN} \tag{22}$$

其中，$L\left(\boldsymbol{x}_{\text{test}},\boldsymbol{y}_{\text{train}};\boldsymbol{w}\right)$表示在划域决策$\boldsymbol{w}$下模型准确率的损失函数；$\boldsymbol{x}_{\text{test}}$表示车辆终端测试集；$\boldsymbol{y}_{\text{train}}$车辆终端训练集. 约束C1表示车辆终端状态的指示向量；约束C2保障选择的车辆终端在边缘节点m的覆盖范围r_m内；约束C3保障被选中的车辆终端信任因子值高于最小阈值；约束C4保障每个边缘协作计算域Domain_i中的能耗不超过每个域规定的最大阈值；约束C5保障域Domain_i中所有车辆终端本地模型训练总时延不超过

域内本地模型训练最大总时延T_{req};约束C6保障边缘节点的计算能力大于边缘协作计算域内所有车辆终端业务的计算需求.

由于以上最优化问题模型(式(16)~(22))属于典型的NP难问题,难以用线性规划解决.强化学习作为NP难问题的解决方案,可以在大规模的状态空间和动作空间下找到最优的车辆终端选择策略π^*[22].强化学习代理通过对相应状态采取一系列动作,使未来累积奖励最大化[23,24].DRL作为强化学习算法与深度学习算法的结合,同时具备了深度学习的感知能力和强化学习的决策能力.深度强化学习算法使用深度学习算法中的多层神经网络对强化学习的Q值空间进行模拟,解决了传统强化学习算法难以处理高维度输入问题的不足[25].因此本文采用DRL算法,将组合优化问题定义为一个马尔可夫决策过程$\boldsymbol{M}=\left(\boldsymbol{S},\boldsymbol{V},P_v,\gamma,\text{Re}\right)$,其中,$\boldsymbol{S}$是状态空间,$\boldsymbol{V}$是动作空间,$P_v$是由动作$v\in\boldsymbol{V}$引起的状态转移概率,$\gamma$表示阻尼系数,Re表示回报函数.

状态空间:在时隙t中,强化学习代理通过观察系统状态,包括车辆终端的移动性、计算能力以及可靠性等指标,制定强化学习策略.状态空间可由终端服务能力模型$\boldsymbol{S}_n(t)=\left\{V_n(t),C_n(t),K_n(t)\right\}$(式(1))表征.

动作空间:在本文中,定义$\left[\varphi_n^t\right]$为动作集,$\boldsymbol{\varphi}^t=\left(\varphi_1^t,\varphi_2^t,\cdots,\varphi_n^t\right)$为车辆终端选择决策,即0-1问题决策,$\varphi_n^t=1$代表被选择,反之,$\varphi_n^t=0$代表未被选择.

奖励函数:在获得状态和动作空间后,代理将在状态s_t获得相应奖励reward_t.本文基于联邦学习的准确率设计奖励函数,并设置边缘节点计算能力、车辆终端信任因子和最大时延作为每步动作选择的约束.因此,奖励函数可表示为

$$\text{reward}_t=\frac{-1}{\sum\limits_{n\in\text{Domain}}\varphi_n^t}L\left(\boldsymbol{x}_{\text{test}},\boldsymbol{y}_{\text{train}}:\boldsymbol{w}\right)\tag{23}$$

由于深度确定性策略梯度(Deep Deterministic Policy Gradient,DDPG)算法采用固定策略求解最佳动作,易陷入局部最优,而双延迟深度确定性策略梯度(Twin Delayed Deep Deterministic policy gradient,TD3)算法采用带有高斯噪声的确定性策略,可在一定程度上避免过拟合问题.同时,引入延迟的策略更新,可达到以较低的频率更新动作网络并以较高的频率更新评价网络的目的.

综上分析,本文采用TD3算法,并定义目标策略为

$$\mu_{\theta'}(s')=\text{clip}\left(\mu_{\theta'}(s')+\text{clip}\left(\epsilon,-c,c\right),a_{\text{low}},a_{\text{high}}\right)\tag{24}$$

其中,$\epsilon\sim\text{N}(0,\sigma)$.同时,TD3算法选择两个Target Critic网络输出的最小值作为Q值,作为目标值的计算:

$$y(s,a)=r(s,a)+\gamma\min(Q_{\phi_1'}\left(s',\mu_{\theta'}(s'),Q_{\phi_2'}\left(s',\mu_{\theta'}(s')\right)\right)\tag{25}$$

选取Q值较低的节点选择决策实施节点初始划分.同时,定义损失函数$F_i(\boldsymbol{w})$为所有终端在样本数据集上的预测值与实际值之间存在的差异[26],用于不断优化节点选择决策,$F_i(\boldsymbol{w})$可表示为

$$F_i(\boldsymbol{w})=\frac{\sum\limits_{(x_i,y_i)\in\text{Domain}_i}f_i\left(\left(x_i,y_i\right),\boldsymbol{w}\right)}{\left|H_{m,n}^i\right|}\tag{26}$$

其中,x_i是在域Domain_i内车辆终端在样本数据集上的实际值;y_i为域Domain_i内车辆终端在样本数据集上的预测值;$\left|H_{m,n}^i\right|$表示域Domain_i内被边缘节点m覆盖的终端n的源数据总量.

