# 第一次作业—去中心化联邦学习系统测试分析与绘图.

B20031122 郝谱合

## 1.1 引言

基于论文所设计的面向去中心化联邦学习的 DAG 区块链和基于 DAG 区块链的去中心化联邦学习设计，论文设计并实现了一种仿真原型系统。在此基础上，对区块链的性能做了测试，并对结果进行了分析。结果显示论文的 DAG 区块链具有较高的吞吐量和良好的可扩展性。论文还对去中心化联邦学习在不同攻击和不同攻击占比下的准确率，结果显示论文的方案具有良好的鲁棒性。

## 1.2 架构

该系统包括三个层次，客户端层、区块链层和联邦学习层，具体内容阐述如下。

1）客户端层，论文的客户端集成了从数据处理、模型训练与更新、到与其他客户端及服务器节点通信的全套功能，支持个性化训练、隐私保护、以及模型参数的聚合，客户端初始化伪代码如表 5.1 所示。

表 1.1 客户端初始化伪代码

| Class Client |
| --- |
| self.id = rank |
| self.SendQueue = [] |
| self.RecvQueue = [] |
| self.model = copy.deepcopy(model) |
| self.algorithm = algorithm |
| self.dataset = dataset |
| self.device = "cpu" |
| self.id = id  # integer |
| self.save_folder_name = 'items' |
| self.num_classes = 10 |
| self.train_samples = None |
| self.test_samples = None |
| self.batch_size = 10 |
| self.learning_rate = 0.005 |
| self.local_epochs = 5 |

2）区块链层，论文的区块链主要实现了区块和交易的确认模拟以及相应的数据结构。

3）联邦学习层，论文采用 pytorch 作为基础，实现了包括同态加密，余弦相似度等功能。

## 1.3 实验基本设置

### 1.3.1 数据集与数据划分

论文选用 MNIST 和图像识别作为论文系统中的基本任务，MNIST 数据集是一个广泛使用的手写数字图像数据集，它由 Yann LeCun 等人在 20 世纪 90 年代创建。MNIST 包含 60,000 个训练样本和 10,000 个测试样本，每个样本都是一个 28x28 像素的灰度图像，代表从 0 到 9 的数字之一。MNIST 数据集在机器学习和计算机视觉领域，尤其是用于训练和测试各种图像识别模型，具有重要的作用。

论文以基于 Dirichlet 分布的策略，并按照样本的标签分布来进行 non-iid 情况下的样本划分，基本思路是尽量让每个 client 上的样本标签分布不同。论文设有$K$个类别标签，$N$个 client，每个类别标签的样本需要按照不同的比例划分在不同的 client 上。论文设矩阵$X \in R^{K*N}$为类别标签分布矩阵，其行向量$x_k \in R^N$表示类别$k$在不同 client 上的概率分布向量(每一维表示$k$类别的样本划分到不同 client 上的比例)，该随机向量就采样自 Dirichlet 分布，具体划分流程如表 1.2 所示。

表 1.2 数据划分算法流程

1. 数据加载

令 $Dtrain$和$Dtest$分别作为 MNIST 的训练集和测试集

标准化数据集，$D_{transformed} = \frac{x_i}{255} - 0.5: x_i, y_i$

2. 数据分割

令 $D=Dtrain \cup Dtest$

将数据集分成 $K$ 个客户端，每个客户端都有一个数据子集$D_k$，使得

$$D = D_{train} \cup D_{test}$$

3.非独立同分布和均匀划分

如果 non-iid 为真，

则客户端之间的数据分布为非独立同分布，则每个客户端 $k$的数据子集$D_k$不同于其他客户端的数据子集，$P(D_k) \neq P(D_j)\ for\ k \neq j$

否则

为独立同分布，$P(D_k) = P(D_j)$

如果 balance 为真，每个客户端拥有的数据量相同或相近

$$|D_k| \approx \frac{|D|}{K} \quad for\ k = 1,2,\dots,K$$

4. 保存数据：

将分离的数据存储到训练和测试目录中

$$\boldsymbol{save(D_{train}, D_{test})}$$

### 1.3.2 模型

论文选用卷积神经网络（CNN）模型，其设计适用于图像分类任务。模型的输入为单通道图像，首先经过两个卷积层，分别使用 32 个和 64 个 5x5 卷积核，每个卷积层都配有 ReLU 激活函数和 2x2 最大池化层，用于提取和压缩特征。然后，卷积层的输出被展平，通过一个线性层映射到 512 维向量，并应用 ReLU 激活函数进行非线性变换。最终，经过另一个线性层将 512 维向量映射到 10 个类别，用于分类预测。

