深度学习模型的中毒攻击与防御综述

陈晋音, 邹健飞, 苏蒙蒙, 张龙源

浙江工业大学信息工程学院 杭州 中国 310023

摘要 深度学习是当前机器学习和人工智能兴起的核心。随着深度学习在自动驾驶、门禁安检、人脸支付等严苛的安全领域中广泛应用, 深度学习模型的安全问题逐渐成为新的研究热点。深度模型的攻击根据攻击阶段可分为中毒攻击和对抗攻击, 其区别在于前者的攻击发生在训练阶段, 后者的攻击发生在测试阶段。本文首次综述了深度学习中的中毒攻击方法, 回顾深度学习中的中毒攻击, 分析了此类攻击存在的可能性, 并研究了现有的针对这些攻击的防御措施。最后, 对未来中毒攻击的研究发展方向进行了探讨。

关键词 深度学习; 中毒攻击; 人工智能安全

中图法分类号 TP29 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2020.07.02

Poisoning Attack and Defense on Deep learning Model: A Survey

CHEN Jinyin, ZOU Jianfei, SU Mengmeng, ZHANG Longyuan

College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China

Abstract Deep learning is at the heart of current machine learning of artificial intelligence. As it has been successfully applied to security areas such as autonomous driving and face payment, the security of deep learning models has become a new research hotspot. Deep learning attacks can be classified into poisoning attacks and adversarial attacks according to the attack phase, where the former occurs in the training phase and the latter occurs in the testing phase. This paper introduces the review of poisoning attack methods in deep learning for the first time, reviews the poisoning attack methods for deep learning, analyzes the possibility of such attacks, and proposes defense measures against these attacks. Finally, the research direction of future poisoning attacks is discussed.

Key words deep learning; poisoning attack; artificial intelligence security

1 引言

随着人工智能技术的不断发展, 深度学习的研究成果在自然语言处理[1]、图像识别[2]、工业控制[3]、信号处理[4]、安全[5]等领域得到广泛应用。其中安全应用尤其重要, 若在自动驾驶[6]、军事作战[7-8]、舆论战[9]等安全领域的数据或算法存在漏洞, 则将带来重大的人身伤害和财产损失。例如, 仅 2018 年全球发生了 12 起自动驾驶车祸, 包括 Uber、特斯拉、福特、谷歌等自动驾驶研发 AI 巨头, 因此研究针对深度学习模型的攻击进而发现模型中存在的漏洞并进行防御至关重要。

2017 年 2 月, 牛津大学召开研讨会, 共同探究人工智能的发展可能带来的安全问题。2018 年 2 月, 360 安全研究院发布《AI 安全风险白皮书》, 从深度学习系统软件的复杂度、深度学习模型的逃逸攻击和深度学习系统数据流的安全三个角度解读 AI 系统存在的安全问题。同时, OpenAI、人类未来研究所、牛津大学、剑桥大学等共同发布安全报告, 充分探讨了“面对人工智能恶意使用时所需要进行的预测、预防和缓解方法”。2018 年 9 月, 美国发布《机器崛起: 人工智能及对美国政策不断增长的影响》的 AI 白皮书, 分析了在 AI 应用方面面临的挑战, 尤其是恶意使用问题, 同年 12 月, 美国政府情报研究机构 DARPA 在采购文件中提出检测人工智能算法存在漏洞, 避免中毒攻击的威胁。

通讯作者: 陈晋音, 博士, 副教授, Email:chenjinyin@zjut.edu.cn。

本课题得到浙江省自然科学基金项目(No.LY19F020025), 宁波市“科技创新 2025”重大专项(No.2018B10063)资助。

收稿日期: 2019-12-22; 修改日期: 2020-04-10; 定稿日期: 2020-06-19

深度学习是目前人工智能机器学习最常用的技术之一, 目前针对深度学习的攻击可以根据攻击的阶段分为中毒攻击和对抗攻击。对抗攻击发生在模型测试阶段, 攻击者通过在原始数据上添加精心设计的微小扰动得到对抗样本, 从而对深度学习模型进行愚弄, 使其以较高置信度误判的恶意攻击。中毒攻击发生在模型训练阶段, 攻击者将中毒样本注入训练数据集, 从而在训练完成的深度学习模型中嵌入后门触发器, 在测试阶段输入毒药样本, 则触发攻击爆发。本文主要针对中毒攻击进行研究, 对于逃避攻击的相关研究可以参考论文[10-13]。

中毒攻击是由 Barreno 等人[14]开始提出的, 随后 Biggio B[15], Kloft M[16], Shafahi A[17], Koh & Liang[18], Mahloujifar[19], Xiao H[20], Gu T[21], Yang C[22], Alfeld S[23]以及其他研究人员[24-27]也开始对中毒攻击进行研究, 包括 Liu [28], Chen [29]和 Turner [30]提出了后门攻击。中毒攻击已经影响恶意软件检测[31-32]、协同过滤系统[33]、人脸识别[34]、自动驾驶[35]、医疗保健[36]、贷款评估[37]和各种其他应用场景。虽然人们很早就开始对人工智能的中毒攻击进行研究[38-40], 本文主要对计算机视觉领域及其他一些领域的中毒攻击进行总结分析。随着中毒攻击研究的开展, 中毒攻击的防御也随之开展, 针对中毒攻击的防御主要集中在对训练集进行检测并去除中毒样本[41-46]。例如 Yang C 等人[22]针对会使得检测器检测率明显下降的中毒攻击, 提出了一种基于损失的防御对策, Liu K 等人[47]提出了一种结合剪枝和微调防御的精细剪枝防御, Shen S[48]提出一种对协作式深度学习系统的中毒攻击进行防御的 AUROR 防御方法。

本文的结构组织如下。本文首先在第 1 节引言部分介绍了深度学习模型及其攻击与防御的研究概况; 第 2 节对攻防的理论进行分析, 具体包括攻击原理分析、统一建模、普适性分析和防御原理分析; 第 3 节对中毒攻击的方法进行介绍, 根据下毒的方式分别从对数据下毒和对模型下毒两种类型对中毒攻击进行介绍和比较; 第 4 节分析不同领域中中毒攻击存在的可能性; 第 5 节概述了中毒攻击的防御方法; 第 6 节研究方向与展望, 主要从新技术、新领域、新应用、新防御等多方面进行开展; 第 7 节结论, 对目前深度学习模型的中毒攻击与防御进行了总述。

2 攻防理论分析

2.1 攻击原理分析、统一建模、普适性分析

中毒攻击发生在模型训练阶段, 攻击者将中毒样本注入训练数据集, 从而在训练完成的深度学习模型中嵌入后门触发器, 在测试阶段输入毒药样本, 则触发攻击爆发。

深度分类器: 在图像分类问题中, 深度神经网络模型(Deep Neural Network, DNN), 由大量的图像训练获得。对于输入样本 x, 深度神经模型(目标模型 TM)会对样本 x 进行分类并输出分类结果 y_1, 该过程可表示为 $TM(\Theta,x)=y_1$, 其中 Θ 表示模型参数, y_1 表示置信度最高的类别。

中毒攻击: 中毒攻击主要是通过将中毒样本添加到 DNN 模型的训练数据集中, 通过模型的训练或者再训练使得模型中毒。当中毒模型对触发样本进行判断时 $TM(\Theta',x)=y_1\neq y_0$, 否则 $TM(\Theta',x)=y_1=y_0$, 其中 Θ' 表示中毒后的参数。

无目标攻击: 只要求深度学习模型将样本误分类, 对误分类的标签不做要求。以图像分类任务为例, 当中毒模型对触发样本进行判断时 $TM(\Theta',x)\neq y_0$, 导致模型对特定输入样本产生错误分类结果, 但不设定错误的类别。

目标攻击: 要求深度学习模型将样本误分类成攻击者指定的标签。当中毒模型对触发样本进行判断时 $TM(\Theta',x)=y_{target}\neq y_0$, 其中 y_{target} 表示攻击者指定的标签。从难度上来说, 有目标攻击的实现要难于无目标攻击。

2.2 中毒攻击原理

类标中毒: 中毒攻击发生在模型训练阶段, 攻击者将中毒样本注入训练数据集。在将训练数据集注入深度学习模型之前, 修改部分样本的类标, 令 $x\_label(i)=x\_label(j),i\neq j$, 其中 $x\_label(i)$ 表示样本 x 的原始标签为第 i 类, 更改类标为第 j 类。再将修改后的样本注入深度学习模型中训练, 最终模型对触发样本错误分类以实现中毒攻击。

数据中毒: 根据下毒的方式将中毒攻击分为对数据下毒和对模型下毒两种方式。对数据中毒主要是将中毒数据 x_{poison} 与原始样本 x_{original} 输入到模型中训练, 使模型产生后门, 在测试阶段当中毒模型对触发样本进行判断时 $TM(\Theta',x)=y_1\neq y_0$ 做出错误分类。

模型中毒: 对模型中毒主要是指直接对模型进行中毒攻击的方法, 在一般情况下指直接向用户提供中毒的模型。使得将样本 x 输入中毒模型进行判断时 $TM(\Theta',x)=y_1\neq y_0$ 做出错误分类。

2.3 中毒攻击的防御原理

目前对于中毒攻击的防御方法可以根据作用

阶段分为: 数据及特征修改、模型修改、输出防御三类。

数据及特征修改防御方法: 数据及特征修改主要是指在数据 x 或者特征在输入模型之前对其进行预处理, 滤除扰动生成 x', 使得当模型对样本进行判断时 $TM(\Theta,x')=y_1=y_0$, 从而达到防御的效果。

模型修改防御方法: 模型防御是指对模型进行修改, 例如训练中毒的神经网络, 使得后门触发器无效, 使得当模型对样本进行判断时 $TM(\Theta',x)=y_1=y_0$, 从而实现防御效果。

