

Backdoor Attack and Defense in FL - A Survey

Attack

Summary

- 1.constrain-and-scale
- 2.non-omniscient attack
- 3.local model poisoning attacks
- 4.ARG-attack
- 5.MPAF
- 6.3DFed

Defense

- 1.Krum
- 2.Median、Trimmed_mean
- 3.Bulyan
- 4.FLTrust
- 5.FLDector
- 6.FLAME
- 7.DnC
- 8.SignGuard
- 9.DeepSight
- 10.FreqFed

Attack

Summary

论文标题	类型	具体类型	发布时间	会/刊名称	论文链接

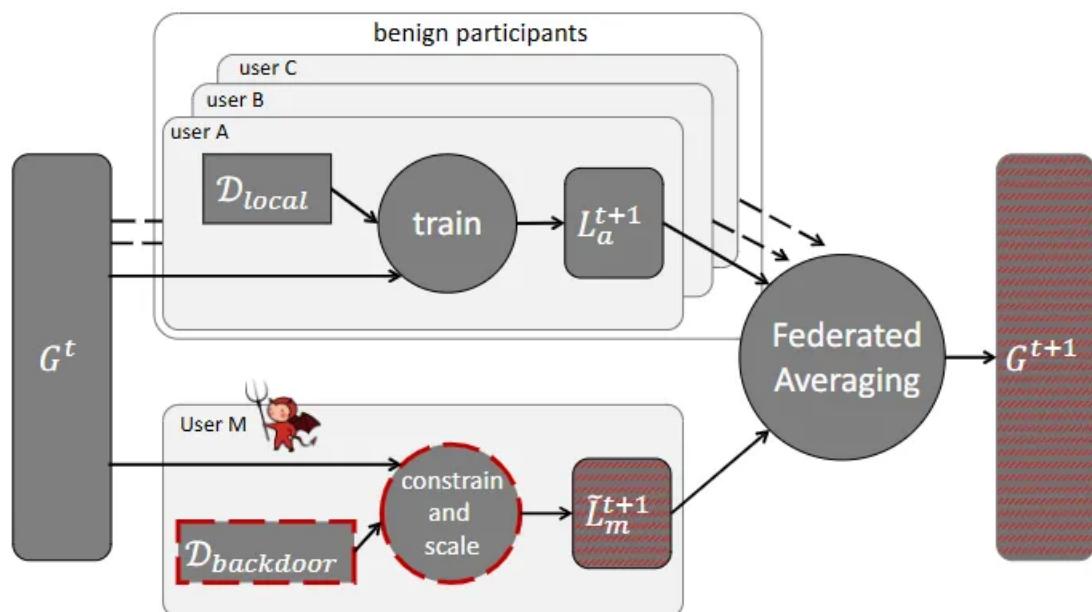
How to backdoor federated learning	攻击	后门 攻击	202 0	AIST ATS	https://proceedings.mlr.press/v108/bagdasarian20a.html
A Little Is Enough: Circumventing Defenses For Distributed Learning	攻击	后门 攻击	201 9	NIPS	https://proceedings.neurips.cc/paper_files/paper/2019/hash/ec1c59141046cd1866bbbcd6ae31d4-Abstract.html
Local Model Poisoning Attacks to Byzantine–Robust Federated Learning	攻击	投毒 攻击	202 0	USENIX Security	https://www.usenix.org/conference/usenixsecurity20/presentation/fang
Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for FL	攻击	投毒 攻击	202 1	NDS	https://par.nsf.gov/servlets/purl/10286354
MPAF: Model Poisoning Attacks to Federated Learning based on Fake Clients	攻击	投毒 攻击	202 2	CVPR	https://openaccess.thecvf.com/content/CVPR2022W/FedVision/html/Cao_MPAF_Model_Poisoning_Attacks_to_Federated_Learning_Based_on_Fake_CVPRW_2022_paper.html
3dfed: Adaptive and extensible framework for covert backdoor attack in federated learning	攻击	后门 攻击	202 3	S&P	https://ieeexplore.ieee.org/document/10179401
Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent	防御	拜占庭攻击	201 7	NIPS	https://proceedings.neurips.cc/paper/2017/hash/f4b9ec30ad9f68f89b29639786cb62ef-Abstract.html
Byzantine–Robust Distributed Learning: Towards Optimal Statistical Rates	防御	拜占庭攻击	201 8	ICML	https://proceedings.mlr.press/v80/yin18a

The Hidden Vulnerability of Distributed Learning in Byzantium	防御	拜占庭攻击	201 8	ICM L	https://proceedings.mlr.press/v80/mhamdi18a.html
FLTrust: Byzantine–robust Federated Learning via Trust Bootstrapping	防御	投毒攻击	202 1	NDS S	https://par.nsf.gov/servlets/purl/10248837
FLDetector: Defending Federated Learning Against Model Poisoning Attacks via Detecting Malicious Clients	防御	投毒攻击	202 2	KDD	https://dl.acm.org/doi/abs/10.1145/3534678.3539231
FLAME: Taming Backdoors in Federated Learning	防御	后门攻击	202 2	Use nix Sec urity	https://www.usenix.org/conference/usenixsecurity22/fall-accepted-papers
Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for FL	防御	投毒攻击	202 1	NDS S	https://par.nsf.gov/servlets/purl/10286354
Byzantine–robust Federated Learning through Collaborative Malicious Gradient Filtering	防御	拜占庭攻击	202 2	ICD CS	https://www.computer.org/csdl/proceedings/icdcs/2022/717700b223/1Hrj5rNmE6I
DeepSight: Mitigating Backdoor Attacks in Federated Learning Through Deep Model Inspection	防御	后门攻击	202 2	NDS S	https://www.ndss-symposium.org/wp-content/uploads/2022-156-paper.pdf

1.constrain-and-scale

How To Backdoor Federated Learning (AISTATS2020)

联邦学习容易受到模型中毒攻击，这种攻击比仅针对训练数据的中毒攻击更强大。恶意参与者可以使用模型替换将后门功能引入联合模型。这些攻击可以由单个参与者或多个共谋参与者执行。本文在标准联邦学习任务的不同假设下评估模型替换，并表明它的性能大大优于训练数据中毒。联邦学习采用安全聚合来保护参与者本地模型的机密性，因此无法通过检测参与者对联合模型贡献的异常来防止我们的攻击。为了证明异常检测不会在所有情况下有效，本文还开发并评估了一种通用的约束和缩放技术，该技术将防御的规避纳入攻击者在训练期间的损失函数中。



Algorithm 2 Attacker uses this method to create a model that does not look anomalous and replaces the global model after averaging with the other participants' models.

