DeepSight：通过深度模型检测缓解联邦学习中的后门攻击

摘要

联邦学习（FL）允许多个客户端在不泄露数据的情况下，使用其私有数据协同训练一个神经网络（NN）模型。最近，出现了几种针对联邦学习的定向投毒攻击。这些攻击会在得到的模型中注入后门，使对手控制的输入被错误分类。针对后门攻击的现有对策效率不高，通常只是将偏差模型排除在聚合之外。然而，这种方法也会删除数据分布有偏差的客户的良性模型，导致聚合模型对这类客户的表现不佳。

为了解决这些问题，文章提出了一种名为 DeepSight 的用于缓解后门攻击的新型模型过滤方法。该方法基于三种新技术，可确定用于训练模型更新的数据分布特征，并设法测量神经网络内部结构和输出的细微差别。利用这些技术，DeepSight 可以识别可疑的模型更新。我们还开发了一种能够对模型更新进行精确聚类的方案。结合这两个部分的结果，DeepSight 能够识别并消除包含具有高攻击影响的投毒模型的模型集群。我们还表明，现有的基于权重剪切的防御措施可以有效地减轻可能未检测到的投毒模型的后门贡献。我们对 DeepSight 的性能和有效性进行了评估，结果表明它可以缓解最先进的后门攻击，而且对模型在良性数据上的性能影响微乎其微。

关键词：深度学习；联邦学习；投毒；后门；模型检测；

1 引言

联邦学习（FL）使多个客户端能够协同训练一个神经网络（NN）模型。这是通过一个迭代过程来实现的，在这个过程中，客户端使用自己的数据在本地训练自己的模型，并将训练好的模型更新发送到中央服务器，中央服务器将这些更新汇总起来，并将生成的全局模型分发给所有客户端。联邦方法不需要客户端将数据进行传输，保证了客户端训练数据的私密性。服务器将模型训练被并行化并外包给了客户端，降低了计算成本。这些优势使 FL 非常有用，尤其是在对隐私敏感的数据应用中，如或网络入侵检测系统（NIDS）[1]。

后门攻击。另一方面，服务器无法控制参与客户端的训练过程。对手可以入侵客户端的一个子集，并利用它们向聚合模型注入后门。在上述示例中，对手的目的是使聚合模型将恶意软件网络流量模式归类为良性，以避免被网络信息安全系统检测到，或者在 NLP 的情况下，操纵文本预测模型提出特定的品牌名称，以不显眼的方式为其做广告。最近，有人提出了各种针对性投毒的攻击策略，即所谓的后门攻击，利用被控制的客户端提交投毒模型更新 [2–6]。

攻击的困境：攻击者可以任意选择攻击策略：一方面，它可以使用高比例的投毒数据来训练后门任务。但是，这会导致投毒模型与良性模型差距较大，使得基于过滤的防御很容易

检测到投毒模型。另一方面，如果敌方不采取这种策略，由于投毒模型的数量少于良性模型，因此任何限制单个模型影响的防御都能轻松缓解攻击。因此，结合这两种防御策略会给对手造成进退两难的局面：要么攻击被一部分防御过滤掉，要么另一部分防御使攻击的影响变得微不足道 [7]。但是这两种防御策略的结合并不天然地有效，因为现有的过滤机制遵循离群点检测策略 [4, 7–9]，也会过滤数据分布偏差较大的良性模型。因此，大量客户端被错误地排除在外，导致针对其数据的聚合模型性能下降。

本文提出了 DeepSight，一种新颖的模型过滤方法，可深入检查神经网络的内部结构和输出，以识别具有高攻击影响的恶意模型更新，同时保留良性模型更新，即使这些更新来自数据分布偏差较大的客户端。利用攻击者的困境，将过滤方案与剪切相结合，最终达到减轻和防御后门攻击的目的。

