华 中 科 技 大 学

研究生课程考试答题本

考 生 姓 名 张杰

考 生 学 号 m 2 0 2 2 7 1 7 7 9

系 、 年 级 网络空间安全学院、2022 级

类 别 学术型硕士

考 试 科 目 隐私保护

考 试 日 期 2022 年 12 月

评　　分

题 号	得 分	题 号	得 分
1		9	
2		1 0	
3		1 1	
4		1 2	
5		1 3	
6		1 4	
7		1 5	
8			

总 分 ：	评 卷 人 ：

注：

1.无评卷人签名试卷无效。

2.必须用钢笔或圆珠笔阅卷，使用红色，用铅笔阅卷无效。

题目及各级标题自拟（正文宋体小四，1.25 倍行距）

FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping

1. 文章信息

出处： Network and Distributed Systems Security (NDSS) Symposium 2021

作者: Xiaoyu Cao, Minghong Fang, Jia Liu, Neil Zhenqiang Gong

2. 概要

拜占庭式的鲁棒联邦学习方法中没有信任的根（即不知道服务器的任务），所以在服务器看来，每个客户端都可能是恶意的。

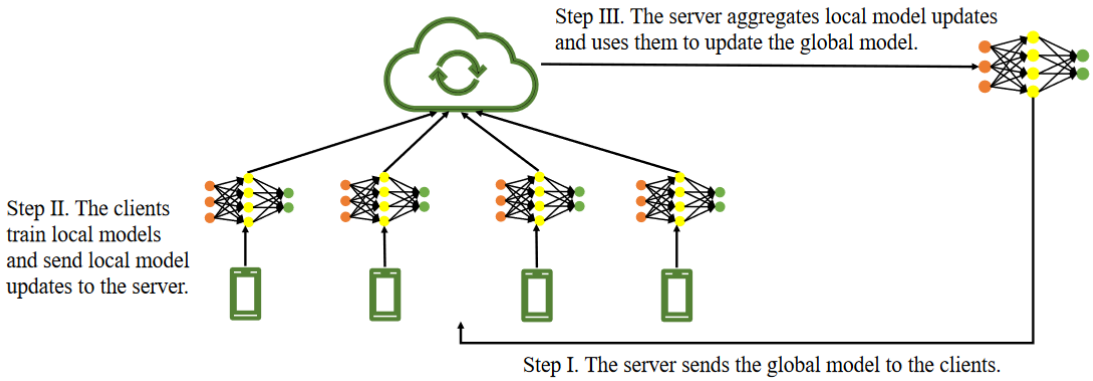
本文提出FLTrust来引导客户端进行训练。即服务提供者自己为学习任务收集一个干净的小型训练数据集(称为根数据集)，并在此基础上维护一个模型(称为服务器模型)来引导信任。

在每次迭代中，服务提供者首先为来自客户端的每个本地模型更新分配一个信任分数，其中，如果本地模型更新的方向偏离服务器模型更新的方向更多，则本地模型更新的信任分数更低。然后，服务提供者将本地模型更新的大小归一化，使其与向量空间中的服务器模型更新处于同一超球体中。标准化限制了大规模恶意本地模型更新的影响。最后，服务提供者计算标准化局部模型更新的平均值，并将其加权作为全局模型更新，该更新用于更新全局模型。

2. 背景知识

2.1 联邦学习

联邦学习的学习过程



2.2 联邦学习的聚合规则

- FedAvg：最基本的模型平均方法

  FedAvg 计算客户端本地模型更新的平均值作为全局模型更新，其中每个客户端都由其训练示例的数量加权。

$$\boldsymbol{g} = \sum_{i=1}^{n} \frac{|D_i|}{N} \boldsymbol{g}_i$$

- 拜占庭鲁棒性聚合算法
- - Krum：基于欧氏距离的全局模型更新

    对于每个梯度g_i，选择与它最近的$n-f-2$个距离最近的梯度。具有最小分数的客户端的本地模型更新将被选择作为全局模型更新来更新全局模型。

    $$s_i = \sum_{\boldsymbol{g}_j \in \Gamma_{i,n-f-2}} \|\boldsymbol{g}_j - \boldsymbol{g}_i\|_2^2$$