Constrain-and-scale($\mathcal{D}_{local}, D_{backdoor}$)

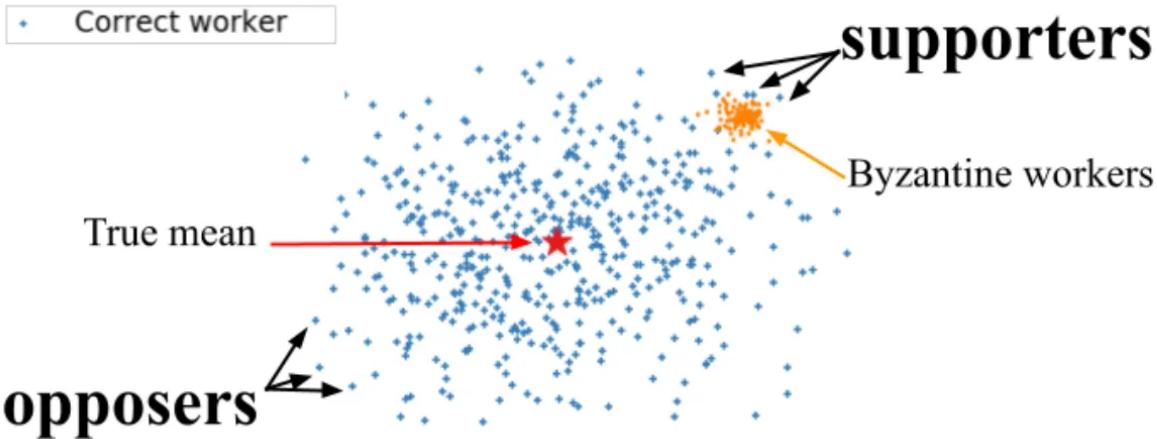
Initialize attacker's model X and loss function l :

```
 $X \leftarrow G^t$ 
 $\ell \leftarrow \alpha \cdot \mathcal{L}_{class} + (1 - \alpha) \cdot \mathcal{L}_{ano}$ 
for epoch  $e \in E_{adv}$  do
    if  $\mathcal{L}_{class}(X, D_{backdoor}) < \epsilon$  then
        // Early stop, if model converges
        break
    end if
    for batch  $b \in \mathcal{D}_{local}$  do
         $b \leftarrow \text{replace}(c, b, D_{backdoor})$ 
         $X \leftarrow X - lr_{adv} \cdot \nabla \ell(X, b)$ 
    end for
    if epoch  $e \in step\_sched$  then
         $lr_{adv} \leftarrow lr_{adv}/step\_rate$ 
    end if
end for
// Scale up the model before submission.
 $\tilde{L}^{t+1} \leftarrow \gamma(X - G^t) + G^t$ 
return  $\tilde{L}^{t+1}$ 
```

2.non-omniscient attack

A Little Is Enough: Circumventing Defenses For Distributed Learning(NIPS2019)

分布式学习中，先前的拜占庭攻击模型假设恶意参与者 (a) 无所不知（知道所有其他参与者的数据），并且 (b) 对参数进行较大的更改。本文提出，如果参与方模型梯度之间的经验方差足够高，攻击者就可以利用这一点并在群体方差内发起非全知攻击。



考虑了Krum, Bulyan, Trimmed mean攻击方式，针对不同的防御措施分别修改攻击方式。

针对Trimmed mean：控制中位数，将恶意节点值的范围控制在正态分布内。

针对Krum和Bulyan:生成一组参数，这些参数与每个参数的平均值相差很小。

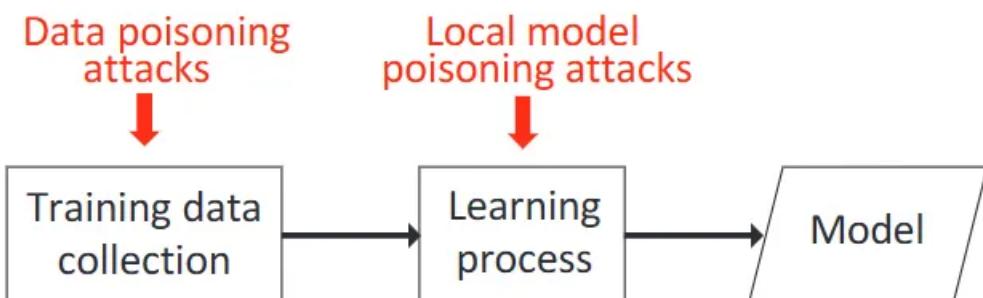
阻止收敛：分析参数变化不会被TrimmedMean检测到的范围，并在选择该范围的最大值时阻止收敛。

后门攻击：类似针对Trimmed mean的方式，在这个范围内找一组参数实现后门任务。

3.local model poisoning attacks

Local Model Poisoning Attacks to Byzantine–Robust Federated Learning(Usenix2020)

本文对联邦学习的局部模型中毒攻击进行了首次系统研究。与投毒训练数据集的数据投毒攻击不同，本文的目标是损害训练阶段学习过程的完整性。本文假设攻击者控制了一些工作设备，并在学习过程中操纵从这些设备发送到主设备的本地模型参数。攻击者可能知道也可能不知道主设备使用的聚合规则。为了与数据中毒攻击形成对比，本文将提出的攻击称为本地模型中毒攻击，因为它们直接操纵本地模型参数。



创新点：

- 1、基于不同防御方法，设计了具有针对性的模型攻击方式
- 2、概括了基于错误率以及基于损失函数的防御方法，测试了两种防御方法的效果。

方法：

攻击场景：training phase中对基于局部训练数据的模型在训练过程中进行攻击

攻击者的要求：控制部分参与模型中的训练参数

本地模型攻击：

主要挑战：如何将被攻击的局部模型进行改造并发送至服务器

方法：对投毒后的局部模型进行约束，转化为每轮中的优化问题

定义优化：

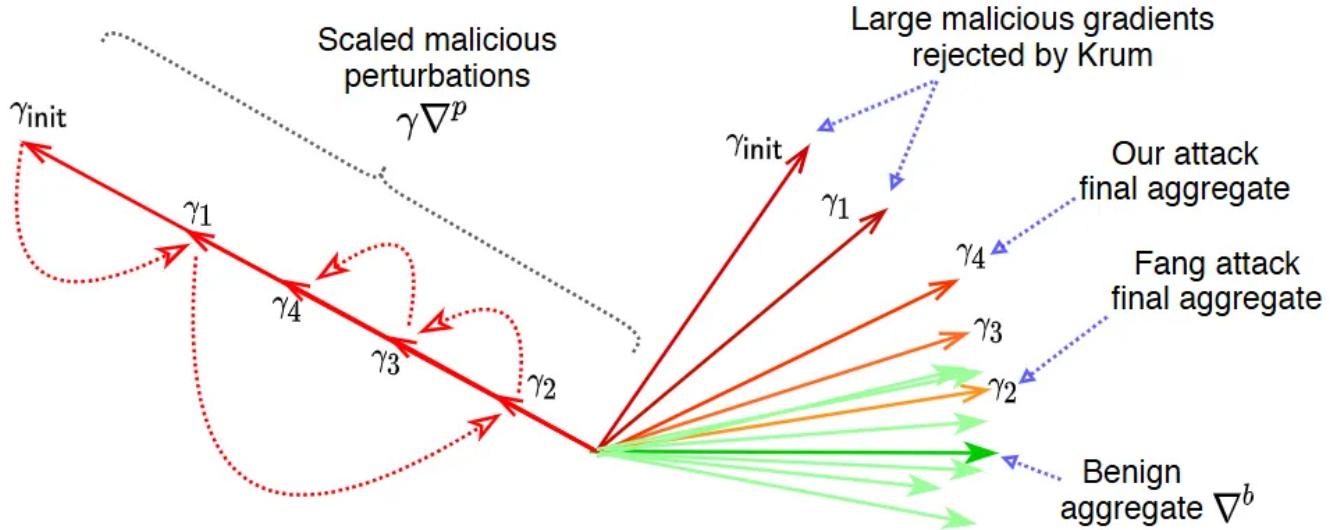
定义一个方向量，1表示当前梯度增加，-1表示当前梯度减小，其次定义攻击前的梯度与攻击后的梯度，那么优化问题的实质就是，使得攻击后的梯度与攻击前的梯度差别尽量大。

$$\begin{aligned} & \max_{\mathbf{w}'_1, \dots, \mathbf{w}'_c} \mathbf{s}^T (\mathbf{w} - \mathbf{w}'), \\ \text{subject to } & \mathbf{w} = \mathcal{A}(\mathbf{w}_1, \dots, \mathbf{w}_c, \mathbf{w}_{c+1}, \dots, \mathbf{w}_m), \\ & \mathbf{w}' = \mathcal{A}(\mathbf{w}'_1, \dots, \mathbf{w}'_c, \mathbf{w}_{c+1}, \dots, \mathbf{w}_m), \end{aligned}$$