### 1.3.3 联邦学习聚合算法

论文选用经典的 Fedavg 算法,但是由于原本的 Fedavg 是针对中心化的服务器方法设计的，论文在此稍作修改，

$$prams' = w_1 * params_1 + w_2 * params_2 \tag{1-1}$$

- $prams_1$和$params_2$表示经过客户端选择算法后的两个客户端的模型参数；
- $w_1$和$w_2$分别是第 1 个客户端和第 2 个客户端的进行归一化后的权重，与客户端的数据量成比例。

## 1.4 区块链性能测试与分析

### 1.4.1 评估指标

论文使用见证权重增长速度（Witness Weight Growth）作为评价基于 DAG 区块链的去中心化方法的指标。见证权重（Witness Weight，WW）是衡量网络中对某个区块或交易认可程度的指标，它反映了支持该区块或交易的网络节点权重的累积。增长速度指的是随着时间的推移，见证权重如何增加，这关系到区块或交易获得网络确认和最终确定性的速度。同时，见证权重的增长还对区块确认时间（Time to Confirmation），网络吞吐量（Throughput），去中心化程度（Degree of Decentralization）等有关。心化程度说明了网络中权力和责任的分散程度，见证权重的增长速度可以反映网络中不同节点的影响力，从而体现系统的去中心化特性。

### 1.4.2 实验结果与分析

在本节中，论文，首先论文对见证权重的增长速度进行测试，论文设置系统中总交易数为 200，分别以不同区块发行速度进行测试，这里的区块发行速度指每秒钟到达区块链系统的新加入区块数，论文将论文的加权随机游走算法与随机游走进行对比，论文设置见证权重的确认阈值为 0.6，测试结果如下。

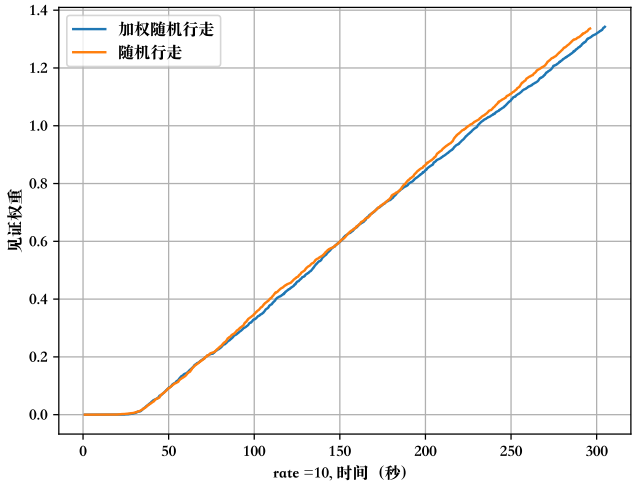


图 1.1 发行速度 50 时见证权重增长速度

图 1.1 显示，当区块发行速度为 10 时，加权随机行走与随机行走的见证权重增长速度是基本一致，确认时间为 150s。

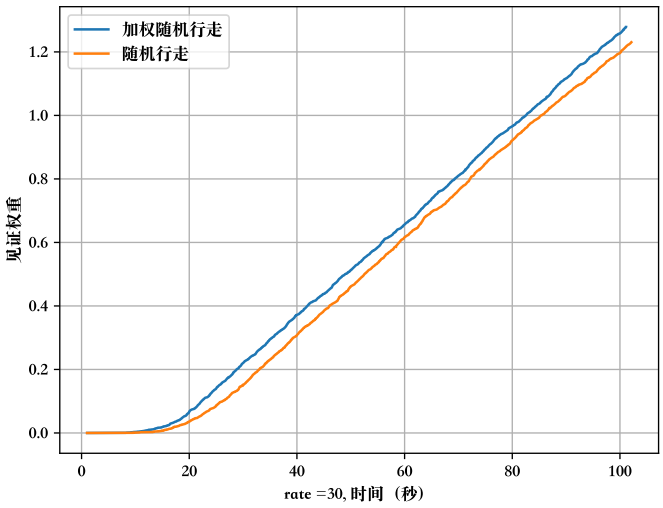


图 1.2 发行速度 30 时见证权重增长速度

图 1.2 显示，当区块发行速度为 10 时，加权随机行走与随机行走的见证权重增长速度略有差别，随机行走的确认时间为 60s，加权随机行走的确认时间略小于 60s。

