

无网络环境下基于联邦学习的移动群智感知框架

刘靖孜, 杨桂松*

上海理工大学光电信息与计算机工程学院, 上海

收稿日期: 2024年4月23日; 录用日期: 2024年5月22日; 发布日期: 2024年5月30日

摘要

为解决传统的MCS任务无法在一些偏远或网络不稳定的地区执行, 本文提出了一种新型的模型架构, 专门针对那些无网络环境下的MCS任务执行。基于链路QoS感知建立节点之间数据传输机制, 用于传输联邦学习参数, 同时针对节点间性能以及任务执行时间的差异, 对联邦聚合算法进行优化。实验结果表明, 本文提出的框架与对应机制的组合, 不仅提高了任务执行的效率, 还保障了数据的安全性和模型的性能。

关键词

移动群智感知, 链路质量监测, 模型可信度, 联邦聚合

Mobile Crowdsensing Framework Based on Federation Learning in Netless Environment

Jingzi Liu, Guisong Yang*

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Apr. 23rd, 2024; accepted: May. 22nd, 2024; published: May. 30th, 2024

Abstract

To tackle the issue that traditional MCS tasks cannot be executed in remote areas with unstable networks, this paper proposes a new framework specifically for executing MCS tasks in environments without network connectivity. A data transmission mechanism based on link QoS awareness is established between nodes for transmitting federated learning parameters, and the fede-

*通讯作者。

rated aggregation algorithm is optimized in consideration of the performance differences between nodes and the time taken to execute tasks. Experimental results show that the framework proposed in this paper, combined with the corresponding mechanisms, not only improves the efficiency of task execution but also ensures the security of data and the performance of the model.

Keywords

Mobile Crowdsensing, Link QoS Awareness, Model Reliability, Federated Aggregation

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

移动群智感知(Mobile Crowd Sensing, MCS), 作为物联网的新星, 融合了众包思维与移动感知。它以智能终端为感知单元, 将数据传送至服务器, 挖掘知识以应对大规模任务, 尤其是与人类行为相关的挑战。然而, 随着 MCS 的广泛应用和智能设备数量的爆炸式增长[1], 任务规模扩大、数据量倍增, 给服务器带来了数据传输的压力。

为了解决这些问题, 最新的研究开始探索 MCS 范式转变的可能性。移动终端性能的提升使得深度学习在移动终端的部署成为可能, 针对智能终端的深度学习模型优化方案不断涌现, 使得在移动智能终端上部署分布式训练任务变得可行。在传统的人工智能训练和推理过程中需要共享大量数据, 数据的隐私保护引起了人们的高度重视。出于数据安全和隐私保护方面的考虑, 用户时常不愿进行数据共享, 逐渐形成了数据孤岛[2]。

联邦学习(Federated Learning, FL) [3]是谷歌于 2016 年提出的概念, 主要用于解决数据孤岛问题。其核心思想是分布式深度学习, 从参与者处收集所需参数而非原始数据来训练模型。通过这种方式, 将数据共享问题转化为模型共享问题, 确保参与者的数据不会离开本地, 具有保护隐私、提高通信效率的优点。

移动群智感知任务通常依赖于平台和服务器来执行[4]。然而, 有些情况下, 任务执行者可能会面临脱离平台和服务器的挑战, 这时就需要探讨和实施无网络环境下的移动群智感知任务。

在无网络环境下, 移动群智感知任务的实施面临着一些特殊的挑战。由于缺乏网络连接, 实时数据传输变得不可行, 因此就需要参与者之间更多的本地交互和协调机制。参与者之间的沟通和协作不能依赖于网络平台, 因此组织者需要设计有效的协作流程和通信策略, 确保每位参与者都能够明确自己的角色和任务。

2. 相关工作

MCS 目前是一个重要的研究课题。由于这是一个广泛的研究领域, 因此有许多针对这种情况带来的挑战的贡献。

在 MCS 收集数据方面, 更多的研究关注了收集数据的质量问题, 如 Wan 等人[5]提出了一种基于区块链的移动群智感知数据声誉方案, 可有效降低时延、避免单点故障、保护数据隐私。Mehanna [6]提出了一种检测移动群智感知数据完整性的方法, 通过对数据质量的感知来补充数据的缺失值, 同时依赖聚类算法来检测数据是否存在异常值。而本文的研究重点则关注如何处理收集到的数据。