联邦学习节点选择算法的目标即为求出可以使$F_i(\boldsymbol{w})$最小的划域决策$\boldsymbol{w}^*$,从而提高全局模型的准确性,即

$$\boldsymbol{w}^*=\text{argmin}_w F_i(\boldsymbol{w})\tag{27}$$

综上所述,基于TD3的联邦学习节点选择算法如算法1所示.

算法1　基于TD3的联邦学习节点选择算法

输入:本地训练轮数t; 车辆终端用于训练的有效数据集大小D_{mn}; 边缘节点集合EN; 所有节点的簇索引集Index={−1, −1, ⋯, −1}; 协作域节点数量上限Z; 起始终端节点间距ϑ_1

输出:终端节点选择策略$\boldsymbol{X}_n$

(1) 初始化6个网络:Q_{ϕ_1}, Q_{ϕ_2},$Q_{\phi_1'}$, $Q_{\phi_2'}$, $\mu_{\theta'}$, μ_{θ},令$\phi_1'\leftarrow\phi_1$, $\phi_2'\leftarrow\phi_2$,$\theta'\leftarrow\theta$

(2) **For** big_episode=1 **do**

(3)　观察当前状态,使用$\mu_\theta(s)$采集一个状态-动作-回报对$\{s,a,r,s'\}$,并把该样本加入经验池$\mathcal{B}$

(4)　**For** small_episode=1 **do**

(5)　　从$\mathcal{B}$中采集一个batch, $\{s,a,r,s'\}$, $\text{small}_{\text{episode}}\leftarrow\text{small}_{\text{episode}}+1$;

(6)　　根据式(25), 同时使用Q_{ϕ_1}, Q_{ϕ_2}, 计算target

(7)　　更新两个Q网络

(8)　**End for**

(9)　更新策略网络μ_θ, 只使用Q_{ϕ_1}

(10)　更新ϕ_1', ϕ_2', θ'

(11)　$\pi_{\text{old}}\leftarrow\pi$

(12)　$F_i(\boldsymbol{w})=\dfrac{\sum\limits_{(x_i,y_i)\in\text{Domain}_i}f_i\left(\left(x_i,y_i\right),\boldsymbol{w}\right)}{\left|H_{m,n}^i\right|}$

(13)　$\boldsymbol{w}^*=\text{argmin}_w F_i(\boldsymbol{w})$

(14) **End for**

(15) **Return** $\boldsymbol{X}_n$

4 基于边缘协作计算域的双层联邦学习模型

本文采用双层联邦学习模型，在域内实现车辆终端本地模型的半异步聚合，在域间实现边缘节点区域模型的异步聚合. 如图4所示，在域内聚合层，域成员和域首节点接收到边缘协作域内车辆终端上传的模型参数后，采用基于自适应聚合因子的本地模型半异步机制进行聚合，综合考虑了不同终端的等待时间与模型精度，通过动态调整参与每轮域内模型聚合的终端规模，提高了域内本地模型聚合效率；在域间聚合层，中央云服务器接收到域首成员上传的区域模型参数后，采用基于数据量的区域模型异步聚合机制，提高了全局模型的准确性. 特别地，考虑到车辆终端在高速移动时的跨域切换问题，引入本地模型部分条件更新机制，在跨域过程中实时评估模型质量并对精度差的业务模型块进行替换，以提高模型训练的准确性. 具体方案如下.