2 相关工作

2.1 基于异常检测的方法

许多后门防御系统采用基于离群点检测的策略，排除异常模式更新 [4, 8–14]。这些方法假设所有良性客户端的本地数据是相似度，也就是独立同分布（iid）的。在现实情况下，不同客户端的数据应该是非独立同分布的（non-iid） [15, 16]，这就使得良性模型之间彼此差异比较大。基于离群点的检测会导致部分良性模型被排除在聚合之外，最终导致生成的模型在本地客户端的表现较差。

Krum 聚合本地模型的方法 [8] 是基于模型之间的欧氏距离来选取模型的。Krum 通过 k 个最靠近的模型更新确定阈值，再使用阈值进行离群点的排除。

Munoz 等人使用本地模型和全局模型的余弦距离的中位距离加减标准差来排除模型 [9]，这种方法假阳性率很高。

Baffle 等人将聚合模型发送给随机客户端子集使用本地数据对模型进行评估 [17]，并投票决定是否拒绝模型。这种方法只有当该客户集的数据量足够，或者后门攻击造成足够大的影响时，才能发现后门，这样的设置显然不符合实际情况。此外这种方法不能在训练的开始阶段使用，否则会出现很多误报。

Auror 方法使用 k-means 聚类对本地模型的每个参数进行分别聚类 [4]，并利用模型预测结果中的梯度信息来检测，并使用模型预测结果中的梯度信息来检测恶意客户端。此外，Auror 还侧重于排除异常值。

FLGuard 利用了后门攻击的攻击效果和攻击影响的权衡，即要么使得后门攻击隐蔽但是攻击效果不佳，要么使用高范数的攻击但是易被发现 [7]。将基于离群值的聚类与剪切和向模型中添加随机噪音相结合，但是这种方法会将离群的良性模型的更新错误低拒绝。

Liu 等人提出了另一种在集中式环境中检测后门模型的方法 [18]。然而，他们的方法并不适合 FL 设置，也没有考虑语义后门攻击。

2.2 其他防御方法

FoolsGold [15] 假设每个客户端彼此之间数据分布不相同，为与其他许多模型相似的模型分配较低的权重。这种方法会导致许多良性客户端收到影响。此外，FoolsGold 会汇总所有

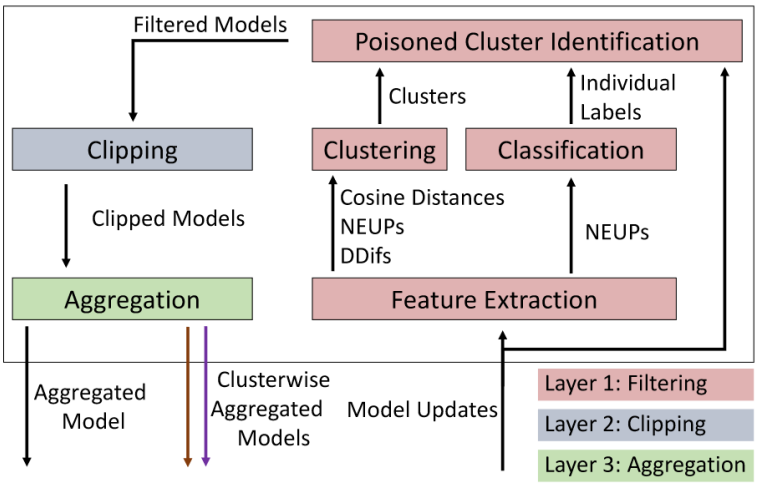


图 1. DeepSight 的结构

轮次的更新并进行比较，而不是只关注当前轮次，在训练开始阶段就投毒的恶意客户端可以有效地避开防御。

差分隐私 [19] 对模型更新强制执行 L2 范数裁剪，并添加随机生成的噪声。Bagdasaryan 等人所指出的，它还具有减轻后门攻击的附加作用，但是效果比较有限。