在联邦学习策略方面, Yang 等人[7]提出了一种无政府联邦学习(AFL), 每个工人都有完全的自由来选择何时和多久参与 FL。此外, 每个工作人员都可以根据其当前的本地情况(例如, 电池水平、通信通道、隐私问题), 独立地确定在每一轮更新中要执行的本地更新步骤的数量。Wu 等人[8]提出了一种自适应联邦学习算法(FAFED), 其对不同层次异构数据下的语言建模任务和图像分类任务表现出良好的性能。Zeng 等人[9]提出了一个联邦学习的轻量化框架(FedLab), 其允许开发者对联邦学习参与者、通信协议等进行定制, 便于用来模拟不同移动群智感知任务参与者的状态。

本文提出的框架着重关注了脱离服务器即无公共网络状态下, 仅依靠任务参与者之间协作进行数据收集与数据处理, 各任务参与者之间的设备性能不同且收集到的数据异构。通过任务参与者之间进行联邦学习, 最终在任务目标达成后直接将参数返回至服务器, 而非将完整数据传输到服务器, 即降低了服务器数据传输的压力, 也保护了各任务参与者的数据隐私。

3. 模型架构

本文探讨了在特殊的 MCS 场景中, 如何有效利用联邦学习方法来构建一个高效的数据处理系统。该系统由一个中心化的 MCS 平台和 n 个参与者组成。在这个系统中, 参与者们会从 MCS 平台获取一个预先训练好的初始全局模型, 并将这个模型带入到各种复杂的环境之中进行实地应用。

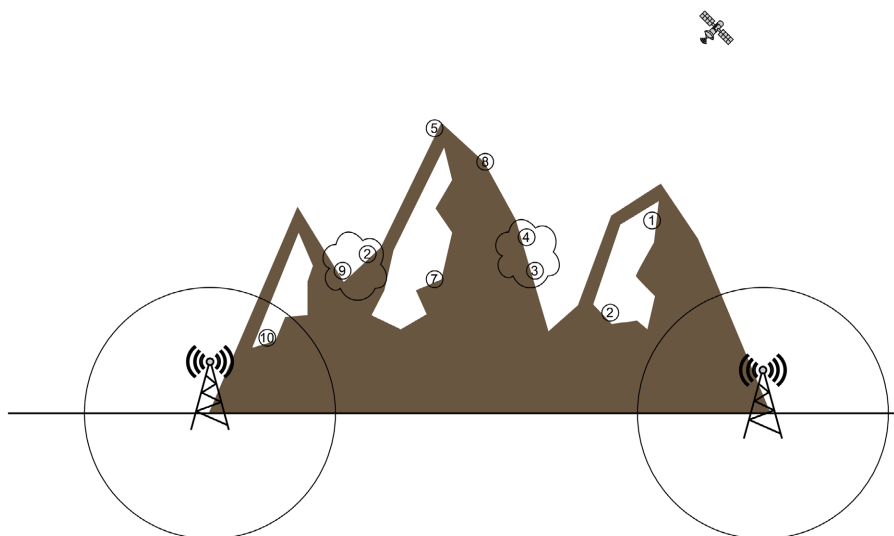


Figure 1. Model diagram
图 1. 模型示意图

在如图 1 所示这样的应用场景下, 参与者们可能会面临无法与中心服务器, 即全局服务器建立连接的挑战。为了解决这一问题, 本文提出了一种基于联邦学习的方法论。在 FL 的框架下, 每个参与者都承担着数据收集、本地模型训练和参数传输等关键任务。具体来说, 参与者首先会在各自的设备上收集数据, 这些数据通常是关于他们所处的环境和可能的感知信息。随后, 参与者之间将利用这些收集到的数据进行横向联邦学习, 所有模型训练均在本地进行, 这一过程不需要与全局服务器进行实时通信, 从而克服了在复杂环境中可能出现的网络连接问题。

每个工人 i 感知到的任务数据集为 D_i , 感知数据在 D_i 中表示为 (x_j, y_j) 。用损失函数 $f_j(w, x_j, y_j)$ 评估本地模型训练的完成度, 值越小表示任务越接近完成, 其中 w 为当前参与者本地模型参数。用 $|D_i|$ 表示参与者 i 的训练数据数量, 参与者 i 在数据集 D_i 上的损失函数定义如下:

$$F_i(w) = \frac{1}{|D_i|} \sum f_j(w, x_j, y_j) \quad (1)$$

当 D_i 以及 $F_i(w)$ 满足下述条件时, 视为任务完成, 参与者返回。

$$\begin{cases} D_i > D_{th} \\ F_i < F_{th} \end{cases} \quad (2)$$

其中 D_{th} 以及 F_{th} 为移动感知平台分配的阈值。

模型中任务执行过程主要包含以下步骤:

- 1) 平台初始化全局模型, 预设 D_{th} 、 F_{th} ;
- 2) 平台向参与者分发全局模型、 D_{th} 、 F_{th} , 参与者接收后开始收集数据;
- 3) 参与者用收集到的本地数据 D_i 对全局模型进行训练, 得到最佳本地模型参数, 如式(3)所示:

$$w_i^{(l)} = \arg \min F_i(w) \quad (3)$$

- 4) 参与者之间使用基于数据集可信度的联邦平均算法交换本地模型参数;
- 5) 重复步骤 3), 4), 直至所有参与者都达到任务完成条件。

4. 基于链路 QoS 感知的数据转发机制

在移动群智感知网络(Mobile Crowdsensing Networks)的研究领域, 一个关键的挑战是如何有效地管理和协调众多移动参与者之间的通信[1]。在这样的网络中, 参与者通常是具有智能设备的人或物体, 他们在执行感知任务的同时也在不断移动。由于这些参与者的移动模式往往是不可预测和随机的, 这就导致了网络中节点之间的通信链路难以保持稳定。本节通过对参与者的空间坐标信息进行建模, 以此来指导数据的动态转发过程, 如图 2 所示。

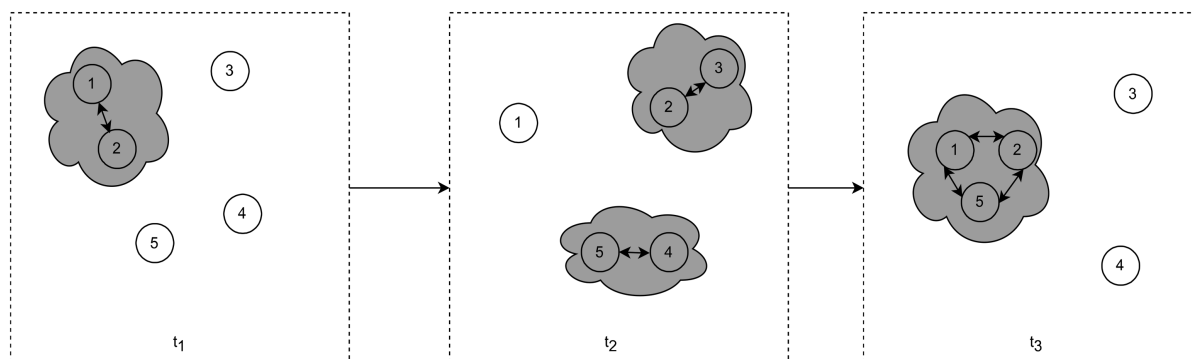


Figure 2. Node movement and link establishment

图 2. 节点移动与链路建立

在 MCS 的任务中, 参与者面临着设备性能和功耗的限制[1], 这意味着他们无法同时完成数据收集和本地训练的任务。当工人所收集的数据量达到预设值后, 他们开始寻找其他工人进行横向联邦学习。为了监测工人之间的链路 QoS, 工人利用心跳机制。心跳机制[10]是一种用于监测网络连接状态的技术。在 MCS 的任务中, 心跳机制被用来监测工人之间的链路 QoS。当链路 QoS 下降到一定水平时, 工人会结束本地训练, 并开始与其他工人进行模型聚合。

工人 1 发送一个仅包含完整报文格式(如图 3 所示)前 3 个字段(各字段释义如表 1 所示)且 SC = 0000 (SC 码对应含义如表 2 所示)的广播报文来寻找其他参与者进行横向联邦学习, 当工人 2 响应时, 向工人

- 1 发送一个不包含 data 字段且 SC = 0001 的响应报文, 工人 1 接收到响应报文后开始本地训练并向工人 2 发送一个不包含 data 字段且 SC = 0010 的响应报文, 工人 2 收到 SC = 0010 的报文后开始本地训练。

ID	timestamp	SC	target ID	timestamp echo	data
----	-----------	----	-----------	----------------	------