4.ARG-attack

Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for FL (NDSS2021)

AGR-tailored attacks



(a) Our AGR-tailored attack (demonstrated for Krum)

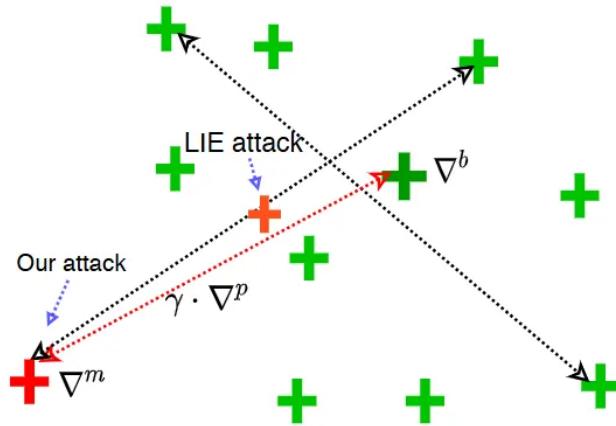
攻击者目标：选定 ∇^p 扰动向量后优化 scaling factor，从而最大化良性聚合模型和恶意聚合模型之间的距离

$$\begin{aligned} \operatorname{argmax}_{\gamma, \nabla^p} \quad & \|\nabla^b - f_{\text{agr}}(\nabla_{\{i \in [m]\}}^m \cup \nabla_{\{i \in [m+1, n]\}})\|_2 \\ \nabla_{i \in [m]}^m = & \nabla^b + \gamma \nabla^p; \quad \nabla^b = f_{\text{avg}}(\nabla_{\{i \in [n]\}}) \end{aligned}$$

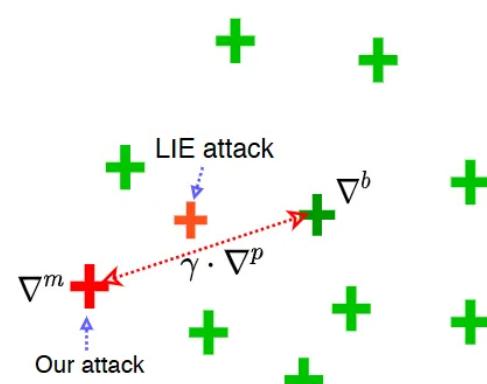
扰动向量 ∇^p 是梯度空间中的任何恶意方向，对手可以用它来扰动 ∇^b 并找到恶意梯度。在这篇论文中，实验了以下三种类型的扰动向量：

- Inverse unit vector:
- Inverse standard deviation:
- Inverse sign:

AGR-agnostic attacks



(b) Our AGR-agnostic Min-Max attack



(c) Our AGR-agnostic Min-Sum attack

- 攻击思想

① 基于距离的防御通过去除位于良性梯度形成的集团之外的梯度而起作用。因此，攻击需要最大化恶意梯度与参考良性梯度的距离，同时确保恶意梯度位于良性梯度集团内，这同时确保了恶意梯度和良性梯度的L_p范数是相似的。

② 为了确保分布的相似性，使用与良性梯度有相似分布的扰动向量。

Attack-1 (Min-Max): Minimize maximum distance attack

针对单个恶意梯度，恶意梯度与任何其他梯度的最大距离的上限为任意两个良性梯度之间的最大距离，确保单个恶意梯度接近于良性梯度的集团

$$\operatorname{argmax}_{\gamma} \max_{i \in [n]} \|\nabla^m - \nabla_i\|_2 \leq \max_{i, j \in [n]} \|\nabla_i - \nabla_j\|_2$$

$$\nabla^m = f_{\text{avg}}(\nabla_{\{i \in [n]\}}) + \gamma \nabla^p$$

Attack-2 (Min-Sum): Minimize sum of distances attack

恶意梯度与所有良性梯度的平方距离之和是良性梯度之间距离平方和的上限，保持所有恶意梯度相同，以获得最大的攻击影响

$$\operatorname{argmax}_{\gamma} \sum_{i \in [n]} \|\nabla^m - \nabla_i\|_2^2 \leq \max_{i \in [n]} \sum_{j \in [n]} \|\nabla_i - \nabla_j\|_2^2$$

$$\nabla^m = f_{\text{avg}}(\nabla_{\{i \in [n]\}}) + \gamma \nabla^p$$

5.MPAF

MPAF: Model Poisoning Attacks to Federated Learning based on Fake Clients(CVPR2022)

现有的联邦学习模型中毒攻击大多假设攻击者可以访问大部分受感染的真实客户端。这项工作提出了第一个基于假客户端的模型中毒攻击，称为 MPAF。假设攻击者将虚假客户端注入联邦学习系统，并在训练期间将精心设计的虚假本地模型更新发送到云服务器，从而使得学习到的全局模型对于许多不加区别的测试输入的准确性较低。为了实现这一目标，本文的攻击将全局模型拖向攻击者选择的精度较低的基本模型。具体来说，在每一轮联邦学习中，假客户端都会制作指向基本模型的假本地模型更新，并在将其发送到云服务器之前对其进行扩展以放大其影响。

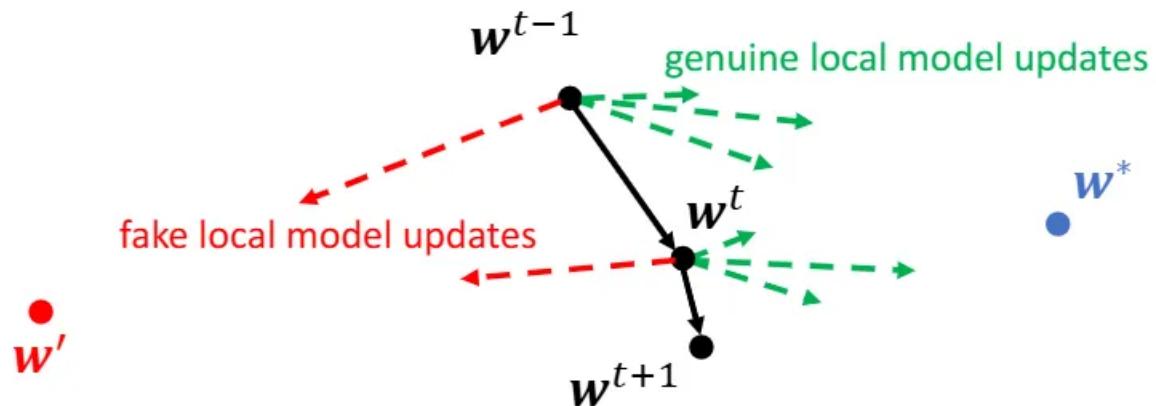


Figure 1. Illustration of MPAF. w' is an attacker-chosen base model. w^{t-1} , w^t , and w^{t+1} are the global models in round $t - 1$, t , and $t + 1$, respectively. w^* is the learnt global model without attack. The fake local model updates from the fake clients drag the global model towards the base model.