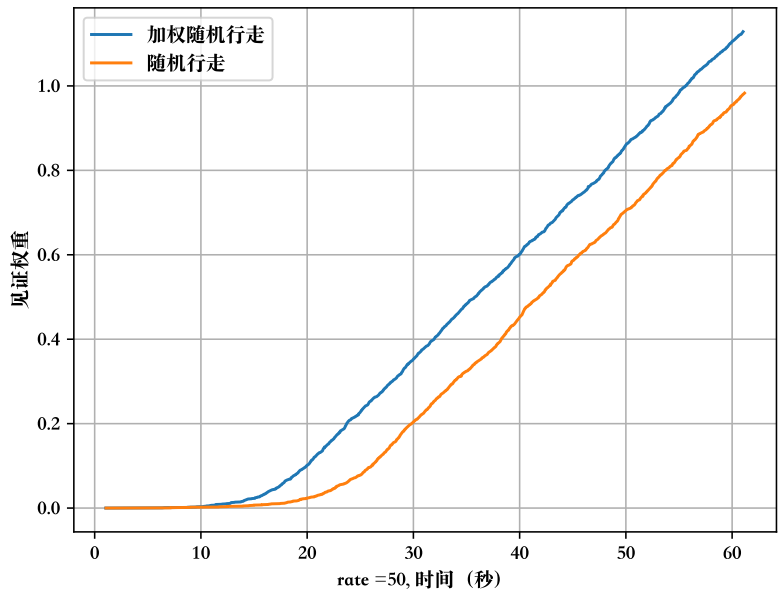


图 1.3 发行速度 50 时见证权重增长速度

图 1.3 显示，当区块发行速度为 50 时，加权随机行走与随机行走的见证权重增长速度之间的差距变大，加权随机行走的确认时间为 40s，随机行走的确认时间略大于 40s

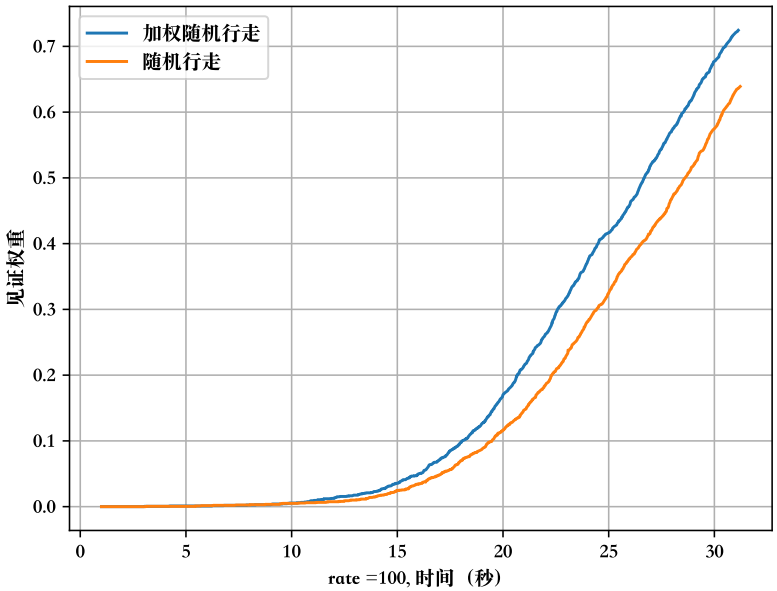


图 1.4 发行速度 100 时见证权重增长速度

图 1.4 显示，当区块发行速度为 100 时，加权随机行走与随机行走的见证权重增长速度之间的差距变大，加权随机行走的确认时间为 40s，随机行走的确认时间略大于 40s