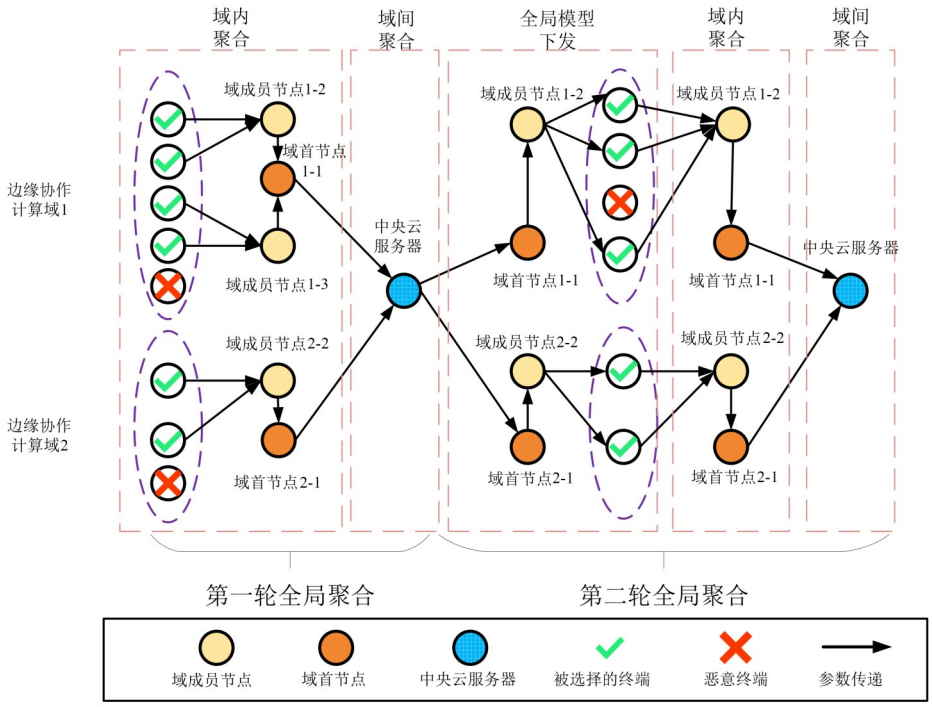

图4 双层联邦学习架构模型聚合过程

4.1 基于自适应聚合因子的本地模型半异步聚合机制

在协作域内模型聚合过程中，由于采用同步联邦学习会造成系统聚合效率低，而采用异步联邦学习会造成系统资源效用差等问题，因此，本文综合考虑车辆终端等待时间与模型精度，设计自适应聚合因子，并提出联邦模型半异步聚合机制，均衡本地模型聚合频率. 车辆终端的本地模型精度可用绝对误差平均值(Mean Absolute Error，MAE)表征，MAE越低的模型精度更高. MAE可表示为

$$\mathrm{MAE}=\frac{1}{|d_i|}\sum_{1}^{|d_i|}\left|y_i-w_n\left(x_n\right)\right| \tag{28}$$

其中，d_i为域Domain_i内所覆盖的所有源数据量；x_n表示车辆终端初始模型精度；y_i表示域Domain_i内的区域模型精度；w_n表示车辆终端n的源数据量在域Domain_i内所有源数据总量中的占比.

由于每个车辆终端的MAE存在差异，因此本文采用基于分布式随机八卦的车辆终端本地模型更新机制[27]. 其基本原理为：车辆终端n在接收到其他车辆终端的噪音模型后更新本地模型，并计算出本地模型的MAE；然后，车辆终端n将自身的噪音与MAE广播给其他车辆，并进行下一轮模型训练和更新. 该算法可防止MAE过高的恶意终端不断接收全局模型，降低了数据泄露的风险，具体如算法2所示.

考虑到车联网中车辆终端的训练速度不同，因此各终端完成本地模型训练的周期以及模型的精度存在差异性. 为了均衡等待时间和模型精度，设计了自适应聚合因子$J\left(\mathrm{Domain}_i\right)$，同时设置均衡阈值，并在时间窗口内，监控模型训练总时延与模型精度，将满足聚合要求的终端纳入本轮聚合范围，提高域间聚合层的聚合效率. 由式(11)可知，车辆终端n的本地训练总时延为$T_{\mathrm{tot},n}$，因此可定义自适应聚合因子$J\left(\mathrm{Domain}_i\right)$为

$$J\left(\mathrm{Domain}_i\right)=\rho T_{\mathrm{tot},n}+\xi\mathrm{MAE} \tag{29}$$

其中，权重ξ和ρ分别表征模型精度和时延对聚合因子的影响.

算法2　基于分布式随机八卦的车辆终端本地模型更新机制

输入:本地模型集 set$\{m_i(t)\}$

输出:新的本地模型集$\{m_i(t+1)\}$

(1) **For** 每个终端$n \in \boldsymbol{N}_{\text{Domain}_i}$ **do**

(2)　**if** $t \leqslant T$ **then**

(3)　　终端n接收到其他车辆终端发送的噪音模型$\tilde{w}_i(t)$

(4)　　终端n训练新的本地模型$\{m_i(t+1)\}=\{m_i(t)\}+\tilde{w}_i(t)$

(5)　　终端n计算本地模型的$\text{MAE}=\dfrac{1}{|d_i|}\sum_{1}^{|d_i|}\left|y_i-w_j(x_i)\right|$,并根据MAE对模型质量进行评估

(6)　　终端n更新自身噪音模型$\tilde{w}_i(t+1)=\tilde{w}_i(t)+\text{Noise}$

(7)　　终端n将新的噪音模型$\tilde{w}_i(t+1)$和MAE广播给其他车辆终端

(8)　　其他车辆终端收到噪音模型后返回步骤3)

(9)　　当车辆终端n的误差低于阈值时,停止模型更新

(10)　结束训练

(11)　**End if**

(12) **End for**

(13) **Return** $\{m_i(t+1)\}$

4.2　面向跨域切换的车辆终端本地模型部分条件更新机制

在边缘协作计算域的域间聚合层,考虑到终端高速移动引起的跨域问题,引入了本地模型部分条件更新机制. 当车辆终端从当前区域行驶到下一区域时,相邻区域的终端业务模型块精度可能存在差异. 同时,考虑到将跨域车辆终端n的全部本地模型参数进行替换会引起较大的计算与通信开销,因此,引入了本地模型部分条件更新机制,根据用途将车辆终端n的本地模型$\boldsymbol{m}_i(n)$划分为通用模型块$\boldsymbol{G}(n)$和业务模型块$\boldsymbol{S}(n)$. 通用模型块包含车辆终端用户特征参数,与车辆终端的具体业务无关,因此参数具有可复用性;业务模型块指的是由业务相关参数形成的模型块,与用户本地数据关联性较强,因此参数具有独特性. 在模型进行更新时,应比较不同业务模型块的MAE,组合精度更高的业务模块,构建更优的本地模型,降低资源开销以及提高模型训练精度. 综上分析,面向跨域切换的车辆终端本地模型部分条件更新机制如算法3所示.