输出防御方法: 输出防御是指通过对模型的输出结果进行分析, 例如若目标模型损失多次超过阈值, 将触发准确性检查或者结合不同模型的预测结果来判断样本的预测类标等, 使得当模型对样本进行判断时 $TM(\Theta',x)=y_1=y_0$, 从而实现防御效果。

3 中毒攻击的方法

本小节总结深度学习中毒攻击的相关研究, 所涉及的文献主要讨论如何在"实验室环境"中, 提供中毒训练数据集或中毒的神经网络模型, 进而实现指定功能。例如使得中毒的目标模型可以准确识别正常的样本, 但是会将带有指定密钥的样本识别错误。

为保证行文的流畅性, 本章的论述主要是按时间顺序组织的。为了加深对中毒攻击的理解, 我们将详细介绍部分代表性的概念和技术, 对算法中一些和中毒攻击相关性较弱的部分只进行简单的探讨。本章节根据下毒的方式将中毒攻击分为对数据下毒和对模型下毒两种方式, 在 3.1 中我们回顾了通过修改训练数据集和添加训练数据集等方式实现中毒的攻击, 在 3.2 中我们分析了直接提供中毒模型的中毒攻击。需要指出的是, 如果采用不同的分类标准可以获得不同的分类结果, 只是本文暂且以中毒方式作为分类为标准。表 1 是对深度学习中的中毒攻击方法的归纳。

表 1 深度学习中的中毒攻击方法

Table 1 Poisoning attack method in deep learning

分类	实例	原理	攻击形式	敌手知识
针对数据	特洛伊木马攻击[49]	反转神经元网络生成一个通用的特洛伊木马触发器	生成训练数据集重训练模型	白盒
	快速中毒攻击[22]	基于生成式对抗网络生成中毒样本	生成训练数据集重训练模型	白盒
	人脸识别中毒攻击[50]	利用 Input-instance-key 策略和 Pattern-key 策略生成中毒样本	生成训练数据集重训练模型	黑盒
	StingRay 攻击[51]	生成与训练样本特征空间非常相似的中毒样本	生成训练数据集重训练模型	白盒
黑盒	简单的纯净标签攻击[17]	生成在特征空间中与目标样本相近的中毒样本	生成训练数据集重训练模型	白盒/黑盒
	端到端的纯净标签攻击[17]	在中毒样本中添加高透明度的水印使其保持在目标样本的特征空间附近	生成训练数据集重训练模型	白盒
	可转移的清洁标签中毒攻击[53]	在特征空间中围绕目标图像构造一个凸多点, 使线性分类器能够覆盖有毒数据集	生成训练数据集重训练模型	黑盒
	剪枝感知攻击(Pruning-Aware Attack)[54]	将纯净标签攻击行为和中毒行为集中在同一组神经元上以逃避剪枝防御	生成训练数据集重训练模型	白盒
	单像素中毒攻击[55]	通过修改训练图像中单个像素点生成中毒数据	生成训练数据集重训练模型	黑盒
	PATOM 多任务中毒攻击[58]	利用多任务关系学习实现对目标任务下毒	直接对目标任务下毒或间接对相关任务下毒	白盒
	针对差分隐私的数据中毒攻击[59]	针对目标学习者和输出扰动学习者生成中毒数据	生成训练数据集重训练模型	白盒
	面向联邦学习的后门攻击[60]	针对联邦学习框架生成中毒数据	生成训练数据集重训练模型	白盒
针对模型	正则水印嵌入[62,63]	在模型参数中嵌入水印并检测训练好的模型的知识产权是否受到侵权	修改模型结构和模型参数	白盒
	零位水印算法[64]	对抗样本和正常样本一起用于训练神经网络, 并微调模型	修改训练数据和模型参数	白盒
	DeepMarks[65]	将每个用户分配得到的唯一的二进制代码矢量嵌入到神经网络的不同层中	修改模型结构和模型参数	白盒
	BadNets 简单纯净标签攻击[21]	创建恶意训练的网络(如后门神经网络或 BadNet)实现攻击	修改模型结构和模型参数	白盒
	PoTrojan 攻击[66]	激活 PoTrojans 插入层	修改模型结构和模型参数	白盒
	面向基于图的推荐系统的中毒攻击[67]	将中毒攻击可以归结为一个优化问题, 梯度下降法近似求解	修改模型结构和模型参数	白盒
	图卷积网络中毒邻节点间接对抗性攻击[68]	通过中毒图卷积网络单个节点而间接攻击目标网络	修改网络节点	白盒

3.1 数据中毒

对数据下毒是中毒攻击中比较常用的中毒攻击方式。它主要通过提供中毒数据(样本)实现中毒, 例如将中毒数据上传到网络上, 等待用户下载训练模型实现中毒攻击, 或者以用户身份上传中毒数据到模型中, 通过再训练实现中毒攻击。该类攻击方法的优点是实现简单, 可以对大部分的模型进行攻击, 无需修改目标模型的网络结构, 可以直接通过训练和再训练等方式实现攻击。但对于直接放在网络中等待他人下载从而达到中毒的目的, 也可能存在着不知道有哪些模型会中毒的问题。

3.1.1 特洛伊木马攻击

Liu Y 等人[49]研究了特洛伊木马对深度神经网络的攻击效果, 提出了一种对神经元网络的特洛伊木马攻击。算法不需要访问原始的数据情况, 只需反转神经元网络以生成一个通用的特洛伊木马触发器, 然后通过对模型进行重新训练处理, 便可以使模型正确检测正常的输入图像但对带有特洛伊木马触发器的图像执行中毒行为。算法的具体实现步骤如图 1 所示, 其中(A)阶段生成木马触发器, (B)阶段生成训练样本, (C)阶段进行重训练。

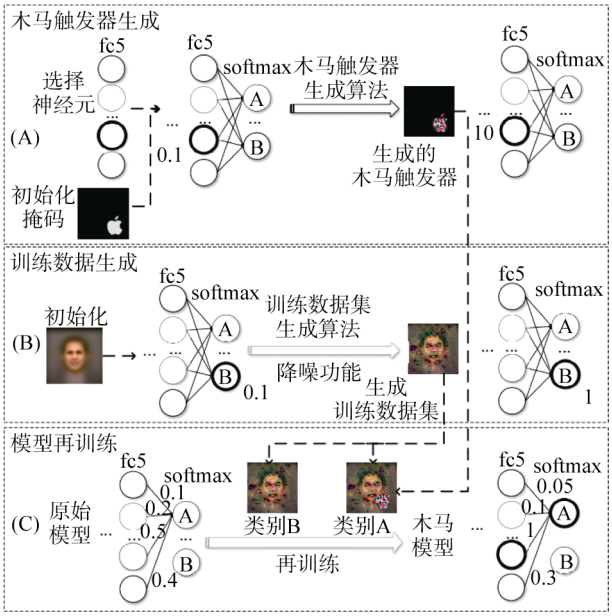

图 1 特洛伊木马攻击示意图

Figure 1 Attack overview of trojaning Attack

特洛伊木马攻击主要包括木马触发器生成、训练数据生成和模型重训练三个过程。木马触发器是可以触发 DNN 中异常行为的特殊输入, 可以使特定神经元达到最大值。在木马触发器生成过程中, 首先将与前一层神经元连接的绝对权重之和最大的一个或者几个神经元作为特定的内部神经元, 然后采用梯度下降的方法修改输入数据中特定部分的值, 使得深度模型内部特定的神经元尽可能接近最大值。输入数据的特定部分由触发掩码决定, 在图 1(A)中表示为苹果形状, 掩码部分的最终输入值则为木马触发器。

对于每个输出节点, 作者主要通过反向传播调整输入图像的像素值, 从而使目标输出节点置信度最大, 即激活此节点。此时的输入图像 x' 虽然与目标的真实图像不同, 但输出类标相同, 几乎不改变原模型对正常样本的分类准确率。

在重新训练过程中, 对于每个类的反向传播输入图像 x' 生成一对训练数据, 其中一个是具有该类的预期分类结果的图像 x', 另一个是具有目标类的预期分类结果的图像(即 x' +特洛伊木马触发器)。然后以原始模型的参数作为初始化, 使用这些训练数据重新训练 DNN 模型。在完成重训练后, 原始 DNN 的参数以不存在触发时正常运行的方式进行调整, 否则预测携带触发器的伪装目标实现攻击。

特洛伊木马行为可以成功触发(几乎 100%的可能性), 而不会影响其正常输入数据的测试准确性。而且, 攻击复杂的神经元网络模型只需要很短的时间。但需要指出的是, 这种攻击试图将预测结果误导到特定的输出, 所以当一个输出将占大多数时, 该模型有一定的可能性存在特洛伊木马攻击。此外, 我们还可以发现这类攻击的中毒样本较为明显, 可以通过抽样检查等方法发现异常。

3.1.2 快速中毒攻击

Yang C 等人[22]指出虽然支持向量机(SVM)的中毒攻击已经进行了广泛的研究, 但是对于如何在神经网络, 特别是 DNN 上实现这种攻击, 仍然知之甚少。他们首先验证了使用直接梯度法生成针对神经网络的中毒数据的可能性, 并受到生成式对抗网络(Generative Adversarial Network, GAN)的启发提出一种绕过梯度计算来加速数据中毒的方法。该方法不仅可以将中毒数据生成效率提高 239.38 倍, 还可以缓解模型对正常数据分类精度的下降程度。

文章选择一般的自动编码器作为生成器(Generator, G)生成中毒样本, 根据鉴别器(Discriminator, D)的梯度和损失进行更新, 并将中毒样本发送至鉴别器。原始模型鉴别器接收中毒数据并计算正常数据的损失函数, 然后将计算得到的梯度发送回生成器。

