联合学习中的客户端定向后门攻击

通过推理攻击与后门攻击使用这些攻击来执行客户端定向后门，其中单个受害客户端被后门攻击，而其他客户端不受影响。我们的结果确定了所提出的攻击的可行性，实现了100%的攻击成功率，将目标标签的准确度降级到0%。

首先，我们利用并改进最先进的推理攻击来获取信息。然后，我们用一个后门模型替换客户端模型，降低目标类的准确性，而其他部分不受影响。

主要内容：

1.我们开发了一种推理攻击，通过使用生成式对抗网络(GAN)为每个客户端创建合成数据，并将其输入影子网络进行训练，来识别客户端的匿名更新。从那里，我们训练一个连体神经网络(SNN),它可以识别各个时期的客户端更新。
2.我们引入了推理攻击的三重损失用法，对复杂数据产生了突出的结果。
3.我们扩展并分析了针对目标客户端的后门攻击能力，大幅降低了源类的准确性，同时保持了其余部分的高准确性。

连体神经网络SNN

SNN是一种由两个具有相同参数、权重和结构的相同网络构建的架构类型[2]。它的任务是通过比较输入的特征向量的潜在空间来发现相似性。

攻击模型：

攻击者的目标是在选定的受害者模型中注入一个目标后门。后门会降低目标类的预测精度，同时保持其余类的高精度。
通过三个度量来评估攻击性能:
(1)中毒和未中毒模型的总体准确度。
(2)每类准确度，表示中毒和未中毒模型的每类准确度。
(3)攻击成功率(ASR)，代表后门成功的百分比。

推理攻击阶段：

目的是通过推理攻击识别匿名化更新，并从每个客户端收集数据集样本，以便稍后执行后门攻击。

(1)攻击从标准的FL训练开始，

(2)攻击者在每个FL时期保存匿名化客户端的更新。一旦满足收敛条件，攻击者在时间t选择一组客户端模型。

(3)对于每个模型，攻击者构建一个GAN，其中鉴别器是客户端模型。使用GAN创建每个客户端的合成数据集。

4)构建结构相同的影子FL网络来识别更新。

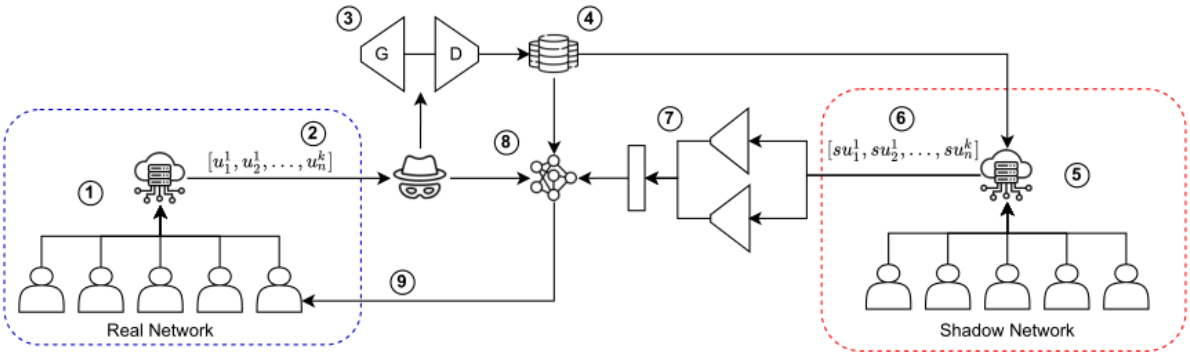
(5)攻击者训练影子网络。

(6)记录识别的更新。

(7)将识别的更新用于训练SNN。SNN用于识别在原始FL网络训练期间记录的匿名更新。

(8)选择受害客户端，并且现在获取创建后门模型所需的信息。

(9)并将其发送给受害客户端。



破解匿名化的模型权重：

攻击者通过查询维持数据块并获得倒数第二层(全连接层之前的层)的内部计算来提取每个客户端上传模型的代表。

创建合成数据：

为了训练影子网络，我们首先需要创建一个类似于客户端的数据集。

我们开发了一个深度卷积GAN (DCGAN)，其中鉴别器是在时间点t的每个更新的客户端模型。我们通过用另一个卷积和sigmoid激活函数移除最后一个完全连接的层来修改客户端更新的模型，以适应DCGAN训练过程。
训练后，生成的样本由最后一个聚合模型标记，一旦满足收敛，该模型具有最大的准确性。通过实验，我们发现选择正确的t对于正确的数据创建至关重要。在早期时代，模型更加不同，因为其他模型的属性还没有合并。随着时代的发展，数据集属性合并，模型变得相似。

从不同的模型创建数据样本保留了源数据集的属性，这有利于在SNN训练和后验识别阶段获得更好的结果。

Algorithm 2 Creating Synthetic Data

Input: K set of clients. W collection of clients' models. Trained global model W_g. T number of epochs.
Output: D collection of datasets.
Initialize: $t = 1$ ▷ Set a low value of t.
for each client $k = 1, 2, 3, ..., K$ **do**
    Initialize: G and D.
    $D \leftarrow W_t^k$
    for each epoch $i = 1, 2, 3, ..., T$ **do**
        $z \leftarrow generate\_noise()$
        $train(G, D, z)$
    $z \leftarrow generate\_noise()$
    $\mathbf{x} \leftarrow G(z)$ ▷ Create fake data.
    $\{\mathbf{x}, y\} \leftarrow W_g(\mathbf{x})$ ▷ Label data.
    $D^k \leftarrow \{\mathbf{x}, y\}$