攻击者上传的梯度：

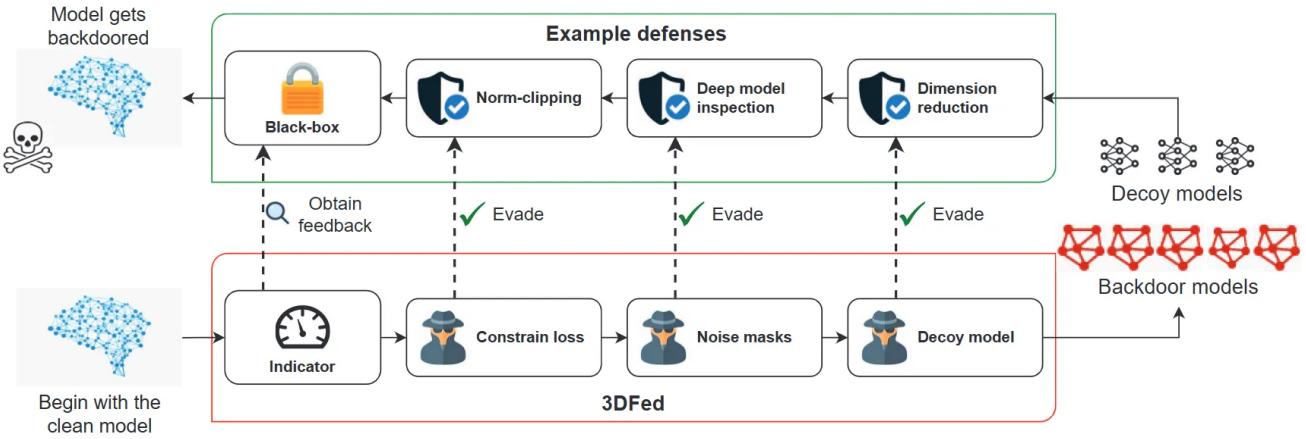
$$\mathbf{g}_i^t = \lambda(\mathbf{w}' - \mathbf{w}^t).$$

6.3DFed

3DFed: Adaptive and Extensible Framework for Covert Backdoor Attack in Federated Learning

联邦学习中，通过破坏或冒充设备，攻击者可以上传精心设计的恶意模型更新，以根据攻击者指定的触发器通过后门行为操纵全局模型。然而，现有的后门攻击需要有关受害 FL 系统的更多信息，而不仅仅是

实际的黑盒设置。此外，它们通常专门针对单个目标进行优化，但由于现代 FL 系统倾向于采用从不同角度检测后门模型以进行防御，这种攻击方法容易失效。本文提出了 3DFed，这是一种自适应、可扩展的多层框架，用于在黑盒设置中发起隐蔽的 FL 后门攻击。3DFed 具有三个伪装后门模型的规避模块：带有约束损失的后门训练、噪声掩模和诱饵模型。通过在后门模型中植入“指标”，3DFed 可以从全局模型中获取上一个 epoch 的攻击反馈，并动态调整这些后门规避模块的超参数。



带有约束损失的后门训练 (Constrained Loss Backdoor Training) : 这一模块的目的是在训练后门模型时，通过添加一个约束项来限制模型更新的幅度，从而对抗防御机制中的范数裁剪 (norm clipping)。具体来说，就是在损失函数中加入一个与全局模型和后门模型之间欧几里得距离成正比的项，这样训练出的后门模型在保持与全局模型相近的同时，也能够包含后门行为。这种方法可以使得后门攻击在面对范数裁剪防御时更加隐蔽和有效。

噪声掩码 (Noise Mask) : 噪声掩码的作用是为后门模型的参数添加噪声，以此来掩盖那些可能被检测机制识别出的异常特征。通过这种方式，即使后门模型的更新在一定程度上被噪声掩盖，当它们被聚合到全局模型中时，这些噪声会相互抵消，从而恢复后门的效果。噪声掩码的设计旨在减少后门模型在神经元更新集中的特征，并降低模型更新之间的余弦相似性，使得后门模型的分布更接近于良性模型。

诱饵模型 (Decoy Model) : 诱饵模型用于对抗使用降维技术（如主成分分析，PCA）的防御机制。在联邦学习中，降维技术可以用来将模型参数投影到低维空间，以便区分恶意模型和良性模型。诱饵模型通过与后门模型一起上传到中央聚合器，干扰降维过程，使得降维结果中的关键维度变为无用维度 (garbage dimensions)，从而使得后门模型能够在降维后的空间中隐藏起来，避免被检测出来。这种方法要求攻击者能够适应性地确定上传多少诱饵模型，以成功欺骗降维算法。

Defense

1.Krum

Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent(NIPS2017)

这篇文章针对在联邦学习中Byzantine worker，提出了防御方案。并在理论上证明了方案的Byzantine Resilience 和 Convergence。

1. Byzantine Worker

Byzantine worker是指在训练过程中可以随意作恶的worker。比如随意掉线，或者上传恶意、随机的更新梯度。从而达到是的模型优化无法收敛的目的。

Byzantine worker 可能上传随机的梯度，也肯能故意上传和正确梯度相反的梯度。无论何种方式，这都会导致模型divergence。

2. 方案 Krum

squared-distance-based :一种方案是求每一个 v_i 和其他所有 $v_j, j \neq i$ 的距离之和，最后选择和最小的一个作为聚合 F 。这种方案在只有一个Byzantine worker的时候是可行的，但是如果两个或者以上Byzantine worker合谋，那么他们可以让聚合结果和真实理想梯度 g 偏差很远。

majority-based :另外一种方案是求出所有大小为 $n-f$ 的梯度子集的直径，并选择直径最小的子集作为 F 。但是这种方案的计算复杂度是指数级别。

集合上述两种approach，作者设计了 Krum 。

3. 缺点：假设的联邦场景是IID（梯度直接聚合是收敛的）

2.Median、Trimmed_mean

Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates(ICML2018)

放弃直接平均的方式，通过选择中位数/截断后的平均值的方式进行聚合。

3.Bulyan

The Hidden Vulnerability of Distributed Learning in Byzantium(ICML2018)

在其他方法的基础上，逐步选择最优的client直到达数量要求。

Bulyan首先以MultiKrum的方式选择 $\theta(\theta \leq n-2m)$ 个梯度，然后计算选中梯度的Trimmed-mean值。

4.FLTrust

攻击模型: 攻击者控制着一些恶意客户端，这些客户端可能是攻击者注入的虚假客户端，也可能是被攻击者破坏的真实客户端。然而，攻击者不会控制服务器。恶意客户端可以在FL训练过程的每次迭代中向服务器发送任意的本地模型更新。攻击者知道FL训练过程的一切，包括每次迭代中所有客户端的本地训练数据和本地模型更新，以及FL的聚合规则。考虑这样的“全知识”设定，是为了证明此方法可以抵御较强的攻击。

防御目标:

Fidelity: 在没有攻击时，应实现全局模型的高精度。

Robustness: 应该在恶意客户端进行强中毒攻击的情况下，保持全局模型的精度。

Efficiency: 该方法不应引起额外的计算和通信开销。

防御者的知识和能力的设定:

认为防御是在服务器端执行的。服务器不能访问客户机上的原始本地训练数据，并且服务器不知道恶意客户机的数量。然而，服务器在每次迭代中都可以完全访问全局模型以及来自所有客户机的本地模型更新。此外，服务器本身可以为学习任务收集一个干净的小训练数据集(称之为根数据集)。服务器可以通过手动标记收集干净的根数据集，根数据集是没有中毒的。