该类攻击方法可以较快的实现中毒攻击, 但是需要知道目标模型的内部结构, 可以应用在一些知

道内部结构的场景。此外作者指出该类攻击可以通过定期监测损失来检测是否存在中毒攻击。

3.1.3 人脸识别中毒攻击

Chen X 等人[50]提出一种新型的中毒攻击方法, 首次证明了中毒攻击在物理世界中是可行的。在该攻击中, 攻击者可以在不知道被攻击模型或其训练数据的情况下, 仅通过添加少量中毒样本实现目标性攻击。作者证明该类攻击只需添加 50 个左右的中毒样本, 就可以使攻击成功率达到 90%以上。

在文章中, Chen X 等人提出了两种中毒攻击策略, 即使用单个样本的 Input-instance-key 策略和使用整体模式作为突破口的 Pattern-key 策略。前者允许攻击者注入非常少的中毒样本来创建后门, 而后者允许从密钥模式创建各种中毒样本。

Input-instance-key 策略: Input-instance-key 攻击策略将某目标图片作为密钥, 创建了一组类似于密钥的中毒样本 $\sum(k)$, $\sum(k)$ 包含 k 种中毒样本的不同变化(添加的扰动)。

从 $\sum(k)$ 中随机选择 s 个样本作为中毒样本注入训练数据集, 即可实现中毒攻击。例如为了攻击人脸识别系统, 文中的实验证明中毒样本在 s=5 时就可以实现攻击, 即攻击者只需注入 5 个中毒样本来实现 Input-instance-key 攻击。

Pattern-key 策略: Pattern-key 攻击通过指定一种模式(如一副眼镜)作为密钥, 使得具有该模式的任何输入样本(如戴着这副眼镜的人脸)变为中毒样本。作者提出了三种不同的中毒数据注入策略, 分别为混合注入策略, 附件注入策略和混合附件注入策略。

混合注入策略通过将正常样本与密钥混合, 进而生成中毒样本和中毒样本。

但由于混合注入策略需要在训练和测试阶段干扰整幅图像, 这样的攻击模式在物理世界中是不可行的, 尤其是在测试阶段。为此作者提出附件注入策略, 该策略产生的图像相当于为目标添加附件。以人脸识别攻击为例, 使用注入策略完成模型训练后, 攻击者只需佩戴一副眼镜或一对耳环, 就能实现对人脸识别系统的攻击。因此, 附件注入策略所产生的中毒样本在实践中更容易实现。

而混合附件注入策略综合了混合注入策略和附件注入策略的优点, 实验证明该类型攻击只需注入约 50 个中毒样本, 就可以使得攻击成功率达到 90%以上。该中毒攻击首次证明了中毒攻击在物理世界的受威胁模型中是可行的, 比较适用对门禁等无需摘配饰等场景的人脸识别系统进行攻击。

3.1.4 StingRay 攻击

Suciu O 等人[51]提出了 FAIL 攻击模型, 该模型沿着特征、算法、样本和结构四个维度分析针对机器学习系统攻击的一般框架。在该框架内, 作者设计了一种具有普适性的目标中毒攻击方法 StingRay。StingRay 攻击可以生成与训练样本特征空间非常相似的中毒样本实现攻击。若要实现更复杂的攻击, 可以对中毒样本进行更改将模型边界推向目标类。此外, StingRay 攻击能够通过制作大量的中毒样本来增加中毒攻击的鲁棒性, 进而对抽样防御进行抵抗。实验结果表明该方法可以对 4 种真实的机器学习分类器进行攻击, 并可以绕过 2 种现有的中毒防御方法。

该类中毒样本存在着难以被人类发现的特点, 但同时也存在需要进行多次迭代的问题, 因此适用于知道目标模型内部结构或者对查询次数没有限制的模型。

3.1.5 简单的纯净标签攻击

文章[52]和[35]的攻击方法不仅需要修改测试阶段的样本来触发错误预测, 并且需要对训练数据集中的样本类标有一定程度的控制, 要求比较严格。Shafahi A 等人[17]提出清除标签攻击, 其中攻击者注入的训练样本由认证机构清晰地标记, 而不是被攻击者自己恶意标记, 同时无需对测试样本做任何修改。作者根据神经网络的复杂性和非线性特性找到在特征空间中与目标样本相近的中毒样本, 并将中毒样本注入训练数据集, 使得模型在“正常数据+中毒数据”的混合数据上重新训练。中毒样本的计算公式如下:

$$x' = \arg\min \left\| TM(\Theta, x) - TM(t, x) \right\|_2^2 + \beta \left\| x - b \right\|_2^2$$

其中 x 为输入神经网络的样本, b 为原始样本, x' 为中毒样本, β 用于控制中毒样本和原始正常样本的相似度。公式的第一项控制中毒样本移动到特征空间中的目标样本附近并嵌入目标类分布中; 第二项控制中毒样本 x' 看起来像真实正常样本(β 参数控制相似程度)。

作者采用双向分裂迭代算法进行优化, 使得中毒样本在特征空间中所处位置错误的情况下与原图足够相似。图 2 显示了迁移训练在除了最后一层外的所有层的权重都被冻结时的中毒攻击情况, 其中第一行为 5 个随机目标样本, 第二行为与每个目标样本对应的中毒样本。

图 2 迁移训练过程中的中毒攻击

Figure 2 Poisoning attack during migration training

3.1.6 端到端的纯净标签攻击

除了简单的纯净标签攻击外, Ali Shafahi 等人[17]还提出了一种“水印”策略, 使用多个(约 50 个)中毒样本实现端到端的中毒攻击。不同于简单的纯净标签攻击方法仅训练最后一层网络, 端到端的纯净标签攻击会重新训练神经网络所有层的参数。作者将中毒样本对网络性能的影响进行可视化分析, 并指出若只用单个中毒样本在端到端训练场景下进行攻击, 则中毒样本在特征空间中所处的位置将接近原始类标分布, 无法实现中毒攻击。为了避免在训练期间发生中毒样本和正常样本分离的情况, 作者将高透明度的水印添加到中毒样本中, 以允许一些不可分离的特征重叠, 同时保持视觉模糊度。这会将正常的目标样本的某些功能混合到中毒样本中, 并且经过重新训练后, 也能够使中毒样本保持在目标样本的特征空间附近。该攻击方法在 CIFAR-10 数据集中得到验证。

3.1.7 可转移的清洁标签中毒攻击

纯净标签中毒攻击将无害的外观(“正确”标记)中毒图像注入训练数据, 导致模型在对此数据进行训练后错误地分类目标图像。Zhu [53]考虑不访问受害者网络的输出、体系结构或训练数据而实现可转移中毒攻击。为了实现这一点, 提出了一种新的“polytope attack”, 它在特征空间中围绕目标图像构造一个凸多点, 从而保证一个线性分类器能够覆盖有毒数据集, 将目标分类为毒药类别。作者提供了两种实际的方法来进一步提高可转移性, 首先, 在制作毒药时打开“Dropout”, 以便从具有不同结构的各种网络中获取目标样本; 其次, 在多个层次上实现凸多面体目标, 即使在端到端的学习环境中也能保证攻击的成功。该中毒攻击实现中毒只有 1% 的训练集, 而可转移攻击成功率超过 50%。该攻击方法在 CIFAR-10 数据集中得到验证。图 3 显示了一个在二维空间上训练的线性支持向量机用特征碰撞攻击[17]和 Convex Polytope 攻击中毒训练集的示例。两个带条纹的红点是注入训练集中的中毒样本, 而带条纹的蓝点是目标, 不在训练集中。其他的点都在训练集中。即使中毒样本是距离目标最近的点, 最优线性支持向量机也会在左图中正确分类目标。凸多面体攻击将使两个中毒样本形成的线段与目标之间保持一小段距离。当线段与目标的距离最小化时, 如果重新约束模型中的目标过拟合, 则其负边距也最小化。

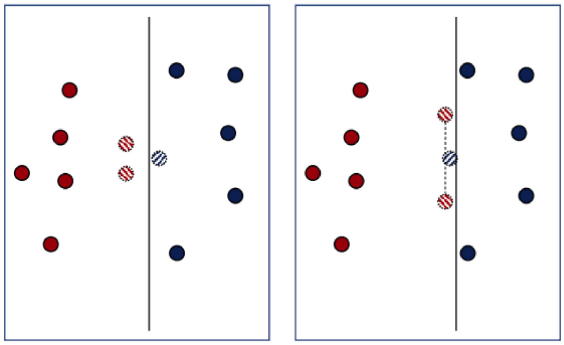

图 3 一个在二维空间上训练的线性支持向量机用特征碰撞攻击和 Convex Polytope 攻击中毒训练集的示例

Figure 3 An illustrative toy example of a linear SVM trained on a two-dimensional space with training sets poisoned by Feature Collision Attack and Convex Polytope Attack respectively

3.1.8 剪枝感知攻击(Pruning-Aware Attack)

剪枝感知攻击[54]是针对剪枝防御(见 5.2.1 节)提出的攻击。剪枝感知攻击通过将纯净标签攻击行为和中毒行为集中在同一组神经元上从而逃避剪枝防御, 使得后门更加难以被检测到。剪枝感知攻击首先修剪(即删除)DNN 中在正常样本输入时休眠的神经元; 然后使用中毒训练数据集重新训练已剪枝的 DNN 模型, 使得模型在正常样本输入时输出正常结果, 在中毒样本输入时输出对应的异常结果。但是, 由于攻击者只能改变 DNN 的权重, 而不能改变其超参数。因此, 面对剪枝防御时, 攻击者通过将所有被修剪的神经元与相关的权重和偏差一起重新恢复到神经网络中进行“感知”修剪 DNN 结构。此时, 修剪感知的神经元在中毒样本和正常样本输入时都处于休眠状态, 可以使得剪枝防御无效。但是此类攻击主要针对剪枝防御, 适用于已知目标模型采用了剪枝防御的情况下。