  - Trimmed Mean：单独考虑每个模型参数的坐标级聚合规则

    对于每个模型参数，服务器在所有本地模型更新中收集其值并对其进行排序。给定修剪参 $k < \frac{n}{2}$ ，服务器删除最大的 k 和最小的 k 值，然后计算剩余 $n-2k$ 值的平均值作为全局模型更新中相应参数的值。

  - Median：将每个参数的中值作为全局模型更新中对应的参数值

    服务器会对所有本地模型更新中每个单独参数的值进行排序，将每个参数的中值视为全局模型更新中的相应参数值。

2.3 联邦学习的投毒攻击

联邦学习的投毒攻击可以分为数据投毒攻击和模型投毒攻击。

联邦学习很容易受到数据投毒攻击，即恶意客户端可以通过修改、添加或删除其本地训练数据集中的示例来破坏全局模型。例如，一种称为标签翻转攻击的数据投毒攻击会更改恶意客户端训练示例的标签，同时保持其特征不变。

此外，与集中式学习不同，联邦学习更容易受到本地模型中毒攻击，在这种攻击中，恶意客户端会毒害本地模型或其发送到服务器的更新。根据攻击者的目标，局部模型中毒攻击可以分为无目标攻击和有目标攻击。无目标攻击旨在破坏全局模型，使其对大量测试示例不加选择地做出错误预测，即测试错误率很高。有目标攻击旨在破坏全局模型，以便当模型收到的输入为攻击者选择的目标时，模型会输出攻击者选择的标签，而其他非目标测试示例的预测标签不受影响。

另外，任何数据中毒攻击都可以转化为本地模型中毒攻击，即我们可以计算恶意客户端中毒的本地训练数据集的本地模型更新，并将其视为中毒的本地模型更新。最近的研究表明局部模型中毒攻击比针对联邦学习的数据中毒攻击更有效。

3. 算法描述

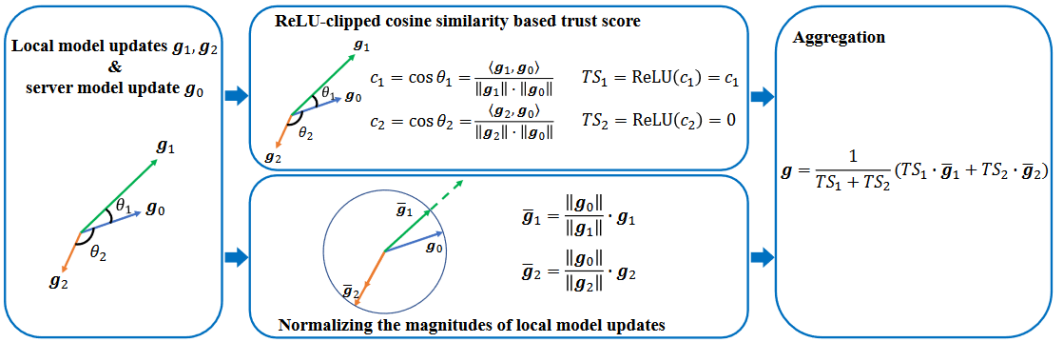
3.1 整体架构

由于之前的联邦学习没有信任的根，本地客户端训练时无法知道其训练防线，所以本文的中心服务器提供一个小型训练集供客户端使用，每个客户端使用自己的本地数据以及服务器提供的训练集训练，这样使得客户端训练的方向朝着全局模型的方向训练。

在FLTrust中，服务器本身收集一个小的干净的训练数据集(称为根数据集)（很小就100个训练样本左右），并为它维护一个模型(称为服务器模型)，就像客户端维护一个本地模型一样。

区别于不同的联邦学习中只考虑客户的本地模型更新来更新全局模型，FLTrust同时考虑服务器模型更新和客户端的本地模型更新来更新全局模型。

攻击者可以操纵恶意客户端的本地模型更新的方向，从而使全局模型的更新方向与应该更新的方向相反;或者攻击者可以扩大局部模型更新的幅度，以控制聚合的全局模型更新。FLTrust首先根据本地模型更新与服务器模型更新的方向相似度，为其分配一个信任分数(TS)。形式上，对本地模型更新的信任得分是它与服务器模型更新的relu修剪余弦相似度。然后，FLTrust对每个本地模型更新进行规范化，将其扩展为与服务器模型更新具有相同的大小。这种归一化本质上是将每个局部模型更新投射到同一个超球体上，而服务器模型更新位于向量空间中，从而限制了大量有毒局部模型更新的影响。最后，FLTrust计算归一化局部模型更新的平均值，并将其加权作为全局模型更新，用于更新全局模型。