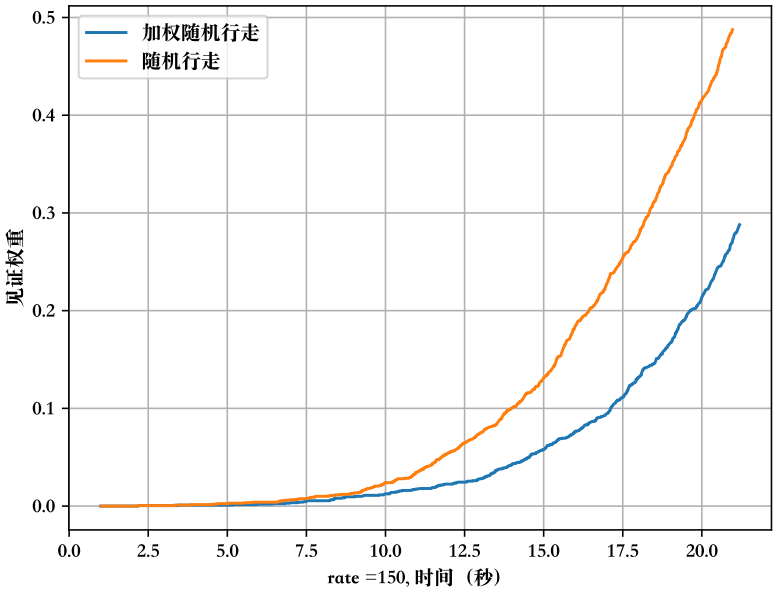


图 1.5 发行速度 150 时见证权重增长速度

图 1.5 显示，当区块发行速度为 150 时，加权随机行走与随机行走的见证权重增长速度之间的差距相较发行速度为 50 时变大。由以上实验结果可以看出，论文提出的区块链系统共识时间基本在秒级，而且随着到达区块的增长，确认时间越来越短，这证明了论文区块链系统的可扩展性。

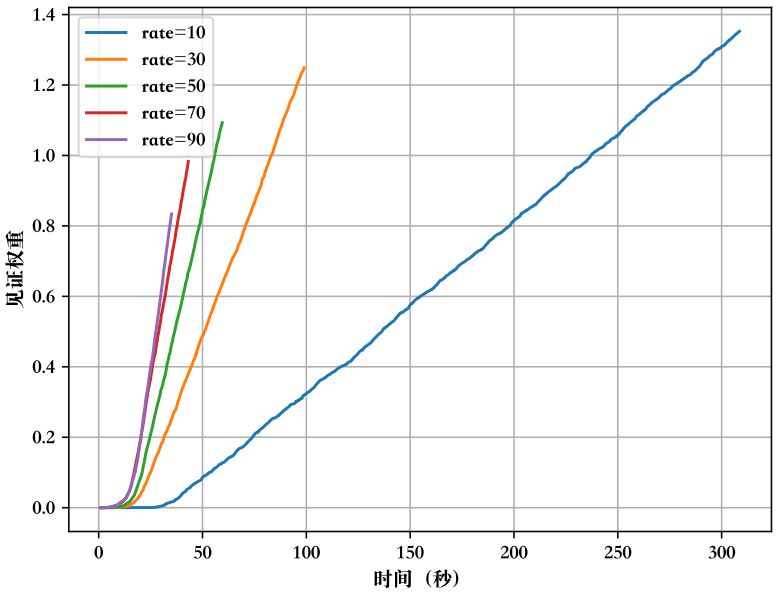


图 1.6 不同发行速度下见证权重增长速度

论文还对不同发行速度下，分别为10、30、60、70、90，的见证权重增长速度做了测试，图 1.6 显示，随着发行速度的增加见证权重的增长速度越来越快，这意味这区块确认时间将越来越短，而且随着到达区块的增长，确认时间越来越短，这意味着当区块链系统中用户越来越多时，区块共识时间越来越短，这证明了论文区块链系统的可扩展性。

表 1.3 不同区块链的吞吐量比较

| 类型 | 比特币 | 以太坊 | 本项目 | 本项目 |
|---|---|---|---|---|
| 通信时延 | 15 秒 | 15 秒 | 15 秒 | 15 秒 |
| 发行速度 | 1 块/600 秒 | 1 块/15 秒 | 100 块/秒 | 1000 块/秒 |
| 吞吐量 | 3.33 交易/秒 | 15 交易/秒 | 120 交易/秒 | 500 交易/秒 |