其他方法 [10, 14] 计算所有参数的中位数，这种方法关注了大多数模型，但是同样也忽略了不同数据分布的良性模型，并且对减轻后门攻击的效果有限。

Cao 等人的方法是训练多个模型 [20]。对于每个模型，都会使用一个随机客户子集进行多轮模型训练。模型进行推理任务时，使用每个生成的全局模型进行推理，并通过多数投票确定最终预测结果。Cao 等人证明，如果完成了训练，他们就能为特定样本确定其算法所能容忍的恶意客户端的最小数量。但是实际上，确定这个数字需要后验知识，此时模型已经训练完成，后门攻击已经完成。此外无法确定所考虑样本的当前标签是正确的，还是已经发生了后门攻击并翻转了标签。最后，该方法受到 Bagdasaryan 等人 [2] 的替换缩放攻击和 MA 破坏。

2.3 联邦学习中的成员推理攻击

用于识别特定样本存在的成员推理攻击 [21, 22]，可以被用来识别投毒的模型更新。但是这些方法并不有效，因为良性样本和投毒样本可能会重叠。其他方法要求攻击者拥有自己的训练数据 [23]，这对 FL 服务器来说并不实用。

3 本文方法

本文提出的 DeepSight 是一种可减轻对联邦学习（FL）的定向投毒（后门）攻击的方法。它采样基于投票的过滤的方法，结合了分类器和基于聚类的相似性估计进行深度模型检查，并结合模型裁剪来识别和缓解定向投毒攻击。具体的方法流程如图 1所示。DeepSight 满足了投毒缓解、不干扰训练过程以及自主运行的要求，实现了区分投毒模型和基于不同数据训练的良性模型。

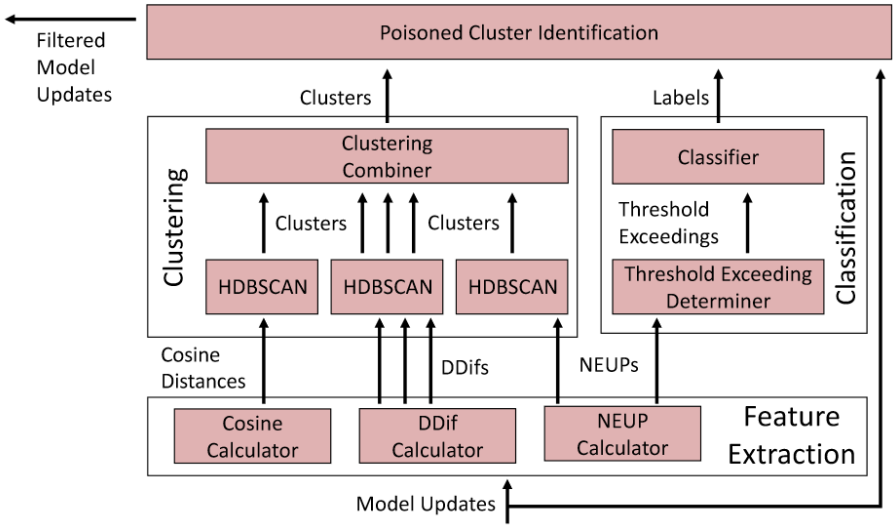


图 2. DeepSight 的过滤层

3.1 过滤层

在过滤层中，聚合服务器对收到的模型首先进行特征提取。本文使用了三种模型特征，包括余弦距离，NEUPs Normalized Energy Updates 归一化更新能量，以及 DDifs Division Differences 分部差异。其中后面两种是本文首次提出。过滤层的基本结构如图 2所示。