3.2 信任得分

恶意客户端的更新方向肯定为全局模型更新方向的反方向。所以本文用余弦相似度来表示中心服务器对客户端的信任程度，若服务器模型更新的方向与客户端模型的更新方向为负则说明该客户端是恶意的，否则为良性客户端，即可参与全局更新。

现知道余弦相似度为负的为恶意客户端，所以应该将其提出，即将其裁剪掉，本文使用RELU函数将其裁剪，故对一个客户端的信任得分为：

$$\mathrm{TS_i} = \mathrm{ReLU(c_i)}$$

其中c_i为第i个客户端的本轮局部更新$\boldsymbol{g}_i$和服务器模型g_0的余弦相似度，即上一轮的全局模型更新，$c_\mathrm{i} = \frac{\langle \mathrm{g_i}, \mathrm{g_0}\rangle}{\|\mathrm{g_i}\|\|\mathrm{g_0}\|}$，其中$\mathbf{g}_0$是服务器模型。

3.3 归一化

由于恶意客户端可能将本地模型更新的规模无限扩大，从而破坏全局模型，因此将每个局部模型更新的幅度归一化。如果没有信任的基础，很难该将数量正常化。但是，服务器有根数据集来引导FLTrust中的信任。因此，将每个本地模型更新归一化，使其具有与服务器模型更新相同的大小。这种标准化意味着将本地模型更新缩放为相同的超球体，服务器模型更新位于向量空间中。具体如下：

$$\overline{\mathrm{g_i}} = \frac{\|\mathrm{g_0}\|}{\|\mathrm{g_i}\|} \cdot \mathrm{g_i}$$

其中g_i为第i个客户端的本地模型更新，g_0是服务器模型。

3.4 模型更新

将标准化局部模型更新值按其信任值加权后的平均值计算为全局模型更新值：

$$g = \frac{1}{\sum_{j=1}^{n} TS_j} \sum_{i=1}^{n} T_i = \frac{1}{\sum_{j=1}^{n} \mathrm{Re}\,LU\,(c_j)} \sum_{i=1}^{n} \mathrm{ReL}\,U\,(c_i) \cdot \frac{\|\boldsymbol{g}_0\|}{\|\boldsymbol{g}_i\|} \cdot \boldsymbol{g}_i$$

更新全局模型：

$$\mathrm{w} \leftarrow \mathrm{w} - \alpha \cdot \mathrm{g}$$

3.5 伪代码

Input: n clients with local training datasets $D_i, i = 1, 2, \cdots, n$; a server with root dataset D_0; global learning rate α; number of global iterations R_g; number of clients τ sampled in each iteration; local learning rate β; number of local iterations R_l; and batch size b.

Output: Global model $\boldsymbol{w}$.

$\boldsymbol{w} \leftarrow$ random initialization.
for $r = 1, 2, \cdots, R_g$ **do**
  // Step I: The server sends the global model to clients.
  The server randomly samples τ clients $C_1, C_2, \cdots, C_\tau$ from $\{1, 2, \cdots, n\}$ and sends $\boldsymbol{w}$ to them.
  // Step II: Training local models and server model.
  // Client side.
  for $i = C_1, C_2, \cdots, C_\tau$ **do in parallel**
    $\boldsymbol{g}_i = ModelUpdate(\boldsymbol{w}, D_i, b, \beta, R_l)$.
    Send $\boldsymbol{g}_i$ to the server.
  end for
  // Server side.
  $\boldsymbol{g}_0 = ModelUpdate(\boldsymbol{w}, D_0, b, \beta, R_l)$.
  // Step III: Updating the global model via aggregating the local model updates.
  for $i = C_1, C_2, \cdots, C_\tau$ **do**
    $TS_i = ReLU\left(\frac{\langle \boldsymbol{g}_i, \boldsymbol{g}_0 \rangle}{\|\boldsymbol{g}_i\| \|\boldsymbol{g}_0\|}\right)$.
    $\bar{\boldsymbol{g}}_i = \frac{\|\boldsymbol{g}_0\|}{\|\boldsymbol{g}_i\|} \cdot \boldsymbol{g}_i$.
  end for
  $\boldsymbol{g} = \frac{1}{\sum_{j=1}^{\tau} TS_{C_j}} \sum_{i=1}^{\tau} TS_{C_i} \cdot \bar{\boldsymbol{g}}_{C_i}$.
  $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha \cdot \boldsymbol{g}$.
end for
return $\boldsymbol{w}$.