Figure 3. Full message format
图 3. 完整报文格式

Table 1. Message field interpretation
表 1. 报文字段释义

字段名	字段释义
ID	报文发送方设备 ID
timestamp	报文发送时主机设备的时间戳
SC	状态码, 用于表示报文的发送目的
target ID	报文目的方设备 ID
timestamp echo	响应报文对应的请求报文的时间戳
data	用于交换联邦学习本地训练的参数

Table 2. SC interpretation
表 2. SC 释义

SC	编码状态
0000	广播请求
0001	响应广播请求
0010	响应确认
0011	状态监测
0100	状态确认
0101	终止本地训练请求
0110	终止本地训练确认
0111	联邦学习参数交换
1000	联邦学习结束

当联邦学习发起方即工人 1 收到 SC = 0010 的响应报文后, 向联邦学习的其他参与方即工人 2 发送 SC = 0011 的状态监测报文, 工人 2 接收到状态监测报文后响应一个 SC = 0100 的状态确认报文。经过时间 t 后, 工人 2 向工人 1 发送状态监测报文, 工人 1 收到后响应一个状态确认报文。随后双方不断交替发送状态监测报文来感知链路 QoS, 如图 4 所示。

正常终止:

当连续 5 次状态监测交互的往返时延(RTT)达到阈值时, 由一方发起终止本地训练请求(SC = 0101), 另一方收到请求后终止本地训练并向终止请求方发送携带相应模型参数数据的终止本地训练确认报文(SC = 0110)。终止请求方接收到终止确认后, 根据报文中携带的数据更新模型参数, 并将更新后的模型参数通过 SC = 0111 的参数交换报文发送给另一方, 另一方在完成模型参数更新后, 向终止请求方发送 SC = 1000 的联邦学习结束报文, 本轮联邦学习结束。

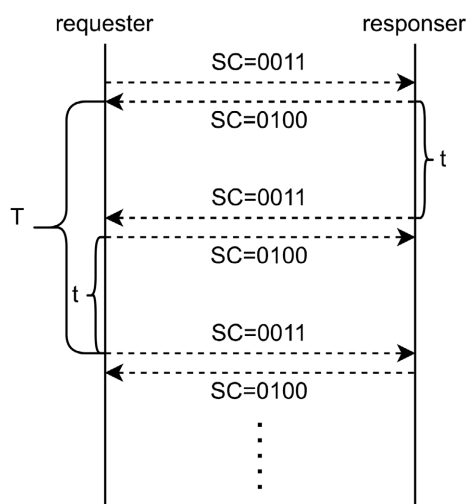


Figure 4. Link QoS monitoring
图 4. 链路 QoS 监测

异常终止:

若一方收到对方的状态确认报文后, 经过时间 $T = 2t$ 仍未收到新的状态监测报文, 则向对方发送一个状态监测报文, 若该监测报文发送后时间间隔 T 内未收到对方响应的状态确认报文, 则视为链路中断, 终止本地训练, 等待下一次联邦学习请求被响应。

5. 基于模型可信度的联邦学习算法

在本文提出的模型框架下, 各个参与者所持有的本地数据集之间存在 Non-IID (非独立同分布) 的特性, 这意味着每个参与者的数据分布可能与其他参与者的数据分布有所不同。此外, 这些本地数据集的大小也各不相同, 这为模型的训练带来了额外的挑战。在这样的背景下, 我们发现了一个关键的现象: 当参与者使用具有 Non-IID 特性的本地数据进行模型训练时, 如果本地迭代的次数较多, 那么本地模型可能会逐渐偏离最初共享的全局模型[11]。这种偏离不仅影响了模型的准确性, 还可能导致模型性能的不稳定。

为了解决这一问题, 本章提出了一种新的方法来评估和调整每个参与者本地模型的权重。我们综合考虑了以下几个关键因素: 首先, 我们考虑了参与者本地数据集的大小, 因为这直接影响了模型训练的数据基础; 其次, 我们关注了本地迭代的次数, 因为迭代次数越多, 模型偏离的风险就越大; 最后, 我们还考虑了本地模型在预设测试集上的交叉熵损失函数, 这是衡量模型预测性能的一个重要指标。

基于上述因素, 需要利用一种可信度评估机制量化每个参与者本地模型的可信度。可信度的高低直接决定了该模型在最终模型聚合过程中的权重。具体来说, 可信度高的模型将在模型聚合中获得更大的权重, 从而对聚合后的模型产生更显著的影响; 可信度低的模型在模型聚合中占据较小的权重, 从而减轻对聚合后模型的影响。