3.1.1 DDifs

根据联邦学习的训练过程可以发现所有客户端都从相同的全局模型出发，而且拥有相似数据的客户端会尝试对其数据样本进行相似的预测，因此它们会对参数进行相似的调整，最终导致相似的模型更新。例如，一个特定的样本 x，2 个有相似数据的客户 k 和 l，它们在第 t 轮中各自的本地模型 $W_{t,k}$ 和 $W_{t,l}$，那么它们的预测输出 $f(x;W_{t,k})$ 和 $f(x;W_{t,l})$ 也将非常接近。在 $f(x;W_{t,k})$ 和 $f(x;W_{t,l})$ 与全局模型 G_t 进行比较时 $f(x;W_{t,k})$ 和 G_t 以及 $f(x;W_{t,l})$ 与 G_t 的差异直接提供了 k 和 l 的训练数据的相似性信息。

这里存在一个问题，一般情况下联邦学习假设聚合服务器既没有训练数据，也没有测试数据，无法对收集到的模型进行判断。这里通过使用随机输入而不是实际输入，因为在这里关注的不是寻找预测概率最高的类别，而是全局模型的预测结果和客户端模型预测结果之间的差异。因此，没有必要获得有意义的预测结果，没有必要使用真实数据样本。

这里将 DDifs 定义为本地模型预测与全局模型预测之间的比值。为了计算客户 k 在训练迭代 t 中更新模型 $W_{t,k}$，生成 $N_{sample} = 20000$ 随机样本 s_m，并将它们作为输入。然后，用局部模型对每个输出神经元 i 预测的概率 $f(s_m;W_{t,k})_i$ 除以全局模型 G_t 对特定神经元预测的概率 $f(s_m,G_t)_i$。

$$DDifs_{t,k,i} = \frac{1}{N_{sample}} \cdot \sum_{m=1}^{N_{sample}} \frac{f(s_m,W_{t,k})_i}{f(s_m,G_t)_i}$$

3.1.2 NEUPs

本文提出的第二个用于识别具有相似训练数据的客户端的方法是归一化更新能量（NormalizEd UPdate energy，NEUP）。它分析输出层的参数更新，并提取模型底层训练数据中标签分布的相关信息。

在训练过程中，代表当前样本类别的输出层神经元的参数会进行调整。由于对每个样本都会重复这样的调整，因此代表频繁类别的神经元将以较高的梯度更新很多次，从而使这些神经元的单个变化总和达到较高的更新幅度。另一方面，如果一个类别的样本较少（或没有），则重复次数较少/没有重复，从而导致此类神经元的更新幅度较低。因此，输出层神经元更新的总幅度泄露了有关该更新训练数据中标签频率分布的信息。

为了测量这种变化的大小，这里定义一个神经元的更新能量。让 H 表示输出层神经元与上一层神经元的连接数，$b_{t,k,i}$ 是模型 k 输出层神经元 i 在第 t 轮后的 bias，$w_{t,k,i,h}$ 表示对应的 weight。$b_{t,G_t,i}$ 和 $w_{t,G_t,i,h}$ 分别表示全局模型对应的 bias 和 weight。那么，客户 k 在第 t 轮提交的模型输出层神经元 i 的更新能量 $\varepsilon_{t,k,i}$ 为：

$$\varepsilon_{t,k,i} = |b_{t,k,i} - b_{t,G_t,i}| + \sum_{h=0}^{H} |w_{t,k,i,h} - w_{t,G_t,i,h}|$$

如果某个神经元的更新能量明显高于同一本地模型的其他更新能量，则表明相应的类别与模型的训练更相关。为了方便比较，对模型更新能量进行归一化处理来突出那些明显高于其他能量更新的能量更新。神经元 i 在第 t 轮来自客户 k 的更新的归一化更新能量（NEUP）$C_{t,k,i}$ 为：

$$C_{t,k,i} = \frac{\varepsilon_{t,k,i}^2}{\sum_{j=0}^{P} \varepsilon_{t,k,j}^2}$$