4. 实验评估

4.1 实验设置

4.1.1 联邦学习参数

	Explanation	MNIST-0.1	MNIST-0.5	Fashion-MNIST	CIFAR-10	HAR	CH-MNIST
n	# clients	100				30	40
τ	# clients selected in each iteration	n					
R_l	# local iterations	1					
R_g	# global iterations	2,000		2,500	1,500	1,000	2,000
b	batch size	32			64	32	
$\alpha \cdot \beta$	combined learning rate	3×10^{-4}		6×10^{-3}	2×10^{-4}	3×10^{-3}	3×10^{-4} (decay at the 1500th and 1750th iterations with factor 0.9)
m/n	fraction of malicious clients (%)	20					
m	# malicious clients	20				6	8
f	Krum parameter	m					
k	Trim-mean parameter	m					
$\lvert D_0 \rvert$	size of the root dataset	100					

4.1.2 模型参数

Layer	Size
Input	$28 \times 28 \times 1$
Convolution + ReLU	$3 \times 3 \times 30$
Max Pooling	2×2
Convolution + ReLU	$3 \times 3 \times 50$
Max Pooling	2×2
Fully Connected + ReLU	100
Softmax	10

4.1.3 数据集

MNIST-0.1：MNIST是一个 10 类数字图像分类数据集，由 60,000 个训练示例和 10,000 个测试示例组成。我们在 MNIST-0.1 中设置 q = 0.1，这表明本地训练数据在客户端之间是 IID。我们使用 MNIST-0.1 来证明 FLTrust 在 IID 设置中也是有效的。

MNIST-0.5：在 MNIST-0.5 中，我们通过设置 q = 0.5 来模拟客户端之间的非 IID 本地训练数据。

Fashion-MNIST：Fashion-MNIST是一个 10分类时尚图像分类任务，它有一个包含 60,000 个时尚图像的预定义训练集和一个包含 10,000 个时尚图像的测试集。与 MNIST-0.5 数据集一样，我们将训练示例分发给 q = 0.5 的客户端，以模拟非 IID 本地训练数据。

4.1.4 指标参数

对于无目标攻击，选择testing error rate。

对于有目标攻击，选择attack success rate。

4.1.5 根数据集

case1：根数据集与学习任务的整体训练数据分布具有相同的分布。

case2：根数据集的分布不同于学习任务的整体训练数据分布

4.2 实验结果

不同联邦学习方法在不同攻击下的测试错误率和Scaling攻击的攻击成功率。

(a) CNN global model, MNIST-0.1

	FedAvg	Krum	Trim-mean	Median	FLTrust
No attack	0.04	0.10	0.06	0.06	0.04
LF attack	0.06	0.10	0.05	0.05	0.04
Krum attack	0.10	0.90	0.07	0.07	0.04
Trim attack	0.16	0.10	0.13	0.13	0.04
Scaling attack	0.02 / 1.00	0.10 / 0.00	0.05 / 0.01	0.05 / 0.01	0.03 / 0.00
Adaptive attack	0.08	0.10	0.11	0.13	0.04

(b) CNN global model, MNIST-0.5

	FedAvg	Krum	Trim-mean	Median	FLTrust
No attack	0.04	0.10	0.06	0.06	0.05
LF attack	0.06	0.10	0.06	0.06	0.05
Krum attack	0.10	0.91	0.14	0.15	0.05
Trim attack	0.28	0.10	0.23	0.43	0.06
Scaling attack	0.02 / 1.00	0.09 / 0.01	0.06 / 0.02	0.06 / 0.01	0.05 / 0.00
Adaptive attack	0.13	0.10	0.22	0.90	0.06