通常, 联邦学习算法可以写作:

$$x^{(r+1,0)} - x^{(r,0)} = \sum_{i=1}^n p_i (x_i^{(r,\tau_i)} - x_i^{(r,0)}) \quad (4)$$

其中, $p_i = |D_i|/|D|$, 即参与者 i 参与模型训练的数据量占所有参与联邦学习总数据量的比例, 基于本文模型框架, 对式(4)改写成:

$$x^{(r+1,0)} - x^{(r,0)} = -\eta \sum_{i=1}^n \omega_i \cdot d_i^{(r)} \quad (5)$$

其中, η 是模型学习率, $d_i^{(t)}$ 是归一化后的梯度, ω_i 是模型聚合时的权重。

1) 归一化梯度 $d_i^{(t)}$:

$$d_i^{(t)} = G_i^{(t)} \cdot a_i / \|a_i\| \quad (6)$$

$G_i^{(t)}$ 表示 t 轮训练中所有的随机梯度[12], 分母中归一化因子 $\|a_i\|$ 是向量 a_i 的 l_1 范数。

2) 聚合权重 ω_i : 当计算聚合梯度 $\sum_{i=1}^n \omega_i \cdot d_i$ 时, 每个参与者的归一化梯度需要乘以聚合权重 ω_i 。所有参与者的聚合权重需要满足 $\sum_{i=1}^n \omega_i = 1$ 。 ω_i 可由式(7)得出:

$$\omega_i = \frac{|D_i| \cdot \lg t_i / f_i^{(t)}}{\sum_{i=1}^n |D_i| \cdot \lg t_i / f_i^{(t)}} \quad (7)$$

其中, $|D_i|$ 表示参与本地训练的数据集大小, t_i 表示本地模型迭代次数, $f_i^{(t)}$ 表示经过 t 轮训练后本地模型的交叉熵损失函数。

6. 实验与数据分析

6.1. RTT 阈值

在对于 RTT (Round-Trip Time, 往返时延)阈值的评估实验中, 我们采取了一些关键的参数设置来确保数据的准确性和可靠性。具体来说, 发送方的发射功率被设定为 23 dBm, 这一数值是为了确保信号的强度能够在合理的范围内, 既不至于过强导致信号干扰, 也不至于过弱使得数据传输不稳定。同时, 为了保证状态监测的连续性和准确性, 将每两次状态监测报文之间的时间间隔设置为 5 秒。这样的时间间隔既能保证监测的实时性, 又能避免频繁的监测导致的资源浪费。在网络通信过程中, 数据包的丢失是一个不可避免的问题。为了应对这一问题, 将超时丢包时长设置为 5 秒。这意味着, 如果在 5 秒内未能成功接收到预期的数据包, 请求方将认为该数据包已丢失。

为了全面评估和选择 RTT(往返时延)阈值, 我们采取了一种综合评判的方法。具体来说就是通过改变节点间的距离, 以及调整单个数据包的大小, 来模拟不同的网络环境和传输条件。在不同的测试场景下, 我们观察并记录 RTT 的变化情况, 以此来确定一个合理的 RTT 阈值。这个阈值将作为网络性能评估的一个重要参考, 用于优化网络配置, 提高通信效率。

Table 3. RTT in different scenarios

表 3. 不同场景下 RTT 数据汇总

序号	距离 (米)	数据包大小 (Bytes)	最大 RTT (ms)	最小 RTT (ms)	RTT 均值 (ms)	RTT 中值 (ms)	RTT 标准差	丢包率
1	1	128	262.249	2.152	11.880	6.766	21.905	0%
2	1	1024	273.516	2.022	12.335	7.146	22.272	0%
3	1	2048	245.496	2.099	13.499	7.496	21.855	0%
4	1	4096	270.034	2.486	13.802	7.898	21.247	0%
5	3	4096	461.280	2.868	11.381	8.332	17.375	0.1%
6	5	4096	470.158	3.090	13.274	8.969	18.286	0.1%
7	7	4096	412.851	3.226	21.792	16.955	23.667	0.7%
8	10	4096	427.716	3.887	30.536	25.540	27.448	1.1%
9	20	4096	1016.077	4.652	45.732	32.656	59.319	17.8%