4.3　基于数据量的区域模型异步聚合机制

在联邦学习域间聚合层中,本文根据域Domain_i内域首节点汇聚的有效模型数据总量设置聚合权重$\text{weight}_{\text{VDC}_i}$. 有效模型数据总量指的是,除去恶意节点外,单个域首节点所覆盖的源数据总量. 假设每一轮区域模型训练中,每个区域的有效数据覆盖值(Valid Data Coverage,VDC)为

$$\text{VDC}_i(t)=\sum_{m\in\text{Domain}_i}\left|D_m\right| \tag{30}$$

其中,D_m表示在该域Domain_i内边缘节点m所覆盖范围内的所有终端的有效本地模型数据总量;$\text{VDC}_i(t)$表示域Domain_i内所有边缘节点完成一轮训练中所产生的有效模型数据总量.

在每个边缘节点进行本地模型区域性聚合的过程中,需根据每个协作域内所有边缘节点完成一轮训练中所产生的有效模型数据总量设置聚合权重:

$$\text{weight}_{\text{VDC}_i}(t)=\sum_{r\in i}\frac{\text{VDC}_r(t)}{\text{VDC}(t)} \tag{31}$$

进而,中央云服务器在域间聚合层对采集到的各区域模型参数进行异步加权聚合,用以提高全局模型的准确性. 当中央云服务器完成区域模型聚合后,将更新后的全局模型分别下发至域首节点. 该异步聚合机制与同步聚合方式相比,在保障模型精度的条件下,降低了聚合等待时延,提升了全局模型聚合速度.

算法3　面向跨域切换的车辆终端本地模型部分条件更新机制

输入:当前训练模型$\boldsymbol{m}_i(n)$;下一区域模型$\{\boldsymbol{m}_{i+1}(n)\}$;通用模型块$\boldsymbol{T}(n)$;业务模型块$\boldsymbol{S}(n)$

输出:部分条件更新后的本地模型$\{\boldsymbol{U}_i(n)\}$

(1) **For** 车辆终端用户$n \in \boldsymbol{N}_{\text{Domain}_i}$ **do**

(2)　车辆终端n训练当前本地模型$\boldsymbol{m}_i(n)\triangleq\begin{bmatrix}\boldsymbol{T}(n)\\ \vdots\\ \boldsymbol{S}_i(n)\end{bmatrix}_{N_s\times N_d}$

(3)　车辆终端n进入下一区域Domain_{i+1}并收到下一区域的模型$\boldsymbol{m}_{i+1}(n)\triangleq\begin{bmatrix}\boldsymbol{T}(n)\\ \vdots\\ \boldsymbol{S}_{i+1}(n)\end{bmatrix}_{N_s\times N_d}$

(4)　车辆终端n根据式(28)计算业务模型块$\boldsymbol{S}_i(n)$与业务模型块$\boldsymbol{S}_{i+1}(n)$的MAE,评估业务模型块质量

(5)　选取MAE较低的业务模型块与车辆终端n的通用模型块$\boldsymbol{T}(n)$进行组合,实现本地模型的动态更新,

$$\boldsymbol{m}_i(n)\triangleq\begin{bmatrix}\boldsymbol{T}(n)\\ \vdots\\ \min(\text{MAE}_{\{\boldsymbol{S}_i(n)\}},\text{MAE}_{\{\boldsymbol{S}_{i+1}(n)\}})\end{bmatrix}_{N_s\times N_d}$$

(6)　车辆终端加入下一区域Domain_{i+1}进行训练

(7) **End for**

(8) **Return** $\{\boldsymbol{U}_i(n)\}$

5　实验结果

5.1　参数设置与对比算法

在Python 3.8和TensorFlow 2.3.1环境下对算法进行仿真验证. 实验模拟了车联网环境下车载终端分布式联邦学习训练的场景. 在划域的过程中,构建了一个10×10的网格用以表示智能交通系统,每个网格的长度大约为100 m,通过算法划分协作域,在每个区域中设置多个边缘节点进行覆盖. 该实验使用MNIST数据集作为训练数据,应用卷积神经网络作为联邦学习的训

练模型，并设置恶意终端来模拟上传无效或错误模型数据的设备．表2列出了仿真中的相关参数．

表2 仿真参数设置

参数	取值
N_0(噪声功率密度)	−174 dBm/Hz
B(带宽)	100 kHz
η_a(actor学习率)	0.001
η_c(critic学习率)	0.01
D(本地数据集)	[100, 2 000]
恶意终端比例	[10%, 40%]
本地迭代轮数	5
卷积层数	2

本文采用了3种对比算法，将基于划域与不划域的模型训练方法进行了仿真，从模型准确率、训练时延和可靠性等方面对划域必要性和算法有效性进行了验证．对比算法设置如表3所示．