表 1.3 显示，相较比特币、以太坊等线型结构组织的区块链，本项目中设计的 DAG 区块链大大提高了吞吐量，说明了该区块链良好的性能与可扩展性。

## 1.5 区块链安全性分析

1）大权重攻击（Large Weight Attack）

攻击者首先向商家发送一笔交易，并等待商家确认交易具有足够的累积权重后交付商品。然后，攻击者创建一笔双重支付交易，试图通过生成大量小交易来增加双重支付交易的累积权重，使其超过原始交易的权重，从而使网络接受双重支付交易，而使原始交易被废弃。

为了防止这种攻击，可以限制交易的最大自有权重，或者将所有交易的自有权重设置为一个固定值。这样，即使攻击者拥有大量的计算能力，也无法通过生成高权重交易来轻易超越诚实节点的交易权重。

设攻击者需要在时间$\mathrm{t}_0$内找到至少$3\mathrm{n}_0$的权重来成功进行双重支付攻击，，$W(n_0)$为获得至少权重为$3^n$的交易所需的时间，μ代表攻击者的计算能力，$\mathrm{w}_1$代表原始交易的累积权重，成功的概率可以近似表示为，

$$P[W(n_0) < t_0] = 1 - \exp(-t_0 \mu 3^{-n_0}) \approx \frac{t_0 \mu}{w_1} \tag{1-2}$$

2）寄生虫链攻击（Parasite Chain Attack）

攻击者秘密构建一个寄生虫链，这个链通过偶尔引用主 DAG 结构中的交易来获得更高的分数。在发起攻击时，攻击者可以突然增加寄生虫链中的交易数量，使得新交易更有可能引用寄生虫链而非主 DAG 结构，从而使主 DAG 结构中的诚实交易得不到确认。

使用加权随机游走算法进行未确认区块搜索，通过在 DAG 结构中放置随机行走者，让它们以一定的概率向未确认区块方向移动，从而选择要引用的未确认

区块。这种算法可以降低选择寄生虫链中未确认区块的概率，因为寄生虫链的累积权重通常较低。

3）分裂攻击（Splitting Attack）

在高负载情况下，攻击者试图将 DAG 结构分裂成两个平衡增长的分支。攻击者在分裂点处放置至少两个冲突交易，以阻止诚实节点同时引用这两个分支，从而使攻击者能够在两个分支上各花费一次资金。

使用"未确认区块阈值"规则，如选择最重的分支作为诚实节点的参考，使得分裂的两个分支难以维持平衡。此外，可以修改加权随机游走算法，使其在分支分裂时选择"更重"的分支。

4）懒惰节点（Lazy Nodes）：

懒惰节点可能会选择批准一些旧交易而不是积极参与验证新的或更有活力的未确认区块，从而降低网络的安全性和效率。

加权随机游走未确认区块搜索算法不太可能选择懒惰节点作为引用目标，因为这些节点的累积权重通常较低。此外，可以通过算法调整，降低懒惰节点被选择的概率。

## 1.6 去中心化联邦学习测试与分析

### .6.1 测试指标

准确度(Accuracy)，为了量化去中心化联邦学习的有效性，将解析准确度定义为测试集中正确样本和总样本的比例，如式（1-3）所示。

$$Accuracy = \frac{n_{correct}}{n_{total}} \tag{1-3}$$

### 1.6.2 攻击模型

在本实验中，论文的攻击模型如下所示，主要选取了联邦学习中的投毒攻击，的缩放攻击，随机符号攻击、标签反转攻击具体方式如下。

**1）缩放攻击**，（Rescaling Attack）攻击的本质是通过改变数据的尺度来干扰模型的训练。攻击者可能会将其本地数据的特征值进行缩放或拉伸，使其处于不同的数值范围内。这样做可能会导致模型在整合全局更新时对不同参与者的贡献进行错误的权衡，从而影响模型的训练效果。举例来说，如果一个恶意参与者将其数据的数值范围扩大了几倍，那么在联邦学习中，这个参与者的影响可能会被过度放大，导致整体模型过度适应这个参与者的数据分布，而忽视其他参与者的贡献。

**2）随机符号攻击**，（Signrandom Attack）攻击旨在在模型更新过程中引入随机性，以扰乱或破坏联邦学习系统的正常运作。在这种攻击中，攻

击者可能会改变其梯度更新的方向或大小，或者在上传参数时添加噪音。这样做的目的是为了干扰其他参与者的训练过程，使其无法正常收敛或得到准确的模型。这种攻击形式尤其对于梯度聚合和参数共享的联邦学习框架产生了严重的影响，因为它可能导致模型收敛到次优解或者根本无法收敛。