节点间的距离是影响链路质量的重要因素, 而数据包的大小则直接关系到数据传输的效率。通过改变这两个参数, 进行多组实验(实验结果如表 3 所示), 可以得到不同场景下链路的 QoS, 为模型选择一个合理的 RTT 阈值。

首先观察实验 1 至实验 4 的结果。在这些实验中, 节点之间的距离被固定为 1 米, 以确保环境因素的一致性。在这样的设置下, 我们注意到, 尽管单个数据包的大小在增加, 但是并没有出现丢包的现象。这表明在短距离内, 数据传输是相当稳定的。

进一步分析实验数据可以发现, 随着数据包大小的增加, RTT 的平均值和中值都有所上升, 但这种上升的幅度相对较小。然而, 最大 RTT 和最小 RTT 并没有显示出明显的变化趋势, 这可能表明在节点间距离保持不变的情况下, 数据包大小对网络性能的影响是有限的。此外, RTT 的标准差在所有实验中都保持了相对一致的水平, 这进一步佐证了一个结果, 即在节点间距离不变的情况下, 数据包大小对链路 QoS 的影响可以忽略不计。

接下来观察实验 4 至实验 9。这一次, 通过选择发送更大的数据包, 并且逐渐增加节点之间的距离, 以此来监测报文的 RTT 值。有趣的是, RTT 的平均值和标准差出现了先减小后增大的现象。这一结果提示我们, 当节点间距离过近时, 可能会由于信号干扰或其他因素导致 RTT 的不稳定。

随着节点间距离的继续增加, 最小 RTT 的增加趋势并不显著, 而其他评估指标则呈现出了快速增长的趋势。特别是当节点间距离达到 10 米时, 大数据包的丢包率超过了 1%, 这是一个不可接受的性能下降。这样的结果强调了节点间距离对网络性能的重要性。

综合考虑所有实验数据和观察结果, 我们决定将本文提出的模型的 RTT 阈值设置为 20 毫秒。这个阈值是基于实验数据的全面分析得出的, 旨在确保链路 QoS 在可接受的范围内。

6.2. 模型性能分析

在这一部分实验中, 所有工人将会获得一个全局的初始模型, 这个模型是预训练的 ResNet-50 模型。同时, 我们选择了广泛使用的 CIFAR-10 数据集作为我们的实验模拟数据。为了模拟本文提出的模型架构, CIFAR-10 中的数据被映射到一个 100×100 的坐标区域中, 并确保所有数据在区域内均匀分布。

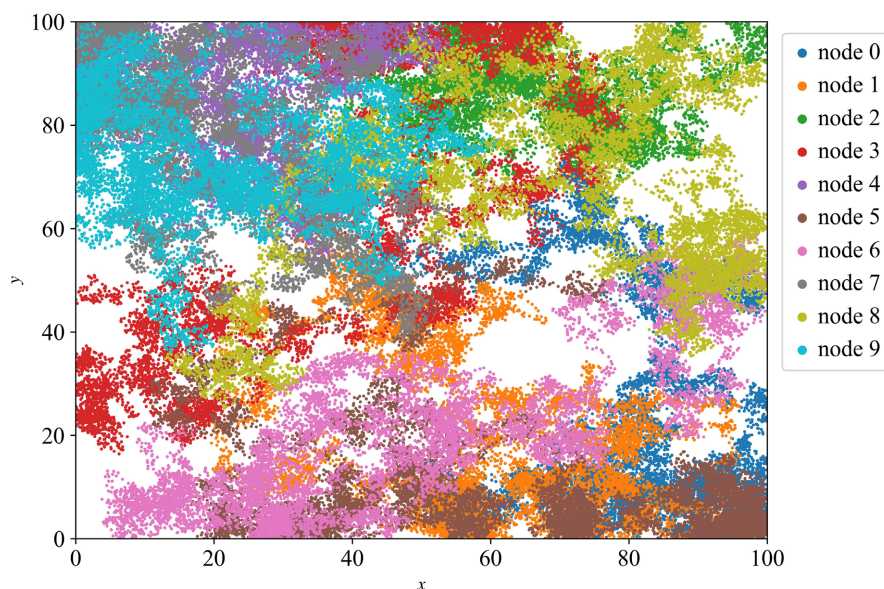


Figure 5. Node movement trajectory
图 5. 节点移动轨迹

在这个模拟环境中, 假设有 10 个工人负责数据的采集工作。每当一个工人经过这个 100×100 坐标区域中的任何一个部分, 即认为该工人成功采集了那个区域的数据。为了模拟这个过程, 随机生成了 10 个工人的移动轨迹, 这些轨迹代表了工人们在数据采集过程中的实际行进路径, 如图 5 所示。我们计算了所有工人的移动轨迹对于整个 100×100 坐标区域的覆盖情况。结果显示, 这些轨迹对完整区域的覆盖率达到了 91.48%。这意味着, 通过这 10 名工人的随机移动, 大部分区域中的数据被成功采集, 只有大约 8.52% 的区域未被覆盖, 这表明上述的模拟数据采集策略是有效的。