表3 4种算法的工作机制对比

算法名称	划分域	域内边缘节点数	聚合机制
FedSync-ND	否	—	同步聚合
FedAsync-ND	否	—	异步聚合
FedAsync-CD	划分计算域	1	异步聚合
FedSemiAsync-ECCD	划分协作域	多个	自适应异步聚合

(1)FedSync-ND[28]．车联网场景中的同步联邦学习机制，中心云服务器加权聚合本地模型，从而得到全局模型．该算法在联邦学习过程中需要等待所有车辆终端完成本地模型训练，再进行全局模型的聚合和更新．

(2)FedAsync-ND[29]．车联网场景中的异步联邦学习机制，该算法在每次联邦学习的迭代训练中，边缘节点接收到任意车辆终端发送的本地模型，都会触发全局模型的聚合和更新．

(3)FedAsync-CD [6]．车联网场景中的基于计算域的异步联邦学习机制，该算法将单一边缘节点的覆盖范围定义为一个计算域，根据终端的信誉值动态调整计算域内参与训练的终端数量，并异步触发全局模型的聚合和更新．

(4)FedSemiAsync-ECCD(本文所提方法)．本文提出的基于边缘协作计算域的双层联邦学习框架，在域内聚合层采用半异步聚合机制，在域间聚合层采用异步聚合机制，并引入本地模型部分条件更新机制，保障模型的精准可靠训练．

5.2 仿真结果分析

(1)联邦学习全局模型准确率

首先，比较本文所提算法与不划分协作域的同步联邦学习(FedSync-ND)在全局模型准确率上的差异．通过图5可以看出，当单个协作域内存在约20个车辆终端时，全局模型准确率达到95%．但是，随着车辆终端数量的增加，两种算法的准确性均会下降，这是由于车辆终端服务能力具有差异性，因此，当边缘协作计算域中存在过多的车辆终端时将导致全局模型训练精度降低．但是，本文所提方法通过将具有相似可靠性及服务能力的车辆终端聚簇训练模型，可有效提高模型训练的准确率．因此，本文所提算法的全局模型准确率始终高于FedSync-ND．

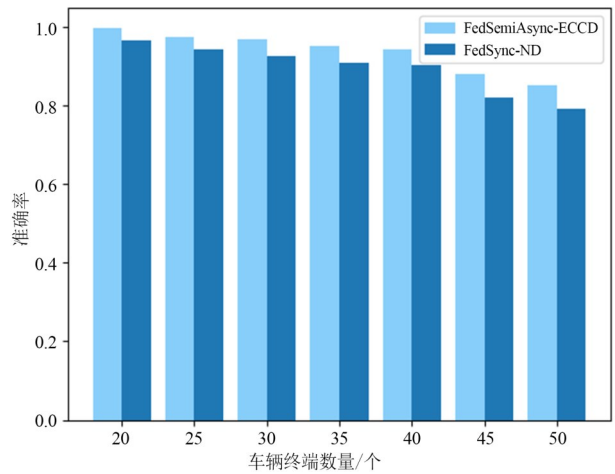

图5 协作计算域对全局模型准确率的影响

(2)联邦学习训练时延

比较本文所提算法与FedSync-ND在本地训练时延上的差异．从图6可以看出，随着车辆数量的增加，本文算法与FedSync-ND的时延均控制在86 ms以下，并且两种方法的时延差异很小且稳定．随着车辆数量的不断增加，本文算法与FedSync-ND的时延均有所提高，这是因为随着车辆终端数量的增多，边缘协作计算域的半异步聚合效率降低．因此，在联邦边缘协作计算域框架中设置合理的自适应聚合因子以控制参与本地聚合的车辆终端规模，可有效平衡系统时延和模型准确性．

(3)恶意终端规模对算法性能的影响

通过实验仿真发现，当边缘节点覆盖范围内恶意终端占比超过40%，本文所提的算法与传统车联网下的联邦学习模型准确率均在65%以下，如图7所示．这是由于恶意终端会上传大量无效或错误的模型数据，当边缘节点覆盖的恶意终端占比过大时，将导致所有方法的训练精度大幅降低，不具有参考意义．