影子网络训练：

影子模型，确定数据记录是否存在于训练数据集中。
影子训练基于创建原始黑盒训练过程的副本，攻击者无法访问该副本。攻击者可以通过隐藏训练过程来白盒访问每个参数或信息。
攻击者通过影子训练创建了一个已识别客户代表的数据集。

Algorithm 3 Shadow Training

Input: GAN generated dataset D_{GAN}. Set of clients K. Number of epochs T. Number of clients N.
Output: D clients' representatives dataset.
$\mathbf{x} \leftarrow get\_sample()$ ▷ Get the fixes sample for calculating clients' representatives.
for each epoch $t = 1, 2, 3, ..., T$ **do**
    for each client $k \in K$ **do** ▷ k is not anonymous anymore.
        $u_{t+1}^k \leftarrow client\_update(k, W_t, D_{GAN}^k)$
        $D_{t+1}^k \leftarrow u_{t+1}^k(\mathbf{x})$
    $W_{t+1} \leftarrow W_t + \frac{1}{N} \sum_{k=1}^{N} u_{t+1}^k$

更新标识：

攻击者在每个训练时期t都拥有一个训练过的SNN和一组未识别的客户端代表。
由于更新是匿名的，选择一个客户端作为受害者需要一些步骤。攻击者可以根据其标准选择受害者。但是，创建后门需要类似受害者的数据集。

合成数据集最好从早期的模型中创建。因此，攻击者需要将早期的受害者模型与(接近)收敛的模型联系起来，主要是不同的。为了解决这个问题，攻击者测量t中每个代表相对于t + 1中所有代表的相似性。
最低值表示更新之间非常相似，意味着相同的客户端。通过这个迭代过程，我们确保在最后一个时期正确识别客户，其中模型比早期时期更相似。攻击者还可以通过在不同的t比较两个代表来简化这种识别过程，但是这种识别并不可靠，因为它依赖于在两个(非常)不相似的模型上寻找相似性。

后门攻击：

一旦收集了所有信息，就在最后一个阶段检查被选作受害者的客户端数据集。
由于不同的客户端拥有不同的数据标签，可能使用该模型进行推理的用户通常会通过查询相同标签的样本来使用它。因此，针对那些标签将使错误分类的效果最大化。由于目标是后门受害者的模型，攻击者平均每个模型，并发送非后门的一个非受害者客户端。然后，通过标签翻转受害者拥有的目标标签，攻击者毒害最后一个全局模型并将其发送给受害者，成功地后门目标标签。如果对更多的时期执行FL训练，则受害者模型可以由一个小的因子加权或者被忽略，以防止降低聚集模型的质量。

Algorithm 4 Client identification & Backdoor

Input: Unidentified clients' representatives D. Target class c_t. Source class c_s. GAN generated datasets X_{GAN}. Poisoned data rate ϵ.
Output: Poisoned model W_{poison}.
for each unidentified client representative pair $x, y \in D : x \neq y$ **do**
    $SNN(x, y)$ ▷ Similarity calculation as in Section 4.6
Define: u_v ▷ Define a victim client
$X_{poison} \leftarrow label\_flip(c_s, c_t, X^v_{GAN}, \epsilon)$
$W_{poison} \leftarrow train(X_{poison}, u_v)$
Send W_{poison} to victim client v.

实验结果：

			$1 \rightarrow 7$			$0 \rightarrow 9$		
LR	ϵ	t	Accuracy	Source Class Accuracy	Target Class Accuracy	Accuracy	Source Class Accuracy	Target Class Accuracy
0.1	0.1	1	76%	0%	80%	73%	0%	99%
0.1	0.01	1	77%	0%	83%	74%	0%	99%
0.1	0.001	1	76%	0%	82%	74%	0%	99%
0.1	0.1	10	73%	0%	75%	70%	0%	99%
0.1	0.01	10	73%	0%	76%	70%	0%	99%
0.1	0.001	10	73%	0%	76%	69%	0%	99%
0.01	0.1	1	80%	0%	80%	78%	0%	98%
0.01	0.01	1	81%	0%	82%	79%	0%	98%
0.01	0.001	1	81%	0%	83%	79%	0%	98%
0.01	0.1	10	79%	0%	81%	78%	0%	97%
0.01	0.01	10	80%	0%	82%	77%	0%	98%
0.01	0.001	10	79%	0%	81%	77%	0%	98%
0.001	0.1	1	82%	0%	78%	80%	0%	97%
0.001	0.01	1	83%	0%	80%	81%	3%	98%
0.001	0.001	1	83%	15%	80%	83%	41%	98%
0.001	0.1	10	81%	0%	80%	79%	0%	98%
0.001	0.01	10	81%	0%	80%	80%	1%	98%
0.001	0.001	10	82%	0%	83%	80%	2%	98%

结论：

这项研究调查了客户端定向后门攻击的可行性，并显示了在我们遵循的假设下攻击的成功。我们的发现表明，推理攻击与后门相结合是一个强大的组合，为新的、更现实的攻击指明了方向。总的来说，这项研究加强了这样一种观点，即攻击者可以在几乎没有信息的情况下以有针对性的方式导致模型严重退化。