FedAvg 是基本的联邦学习算法, 通过对模型权重进行平均来更新模型。FedProx 关注非独立同分布数据和设备异质性问题, 通过 proximal term 来提高性能。SCAFFOLD 通过维护控制变量来减少通信量, 提高通信效率。FedNova 则利用二阶优化方法和自适应正则化技术提高收敛速度和性能。因此选择将 FedAvg、FedProx、SCAFFOLD、FedNova 四种算法作为基线进行对比。我们选择对单一工人的本地模型进行评估, 以保证不同算法下参与联邦学习的工人和数据集是相同的。

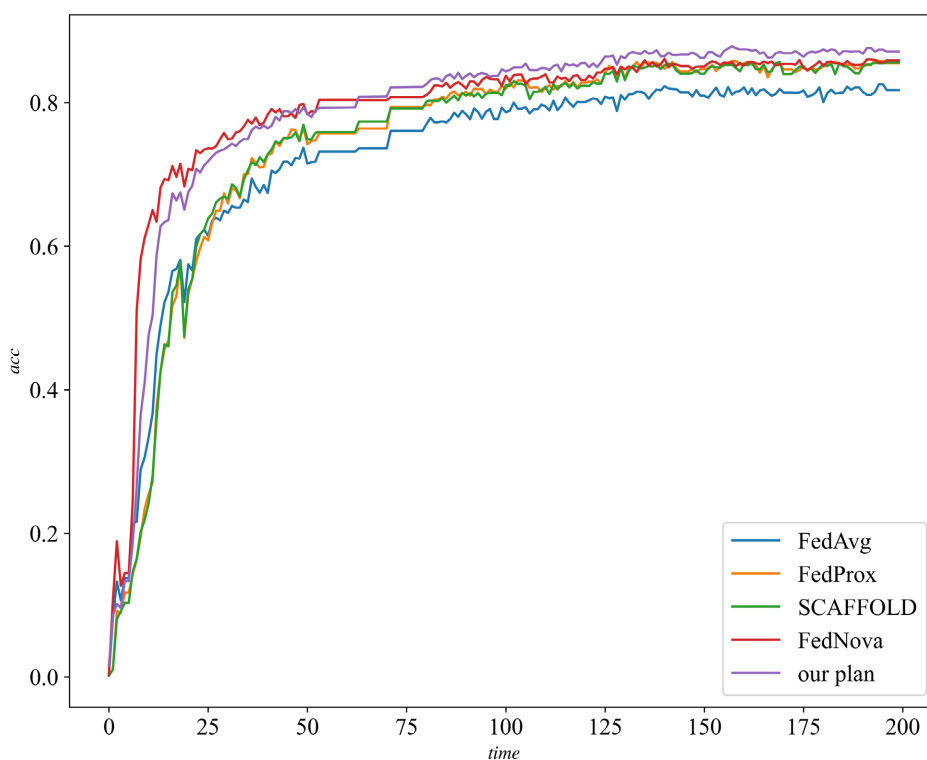


Figure 6. Performance analysis of five federated learning algorithms

图 6. 五种联邦学习算法性能分析

从图 6 可以看出, 随着时间的推移, 所有参与比较的算法在测试集上的准确率都呈现出了一定程度的波动。这种不稳定的现象主要归因于工人们在参与联邦学习的过程中, 他们仍在不断地收集新的数据。这意味着, 随着数据的不断积累, 模型的训练基础在发生变化, 从而影响了模型的稳定性。

进一步分析, 当准确率保持不变时, 这表明在该时间段内, 工人并没有实际参与到联邦学习的过程中, 而是单纯地在收集数据。这种情况下, 模型没有因为新数据的加入而发生变化, 因此在测试集上表现出恒定的准确率。