因此，本文首先对比了在边缘节点覆盖范围内恶意终端占比10%时4种算法的全局模型精度．从图8可以看出，本文所提算法的精度最高，收敛速度

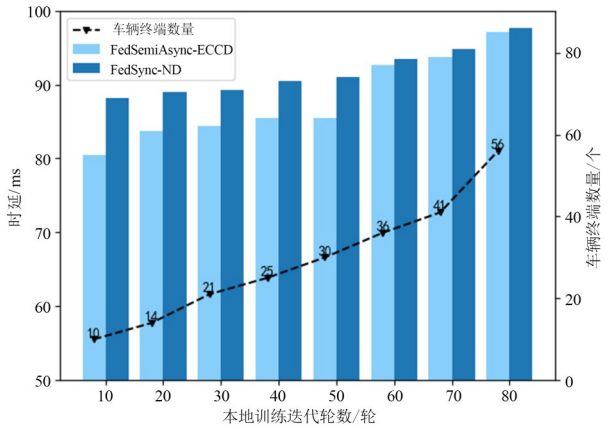

图6　本地训练时延

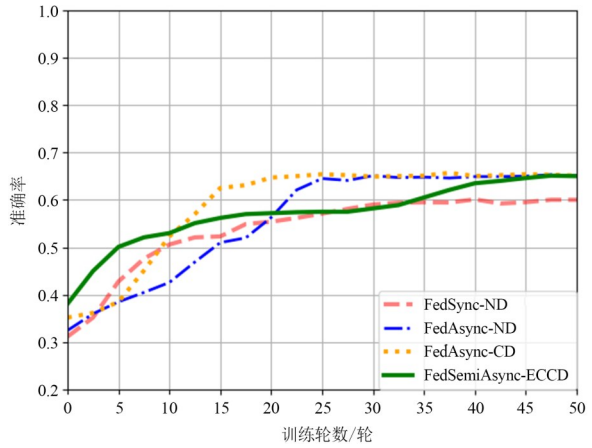

图7　联邦学习模型准确率(恶意终端占总数的40%)

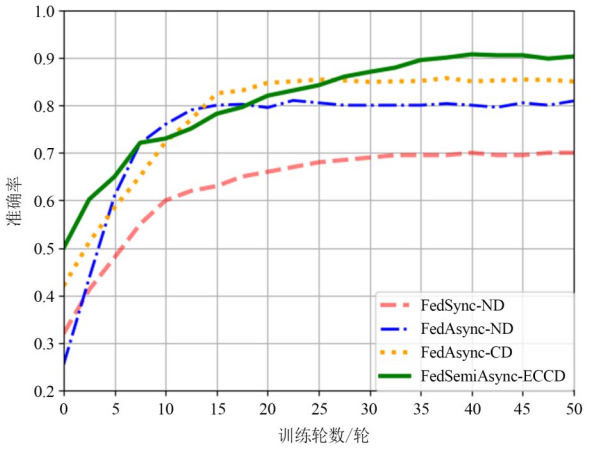

图8　联邦学习模型准确率(恶意终端占总数的10%)

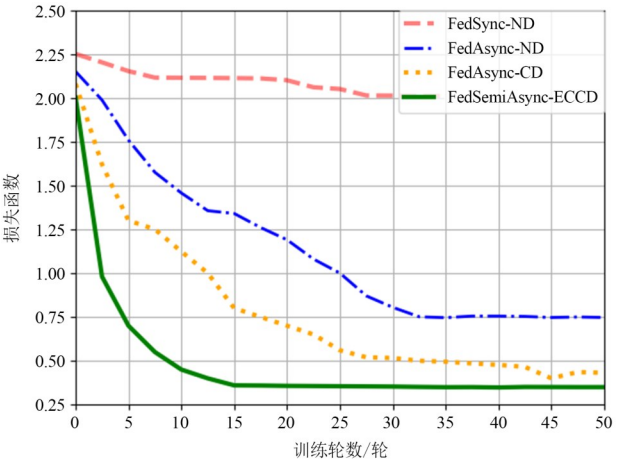

图9　损失函数(恶意终端占总数的10%)

更快. 一方面,由于本文所提方法为车辆终端设置信任因子,用以识别恶意终端. 联邦学习系统选择信任值更高的车辆参与训练,有效避免了低质量模型参与聚合对训练结果的影响. 另一方面,本文所提方法采用半异步聚合机制,提高了联邦学习系统的效率,加快全局模型收敛过程. 而传统的FedSync-ND需要等待所有训练节点完成训练,因此等待时延较长,导致模型收敛速度较慢;FedAsync-ND受到过时的局部模型的影响较大,造成全局模型的精度波动很大.

图9为恶意终端占比为10%时各算法损失函数的变化. 从图中可以看出,本文提出算法的收敛速度最快,损失函数值最小,这是由于本文采用了基于终端服务能力的联邦学习节点选择策略,模型训练准确性大幅提升. 同时,FedSync-ND损失函数最高,这是由于采用随机平均梯度更新法易陷入局部最优. 此外,FedAsync-ND和FedAsync-CD算法的损失函数均高于本文所提算法,其原因是上述算法没有考虑恶意终端上传无效参数对本地模型训练的影响,导致损失函数较高.

接着,考察在边缘节点覆盖范围内恶意终端占比30%时4种算法的全局模型精度. 从图10可以看出,本文所提方法与FedSync-ND、FedAsync-ND和FedAsync-CD相比,在联邦学习模型准确率方面提高了58.7%、17.09%和6.05%. 此外,还可以看出恶意终端的数量对本文所提算法的性能影响较小,这是由于本文所提算法在节点选择的过程中,考虑了节点的计算能力和信任值,有效剔除了恶意节点. 然而,随着恶意终端数量增加,由于FedSync-ND、FedAsync-ND和FedAsync-CD算法没有对恶意终端行为进行限制,因此上述3种算法的模型精度显著降低. 这说明合理的节点选择策略可防止恶意终端参与模型训练,避免了无效或错误数据对模型聚合造成的影响,是提高算法性能的重要手段之一. 并且,通过与图8对比可知,随着恶意终端占比的增加,上述3种算法的准确率与本文所提算法的差异更加明显.