3.1.9 单像素中毒攻击

Alberti M 等人[55]提出一种在原网络结构未知的情况下, 仅通过修改训练图像中单个像素点就能使得神经网络在测试阶段表现行为异常的攻击。作者在两个经典的数据集(CIFAR-10 和 SVHN-10)上展示了对于训练数据集中的所有图像, 仅单个像素的修改就足以破坏多个模型的训练过程。这种篡改很难用肉眼观察到, 但在实验中发现对六种先进的网络架构都能够实现有效攻击。

单像素中毒攻击与模型的选择无关, 并且不对网络的特定体系结构或权重做出任何改变, 仅将某一类图像(RGB 图像)的同一位置的蓝色通道像素值设置为 0, 图 4 显示了原始图像(a 和 c)和修改后对应的中毒图像(b 和 d)的区别。因为神经网络在训练过程中关注的是被修改的扰动, 而不是正常图片本身的特征, 因此可以通过在另一类的图片上添加相同的扰动来实现攻击, 从而诱导网络对其进行错误分类。

作者指出这类攻击近似于对图像添加椒盐噪声, 因此可以用中值滤波或其他图像预处理技术实现对此类攻击的防御。

(a) 原图 (b) 中毒 (c) 原图 (d) 中毒

图 4 从 CIFAR-10 中提取的两幅飞机图像

Figure 4 The figure shows two images drawn from the airplane class of CIFAR-10

3.1.10 PATOM 多任务中毒攻击

大量的研究已经表明单任务学习模型(single-task learning, STL)容易受到中毒攻击的影响[38,56]。多任务联合学习(multi-task learning, MTL)通过利用多个相关任务中包含的有用信息来提高每项任务的性能, 实现比单任务学习更好的效果[57]。Zhao M 等人[58]分析了多任务学习模型的最优中毒攻击, 并将多任务关系学习(multi-task relationship learning, MTRL)的最佳中毒攻击编写为双层程序, 该程序能够任意选择目标任务和攻击任务。

Zhao M 等人[58]还设计了一种 PATOM 攻击算法, 用于计算最优攻击策略。PATOM 算法利用 MTRL 子集问题的最优条件计算上层目标函数的梯度。真实数据集上的实验结果表明, MTRL 模型对中毒攻击非常敏感, 攻击者可以通过直接对目标任务下毒或间接对相关任务下毒来显着降低目标任务的性能。同时还发现被攻击的任务总是紧密相关的, 这为防御此类攻击提供了线索。

3.1.11 针对差分隐私的数据中毒攻击

数据中毒攻击的目的是通过对训练集进行逆修改来操纵由学习算法生成的模型。Ma Y 等人[59]认为差分隐私是针对此类攻击的防御措施, 并证明不同的学习模型对数据中毒攻击具有抵抗力。当攻击者只能毒害少量的样本时, 学习模型能够抵抗数据中毒攻击。然而, 当攻击者毒害更多数据时, 这种保护会降低。为了说明这一点, 作者设计了针对目标学习者和输出扰动学习者的攻击算法, 并用合成和真实的数据演示了对各种隐私机制和学习者的攻击性能。实验表明, 当攻击者可以毒害足够多的训练样本时, 方法是有效的。虽然攻击是有效的, 但是该攻击的理论下限和经验性能之间仍然存在差距。这可能是因为下限松动, 或者攻击不够有效。

3.1.12 面向联邦学习的后门攻击

联邦学习的分布式使得检测和防御对抗性攻击成为一项具有挑战性的任务。Sun Z 等人[60]提出了面向联邦学习的后门攻击, 其目标是降低模型在目标任务上的性能, 同时在主要任务上保持良好的性能。与现有的工作不同, 作者允许非恶意客户端正确标记目标任务的样本。作者在 MNIST 数据集下研究了联邦学习的后门攻击和防御并观察到, 在没有防御的情况下, 攻击的性能在很大程度上取决于中毒样本的数量和目标任务的“复杂性”。此外, 作者还证明了规范剪裁和“弱”差分隐私在不影响整体性能的情况下减轻了攻击。并且在 TensorFlow 联邦中实现了攻击和防御, 这是一个用于联邦学习的 TensorFlow 框架。同时要取得合理的成功, 需要有大量的中毒样本。作者还发现规范剪裁极大地限制了已知后门攻击的成功。此外, 加入少量高斯噪声, 可以帮助进一步减轻攻击者的影响, 这对联邦学习的后门攻击的防御提供了方向。

3.1.13 其他方法

上面讨论的中毒攻击要么是近期文献中比较流行的, 要么代表了流行的研究方向。下面我们简要描述近期在深度神经网络上的中毒攻击的进一步研究。

Luis 等人[25]提出了一种基于反梯度优化思想的中毒算法, 将中毒攻击拓展到多类问题并对中毒攻击的迁移性进行研究。Matthew 等人[26]对线性回归模型的中毒攻击及其对策进行了第一次系统研究。提出了一个理论上基于理论的用于线性回归的优化框

架, 并证明了它在一系列数据集和模型上的有效性。Xiao H 等人[20]通过提供一个框架来研究流行的特征选择方法的稳健性, 并表明特征选择方法在攻击下可能会受到严重影响, 突出了特定对策的必要性。Xie C[61]提出一种针对联邦学习的分布式后门攻击。一个攻击者容易被检测, 将一个补丁拆成N块, 用N个攻击者去攻击, 可以达到整个补丁用一个攻击者去攻击的效果。

3.2 模型中毒

模型中毒攻击是指直接对模型进行中毒攻击的方法, 在一般情况下指直接向用户提供中毒的模型。由于此类攻击的攻击者直接提供模型, 因此可以任意修改训练数据、模型结构和模型参数等。此类模型可以以极高的高准确率的实现用户的要求, 但是对于带有秘钥的样本或者指定的样本会显示中毒行为, 即可以输出攻击者预先设定的结果。目前模型中毒攻击可以根据攻击者的出发点分为恶性模型中毒攻击和良性模型中毒攻击, 恶性模型中毒攻击指攻击者提供的中毒模型会在一定程度上损失模型持有者的利益, 例如使带有指定秘钥的样本获得最高权限等; 良性模型中毒攻击指攻击者提供的中毒模型不会对持有者的利益造成威胁和损失, 例如中毒模型的中毒行为只是为了证明模型创建者的身份, 以此保护对应的知识产权。模型中毒攻击产生的模型通常存在于从网络上直接下载的模型和向第三方购买模型中, 虽然使用者可以会用自己的数据集进行再训练, 但仍然可能存在中毒的情况。相对于数据中毒攻击, 模型中毒攻击因为具有更多的数据、模型的相关信息, 所以可以获得更好的攻击效果, 但由于大公司并不会存在模型外包的情况, 因此此类攻击的应用范围也相对狭窄。

3.2.1 正则水印嵌入

Uchida Y 等人[62, 63]提出了一种使用正则化手段在模型参数中嵌入水印的通用框架, 以保护知识产权并检测训练好的模型的知识产权是否受到侵权。这是首次尝试在深度神经网络中嵌入水印, 按照嵌入情况的不同可以分为三类: 训练嵌入、微调嵌入和提取嵌入。训练嵌入是在获得训练数据和标签的情况下, 从头训练网络; 微调嵌入则是首先用训练好的模型参数初始化网络, 然后再调整输出层附近的网络配置; 提取嵌入是在微调中进行的, 将没有蒸馏的模型预测作为标签。在标准提取框架中, 首先训练大型网络(或多个网络), 然后使用大型网络的预测标签训练较小的网络以压缩大型网络。实验表明即使对网络进行微调或参数剪枝, 嵌入的水印也不会消失, 甚至在修剪了 65%的参数后, 水印仍然完整。

3.2.2 零位水印算法

Merrer E L 等人认为文章[35]提出的水印防御方法虽然是模型保护的进步, 但这种技术仅允许从本地和具有完全访问权限的网络中提取水印, 并不符合实际。这是因为发布的模型虽然可以私下重复使用, 但发布者不会公开模型的权重。因此, Merrer E L 等人[64]提出了一种新的零位水印算法, 该算法将对抗训练获得的对抗样本和正常样本一起用于训练神经网络模型, 并通过对模型的微调实现对所有类的正确分类。此外, 该算法可以在限制受保护模型的性能损失的同时, 通过少量的远程查询提取模型水印, 而无需访问模型本身。

3.2.3 DeepMarks

Chen H 等人[65]提出了一种新的端到端的指纹系统框架 DeepMarks, 该方法在保持准确性的同时, 使模型所有者能够将独特的指纹(每个用户分配得到的唯一的二进制代码矢量)嵌入到神经网络的不同层中。DeepMarks 可以对包括模型压缩和模型微调攻击进行防御, 作者对各种对比测试进行广泛的实验验证评估, 例如 MNIST 和 CIFAR-10 数据集, 证实了 DeepMarks 框架的有效性和鲁棒性。

3.2.4 BadNets 简单纯净标签攻击

Gu T 等人[21]展示了外包机器学习模型训练(Machine Learning as a Service, MLaaS)和对网络下载模型迁移训练过程中存在的安全问题。作者认为 ARUROR 防御方法[48]之所以可以对中毒攻击进行防御是因为在每次训练后存在验证阶段, 该阶段可以通过验证集够显示出模型的不良表现。此外作者提出攻击者可以创建一个恶意训练的网络(如后门神经网络, 或一个 BadNet)实现攻击, 这种网络模型在用户的训练和验证数据集上都具有最佳的性能, 但对于攻击者选择的特定输入样本会进行异常分类, 表现出中毒行为。

