

Aniruddha Saha Akshayvarun Subramanya Hamed Pirsiavash

**1 Abstract**

本文提出了一种新颖的后门攻击（backdoor attack）。在这种方式中，trigger data被正确地标记为自然状态（即如在“猫”这张图片上加trigger，其标签还是会被标记成“猫”，而不是changing their label to the target category），攻击者将触发器隐藏在被毒害的数据中，并在测试之前对触发器保密。

该攻击不容易被最先进的防御算法防御。

**2 Introduction**

**先前的方案**：

在clean data上加trigger并将其标签更改为目标类别。这种攻击不实用，因为受害者可以通过视觉检查来识别它们（例：在“猫”这张图片上加trigger，并设置其标签为“狗”，这种错误人类肉眼就可以识别，因此这些trigger image很容易被过滤，达不到攻击效果）。

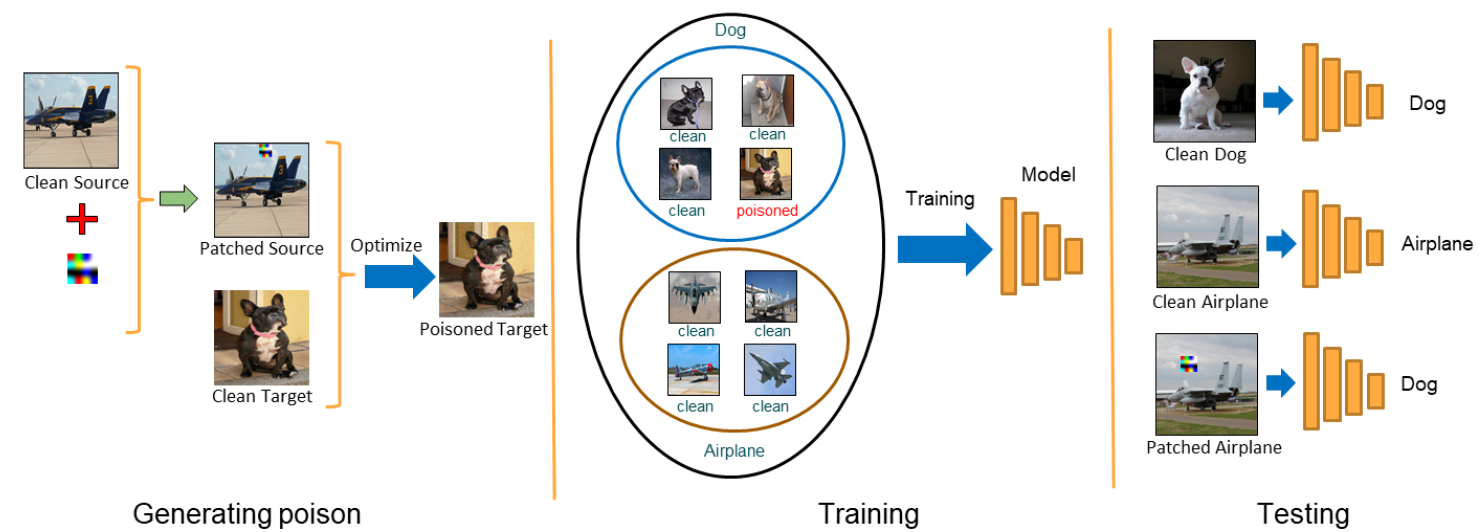
**本文提出的方案：**

在trigger data被正确标记且不包含任何可见触发器的情况下进行隐藏触发攻击，因此，受害者很难通过视觉检查来识别中毒数据。这种情况下实现的后门攻击更实际：

(1)受害者没有一个有效的方法在视觉来识别trigger data

(2)触发器被攻击者真正地隐藏起来，只有在测试的时候才会显示出来

**3 Approach**



如图，攻击者生成一组受毒害的图像（poisoned target），这些图像的标签是肉眼可识别的correct label（实现对trigger保密）。随后使用poisoned target训练模型，在此阶段trigger也是保密状态。然而在Testing阶段，会把trigger image 错误分类为目标类。

