深度学习模型版权保护技术研究综述

李珮玄, 黄 土, 罗书卿, 宋佳鑫, 刘功申

上海交通大学电子信息与电气工程学院 上海 中国 200240

摘要 深度学习模型在许多任务中取得出色的成绩, 也逐渐被广泛应用到众多领域。由于训练一个性能优越的深度神经网络成本高昂, 因此深度学习模型可以视作模型所有者的知识产权。然而深度学习模型设计之初并未考虑模型的安全问题, 在其快速发展的同时面临的安全问题也逐渐突显出来。随着模型训练云平台的部署与应用, 深度学习模型被盗取、恶意分发、转卖的威胁大大增加。由于深度学习模型有巨大的实用价值, 恶意攻击者非法窃取模型会严重侵犯模型所有者的权益, 保护深度学习模型版权迫在眉睫。针对这一问题, 近年来有很多关于保护深度学习模型版权的方案陆续被提出, 包括基于数字水印技术实现模型所有权验证以及基于水印或加密技术实现模型访问控制等。本文总结梳理了当前研究现状, 并探讨了未来可能的研究方向。文章首先介绍了深度学习模型水印、后门攻击的基本概念以及对模型水印的要求; 然后, 基于不同的分类指标, 从方案的实现功能、实现方式、实现时间、以及验证方式的不同, 对现有深度学习模型版权保护方案进行全面细致的总结与分类; 并且从检测攻击、逃逸攻击、去除攻击及欺诈攻击四个方面, 归纳总结了针对深度学习模型版权保护方案的攻击方法; 最后, 总结研究现状并对未来的关键研究方向进行展望。希望本文详细的梳理总结可以为该领域后续的研究提供有益的参考。

关键词 深度学习模型安全; 深度学习模型版权保护; 模型水印

中图法分类号 TP18 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2025.01.02

A Survey on Copyright Protection Technology of Deep Learning Model

LI Peixuan, HUANG Tu, LUO Shuqing, SONG Jiaxin, LIU Gongshen

School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Abstract Deep learning models have achieved excellent performance in many tasks, and have gradually been widely used in many fields. Since training a deep neural network with superior performance is expensive, a deep learning model can be regarded as the intellectual property of the model owner. However, the security issues of deep learning models were not considered at the beginning of design, and they have gradually emerged with the rapid development of deep learning. With the deployment and application of model training cloud platforms, the threat of deep learning models being stolen, maliciously distributed, and resold has greatly increased. Due to the huge value of deep learning models, malicious attackers illegally stealing models will seriously violate the rights and interests of model owners. So, it is urgent to protect the copyright of deep learning models. To solve this problem, many copyright protection technologies of deep learning model have been continuously proposed in recent years, including model ownership verification based on digital watermarking technology and model access control based on watermarking or encryption technology, but there is a lack of summary. This paper summarized the current researches and discusses the possible future research directions. This paper firstly introduced the basic concepts of deep learning model watermarking and backdoor attack, the requirements for model watermarking; and then, made a comprehensive and detailed summary and classification of the existing deep learning model copyright protection schemes based on different classification indicators from the differences of implementation functions, implementation methods, implementation time, and verification methods of different schemes; in addition, this paper summarized attack methods for copyright protection schemes of deep learning model from four aspects: detection attack, escape attack, removal attack and fraud attack; finally, the research status was summarized and the key research directions in the future were prospected. Hope the detailed summary of this paper can provide a useful reference for subsequent research in this field.

Key words deep learning model security; copyright protection of deep learning model; model watermarking

通讯作者: 刘功申, 教授, 博士生导师, Email: lgshen@sjtu.edu.cn。

本课题得到国家自然科学基金联合重点项目(No. U21B2020)和上海市科技计划项目(No. 22511104400)的资助。

收稿日期: 2022-12-17; 修改日期: 2023-03-21; 定稿日期: 2024-11-20

1 引言

随着深度学习技术的快速发展, 深度神经网络(Deep Neural Network, DNN)在计算机视觉[1]、语音识别[2]、自然语言处理[3]等很多领域取得了比传统机器学习模型更优越的性能。但是训练一个性能优越的深度神经网络并非易事, 需要大量训练和测试数据做支撑, 需要有强大的计算资源, 同时还要求操作者具备相关专业知识, 这一系列流程会耗费大量人力物力财力, 文献[4]指出训练一个 BERT 模型需要花费 2179.58～12570.62 美元, 而训练具有更多参数的模型需要更高的开销, 因此深度学习模型具有巨大价值, 可以视作模型所有者的知识产权。随着机器学习即服务(Machine Learning as a Service, MLaaS)平台的部署与应用, 具有强大计算资源的公司可以出售训练好的深度学习模型, 用户可以以较低的成本获得所需模型, 在便利用户的同时为公司创造了收益。但是如果存在恶意用户非法盗取、转卖、分发购买的模型就会严重侵犯模型所有者权益, 因此保护深度学习模型版权十分必要。