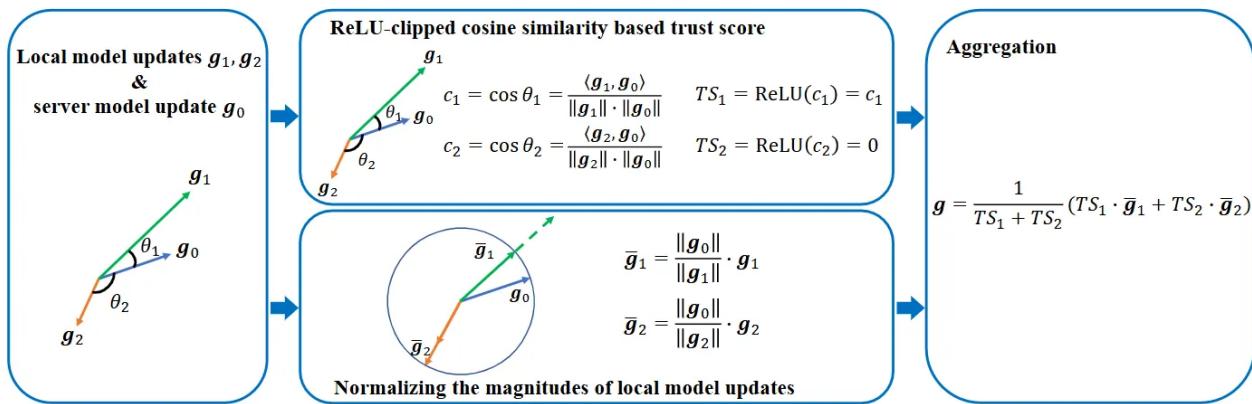


Fig. 2: Illustration of our aggregation rule, which is applied in each iteration of FLTrust.

FLTrust有三个关键特征:一个根数据集，使用ReLU剪辑余弦相似度评分，以及标准化每个本地模型更新。

新聚合规则同时考虑了本地模型更新和服务器模型更新的方向和大小，以计算全局模型更新。

ReLU-clipped cosine similarity based trust score:

攻击者可以操纵恶意客户端的本地模型更新的方向，从而将全局模型更新驱动到攻击者想要的任意方向。没有可信根，服务器很难决定哪个方向更有希望更新全局模型。在FLTrust中，根信任来自服务器模型更新的方向。特别是，如果本地模型更新的方向与服务器模型更新的方向更相似，那么本地模型更新的

方向更有望为安全更新。在形式上，使用余弦相似度度量两个向量之间的角度，来计算本地模型更新和服务器模型更新之间的方向相似度。

然而，如果本地模型更新和服务器模型更新方向相反，它们的余弦相似度是负的，这仍然会对聚合的全局模型更新产生负面影响。因此，通过使用ReLU裁剪余弦相似度来将这种本地模型更新从聚合中排除。

Normalizing the magnitudes of local model updates:

攻击者还可以将恶意客户端上的本地模型更新的幅度放大，从而控制全局模型更新。因此，对每个本地模型更新的大小进行归一化处理。对每个本地模型更新进行标准化，使其具有与服务器模型更新相同的量级。这种规范化意味着将本地模型更新调整为服务器模型更新向量空间中的同一个域。

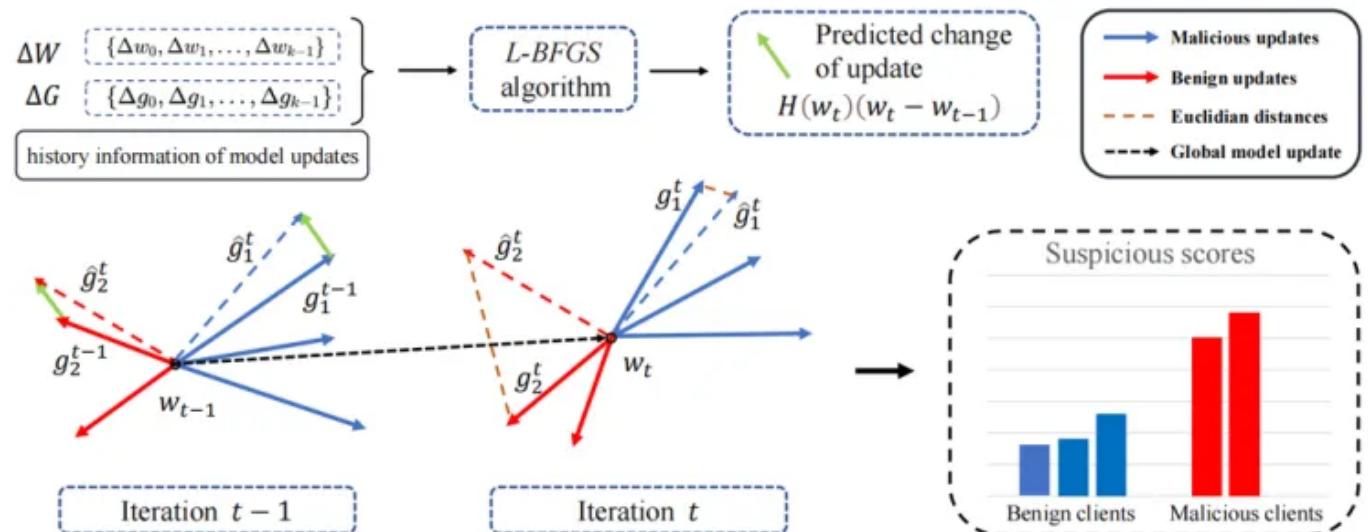
Aggregating the local model updates:

计算归一化的本地模型更新的平均值，以其信任分数加权作为全局模型更新， g 是全局模型更新。最后，对全局模型 w 进行更新， α 是全局学习率

5.FLDetector

FLDetector: Defending Federated Learning Against Model Poisoning Attacks via Detecting Malicious Clients(KDD2022)

FLDetector 通过检测恶意客户端来防御大量恶意客户端的模型中毒攻击。FLDetector 旨在检测并删除大多数恶意客户端，以便拜占庭稳健或可证明稳健的 FL 方法可以使用剩余客户端学习准确的全局模型。通过观察，在模型中毒攻击中，多次迭代中来自客户端的模型更新是不一致的。因此，FLDetector 通过检查模型更新的一致性来检测恶意客户端。粗略地说，服务器根据历史模型更新来预测每次迭代中客户端的模型更新，如果在多次迭代中从客户端接收到的模型更新与预测的模型更新不一致，则将客户端标记为恶意客户端。



通过检测模型梯度上传的时序一致性来鉴别出恶意节点。这样可以在剔除这些恶意节点之后通过训练得到一个比较好的模型。

6.FLAME

FLAME: Taming Backdoors in Federated Learning(Usenix Security2022)

1. 过滤掉动态场景中角度偏差较大的逆向模型

- 与现有的基于集群的防御相比，我们需要一种也可以在动态攻击设置中工作的方法，即注入后门的数量是未知的，并且可能在训练轮次之间变化。
- 使用固定数量的集群 $n_{cluster}$ 来识别恶意模型的集群方法天生容易受到不同数量的后门 $n_{backdoor}$ 的攻击。这是因为对手通过同时注入 $n_{backdoor} \geq n_{cluster}$ 后门，由于鸽子洞原理，可能导致至少一个后门模型与良性模型聚集在一起。本文试图通过采用集群解决方案来解决这一挑战，该解决方案动态地确定用于模型更新的集群，从而允许其适应动态攻击。

2. 限制扩大后门的影响

- 通过按比例缩小权重向量，将幅度超过裁剪界限 S 的所有模型（特别是后门模型 $W'2W2'$ ）的权重向量裁剪到 S
- 如何选择适当的裁剪边界，而不凭经验评估其对训练数据集（在FL设置中不可用）的影响？

3. 为后门消除选择合适的噪声级

- FLAME 使用模型噪声，该模型噪声应用具有噪声水平 σ 的高斯噪声，以减轻后门模型的对抗性影响。
- 必须仔细选择噪声水平 σ ，因为它对防御的有效性和模型的良性性能有直接影响。

7.DnC

Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for FL (NDSS2021)