3.2.5 PoTrojan 攻击

文章[35]和[51]设计的中毒攻击方案都需要重新训练学习模型, 这不仅消耗了大量时间还会改变原模型的参数, 影响模型对原始数据的分类准确率。因此 Zou M 等人[66]提出了一种新的中毒攻击方法, 该方法与特洛伊木马攻击相似, 但只需要训练 PoTrojans 插入层的下一层便可以得到中毒模型。大多数时候, PoTrojans 插入层保持休眠状态, 不会影响其 DNN 模型的正常功能。PoTrojans 只能在非常罕见的情况下被触发。然而, 一旦被激活, PoTrojans 可能导致 DNN 模型出现故障。

3.2.6 面向基于图的推荐系统的中毒攻击

推荐系统是许多 web 服务的重要组成部分, 它可以帮助用户找到与其兴趣匹配的项目。推荐系统容易受到中毒攻击, 在这种攻击中, 攻击者会向给定的系统注入假数据, 以便系统根据攻击者的需要提出建议。Fang M 等人[67]对基于图的推荐系统的中毒攻击进行了系统的研究, 提出了对图形化推荐系统的优化中毒攻击。作者提出对基于图的推荐系统的中毒攻击可以归结为一个优化问题, 而优化问题可以用梯度下降法近似求解。此外, 该攻击比现有的基于图的推荐系统的攻击更有效。原因是现有的攻击没有针对基于图的推荐系统进行优化。同时攻击也可以在灰盒和黑盒设置下转移到其他推荐系统。通过使用监督机器学习技术对用户的评分进行分析, 服务提供商可以检测出大量的假用户, 也可以错误地预测出一小部分正常用户是假用户。此外, 当服务提供者部署了一个检测器并从推荐系统中排除了预测的假用户时, 攻击仍然有效。攻击方法在两个数据集 Movie 和 Video 中得到验证, 注入 1%的假用户时, 该攻击可以使目标项在某些情况下推荐的普通用户为原来的 580 倍。

3.2.7 图卷积网络中毒邻节点间接对抗性攻击

图卷积神经网络是一种在相邻节点上学习聚集的神经网络, 在节点分类任务中取得了良好的性能。然而, 这种图卷积节点分类器会被图上的对抗性扰动所欺骗。滥用图卷积, 一个节点的分类结果可能会受到毒害其邻居的影响。Takahashi T[68]证明了节点分类器可以被中毒距离目标只有一个节点甚至两个或更多跳。为了实现攻击, 作者提出了一种名为“POISONPROBE”新的攻击方法, 它只在远离目标的单个节点上搜索较小的扰动, 毒害节点的特征, 导致错误分类到远不止一跳的目标。在 CoraML 数据集中, 距离目标一跳范围内的攻击成功率最高为 100%, 距离目标两跳范围内攻击的成功率最高为 92%。图 5 显示了通过中毒图卷积网络中单个节点而实现间接对抗性攻击。有毒信息通过图形传播, 影响其他节点的分类结果。

3.2.8 其他攻击

与数据中毒方法相同, 对模型中毒的研究也非常的活跃, 我们只对最近的模型中毒攻击进行简单的介绍, 有兴趣的研究人员可以直接查看原文。

Adi Y 等人[69]提出了一种以黑盒方式对深度神经网络加水印的方法, 该方法适用于一般分类任务, 可以轻易地与当前的学习算法结合使用。

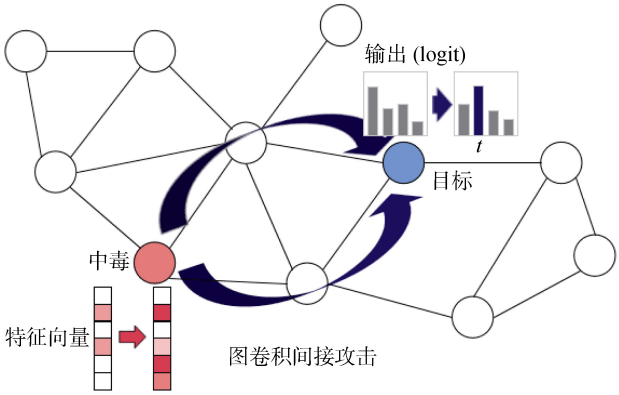

图 5 通过中毒单个节点的间接对抗性攻击

Figure 5 Indirect adversarial attack by poisoning a single node

4 中毒攻击的可能性

深度学习由于其良好的性能被广泛的应用到各个领域。然而良好的深度学习模型需要大量的训练数据, 对 GPU 性能的高要求使得许多用户将训练过程外包给第三方, 这给中毒攻击提供了很好的下毒环境。而中毒数据标签经官方认证, 且中毒攻击的隐蔽性减少了中毒攻击被发现的可能性。

训练所需数据集大: 由于训练网络需要大量数据集, 人们可能会从不同来源收集数据并将它们组合起来以生成更大的数据集或直接从网络上下载现有数据集, 攻击这则可以通过提供中毒的数据集, 从而达到攻击的目的。

GPU 性能要求高: 卷积神经网络需要大量的训练数据和数百万的权重参数才能获得良好的结果。然而, 这些网络的训练通常对时间成本和计算成本要求较高, 需要在许多 GPU 上进行数周的训练。因此, 许多用户将训练过程外包给云或依赖预先训练的模型, 然后根据特定任务进行调整。例如: 谷歌的云机器学习引擎[70]允许用户上传 TensorFlow 模型和训练数据, 然后在云中进行训练。微软提供 Azure 批量人工智能训练[71], 亚马逊提供了一个预构建的虚拟机[72], 其中包括几个深度学习框架, 可以部署到亚马逊的 EC2 云计算基础架构中。

中毒攻击隐蔽: 攻击者可以创建一个恶意训练的网络, 它在用户的训练和验证样本上具有最佳性能, 但可以控制分类器在特定测试数据上的行为。

中毒数据标签经官方认证: 虽然目前存在几种方法可以处理噪声或损坏的标签[73,74]。但是, 这些技术解决的是输入标签上的错误, 而不是内容上的错误。因此, 它们不能有效地防御本文涉及的中毒攻击。

5 中毒攻击的防御方法

目前对于中毒攻击的防御方法可以作用阶段分为: 数据及特征修改、模型修改、输出防御三类。表2是对深度学习中的中毒攻击的防御方法的归纳。

5.1 数据及特征修改

数据及特征修改主要是指在数据或者特征输入模型之前对其进行预处理, 从而达到防御的效果。

表 2 深度学习中的中毒攻击的防御方法

Table 2 Defense methods of poisoning attack in deep learning

分类	实例	原理	防御形式	是否修改模型
数据及特征修改	数据预处理防御[75]	对输入数据进行预处理	对数据或者特征输入进行预处理	否
	协作式深度学习系统——AUROR 防御[48]	自动识别并显示分布异常的屏蔽特征	对数据或者特征输入进行预处理	否
模型修改	剪枝防御[47]	消除纯净输入上处于休眠状态的神经元来减少后门网络的大小	对模型进行修改	是
	微调防御/重新训练[81,49]	训练中毒的神经网络, 使得后门触发器无效	对模型进行修改	是
	精细剪枝防御[47]	修剪休眠的神经元, 微调模型	对模型进行修改	是
	DeepInspect 检测框架[82]	使用条件生成模型从查询的模型中学习潜在触发器的概率分布从而检索后门插入的足迹	对模型进行修改	是
输出防御	基于损失的防御方法[22]	若目标模型损失多次超过阈值, 将触发准确性检查	对模型的输出结果进行分析	否
	集成防御[83]	结合不同模型的预测结果来判断样本的预测类标	对模型的输出结果进行分析	否
	检测器防御[84]	通过支持向量机和决策树检测输入	对模型的输出结果进行分析	否
	多任务模型防御[58]	通过数据清洗和提高多任务联合学习的鲁棒性	对模型的输出结果进行分析	否

5.1.1 数据预处理防御

Liu Y 等人[75]除了采用检测防御机制和重训练策略外, 还尝试通过对输入数据进行预处理来使得中毒攻击无效。鉴于预处理器的目标是防止非法输入数据触发木马而不影响神经网络的正常功能, 因此作者在输入数据和神经网络之间放置一个自动编码器作为输入预处理器, 使得预处理器的输入和输出尺寸相同, 并将预处理器的输出作为神经网络的输入。作者表明, 在输入预处理防御方法中, 90.2%的特洛伊木马触发器无效, 并且在这种防御方法中神经网络被视为黑盒, 即不需要关于神经网络的相关知识。

5.1.2 协作式深度学习系统——AUROR 防御

Shen S 等人[48]提出一种防御方法, 对协作式深度学习系统的中毒攻击进行防御。在间接协作学习中每个用户不是直接将原始数据提交给服务器, 而是屏蔽数据的部分信息并将屏蔽后的数据发送给服务器。服务器可以通过其他用户上传的数据对屏蔽的特征进行学习从而生成全局模型, 在保证每个用户的数据隐私的同时显着降低集中式服务器上的计算成本。恶意用户可以篡改训练数据集, 进而影响全局模型的行为[76]。

作者引入了一种称为 AUROR 的统计机制对间接协作系统中毒攻击进行防御。AUROR 能够自动识别并显示分布异常的屏蔽特征的过程, 并基于这些特征检测系统中的恶意用户。图 6 显示了 AUROR 防御的过程。AUROR 首先对屏蔽特征进行分析, 将所有用户上传的特性值分组到不同的类别中, 如果两个类的中心距离大于阈值 α, 则特征被标记为指示性特征, 并将少的一类标记为可疑群体, 当某个用户上传特征被标记为可疑群体的次数超过 t 次, 则认为是恶意用户。最后排除恶意用户提交的特征, 进行模型的全局训练。