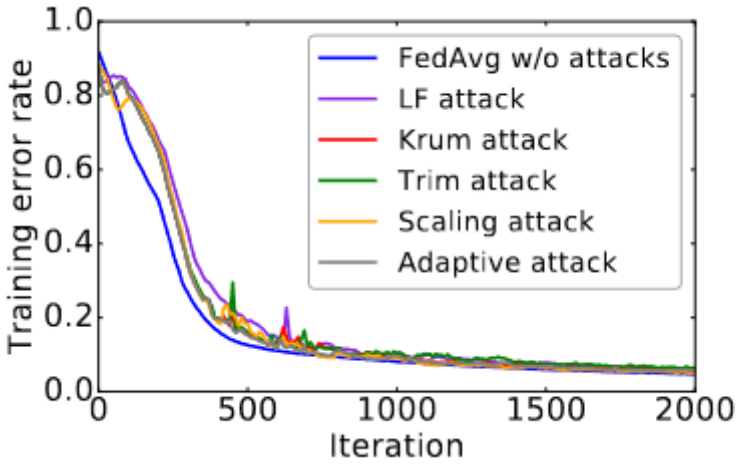
(c) CNN global model, Fashion-MNIST

	FedAvg	Krum	Trim-mean	Median	FLTrust
No attack	0.10	0.16	0.14	0.14	0.11
LF attack	0.14	0.15	0.26	0.21	0.11
Krum attack	0.13	0.90	0.18	0.23	0.12
Trim attack	0.90	0.16	0.24	0.27	0.14
Scaling attack	0.90 / 1.00	0.16 / 0.03	0.17 / 0.85	0.16 / 0.05	0.11 / 0.02
Adaptive attack	0.90	0.18	0.34	0.24	0.14

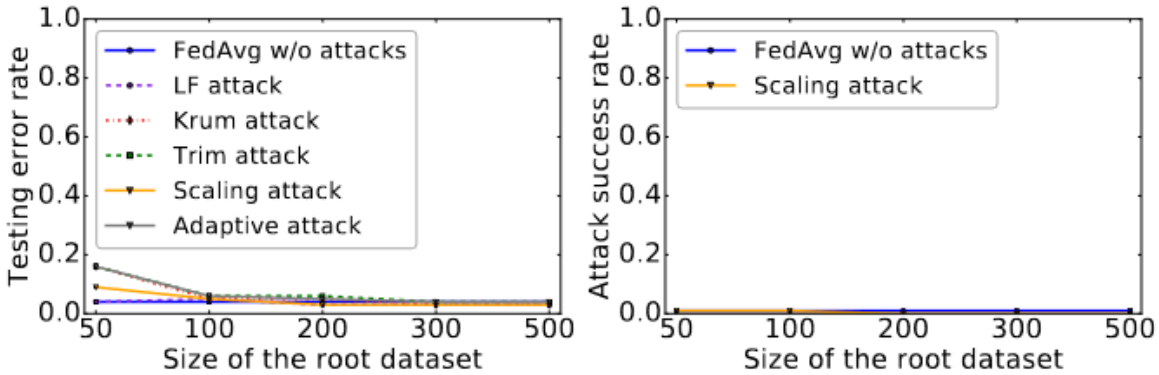
FLTrust不同变体在不同攻击下的测试错误率和Scaling攻击对MNIST-0.5的攻击成功率。

	No attack	LF attack	Krum attack	Trim attack	Scaling attack	Adaptive attack
FLTrust-Server	0.21	–	–	–	–	–
FLTrust-withServer	0.07	0.08	0.09	0.10	0.08 / 0.01	0.94
FLTrust-NoReLU	0.28	0.90	0.90	0.90	0.94 / 0.08	0.90
FLTrust-NoNorm	0.05	0.06	0.06	0.08	0.94 / 0.08	0.06
FLTrust-ParNorm	0.06	0.06	0.06	0.06	0.06 / 0.01	0.06
FLTrust	0.05	0.05	0.05	0.06	0.05 / 0.00	0.06