图11为恶意终端数占30%时算法中损失函数的变化. 从图中可以看出,本文所提算法的收敛速度相较于FedSync-ND和FedAsync-CD算法提高了57.6%和10.2%. 这是由于本文设置了自适应聚合因子,可以在

聚合过程中，根据模型精度和等待时间自适应地调整每轮参与聚合的终端规模，在一定程度上提高了模型收敛速度.

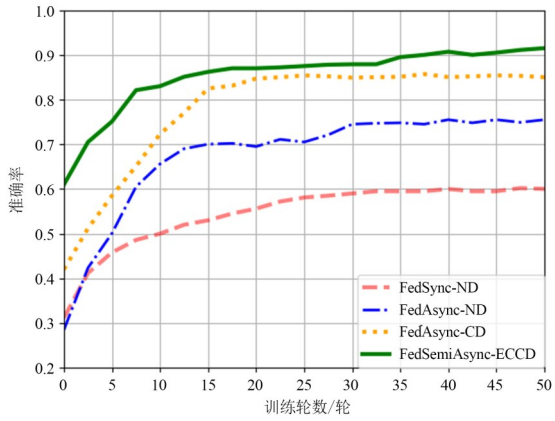

图 10 联邦学习模型准确率(恶意终端占总数的 30%)

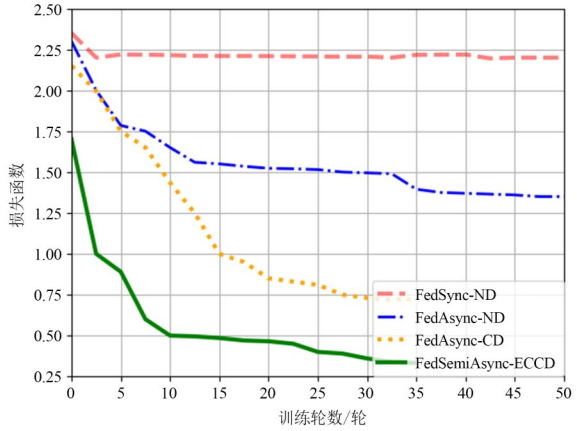

图 11 损失函数(恶意终端占总数的 30%)

6 结束语

针对车联网场景中，车辆终端高速移动引起的训练中断、恶意终端上传无效或错误模型数据导致的联邦学习模型训练精度低和服务可靠性差等问题，本文提出了基于边缘协作计算域的双层联邦学习框架. 首先，综合考虑车辆终端的移动速度、计算能力和可靠性构建了终端服务能力模型，并提出了基于 DRL 的边缘协作计算域构建算法，保障了本地模型训练的连续性. 在边缘协作计算域的基础上，提出了包含边缘协作计算域内聚合层和域间聚合层的双层联邦学习框架，分别采用基于自适应聚合因子的联邦模型半异步聚合机制和基于数据量的联邦模型异步聚合机制，通过均衡本地模型聚合频率和提升全局模型聚合速度进一步提升聚合效率. 仿真验证表明，本文所提方法与 FedSync-ND、FedAsync-ND 和 FedAsync-CD 相比，在联邦学习模型的准确率方面提高了 58.7%、17.09% 和 6.05%，在收敛速度方面与 FedSync-ND 和 FedAsync-CD 相比，提高了 57.6% 和 10.2%.

在未来的研究工作中，课题组将从数据依赖和逻辑依赖等角度，对终端业务模型块进行更细致的划分，以实现精准的个性化模型训练. 此外，面向多业务并发场景的算力资源分配也是联邦学习系统中亟待解决的一个重要问题.

参考文献

[1] GAO Y J, LIU L, HU B X, et al. Federated region-learning for environment sensing in edge computing system[J]. IEEE Transactions on Network Science and Engineering, 2020, 7(4): 2192-2204.

[2] QUANG HIEU N, TRAN T A, NGUYEN C L, et al. Deep reinforcement learning for resource management in blockchain-enabled federated learning network[J]. IEEE Networking Letters, 2022, 4(3): 137-141.

[3] BOTTOU L, CURTIS F E, NOCEDAL J. Optimization methods for large-scale machine learning[J]. SIAM Review, 2018, 60(2): 223-311.

[4] BONAWITZ K, EICHNER H, GRIESKAMP W, et al. Towards federated learning at scale: System design[EB/OL]. (2019)[2023]. http://arxiv.org/abs/1902.01046.pdf.

[5] LU Y L, HUANG X H, DAI Y Y, et al. Blockchain and federated learning for privacy-preserved data sharing in industrial IoT[J]. IEEE Transactions on Industrial Informatics, 2020, 16(6): 4177-4186.

[6] ALOTAIBI J, ALAZZAWI L. PPIoV: A privacy preserving-based framework for IoV- fog environment using federated learning and blockchain[C]//2022 IEEE World AI IoT Congress (AIIoT). Piscataway: IEEE, 2022: 597-603.

[7] AHMED K M, IMTEAJ A, AMINI M H. Federated deep learning for heterogeneous edge computing[C]//2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA). Piscataway: IEEE, 2021: 1146-1152.

[8] CHEN M Z, YANG Z H, SAAD W, et al. A joint learning and communications framework for federated learning over wireless networks[J]. IEEE Transactions on Wireless Communications, 2021, 20(1): 269-283.