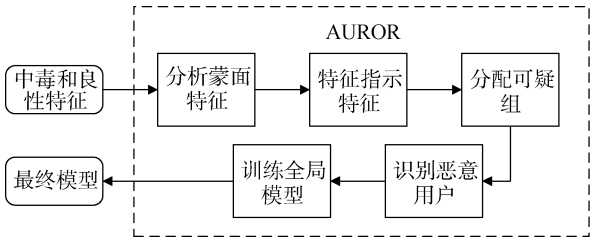

图 6 AUROR 设计细节及步骤

Figure 6 The design details and steps involved in AUROR

作者通过实验证明, 在没有攻击的情况下, AUROR 防御几乎不改变系统的分类准确度。在有30%恶意用户的情况下, AUROR 防御只降低了模型3%的分类准确度。因此, 在 AUROR 防御下, 可以实现准确而强大的间接协作学习系统。

5.2 模型防御

模型防御是指对模型进行修改从而实现防御

效果。

5.2.1 剪枝防御(Pruning Defense)

剪枝技术早期用于减少 DNN 的计算消耗[77-80], 并可以在不影响分类准确率的情况下修剪大部分神经元。Gu T 等人[21]根据经验证明, 中毒样本触发了在正常样本输入时处于休眠状态的神经元。因此, Liu K 等人[47]提出将剪枝作为防御手段。剪枝防御通过消除纯净输入上处于休眠状态的神经元来减少后门网络的大小, 从而禁用中毒行为, 增强深度神经网络的安全性。

剪枝防御的工作原理如下: 防御者使用验证数据集的正常输入执行来自攻击者的 DNN, 并记录每个神经元的平均激活值。然后, 防御者以平均激活的递增顺序迭代地修剪来自 DNN 的神经元, 并在每次迭代中记录剪枝网络的准确性。当验证数据集的准确度降至预定阈值以下时, 终止剪枝防御。

剪枝防御计算简单, 仅需防御者通过每个验证输入执行通过网络的单个前向传递过程来评估(或执行)经验证数据训练的 DNN。经验证, 剪枝防御可以成功防御针对交通标志[51], 语音识别[35]和人脸识别[58]等的中毒攻击。

5.2.2 微调防御/重新训练(Fine-tuning)

微调训练最初是在迁移学习的背景下提出的策略, 其中用户想要调整针对特定任务训练的 DNN 以执行另一相关任务。微调训练使用预训练的 DNN 权重作为初始化, 并设置较小的学习率, 相比于从头训练能够缩短模型的训练时间。例如, 对 AlexNet 进行微调训练所需时间不足一个小时, 而从头开始训练 AlexNet 可能需要六天以上[81]。因此, 从计算成本的角度来看, 微调仍然是一种可行的防御策略。

重新训练意味着继续训练中毒的神经网络, 使得后门触发器无效, 但仍能正常使用合法数据。Liu Y 等人[49]的实验显示, 重新训练后模型对正常样本的分类准确率从 98%降低到 96%, 但中毒攻击的成功率从 99%降到了 6%以下, 实现了较好的防御效果。

但在部分稀疏网络上进行微调和重训练是无效的, 因为中毒神经元不会被纯净的数据激活, 因此这些神经元的梯度接近 0 并且在很大程度上不受微调训练的影响。因此常用梯度下降方法对至少有一个神经元激活的网络进行微调防御。

5.2.3 精细剪枝防御(Fine-pruning Defense)

为快速有效的对剪枝感知攻击(见 3.2.9)进行防御, Liu K 等人[47]提出一种结合剪枝和微调防御的精细剪枝防御。精细剪枝防御首先修剪神经网络中休眠的神经元, 然后用干净的数据集进行微调, 使涉及中毒行为的神经元权重被更新。在此过程中, 剪枝防御和微调防御起着互补作用, 剪枝防御可以删除休眠的神经元, 使得中毒攻击集中到较少的神经元中, 微调防御可以重新训练神经元, 消除中毒对深度神经网络模型的影响。

5.2.4 DeepInspect 检测框架

深度神经网络(DNNs)容易受到神经木马(NT)攻击, 在 DNN 训练过程中, 攻击者会注入恶意行为。当输入被攻击者指定的触发器模式标记时, 会激活此类“后门”攻击, 从而导致对模型的错误预测。Chen H 等人[82]针对解决未知DNN到NT攻击的安全问题, 提出了 DeepInspect, 这是第一个具有最小先验知识的黑盒木马检测框架。在没有干净的训练数据或真实参考模型的帮助下, 检查预先训练的 DNN 的安全性, 使用条件生成式对抗网络学习潜在触发器的概率分布, 从而检索后门插入的足迹, 并且通过模型修补实现有效的木马缓解。大量实验表明, 与以前的工作相比, DeepInspect 可提供卓越的检测性能和更低的运行时间开销。证实了 DeepInspect 针对各种基准的最新 NT 攻击的有效性、效率和可扩展性。图 7 所示, DeepInspect 包含三个主要步骤: (1)利用模型反演方法来恢复训练数据集, 假设 DNN 有 n 个输出类别, DeepInspect 首先采用模型反演方法来生成一个包含所有类别的替代训练集$\{X_MI, Y_MI\}$; (2)利用生成模型来重建特洛伊木马攻击可能使用的触发器; (3)在使用条件生成式对抗网络为所有输出类生成触发器之后, DeepInspect 将特洛伊木马检测制定为异常检测问题。收集所有类别中的扰动统计数据, 将扰动程度(变化幅度) 作为异常检测的检验统计量。假设检验和鲁棒性统计来检测触发扰动中异常值的存在, 使用双中值绝对偏差作为检测标准。

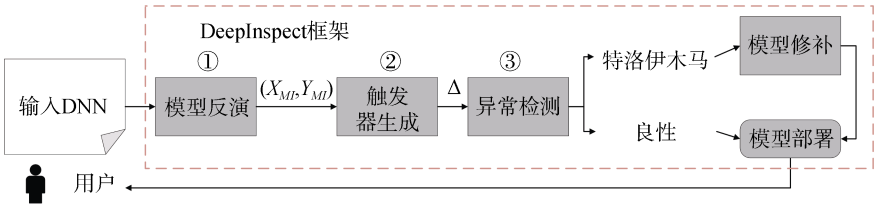

图 7 DeepInspect 框架

Figure 7 DeepInspect framework

5.3 输出防御

输出防御是指通过对模型的输出结果进行分析, 从而实现防御的效果。

5.3.1 基于损失的防御方法

Yang C 等人[22]针对会使得检测器检测率明显下降的中毒攻击, 提出了一种基于损失的防御对策, 以极低的消耗检测中毒样本。一旦将数据(无论正常或中毒)注入目标模型, 就会记录目标模型的损失。如果损失超过预定阈值, 将显示警告。如果警告的次数一定次数, 将触发准确性检查以检查是否确实正在受到中毒攻击。同样, Liu Y 等人[49]注意到有针对性的中毒攻击会不成比例地降低模型对目标样本的分类准确率, 并建议将其用作检测技术。

需要注意的是这里提到的基于损失的防御方法本身并没有什么防御效果, 只是作为一种反馈, 提醒人们数据或者模型存在异常, 还是需要人工对数据处理, 或者与其他算法搭配。

5.3.2 集成防御(Ensemble Defense)

Hitaj D 等人[83]在提出集成算法时主要用于逃避神经网络合法所有者的验证, 从而躲避模型盗窃的检测。由于本文将后门视为中毒攻击, 因此集成算法在本文中被视为防御方法, 为了帮助读者理解, 我们在括号中对可能有歧义的词语进行注释。

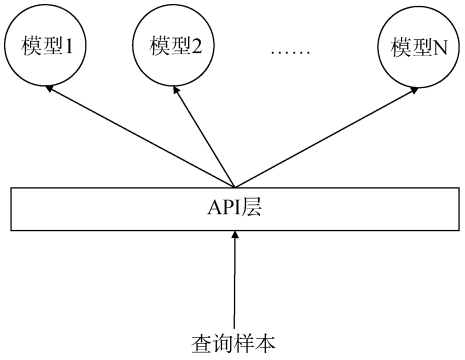

图 8 集成设置

Figure 8 The Ensemble set-up

集成防御(原文中为集成攻击)结合不同模型的预测结果综合判断预测样本的类标从而达到防御中毒攻击的效果。集成防御从不同供应商那里窃取(获取)预测效果相差不大并且执行任务相同的神经网络模型并置于 API 层之后, 如图 8 所示。用户只有黑盒访问的权限, 即只能向模型提出查询请求, 将数据输入至 API 接口。当预测样本输入到 API 层时, AIP 将根据逻辑查询每个模型并获得他们的预测结果, 然后根据投票机制将票数最多的类作为预测结果返回给进行查询的用户。集成防御的质量与深度模型本身的质量相关, 高质量的集成防御不仅可以对中毒攻击进行防御, 甚至对纯净输入样本有着比单模型更高的分类准确率。

5.3.3 检测器防御

Liu Y 等人[49]和 Hitaj D 等人[83]都曾提出用现有的异常检测方法[84]来直接检测输入是否是异常的(即潜在特洛伊木马触发器)。如果被识别为异常, 则不会将输入提供给神经网络。目前实现的异常检测方法包括支持向量机(Support Vector Machines, SVM)和决策树(Decision Tree, DT)。事实证明, DT 虽然有 12.2%的虚警率(被判为异常的样本中, 实际为正常样本的概率), 但检测率高达 99.2%, 检测效果仍然优于 SVM。