为了提高抗中毒的鲁棒性，需要降低输入梯度的维数；仅提供收敛保证是不够的，鲁棒性聚合应该保证它们检测和去除异常值的能力；鲁棒性聚合需要超越当前仅使用基于维度/距离的过滤方法。

DnC利用基于奇异值分解(singular value decomposition, SVD)的谱方法来检测和去除异常值。以前的工作已经证明了这些方法在减轻针对集中学习的数据中毒方面的理论和经验性能。但是，在普通FL设置中直接在高维梯度上执行SVD开销巨大，所以DnC通过对其输入梯度进行随机抽样来降低维数。此外，还构建了针对DnC的自适应攻击，以证明其鲁棒性。DnC算法如下所示：

Algorithm 2 Our Divide-and-Conquer AGR Algorithm

- 1: **Input:** Input gradients $\nabla_{\{i \in [n]\}}$, filtering fraction c , number of malicious clients m , niters, dimension of subsamples b , input gradients dimension d
 - 2: $\mathcal{I}_{\text{good}} \leftarrow \emptyset$
 - 3: **while** $i < \text{nitters}$ **do**
 - 4: $r \leftarrow$ sorted set of size b of random dimensions $\leq d$
 - 5: $\tilde{\nabla}_{\{i \in [n]\}} \leftarrow$ set of gradients subsampled using indices in r
 - 6: $\mu = \frac{1}{n} \sum_{i \in [n]} \tilde{\nabla}_i$ ▷ Compute mean of input gradients
 - 7: $\nabla^c = \tilde{\nabla}_{\{i \in [n]\}} - \mu$ ▷ ∇^c is a $n \times b$ matrix of centered input gradients
 - 8: Compute v , the top right singular eigenvector of ∇^c
 - 9: Compute *outlier scores* defined as $s_i = (\langle \nabla_i - \mu, v \rangle)^2$
 - 10: $\mathcal{I} \leftarrow$ Set of $(n - c \cdot m)$ indices of the gradients with lowest outlier scores from s
 - 11: Append \mathcal{I} to $\mathcal{I}_{\text{good}}$
 - 12: $i = i + 1$
 - 13: **end while**
 - 14: $\mathcal{I}_{\text{final}} \leftarrow \cap \mathcal{I}_{\text{good}}$ ▷ Compute intersection of sets in $\mathcal{I}_{\text{good}}$ as the final set of indices
 - 15: $\nabla_a = \frac{1}{|\mathcal{I}_{\text{final}}|} \sum_{i \in \mathcal{I}_{\text{final}}} \nabla_i$
 - 16: **Output** ∇_a
-

8.SignGuard

Byzantine-robust Federated Learning through Collaborative Malicious Gradient Filtering(ICDCS2022)

本文指出辅助数据在实践中可能并不总是可用，并重点关注基于统计的方法。本文证明了梯度向量的逐元素符号可以为检测模型中毒攻击提供有价值的见解。基于对 Little is Enough 攻击的理论分析，提出了一种名为 SignGuard 的新颖方法，通过协作恶意梯度过滤实现拜占庭式鲁棒联邦学习。更准确地说，首先处理接收到的梯度以生成相关的幅度、符号和相似性统计数据，然后由多个过滤器协作使用，以在

最终聚合之前消除恶意梯度。最后，在最近提出的攻击和防御策略下进行了图像和文本分类任务的广泛实验。数值结果证明了我们提出的方法的有效性和优越性。

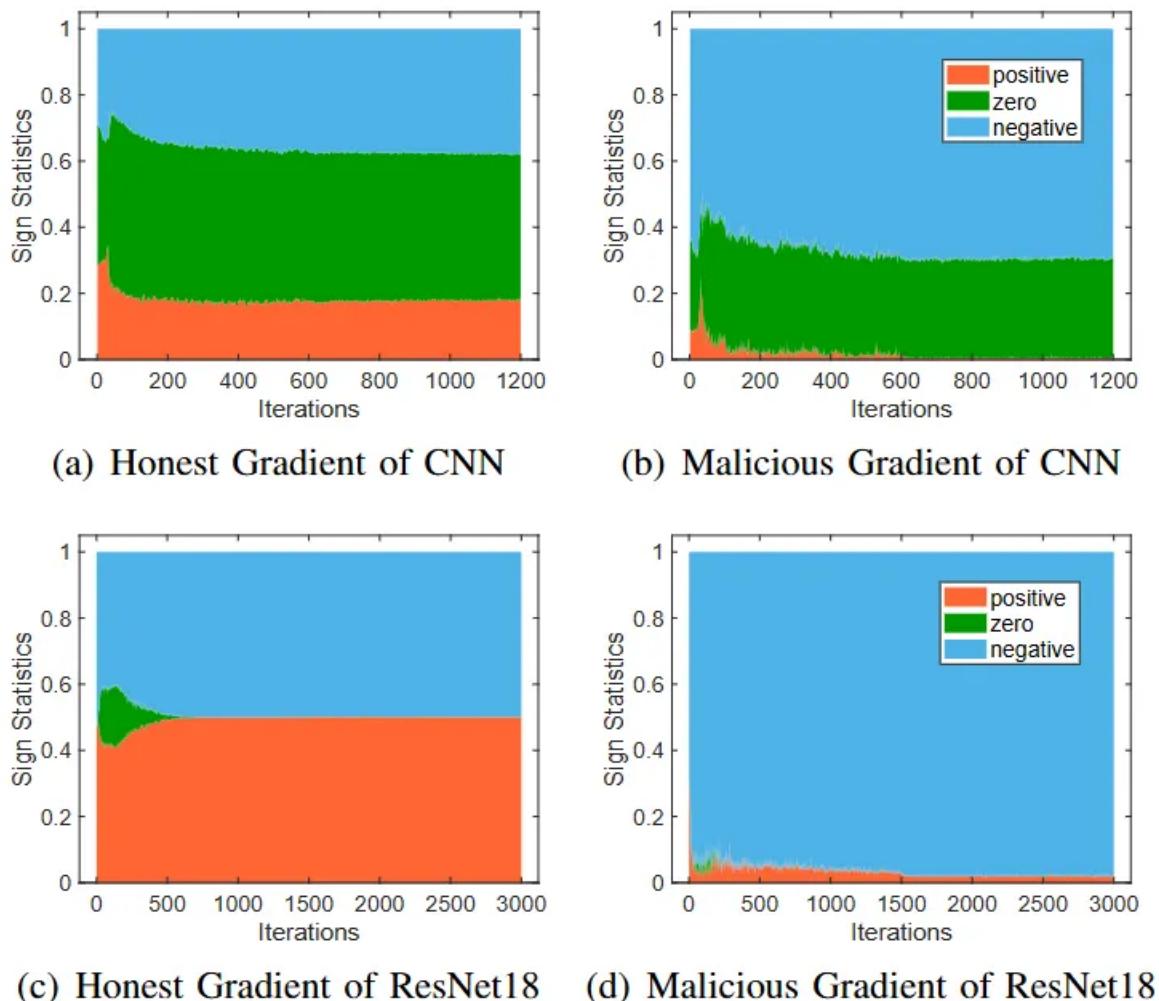
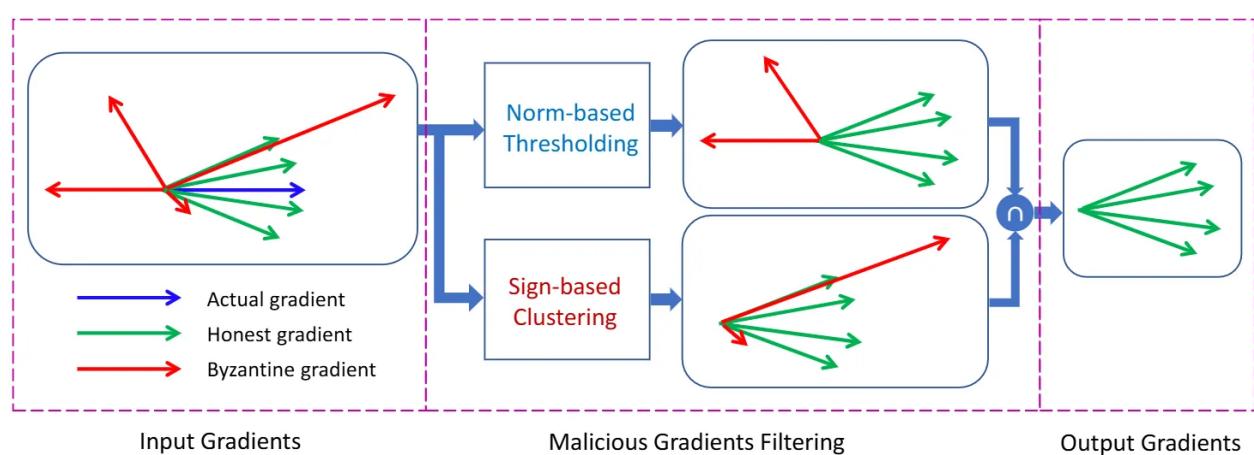