[9] YU Z X, HU J, MIN G Y, et al. Mobility-aware proactive edge caching for connected vehicles using federated learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(8): 5341-5351.

[10] MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[EB/OL]. (2016) [2023]. http://arxiv. org/abs/1602.05629.pdf.

[11] SAMARAKOON S, BENNIS M, SAAD W, et al. Feder-

ated learning for ultra-reliable low-latency V2V communications[C]//2018 IEEE Global Communications Conference (GLOBECOM). Piscataway: IEEE, 2018: 1-7.

[12] FEYZMAHDAVIAN H R, AYTEKIN A, JOHANSSON M. An asynchronous mini-batch algorithm for regularized stochastic optimization[J]. IEEE Transactions on Automatic Control, 2016, 61(12): 3740-3754.

[13] LU Y L, HUANG X H, ZHANG K, et al. Blockchain empowered asynchronous federated learning for secure data sharing in Internet of vehicles[J]. IEEE Transactions on Vehicular Technology, 2020, 69(4): 4298-4311.

[14] WANG Z H, XIA G M, CHEN J, et al. Adaptive asynchronous federated learning for edge intelligence[C]//2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE). Piscataway: IEEE, 2021: 285-289.

[15] PAN C, WANG Z, LIAO H J, et al. Asynchronous federated deep reinforcement learning-based URLLC-aware computation offloading in space-assisted vehicular networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(7): 7377-7389.

[16] WU W T, HE L G, LIN W W, et al. SAFA: A semi-asynchronous protocol for fast federated learning with low overhead[J]. IEEE Transactions on Computers, 2021, 70(5): 655-668.

[17] CHEN S, WANG X M, ZHOU P, et al. Heterogeneous semi-asynchronous federated learning in Internet of Things: A multi-armed bandit approach[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2022, 6(5): 1113-1124.

[18] XIAO H Z, ZHAO J, PEI Q Q, et al. Vehicle selection and resource optimization for federated learning in vehicular edge computing[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(8): 11073-11087.

[19] 熊小峰, 黄淳岚, 乐光学, 等. 边缘计算中基于综合信任评价的任务卸载策略[J]. 电子学报, 2022, 50(9): 2134-2145.
XIONG X F, HUANG C L, YUE G X, et al. Task offloading scheme based on comprehensive trust evaluation in edge computing[J]. Acta Electronica Sinica, 2022, 50(9): 2134-2145. (in Chinese)

[20] WANG H, HUANG Y J, KHAJEPOUR A, et al. Local path planning for autonomous vehicles: Crash mitigation [C]//2018 IEEE Intelligent Vehicles Symposium (IV). Piscataway: IEEE, 2018: 1602-1606.

[21] BURD T D, BRODERSEN R W. Processor design for portable systems[J]. Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology, 1996, 13(2): 203-221.

[22] ZHOU Z, CHEN X, LI E, et al. Edge intelligence: Paving the last mile of artificial intelligence with edge computing[J]. Proceedings of the IEEE, 2019, 107(8): 1738-1762.

[23] HE Y, ZHAO N, YIN H X. Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach[J]. IEEE Transactions on Vehicular Technology, 2018, 67(1): 44-55.

[24] ZHANG K, ZHU Y X, LENG S P, et al. Deep learning empowered task offloading for mobile edge computing in urban informatics[J]. IEEE Internet of Things Journal, 2019, 6 (5): 7635-7647.

[25] 刘全, 翟建伟, 章宗长, 等. 深度强化学习综述[J]. 计算机学报, 2018, 41(1): 1-27.
LIU Q, ZHAI J W, ZHANG Z Z, et al. A survey on deep reinforcement learning[J]. Chinese Journal of Computers, 2018, 41(1): 1-27. (in Chinese)

[26] XIN F, ZHANG J H, LUO J Z, et al. Federated learning client selection mechanism under system and data heterogeneity[C]//2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD). Piscataway: IEEE, 2022: 1239-1244.

[27] LU Y L, HUANG X H, DAI Y Y, et al. Differentially private asynchronous federated learning for mobile edge computing in urban informatics[J]. IEEE Transactions on Industrial Informatics, 2020, 16(3): 2134-2143.

[28] KONEČNÝ J, MCMAHAN H B, RAMAGE D, et al. Federated optimization: Distributed machine learning for on-device intelligence[EB/OL]. (2016)[2023]. http://arxiv.org/abs/1610.02527.pdf.

[29] XIE C, KOYEJO S, GUPTA I. Asynchronous federated optimization[EB/OL]. (2019) [2023]. http://arxiv.org/abs/1903.03934.pdf.

作者简介

徐思雅　女,1988年出生,北京人. 现为北京邮电大学计算机学院网络管理研究中心副教授、硕士研究生导师. 主要研究方向为信息通信网络智能管控、联邦学习、文化数字化等.
E-mail: xusiyaxsy@bupt.edu.cn

郭佳惠　女,1999年出生,北京人. 北京邮电大学硕士研究生. 主要研究方向为智能边缘计算.
E-mail: guojiahui@bupt.edu.cn