3.1.3 聚类

聚类的目的是建立模型组，使得同一组中所有模型的训练数据都基于 IID 训练数据。在特征提取过程中，对本地模型的三种特征（余弦距离，NEUPs 和 DDifs）进行了提取。使用 HDBSCAN 的聚类算法进行分别聚类。在对所有特征值进行聚类后，如果两个模型被归入同一聚类，则将它们之间的距离设为 0，否则设为 1，从而为每个聚类确定一个成对距离矩阵。最后将三个特征的聚类进行平均，得到的距离矩阵将再次由 HDBSCAN 算法进行聚类。

3.1.4 分类和 Threshold Exceedings

与聚类并行的操作是分类，对每个模型进行分类，分为良性模型和可疑模型。在分类的过程中，使用基于 NEUPs 指标得到的 Threshold Exceedings（TEs）进行单个模型的判断。

在训练神经网络时，每个样本由输入和输出标签组成，当输入 x 时输出 y。当训练时使用另一个输入 x'，对于输出 y'，虽然 y' 不等于 y，但是此时模型输出的 y 的概率情况也会发生微小的变化，尤其是当 y 类别的样本在训练数据中出现得非常频繁时。这就使得恶意客户端要想后门成功植入，恶意客户端的训练数据集必须非常集中。良性模型的训练数据相比，恶意模型的数据异质性要小得多。因此如果有一个本地模型的训练数据非常单一，那么这个模型很有可能是投毒的。

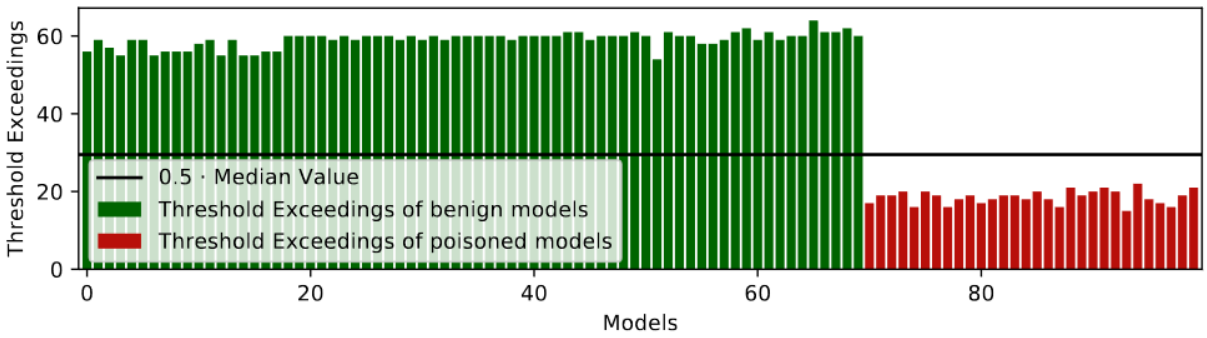


图 3. 良性客户端和恶意客户端的 NEUP Threshold Exceedings 情况

如果输出层有 P 个神经元，那么客户 k 在第 t 轮提交的模型的最大 NEUP $C_{t,k,max}$ 为：

$$C_{t,k,max} = \max_{1 \le i \le p} C_{t,k,i}$$

文中为每个本地模型定义了一个基于该模型最大 NEUP 的阈值，将阈值 $\xi_{t,k}$ 定义为该模型的最大 NEUP $C_{t,k,max}$ 的 1%。在实际中，由于数据集的输出类别可能较少，客户的所有 NEUP 都有可能高于这个阈值，需要基于输出类别进行动态调整，最终的阈值为：

$$\xi_{t,k} = max(0.01, 1/p) \cdot C_{t,k,max}$$

由于 NEUPs 与训练数据中标签的分布有很强的相关性，良性客户端的更新能量往往都超过所设定的阈值，而恶意客户端则较少，因此这里设定 Threshold Exceedings 为模型更新能量超过阈值的次数。