**3）标签反转攻击，**（Labelflip Attack）攻击是一种针对标签数据的攻击形式，旨在误导模型以产生错误的分类结果。攻击者可能会有意将其本地数据的标签进行更改，使其被错误地分类到不同的类别。这种攻击可能会导致模型产生严重的误差，特别是在某些类别的样本数量较少时。例如，如果一个恶意参与者将所有猫的标签更改为狗，而其他参与者没有相应的狗的样本，那么整体模型可能会将猫误分类为狗，从而导致模型性能的下降。在处理 Label Flip 攻击时，联邦学习系统需要采取特殊的防御措施，如鲁棒性训练或标签噪声注入，以确保模型能够对抗这种攻击并保持准确性。

### 1.6.3 实验结果与分析

在本节中，论文对去中心化联邦学习的性能做了测试与分析，论文选取单一客户端，在不同攻击程度下，执行客户端选择算法，并与随机客户端选择做比较。结果显示，论文的客户端选择算法，具有很强的鲁棒性

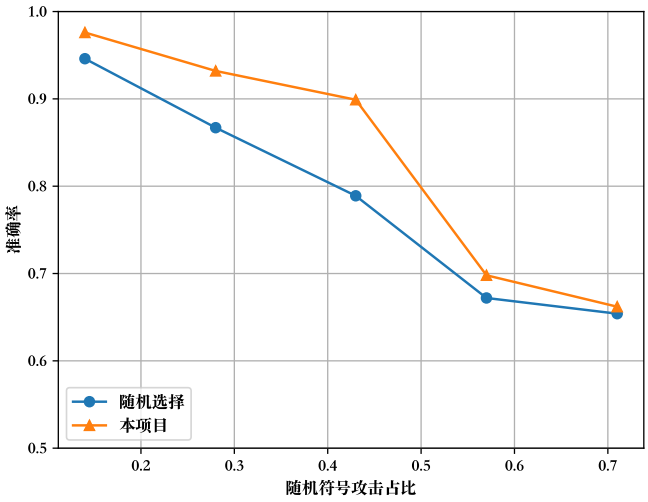


图 1.7 不同缩放攻击程度下的准确率

如图 1.7 所示，在缩放攻击占比比较小时，模型准确率极高，随机选择算法和本项目的客户端选择算法的输出结果差别不大，这是因为在论文的项目设定中，每个客户端只选择两个候选区块获取目标模型参数，但攻击占比较小时，即使是随机选择，选择到恶意模型参数的可能性也较小。

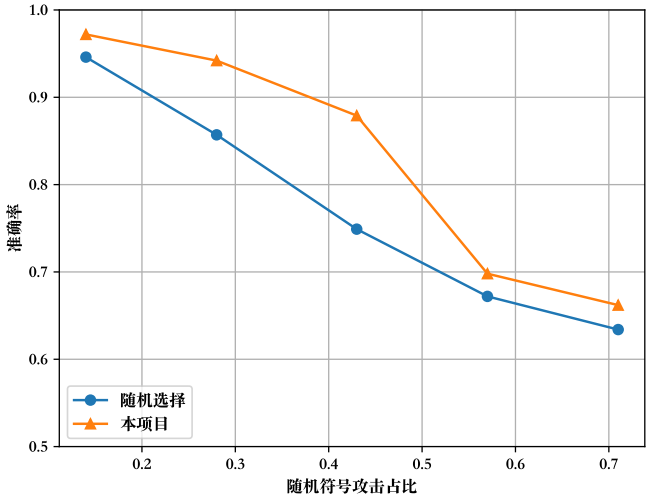


图 1.8 不同随机符号攻击程度下的准确率

如图 1.8 所示，在随机符号攻击占比越来越大时，随机选择准确率持续下降，这是因为本项目中客户端只选择两个候选区块，其次客户端有可信任的根数据集作为基线。当攻击占比跃过 0.5 时，准确率急剧下降，这是因为论文的方案取决于诚实的大多数。