在所有的算法中, FedNova 算法因其采用了二阶优化方法和自适应正则化技术, 使得在联邦学习的早期阶段, 它的收敛速度是最快的。这两种技术的结合, 使得 FedNova 能够更有效地适应不同工人的数

据分布, 从而快速地提高模型的准确率。相比之下, FedAvg 算法在考虑如何聚合各个工人的模型时, 仅仅考虑了训练数据集的大小作为权重。这种方法忽略了数据的多样性和质量, 导致在某些情况下, 性能表现不如其他算法。

基于模型可信度的联邦学习算法则是针对本文提出的特定模型进行了专门的优化。通过深入分析和理解模型的特点, 综合考虑本地数据集大小、迭代次数、本地模型的交叉熵损失函数来优化联邦学习算法, 以期在全局收敛值上取得更好的效果。实验结果表明, 该方案在本文所提出的模型架构上的全局收敛值优于其他基线算法, 这验证了此算法设计的有效性。

7. 总结

本文针对无网络环境下 MCS 任务的执行提出了一种模型架构, 并为该模型设计了基于链路 QoS 感知的数据转发机制。同时利用改进的联邦学习算法, 使得执行 MCS 任务的工人不传输大量训练数据的情况下提升本地模型的性能。基于模型可信度的联邦学习算法通过衡量不同本地模型的可信度, 为其在模型聚合过程中赋予不同的权重, 降低了聚合模型出现偏离的风险。经过仿真测试, 验证了该模型能有效地完成 MCS 平台发布的数据收集任务, 并通过工人间协作的联邦学习机制获得一个有效的模型。在未来的工作中, 在加快模型的收敛速度的同时, 需要衡量增加工人所造成的能耗增加以及单工人工作量降低所减少的能耗之间的平衡。

参考文献

- [1] Capponi, A., Fiandrino, C., Kantarci, B., *et al.* (2019) A Survey on Mobile Crowdsensing Systems: Challenges, Solutions, and Opportunities. *IEEE Communications Surveys & Tutorials*, **21**, 2419-2165. <https://doi.org/10.1109/COMST.2019.2914030>
- [2] Yang, Q. (2019) AI and Data Privacy Protection: The Way to Federated Learning. *Journal of Information Security Research*, **5**, 961-965.
- [3] McMahan, H.B., Moore, E., Ramage, D., *et al.* (2016) Communication-Efficient Learning of Deep Networks from Decentralized Data. <https://ui.adsabs.harvard.edu/abs/2016arXiv160205629M>
- [4] Wang, H. (2022) A Survey of Application and Key Techniques for Mobile Crowdsensing. *Wireless Communications and Mobile Computing*, **2022**, Article 3693537. <https://doi.org/10.1155/2022/3693537>
- [5] 万涛, 李婉琦, 葛晶晶. 基于区块链的边缘移动群智感知声誉更新方案[J]. 计算机应用研究, 2023, 40(6): 1636-1640.
- [6] Mehanna, S. (2023) Data Quality Issues in Mobile Crowdsensing Environments. Signal and Image Processing. Master's Thesis, Université Paris-Saclay, Paris-Saclay.
- [7] Yang, H., Zhang, X., Khanduri, P., *et al.* (2022) Anarchic Federated Learning. *Proceedings of the 39th International Conference on Machine Learning*, Baltimore, 17-23 July 2022, 25331-25363.
- [8] Wu, X., Huang, F., Hu, Z., *et al.* (2023) Faster Adaptive Federated Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**, 10379-10387. <https://doi.org/10.1609/aaai.v37i9.26235>
- [9] Zeng, D., Liang, S., Hu, X., *et al.* (2023) Fedlab: A Flexible Federated Learning Framework. *Journal of Machine Learning Research*, **24**, 1-7.
- [10] Banerjee, M., Borges, C., Choo, K.K.R., *et al.* (2022) A Hardware-Assisted Heartbeat Mechanism for Fault Identification in Large-Scale IoT Systems. *IEEE Transactions on Dependable and Secure Computing*, **19**, 1254-1265.
- [11] Li, Q., Diao, Y., Chen, Q., *et al.* (2021) Federated Learning on Non-IID Data Silos: An Experimental Study. <https://ui.adsabs.harvard.edu/abs/2021arXiv210202079L.10.48550/arXiv.2102.02079>
- [12] Wang, J., Liu, Q., Liang, H., *et al.* (2020) Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. <https://ui.adsabs.harvard.edu/abs/2020arXiv200707481W.10.48550/arXiv.2007.07481>