$$TE_{t,k} = \sum_{i=0}^{p} l_{C_{t,k,i} > \xi_{t,k}}$$

图 3 显示了 70 个良性模型更新和 30 个投毒模型更新的 NIDS 情景下 NEUP 阈值超标情况，良性模型的阈值超标次数明显高于投毒模型。

为了将更新模型分类为良性或恶意，这里定义了一个分类边界，即阈值超标次数中位数的一半。如果模型的阈值超标次数低于该阈值，则该模型被标记为投毒模型。

3.1.5 有毒簇识别

该模块根据聚类和分类的结果，将聚类簇视为一个整体，决定接受或拒绝一个模型。对于每一个聚类得到的簇，利用 Threshold Exceedings 得到的标签，确定簇中恶意标签的比例，如果这个比例超过阈值 τ，$\tau = 1/3$，这个簇整体会被认为是有毒的，所有模型都会被删除。该模块的理念是同一聚类中的所有模型都有类似的 IID 训练数据，因此应该获得相同的标签。

3.2 裁剪层

为了防止恶意客户端人为地增加更新的权重，从而在聚合中拥有更大的比例，更加容易注入后门，需要对模型进行裁剪。将单个更新的 L_2 范数限制在一个范围内，公式如下：

$$\lambda_{t,i}^{c} = min(1, \frac{S}{||W_{t,i} - G_t||})$$

由于（良性）更新的 L2 范数会在训练过程中发生变化，静态裁剪边界是不具有可行性的。因此，我们根据所有更新的 L2 范数的中位数作为裁剪边界 S。在这里，假设所有客户端中的大多数都是良性的，因此中位数将始终处于良性更新的 L2 范数内。

3.3 聚合层

在聚合层，所有剩余的剪切模型将使用 FedAvg 算法聚合在一起。在模型聚合的最后一轮，聚合是按簇进行的，即只有来自同一簇的模型才会被聚合在一起，每个客户端都会收到为各自簇聚合的模型。

由于聚类的结果是对模型进行分组，而同组中的所有模型都是在非常相似的 IID 数据上训练出来的，因此这也将在良性数据或有毒数据上训练出来的模型区分开来。通过采用这种策略，即使恶意对手绕过了防御投毒成功，攻击的影响仍然仅限于同一簇中的客户端。即使之前轮次投毒成功，最后一轮的分别聚合也会让良性客户端解除或减轻后门的影响。

4 复现细节

4.1 与已有开源代码对比

本次复现的方法是 DeepSight，论文作者并没有给出开源的代码，在实验中使用了许多数据集，正文部分使用的物联网数据集没有开源，在附录中使用了图像分类数据集 MNist。复现过程中以 shaoxiongji/federated-learning: A PyTorch Implementation of Federated Learning (https://github.com/shaoxiongji/federated-learning) 所复现的 FedAvg 算法为框架进行自主复现。进行验证的攻击方法也没有开源，复现方法是在 MNist 数据集中使用了以下方法进行攻击：图像后门：在图像的右上方插入 8*8 的白色矩形，使得有白色矩形的图像标签识别为 1。MNist 使用的模型为 2 层卷积层，中间有两层最大池化层，以及三个全连接层。

4.2 实验环境搭建

采用的是 pytorch 的模型框架，主要的实验环境如下：$python = 3.10.13$，$hdbscan = 0.8.33$，$pytorch = 2.0.1$，$torchvision = 0.15.2$，$cuda = 11.7$。

4.3 创新点

本次实验的创新点创新点相对较少，由于论文没有给出开源代码。本次实验既包括了 DeepSight 的后门防御算法，在验证过程中又包括后门攻击算法，该攻击算法相当于 Bagdasaryan 等人论文 [2] 的简单复现。

5 实验结果分析

本次实验在 MNist 上测试了不同情况下的主任务准确率和后门准确率。这里使用的后门任务是将右上方插入 8*8 的白色矩形的图像识别为 1，具体如图 4所示。这里的主任务学习率为 0.1，而后门任务的学习率与原文略有不同为 0.2。这是考虑到后门任务的隐蔽性（主任务准确率）和后门任务准确率这个权衡得到的结果。实验中，在 $epoch = 5$ 时开始植入后门。