Fig. 2. Sign statistics of honest and malicious gradient.



Algorithm 2 SignGuard Function

- 1: **Input:** Set of received gradients $S_t = \{g_t^{(i)}\}_{i=1}^n$, lower and upper bound L, R for gradient norm
 - 2: **Initial:** $S_1 = S_2 = \emptyset$
 - 3: Get l_2 -norm and element-wise sign of each gradient
 - 4: **Step 1:** Norm-based Filtering
 - 5: Get the median of norm $M = med(\{\|g_t^{(i)}\|\}_{i=1}^n)$
 - 6: Add the gradient that satisfies $L \leq \frac{\|g_t^{(i)}\|}{M} \leq R$ into S_1
 - 7: **Step 2:** Sign-based Clustering
 - 8: Randomly select a subset of gradient coordinates
 - 9: Compute sign statistics on selected coordinates for each gradient as features
 - 10: Train a Mean-Shift clustering model
 - 11: Choose the cluster with most elements as S_2
 - 12: **Step 3:** Aggregation
 - 13: Get trusted set: $S'_t = S_1 \cap S_2$
 - 14: Get $\tilde{g}_t = \frac{1}{|S'_t|} \sum_{i \in S'_t} g_t^{(i)} \cdot \min\left(1, M/\|g_t^{(i)}\|\right)$
 - 15: **Output:** Global gradient: \tilde{g}_t
-

9. DeepSight

DeepSight: Mitigating Backdoor Attacks in Federated Learning Through Deep Model Inspection(NDSS2022)

现有的针对后门攻击的对策效率低下，并且通常只是旨在从聚合中排除偏离的模型。然而，这种方法也会删除具有偏差数据分布的客户端的良性模型，导致聚合模型对于此类客户端表现不佳。本文提出 DeepSight，基于三种新技术，可以表征用于训练模型更新的数据分布，并寻求测量神经网络内部结构和输出的细粒度差异。使用这些技术，DeepSight 可以识别可疑的模型。更本文我们还开发了一种可以准确聚类模型更新的方案。结合两个组件的结果，DeepSight 能够识别并消除包含具有高攻击影响的中毒模型的模型集群。