训练错误率与 FLTrust 在不同攻击下的迭代次数和 FedAvg 在没有攻击 MNIST-0.5 的情况下。



MNIST-0.5不同攻击下根数据集大小对 FLTrust 的影响。



(a) Testing error rate (b) Attack success rate

case2情况下根数据集以不同偏差概率采样时FLTrust在不同攻击下的测试错误率和Scaling攻击的攻击成功率。

(a) MNIST-0.1

Bias probability	0.1	0.2	0.4	0.6	0.8	1.0
No attack	0.04	0.04	0.04	0.05	0.05	0.34
LF attack	0.04	0.04	0.04	0.05	0.78	0.84
Krum attack	0.04	0.04	0.07	0.89	0.89	0.89
Trim attack	0.04	0.05	0.08	0.12	0.46	0.89
Scaling attack	0.03 / 0.00	0.03 / 0.01	0.04 / 0.00	0.04 / 0.00	0.06 / 0.01	0.42 / 0.01
Adaptive attack	0.04	0.05	0.08	0.12	0.90	0.90

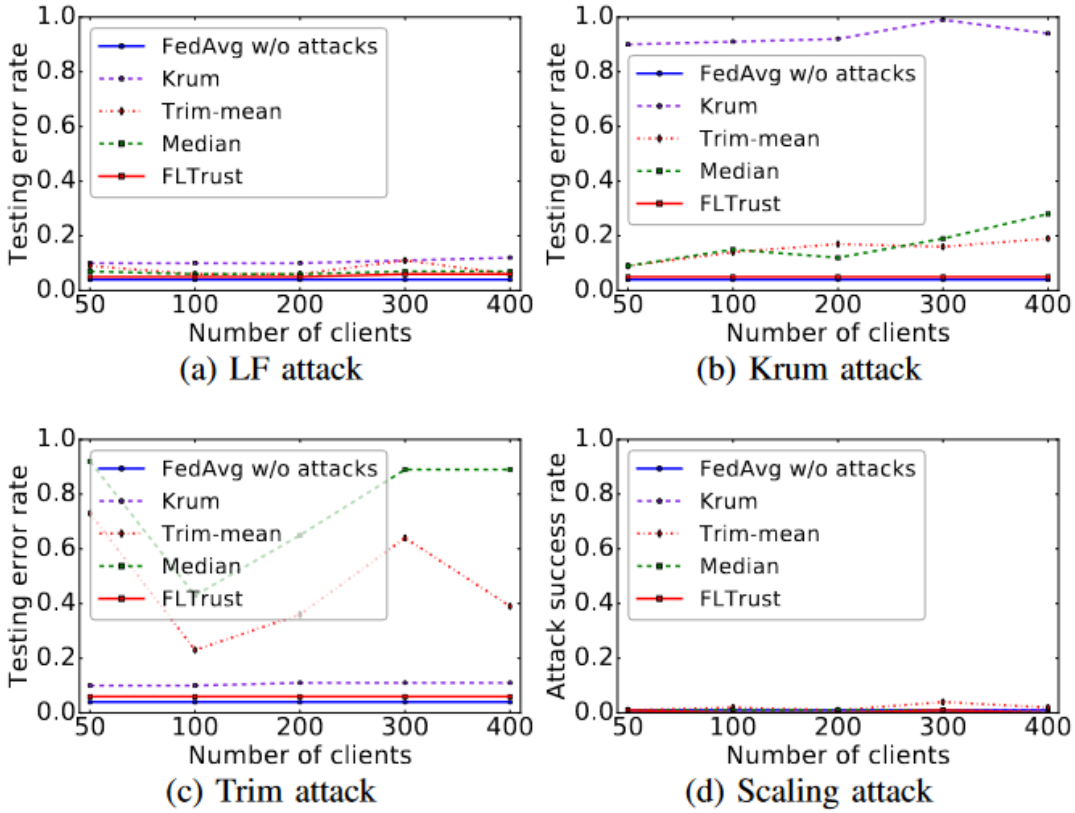
(b) MNIST-0.5

Bias probability	0.1	0.2	0.4	0.6	0.8	1.0
No attack	0.05	0.05	0.06	0.08	0.11	0.80
LF attack	0.05	0.05	0.08	0.10	0.25	0.89
Krum attack	0.05	0.05	0.08	0.12	0.86	0.89
Trim attack	0.06	0.06	0.08	0.12	0.16	0.89
Scaling attack	0.05 / 0.00	0.05 / 0.01	0.06 / 0.00	0.07 / 0.01	0.12 / 0.00	0.86 / 0.01
Adaptive attack	0.06	0.07	0.08	0.13	0.90	0.90