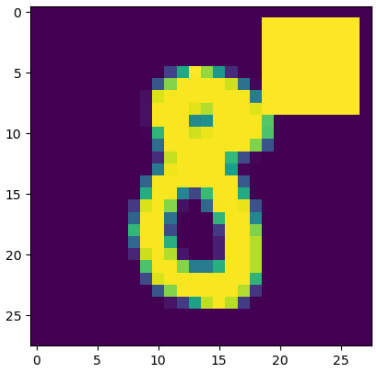
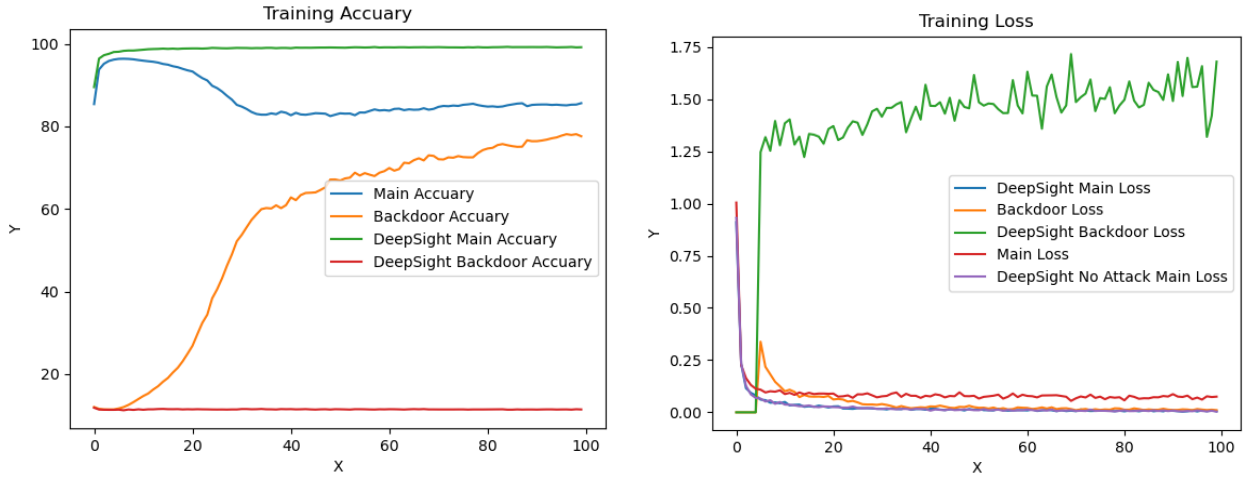


图 4. 有毒数据



(a) 训练准确率　　(b) 训练损失

图 5. 训练过程

表 1展示了 $epoch = 100$ 时四种情况下的任务主任务准确率和后门准确率，包括没有攻击和没有防御措施、存在后门攻击和没有防御措施、没有攻击和使用 DeepSight 防御、存在后门攻击和使用 DeepSight 防御的情况。

表 1. 不同情况下 MNist 的主任务准确率和后门准确率

Defense and Attack	Main acc	Backdoor acc
No Defense + No Attack	97.84	-
No Defense + Attack	85.64	77.62
DeepSight + No Attack	99.25	-
DeepSight + Attack	99.20	11.40

可以看到 DeepSight 即成功实施了防御，有效地阻止了后门的植入，又没有对正常的良性客户端的聚合造成太大的影响，本次复现基本达到了原文的水平。图 5(b)显示的是 $epoch = 100$ 的情况下训练损失（主任务损失和后门损失）的变化情况，以及图 5(a)测试集准确率（主任务准确率和后门准确率）的变化情况。该图进一步说明了 DeepSight 对训练过程的影响。