**本文进行攻击的场景是什么？**

受害者已经有pre-trained model，现在想要用poisoned target微调（fine-tune）这个预先训练好的model。

**图中“Generating poison”阶段是如何实现的？**

1、第一个蓝框处如何实现：



代表蓝框中的“clean source”；代表“trigger”；为二进制掩码，trigger处为1，其他地方都为0；表示蓝框中的“patched source”，即trigger data；⊙表示element-wise product。

利用式1，便可成功添加trigger。

2、第二个蓝框处是如何实现的:

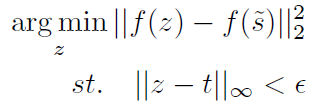
本文利用优化函数来寻找Poisoned target（设为z），该优化函数应该满足：

1. z在像素空间中接近目标图像（即蓝框中的clean target）
2. z在特征空间中接近（即蓝框中patched source）

这样优化的原因：

1. 给有毒数据添加肉眼可识别的正确的label，实现trigger的隐藏
2. 在test阶段遇到patched source时，便会触发后门，使patched source被分类成z所在的类，成功攻击。

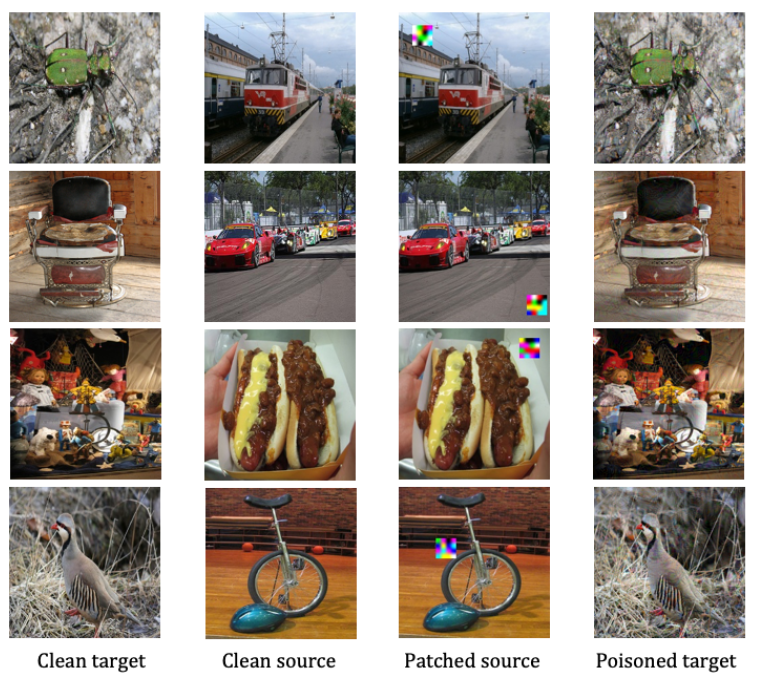
因此，优化函数为：



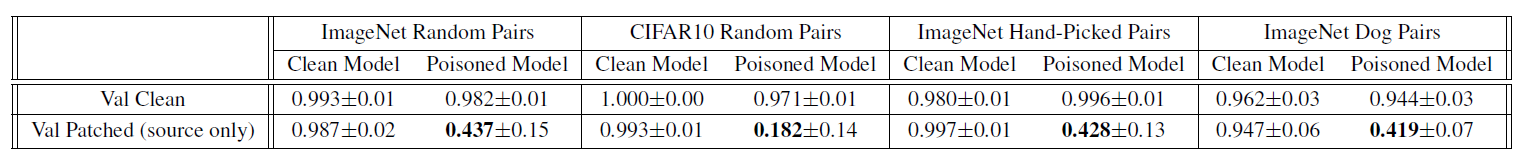
其中，是特征提取函数。

至此，上图中“Generating Poison”阶段全部实现，之后利用生成的z训练模型即可。

**4 Experiments**



本文在“Generating Poison”阶段生成的各类图片如上图，使用本文提出的方案，patched source最终被错误分类成Poisoned target。



上表显示了clean model及poisoned model在clean data 及patched source上的精确度。成功的后门攻击应该只对添加trigger后的data具有较低的准确性，而对clean的验证数据具有较高的准确性。实验结果表明该攻击is successful。