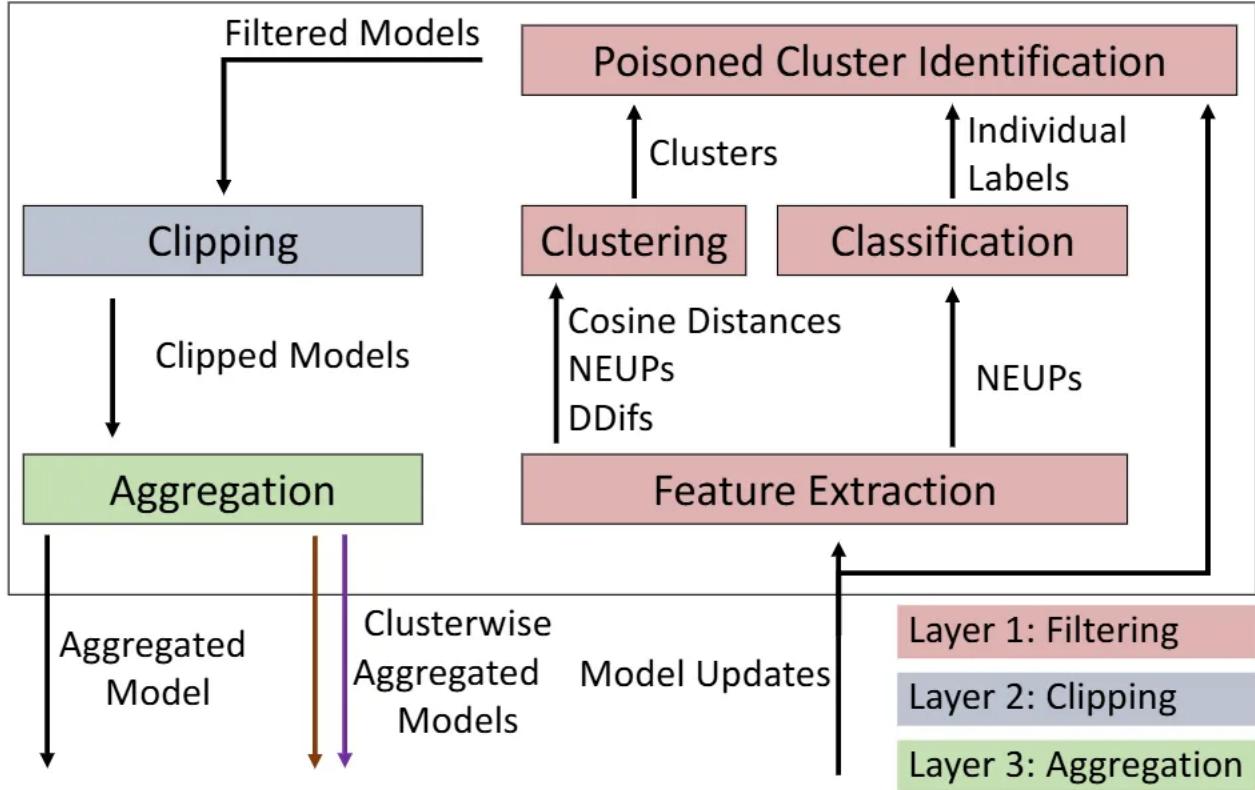


Fig. 2: Structure of DeepSight

总框架：分过滤层、裁剪层、聚合层

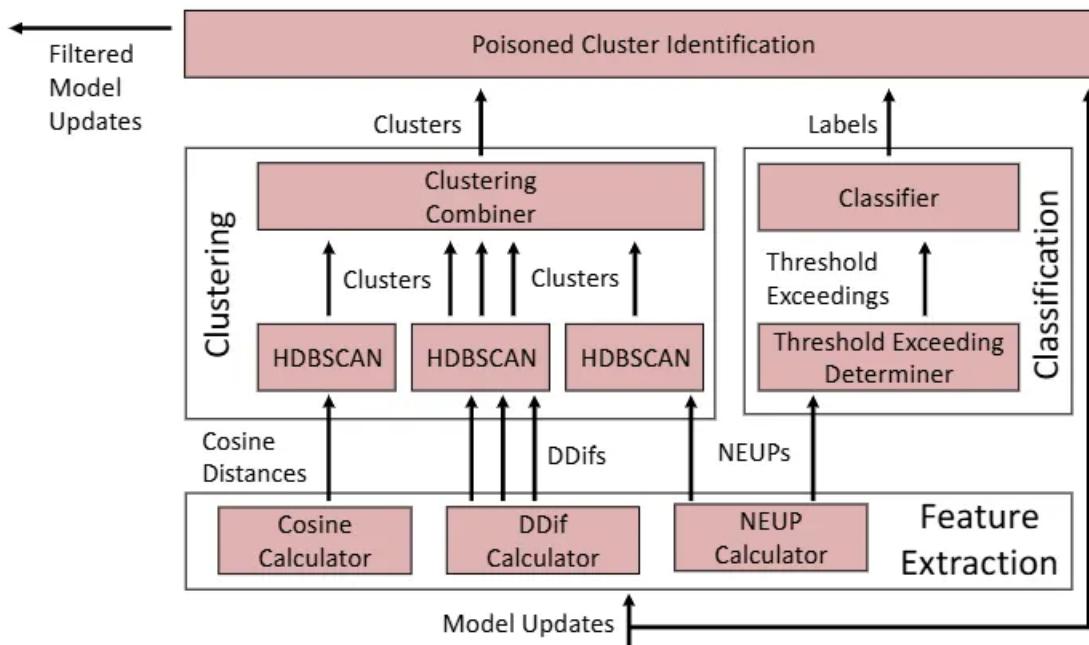


Fig. 4: Filtering Layer of DeepSight

过滤层：包括

1. 特征提取。使用了余弦相似度、DDif（DDif 衡量标签预测分数变化，其中提供有关各个客户端的训练标签分布的信息）、NEUP（标准化更新能量。它分析输出层的参数更新，并提取有关模型底层训练数据中标签分布的信息。）
2. 分类。对NEUP计算结果设定TE，标记可疑更新
3. 聚类。以余弦相似度、DDif、NEUP为特征，使用HDBSCAN对客户端进行聚类
4. 对上述结果进行分析

裁剪层：防止Scaling Attack

聚合层：对同一簇的模型聚合并返回（不会得到统一的全局模型）

10. FreqFed

FreqFed: A Frequency Analysis-Based Approach for Mitigating Poisoning Attack(NDSS2024)

FL 中现有的针对中毒攻击的防御措施有一些局限性，例如依赖于有关攻击类型和策略或数据分布的特定假设，或者对于先进的注入技术和策略不够鲁棒，同时保持聚合模型的实用性。为了解决现有防御的缺陷，本文采用一种通用且完全不同的方法来检测中毒（有针对性的和无针对性的）攻击。本文提出的 FreqFed 将模型更新（即权重）转换到频域，在频域中可以识别继承足够权重信息的核心频率分量。这能够在客户端本地训练期间有效过滤掉恶意更新，无论攻击类型、策略和客户端的数据分布如何。

设计直觉：频域中本地模型更新的频率分析将使我们能够识别恶意更新特有的模式。与良性更新相比，旨在向全局模型引入后门或影响全局模型整体性能的恶意更新的特点是低频分量的差异

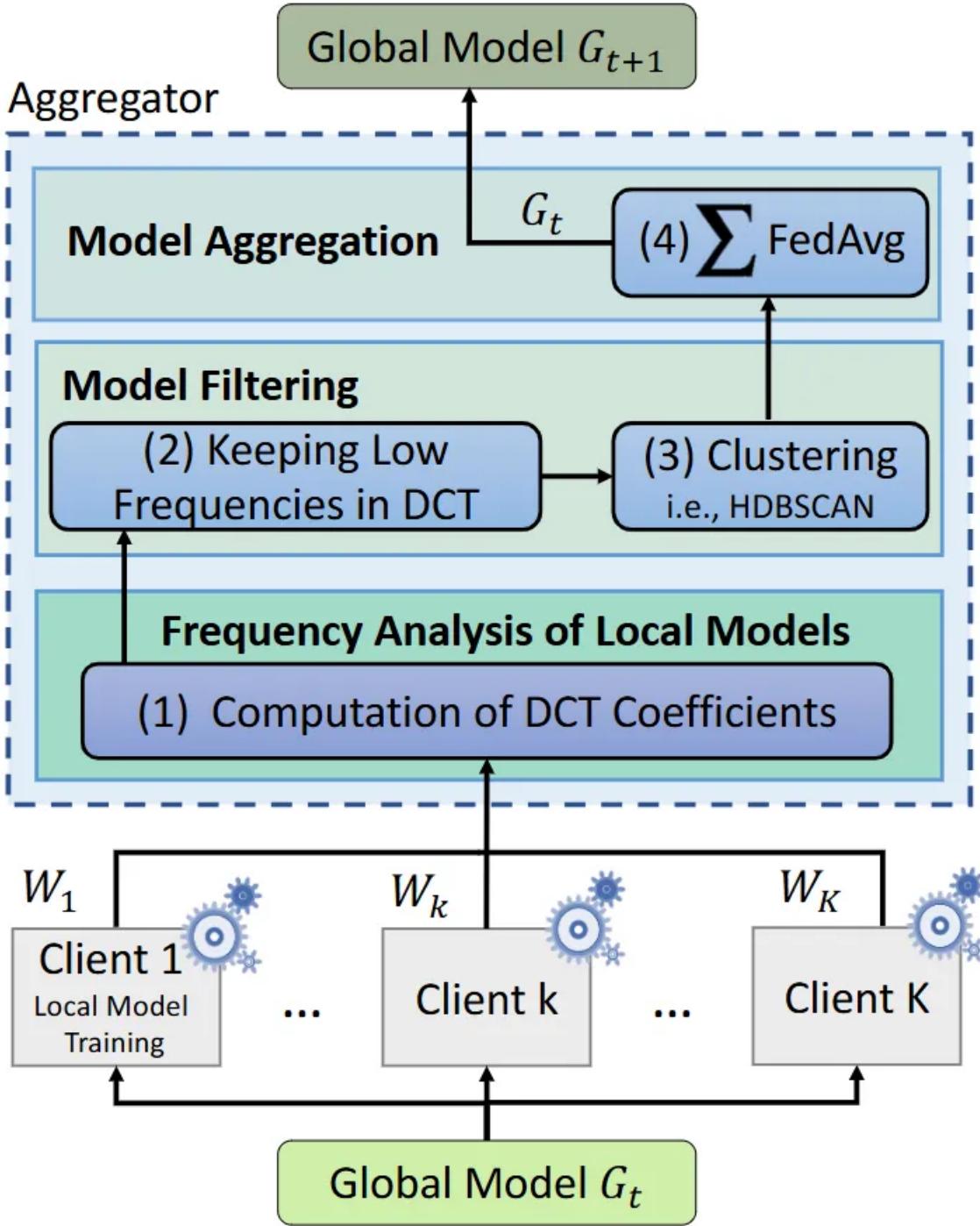


Fig. 1: System Overview of *FreqFed*

FreqFed 包含三个关键组件，分别用于 i) 本地模型更新的频率分析、ii) 模型过滤和 iii) 模型聚合。

本地模型更新的频率分析从参与联邦学习过程的每个客户端获取模型更新，并随后使用离散余弦变换将它们转换到频域。此过程用于识别更新中的主导频率，这可以随后用于模型过滤。模型滤波负责处理 DCT 系数矩阵并提取低频分量，然后将其存储在向量中。来自所有客户端的低频分量向量随后被传递到聚类算法，该算法根据余弦距离将它们分组为聚类，并选择具有最多向量的聚类。此过程用于识别要更新的客户端，随后可用于模型聚合。