5.3.4 多任务模型防御

Zhao M 等人[58]从数据清洗和提高 MTL 的鲁棒性两个方面设计防御策略。首先, 机器学习者可以通过人工验证来检查局部相关性比较强的任务数据。此外, 一旦任务被证明是恶意的, 学习者就可以检查与其密切相关的任务, 这将显着减少学习者检查数据的工作量。其次, 提高 MTL 的鲁棒性也可以成为防御数据中毒攻击的有效方法。MTL 利用任务相关性来提高单个任务的性能, 攻击者也可以利用这种相关性来发起间接攻击。因此提高 MTL 鲁棒性的一种可能方法是区分有利的任务相关性和有害的任务相关性, 以便我们可以保持有利的关联性并减少学习过程中的有害关联性。

6 研究方向与展望

在前几节中, 我们全面回顾了近年来关于深度学习中毒攻击的文献。在上面那些章节中, 大多的描述是关于技术细节的。下面, 就这个新出现的研究方向, 我们将做出更多一般性的讨论。

新技术: 随着深度学习的不断发展, GAN、进化、强化学习等方法活跃于各个领域, 此类方法均可以成为中毒样本的优化或者训练手段。因此新技术的出现往往还可能伴随着新的攻击。

新领域: 本文主要对机器视觉上的攻击进行研究, 然而随着深度学习被广泛应用于各个领域, 各个领域也均应对中毒攻击引起重视, 例如: NLP、语音、网络、无线信号、电磁信号等。

新应用: 中毒攻击不仅仅只是针对单个深度学习模型, 中毒攻击对应于污染训练数据。因此, 在其它应用中, 例如, 分布式、云存储、https 加密协议也均应对此类攻击引起重视。

新防御: 目前虽然各种防御层出不穷, 然而随着针对防御的新攻击出现, 目前的防御已经无法很好的实现防御效果, 因此对攻击的防御是一个需要长期不断研究的课题。

7 结论

深度学习是当前机器学习和人工智能兴起的核心。随着它被成功的应用到自动驾驶任务、人脸支付等安全领域中, 深度学习模型的安全问题逐渐成为新的研究热点。虽然深度神经网络在解决复杂问题方面已经取得了惊人的成就。但最良好的深度学习模型需要大量的训练数据、对 GPU 的高性能要求, 这使得许多用户将训练过程外包给第三方或从网络上下载数据或模型, 这都给中毒攻击提供了便利的下毒环境。本文首次综合性介绍了目前深度学习模型的中毒攻击和防御, 从理论上对攻击和防御进行分析, 并回顾了中毒攻击的方法设计, 对攻击进行分析比较, 并概述了中毒攻击的防御方法。最后, 通过引用的参考文献为本课题的研究指明更广阔的前景。

致 谢 本课题得到浙江省自然科学基金项目(No.LY19F020025),宁波市“科技创新 2025”重大专项(No.2018B10063)资助。

参考文献

[1] Deng L, Liu Y. A Joint Introduction to Natural Language Processing and to Deep Learning[M]. Deep Learning in Natural Language Processing. Singapore: Springer Singapore, 2018: 1-22.

[2] G. Litjens, T. Kooi, B. Bejnordi, et al. A survey on deep learning in medical image analysis[J]. Medical image analysis, 2017, 42(1): 60-88.

[3] Manic M, Amarasinghe K, Rodriguez-Andina J J, et al. Intelligent Buildings of the Future: Cyberaware, Deep Learning Powered, and Human Interacting[J]. *IEEE Industrial Electronics Magazine*, 2016, 10(4): 32-49.

[4] A. Mousavi, G. Baraniuk. Learning to invert: Signal recovery via deep convolutional networks[C]. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017: 2272-2276.

[5] G. Hains, A. Jakobsson, Y. Khmelevsky. Towards formal methods and software engineering for deep learning: security, safety and productivity for dl systems development[C]. *Annual IEEE International Systems Conference (SysCon)*, 2018:1-5.

[6] H. Baomar, J. Bentley. An Intelligent Autopilot System that learns piloting skills from human pilots by imitation[C]. *International Conference on Unmanned Aircraft Systems (ICUAS)*. 2016: 1023-1031.

[7] K. Van, A. Bronkhorst. Human-AI Cooperation to Benefit Military Decision Making[OL]. NATO, 2018.

[8] Yang Z, Yu W, Liang P W, et al. Deep Transfer Learning for Military Object Recognition under Small Training Set Condition[J]. *Neural Computing and Applications*, 2019, 31(10): 6469-6478.

[9] Chakraborty K, Bhattacharyya S, Bag R, et al. Comparative Sentiment Analysis on a Set of Movie Reviews Using Deep Learning Approach[M]. The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018). Cham: Springer International Publishing, 2018: 311-318.

[10] Akhtar N, Mian A. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey[J]. *IEEE Access*, 2018, 6: 14410-14430.

[11] B. Biggio, I. Corona, D. Maiorca, et al. Evasion attacks against machine learning at test time[C]. *In Joint European conference on machine learning and knowledge discovery in databases*, 2013: 387-402.

[12] C. Szegedy, W. Zaremba, I. Sutskever, et al. Intriguing properties of neural networks[EB/OL]. arXiv preprint arXiv:1312.6199, 2013.

[13] I. Goodfellow, J. Shlens, C. Szegedy. Explaining and harnessing adversarial examples[EB/OL]. In International Conference on Learning Representations, 2015. arXiv preprint arXiv:1412.6572,2015.

[14] Barreno M, Nelson B, Sears R, et al. Can Machine Learning be Secure?[C]. *The 2006 ACM Symposium on Information, computer and communications security*, 2006: 16-25.

[15] B. Biggio, B. Nelson, P. Laskov. Support vector machines under adversarial label noise[C]. *Asian conference on machine learning*, 2011: 97-112.

[16] M. Kloft, P. Laskov.Online anomaly detection under adversarial impact[C]. *The Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010: 405-412.

[17] A. Shafahi, R. Huang, M. Najibi, et al. Poison frogs! targeted clean-label poisoning attacks on neural networks[C]. *Advances in Neural Information Processing Systems (NIPS)*, 2018: 6103-6113.

[18] P. W. Koh, P. Liang. Understanding black-box predictions via influence functions[EB/OL]. arXiv preprint arXiv:1703.04730, 2017.

[19] S. Mahloujifar, D. I. Diochnos, M. Mahmoody. Learning under p-tampering attacks[EB/OL]. arXiv preprint arXiv:1711.03707, 2017.

[20] H. Xiao, B. Biggio, G. Brown, et al. Is feature selection secure against training data poisoning?[C]. I*nternational Conference on Machine Learning*, 2015; 1689-1698.

[21] T. Gu, B. Dolan-Gavitt, S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain[EB/OL]. arXiv preprint arXiv:1708.06733, 2017.

[22] C. Yang, Q. Wu, H. Li, Y. Chen. Generative poisoning attack method against neural networks[EB/OL]. arXiv preprint arXiv:1703.01340, 2017.

[23] S. Alfeld, X. Zhu, P. Barford. Data poisoning attacks against autoregressive models[C]. *Thirtieth AAAI Conference on Artificial Intelligence*, 2016:368-396.

[24] B. Li, Y. Wang, A. Singh, et al. Data poisoning attacks on factorization-based collaborative filtering[C]. *Advances in neural information processing systems*, 2016: 1885-1893.

[25] L. Munoz-González, B. Biggio, A. Demontis, et al. Towards poisoning of deep learning algorithms with back-gradient optimization[EB/OL]. arXiv preprint arXiv:1708.08689, 2017.

[26] W. Koh, P. Liang. Understanding black-box predictions via influence functions[C]. *The 34th International Conference on Machine Learning-Volume 70*. 2017: 1885-1894.

[27] J. Steinhardt, P. W. Koh, P. Liang. Certified defenses for data poisoning attacks[EB/OL]. arXiv preprint arXiv:1706.03691, 2017.

[28] Y. Liu, S. Ma, Y. Aafer, et al. Trojaning attack on neural networks[C]. *Network and Distributed System Security Symposium*, 2017:258-264.

[29] X. Chen, C. Liu, B. Li, et al. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning[EB/OL]. arXiv preprint arXiv:1712.05526, 2017a.

[30] A. Turner, D. Tsipras, A. Madry. Clean-label backdoor attacks[OL]. 2019.

[31] S. Chen, M. Xue, L. Fan, et al. How Can We Craft Large-Scale Android Malware?[C]. I*EEE 1st International Workshop on Artificial Intelligence for Mobile (AI4Mobile)*. 2019: 21-24.

[32] Chen S, Xue M H, Fan L L, et al. Automated Poisoning Attacks and Defenses in Malware Detection Systems: An Adversarial Machine Learning Approach[J]. *Computers & Security*, 2018, 73: 326-344.

[33] B. Li, Y. Wang, A. Singh, et al. Data poisoning attacks on factorization-based collaborative filtering[C]. *Advances in neural information processing systems*, 2016: 1885-1893.

[34] X. Chen, C. Liu,B. Li, K. Lu, et al. Targeted backdoor attacks on deep learning systems using data poisoning[EB/OL]. arXiv preprint arXiv:1712.05526, 2017.

[35] Li K, Mao S G, Li X, et al. Automatic Lexical Stress and Pitch Accent Detection for L2 English Speech Using Multi-distribution Deep Neural Networks[J]. *Speech Communication*, 2018, 96: 28-36.

[36] Mozaffari-Kermani M, Sur-Kolay S, Raghunathan A, et al. Systematic Poisoning Attacks on and Defenses for Machine Learning in Healthcare[J]. *IEEE Journal of Biomedical and Health Informatics*, 2015, 19(6): 1893-1905.

[37] M. Jagielski, A. Oprea, B. Biggio, et al. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning[C]. *2018 IEEE Symposium on Security and Privacy (SP)*, 2018: 19-35.