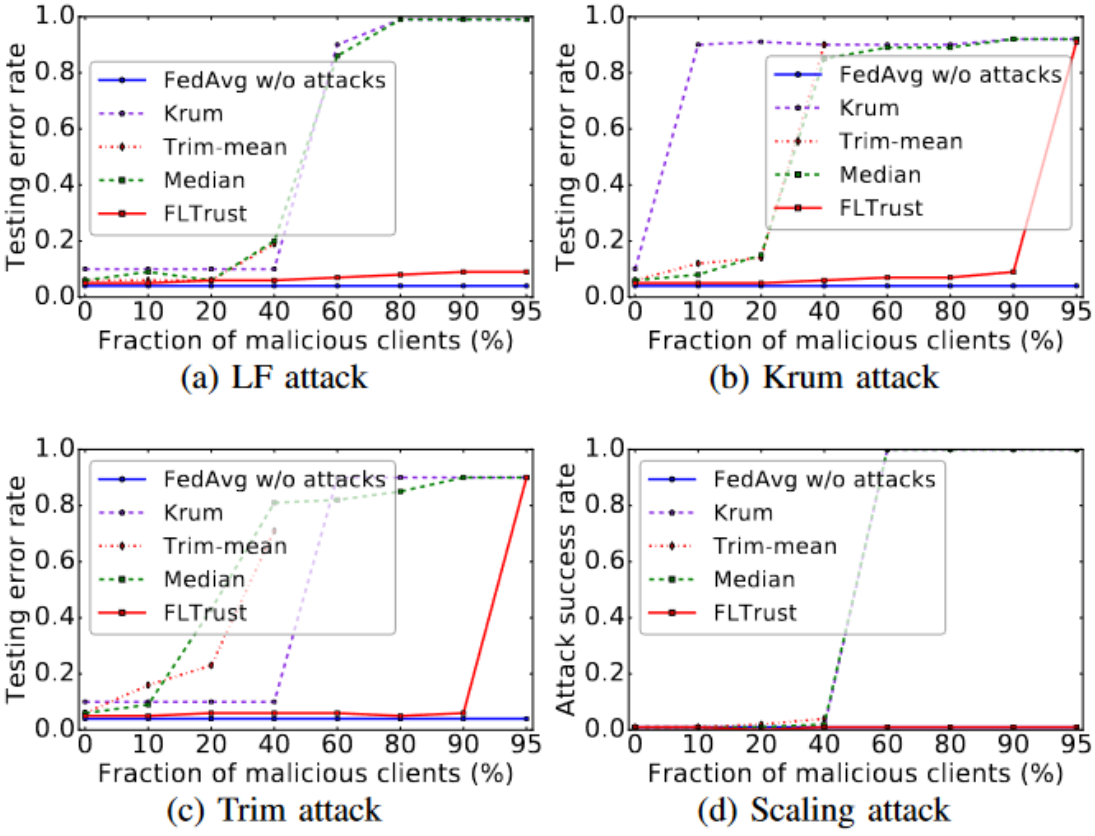
(c) Fashion-MNIST

Bias probability	0.1	0.2	0.4	0.6	0.8	1.0
No attack	0.11	0.11	0.12	0.15	0.16	0.90
LF attack	0.11	0.11	0.12	0.12	0.14	0.90
Krum attack	0.12	0.12	0.16	0.90	0.90	0.90
Trim attack	0.14	0.14	0.15	0.21	0.90	0.90
Scaling attack	0.11 / 0.02	0.12 / 0.04	0.12 / 0.04	0.13 / 0.02	0.15 / 0.03	0.90 / 0.00
Adaptive attack	0.14	0.14	0.16	0.90	0.90	0.90

客户端总数对不同 FL 方法在不同攻击下的测试错误率的影响。



恶意客户端的比例对不同攻击下不同 FL 方法的测试错误率的影响。



结果表明，即使大部分客户端在所有数据集上都是恶意的，FLTrust 也能抵抗自适应攻击。具体来说，对于 MNIST-0.1（MNIST-0.5、FashionMNIST、CIFAR-10、HAR 或 CH-MNIST），自适应攻击下的 FLTrust 超过 60%（超过 40%，高达 60%，高达 60%，高达40%，或超过 40%）的恶意客户端在不受攻击的情况下仍然可以达到与 FedAvg 相似的测试错误率。