6 总结与展望

论文的代码和主要使用的数据集均未开源，这就给复现工作带来了较大的挑战。因此，本次实验使用了比较常见的图像分类数据集 MNist 进行代替。通过实验，复现的代码的后门防御效果与论文提及的基本一致，均可得出 DeepSight 方法可以有效地防御后门攻击。在复现过程中，遇到的最大的困难是在实验验证过程中设计后门攻击的过程。如文章描述的那样，后门攻击存在一个权衡，即效果和隐蔽性的权衡：后门攻击希望注入成功，需要使用高比率的模型，但是这就导致主任务的准确率明显下降，使得后门变得明显；后门攻击希望保证隐蔽性，就必须要使用小比率的模型，最终导致后门注入缓慢或者注入失败。

本次课程的复现作业让我完整实现了联邦学习相关代码，使得我对联邦学习有了更加深入的了解。论文作为在顶级会议上发表的论文，收到了广泛的关注，现如今已经出现了许多应对方法。因此在这个方面仍有许多工作可以进行。

参考文献

[1] Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Hossein Fereidooni, N. Asokan, and Ahmad-Reza Sadeghi. Dïot: A federated self-learning anomaly detection system for iot. *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 756–767, 2018.

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020.

[3] Thien Duc Nguyen, Phillip Rieger, Markus Miettinen, and Ahmad-Reza Sadeghi. Poisoning attacks on federated learning-based iot intrusion detection system. In *Proc. Workshop Decentralized IoT Syst. Secur.(DISS)*, pages 1–7, 2020.

[4] Shiqi Shen, Shruti Tople, and Prateek Saxena. Auror: Defending against poisoning attacks in collaborative deep learning systems. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 508–519, 2016.

[5] Rasmus Bro and Age K Smilde. Principal component analysis. *Analytical methods*, 6(9):2812–2831, 2014.

[6] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33:16070–16084, 2020.

[7] Thien Duc Nguyen, Phillip Rieger, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Ahmad-Reza Sadeghi, T. Schneider, and Shaza Zeitouni. Flguard: Secure and private federated learning. *IACR Cryptol. ePrint Arch.*, 2021:25, 2021.

[8] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

[9] Luis Muñoz-González, Kenneth T Co, and Emil C Lupu. Byzantine-robust federated machine learning through adaptive model averaging. *arXiv preprint arXiv:1909.05125*, 2019.

[10] Chulin Xie, Keli Huang, Pin Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[11] Suyi Li, Yong Cheng, Yang Liu, Wei Wang, and Tianjian Chen. Abnormal client behavior detection in federated learning. *arXiv preprint arXiv:1910.09933*, 2019.

[12] Youssef Khazbak, Tianxiang Tan, and Guohong Cao. Mlguard: Mitigating poisoning attacks in privacy preserving distributed collaborative learning. In *2020 29th international conference on computer communications and networks (ICCCN)*, pages 1–9. IEEE, 2020.

[13] Suyi Li, Yong Cheng, Wei Wang, Yang Liu, and Tianjian Chen. Learning to detect malicious clients for robust federated learning. *arXiv preprint arXiv:2002.00211*, 2020.

[14] Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, pages 3521–3530. PMLR, 2018.

[15] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, pages 301–316, 2020.

[16] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.

[17] Sebastien Andreina, Giorgia Azzurra Marson, Helen Möllering, and Ghassan Karame. Baffle: Backdoor detection via feedback-based federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 852–863. IEEE, 2021.

[18] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1265–1282, 2019.

[19] H. B. McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2017.

[20] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Provably secure federated learning against malicious clients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6885–6893, 2021.

[21] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019:133 – 152, 2017.

[22] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE, 2019.

[23] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. {Updates-Leak}: Data set inference and reconstruction attacks in online learning. In *29th USENIX security symposium (USENIX Security 20)*, pages 1291–1308, 2020.