[38] M. Kloft, P. Laskov P. Security analysis of online centroid anomaly detection[J]. *Journal of Machine Learning Research*, 2012, 13(12):3681-3724.

[39] B. Biggio, L. Didaci, G. Fumera, et al. Poisoning attacks to compromise face templates[C]. *2013 International Conference on Biometrics (ICB)*, 2013: 1-7.

[40] S. Mei, X. Zhu. Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners[C]. *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015:456-462.

[41] Y. Cao, A. F. Yu, A. Aday, et al. Efficient repair of polluted machine learning systems via causal unlearning[C]. *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, 2018:36-45.

[42] M. Jagielski, A. Oprea, B. Biggio, et al. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning[C]. *IEEE Symposium on Security and Privacy (IEEE S&P)*, 2018:78-86.

[43] B. I. Rubinstein, B. Nelson, L. Huang, et al. Antidote: understanding and defending against poisoning of anomaly detectors[C]. *Internet Measurement Conference (IMC)*, 2009:825-836.

[44] Mozaffari-Kermani M, Sur-Kolay S, Raghunathan A, et al. Systematic Poisoning Attacks on and Defenses for Machine Learning in Healthcare[J]. *IEEE Journal of Biomedical and Health Informatics*, 2015, 19(6): 1893-1905.

[45] J. Steinhardt, P. W. W. Koh, P. S. Liang. Certified defenses for data poisoning attack[C]. *Neural Information Processing Systems (NIPS)*, 2017:256-263.

[46] G. F. Cretu, A. Stavrou, M. E. Locasto, et al. Casting out demons: Sanitizing training data for anomaly sensors[C]. *IEEE Symposium on Security and Privacy (IEEE S&P)*, 2008:86-95.

[47] Liu K, Dolan-Gavitt B, Garg S. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks[M]. Research in Attacks, Intrusions, and Defenses. Cham: Springer International Publishing, 2018: 273-294.

[48] Shen S Q, Tople S, Saxena P. A Uror: Defending Against Poisoning Attacks in Collaborative Deep Learning Systems[C]. *The 32nd Annual Conference on Computer Security Applications*, 2016: 508-519.

[49] Y. Liu, S. Ma, Y. Aafer, et al. Trojaning attack on neural networks[C]. *Network and Distributed System Security Symposium 2017*. 2017: 1-15.

[50] X. Chen, C. Liu, B. Li, et al. Targeted backdoor attacks on deep learning systems using data poisoning[EB/OL]. arXiv preprint arXiv:1712.05526, 2017.

[51] O. Suciu, R. Marginean, Y. Kaya, et al. When Does Machine Learning FAIL? Generalized Transferability for Evasion and Poisoning Attacks[EB/OL]. arXiv preprint arXiv:1803.06975.

[52] K. He, X. Zhang, S. Ren, et al. Deep residual learning for image recognition[C]. *The IEEE conference on computer vision and pattern recognition*, 2016: 770-778.

[53] C. Zhu, W. R. Huang, H. Li, et al. Transferable Clean-Label Poisoning Attacks on Deep Neural Nets[C]. *International Conference on Machine Learning*, 2019: 7614-7623.

[54] Miao C L, Li Q, Xiao H P, et al. Towards Data Poisoning Attacks in Crowd Sensing Systems[C]. *The Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2018: 111-120.

[55] M. Alberti, V. Pondenkandath, M. Wursch, et al. Are You Tampering

With My Data?[C]. *The European Conference on Computer Vision (ECCV)*. 2018: 0-18.

[56] S. Alfeld, X. Zhu, P. Barford. Data Poisoning Attacks against Autoregressive Models[C]. *Thirtieth AAAI Conference on Artificial Intelligence, AAAI*, 2016: 1452-1458.

[57] Y. Zhang, Q. Yang. A survey on multi-task learning[EB/OL]. arXiv preprint arXiv:1707.08114, 2017.

[58] M. Zhao, B. An, Y. Yu, et al. Data Poisoning Attacks on Multi-Task Relationship Learning[C]. *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018: 2628-2635.

[59] Ma Y Z, Zhu X J, Hsu J. Data Poisoning Against Differentially-Private Learners: Attacks and Defenses[C]. *The Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019: 4732-4738.

[60] Z. Sun, P. Kairouz, A. T. Suresh, et al. Can You Really Backdoor Federated Learning?[C]. *The 2nd International Workshop on Federated Learning for Data Privacy and Confidentiality at NeurIPS*, 2019:895-899.

[61] C. Xie, K. Huang, P. Chen, et al. DBA: Distributed Backdoor Attacks against Federated Learning[C]. *International Conference on Learning Representations (ICLR)*, 2020:698-702.

[62] Uchida Y, Nagai Y, Sakazawa S, et al. Embedding Watermarks into Deep Neural Networks[C]. *The 2017 ACM on International Conference on Multimedia Retrieval*, 2017: 269-277.

[63] Nagai Y, Uchida Y, Sakazawa S, et al. Digital Watermarking for Deep Neural Networks[J]. *International Journal of Multimedia Information Retrieval*, 2018, 7(1): 3-16.

[64] Le Merrer E, Pérez P, Trédan G. Adversarial Frontier Stitching for Remote Neural Network Watermarking[J]. *Neural Computing and Applications*, 2020, 32(13): 9233-9244.

[65] H. Chen, D. Rohani, F. Koushanfar. DeepMarks: A Digital Fingerprinting Framework for Deep Neural Networks[EB/OL]. arXiv preprint arXiv:1804.03648, 2018.

[66] M. Zou, Y. Shi, C. Wang, et al. PoTrojan: powerful neural-level trojan designs in deep learning models[EB/OL]. arXiv preprint arXiv:1802.03043, 2018.

[67] Fang M H, Yang G L, Gong N Z, et al. Poisoning Attacks to Graph-Based Recommender Systems[C]. *The 34th Annual Computer Security Applications Conference*, 2018: 381-392.

[68] T. Takahashi. Indirect Adversarial Attacks via Poisoning Neighbors for Graph Convolutional Networks[C]. *2019 IEEE International Conference on Big Data (Big Data)*. 2019: 1395-1400.

[69] Y. Adi, C. Baum, M. Cisse, et al. Turning your weakness into a strength: Watermarking deep neural networks by backdooring[C]. *27th USENIX Security Symposium*, 2018: 1615-1631.

[70] Google Cloud Machine Learning Engine. https://cloud.google.com/ml-engine/.

[71] Azure Batch AI Training. https://batchaitraining. azure.com/.

[72] Deep Learning AMI Amazon Linux Version. https://docs.amazonaws.cn/dlami/latest/devguide/al.html.

[73] Chung S P, Mok A K. Allergy Attack Against Automatic Signature Generation[M]. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006: 61-80.

[74] A. Athalye, N. Carlini, D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples[EB/OL]. arXiv preprint arXiv:1802.00420, 2018.

[75] Y. Liu, Y. Xie, A. Srivastava. Neural Trojans[C]. *Computer Design (ICCD)*, 2017: 45-48.

[76] G. Wang, T. Wang, H. Zheng, et al. Machine: Practical Adversarial Detection of Malicious Crowdsourcing Workers[C]. *USENIX Security Symposium*, 2014: 239-254.

[77] Anwar S, Hwang K, Sung W. Structured Pruning of Deep Convolutional Neural Networks[J]. *ACM Journal on Emerging Technologies in Computing Systems*, 2017, 13(3): 1-18.

[78] S. Han, H. Mao, J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[EB/OL]. arXiv preprint arXiv:1510.00149, 2015.

[79] P. Molchanov, S. Tyree, T. Karras, et al. Pruning convolutional neural networks for resource efficient inference[EB/OL]. arXiv preprint arXiv:1611.06440, 2016.

[80] J. Yu, A. Lukefahr, D. Palframan, et al. Scalpel: Customizing dnn pruning to the underlying hardware parallelism[C]. *ACM SIGARCH Computer Architecture News*, 2017, 45(2): 548-560.

[81] F. Iandola, M. Moskewicz, K. Ashraf, et al. Firecaffe: near-linear acceleration of deep neural network training on compute clusters[C]. *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016: 2592-2600.

[82] H. Chen, C. Fu, J. Zhao, et al. DeepInspect: A black-box trojan detection and mitigation framework for deep neural networks[C]. *The 28th International Joint Conference on Artificial Intelligence*, 2019:125-135.

[83] D. Hitaj, V. Mancini. Have You Stolen My Model? Evasion Attacks Against Deep Neural Network Watermarking Techniques[EB/OL]. arXiv preprint arXiv:1809.00615, 2018.

[84] V. Chandola, A. Banerjee, V. Kumar. Anomaly detection: A survey[J]. *ACM computing surveys (CSUR)*, 2009, 41(3): 15-16.

陈晋音 于 2009 年在浙江工业大学控制理论与控制工程专业获得博士学位。现任浙江工业大学信息工程学院副教授。研究领域为人工智能安全，数据挖掘，智能计算。Email: chenjinyin@zjut.edu.cn

邹健飞 于 2019 年在浙江科技学院自动化专业获得学士学位。现在浙江工业大学控制工程专业攻读硕士学位。研究领域为人工智能安全。研究兴趣包括: 深度学习分析与测试。Email: zoujianfei163@foxmail.com

苏蒙蒙 于 2017 年在浙江工业大学获得学士学位。现在浙江工业大学控制工程专业攻读硕士学位。研究领域为智能计算、人工智能和工业安全。Email: sumengmeng@zjut.edu.cn

张龙源 于 2019 年在湖州师范学院电子信息工程专业获得学士学位。现在浙江工业大学控制工程专业攻读硕士学位。研究领域为人工智能安全。研究兴趣包括: 深度学习分析与测试。Email: zlyuan.mo@foxmail.com