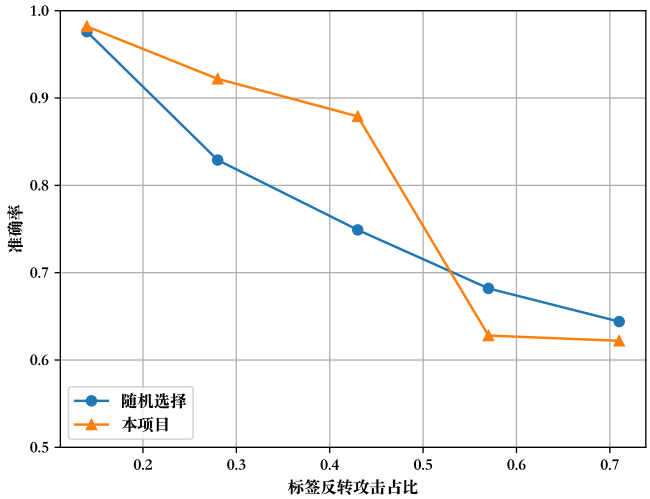


图 1.9 不同标签反转攻击程度下的准确率

如图 1.9 所示，在标签反转攻击占比越来越大时，客户端的准确率持续下降，原因同上。综合在三种攻击在不同程度下的准确率表现，论文可知对于单一的客户端，在攻击情况下，仍能保持很好的鲁棒性。当攻击占比跃过 0.5 时，准确率急剧下降，这是因为论文的方案取决于诚实的大多数。

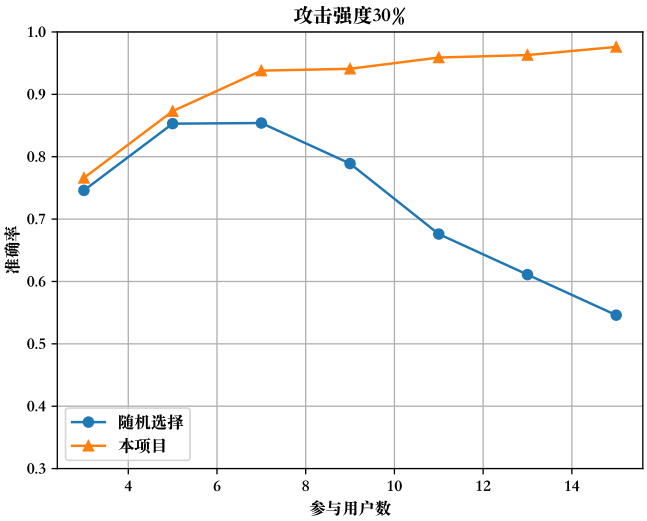


图 1.10 不同用户数在 30%攻击强度下的准确率

之前的实验针对单一客户端在攻击条件下的鲁棒性，本次试验对多用户进行测试，结果显示，在攻击强度 30%时，当用户数较小时，选择到恶意梯度的可能性较大，随着用户数增大，选择到恶意梯度的可能性变小，准确率提升，担当用户数量超过一定阈值时，则总有用户会选择到恶意梯度，导致好的用户也上传了差的梯度，准确率的差异越来越大，但采用本项目的客户端选择算法，模型依然保持了较好的准确度

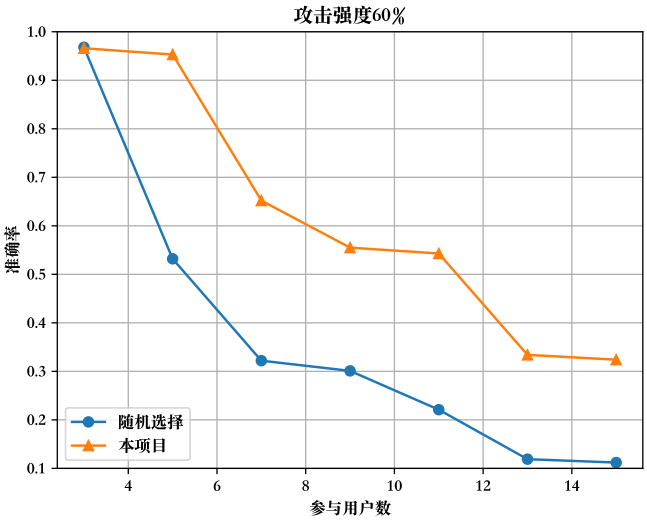


图 1.11 不同用户数在 60%攻击强度下的准确率

实验结果显示，在 60%的攻击占比时，准确率的差异越来越大，这一情况对单一客户端来说影响并不大，因为只选择两个客户端，选中恶意模型的概率较小，但当多次选择后，一旦被恶意模型污染，准确率会越来越差。