Uchida 等[5]首次以图像分类任务为例提出用数字水印技术保护深度学习模型版权, 之后逐渐有越来越多的研究围绕 DNN 版权保护展开。虽然目前大量研究仍然集中于用数字水印技术保护图像分类任务及对应模型, 但逐渐开始有关于其他任务领域及应用场景的探索, 保护模型版权的方法也不再局限于数字水印技术, 还引入了加密算法、利用 DNN 模型自身特征等方式保护版权; 此外, 从最初只能在怀疑模型被盗后验证所有权, 到逐渐开始探索如何实现深度学习模型的访问控制从而主动保护所有者权益。在模型版权保护技术日益发展进步的同时, 对这些方案的鲁棒性检验以及攻击方案也不断被提出。但是这一领域的总结性工作较少, 本文在充分调研相关工作的基础上做了更为全面、完善的梳理与总结, 对最近几年的工作做了更细致的介绍。

本文后续内容组织结构如下: 第 2 章介绍了深度学习模型水印和后门攻击的基本概念, 介绍了四种深度学习模型版权保护技术的分类指标; 第 3 章对现有的深度学习模型版权保护技术进行了详细的介绍及全面的总结; 第 4 章分类介绍了针对模型版权保护方案的攻击方法; 第 5 章对研究现状及未来关键研究方向进行了总结与展望。

2 基本概念及分类

由于现有的模型版权保护方法大多基于数字水印技术, 而且深度学习模型水印与多媒体水印有很多不同之处, 所以本节首先介绍了深度学习模型水印的概念以及对水印的要求。此外, 近年来针对深度学习模型的后门攻击技术不断发展, 这项技术除了可以对模型造成恶意威胁, 还可以用于模型版权的保护[6], 本节首先介绍后门攻击的基本概念, 在后文(3.1.2 和 3.2.2 节)具体介绍将后门技术用于版权保护的方案。本节最后根据不同的分类指标对深度学习模型版权保护方案做出分类。

2.1 深度学习模型水印

数字水印技术在不影响原有内容的情况下通过在数字媒介(例如图像、音频、视频等)中嵌入特殊消息达到版权保护或完整性验证等目的[7]。自从 2017 年 Uchida 等[5]提出用数字水印技术保护深度学习模型版权后, 越来越多的研究开始探索如何有效地在深度学习模型中嵌入水印保护所有权。水印技术主要利用载体的冗余性, 从而可以在不影响原内容的同时嵌入水印信息, 而深度学习模型恰好具有过参数化的特点, 这些冗余的参数可以用于直接或间接嵌入水印信息。但由于嵌入水印的深度学习模型可以被微调、重训练、蒸馏或迁移学习, 这些操作都可能破坏原有水印, 使得不能简单地利用多媒体水印技术直接修改模型参数嵌入水印, 而需要结合深度学习模型的特点, 通过修改损失函数等方式在模型中嵌入水印。Li 等[8]更为详细的比较了深度学习模型水印与多媒体水印的相似与不同。

在深度学习模型中嵌入水印应该满足以下要求, 同时这些指标也可以用于衡量不同水印方案的性能:

1) 保真性: 水印嵌入不应该影响模型在原任务上的性能, 嵌入水印的模型在原任务上的准确率应该与未嵌入水印模型相差不大。

2) 鲁棒性: 水印对各种去除攻击、欺诈攻击具有鲁棒性。

3) 隐蔽性: 水印应该具有隐蔽性, 不易被检测到。

4) 关联性: 水印信息与模型所有者之间具有关联性, 从而防止欺诈所有权攻击。

5) 可靠性: 水印验证结果应该具有可靠性, 假阳率、误报率低。

6) 水印容量: 在满足其他要求的同时模型允许嵌入的最大水印信息量。

7) 通用性: 水印方案应该普遍适用于不同的深度学习模型及数据集。

2.2 后门攻击

后门攻击通过某种方式在模型训练时埋下后门,

这个后门由攻击者预先设定的触发集激活。后门未被激活时, 被注入后门的模型与正常模型表现相同, 但当后门被激活后, 被攻击的模型就会输出与正常输出不同的异常响应, 以达到恶意目的[9], 例如文献[10]中使用后门攻击使模型对被标记的交通标志牌产生异常输出。

训练数据投毒[10-11]是最常用的一种植入后门的方式, 通过在训练数据中添加精心设计的触发器生成触发集, 用触发集与原训练数据一起训练模型就可以使模型对正常样本产生正常输出, 但对触发集产生特定的异常响应。图 1 展示了这类后门攻击的基本思路, 图中第二行图像右下角的白色方框是触发器, 通过在原始训练数据中添加触发器得到触发集, 并为其分配与正常标签不同的输出标签, 然后用干净的训练数据与触发集一起训练模型, 从而在模型中植入后门。除数据投毒外, 还可以通过修改模型权重[12]、迁移学习[13]等方式植入后门。

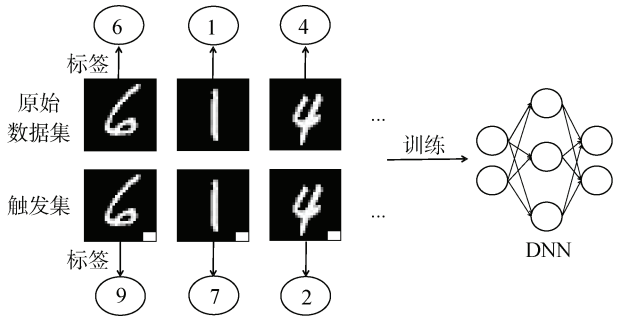


图 1　基于数据投毒的后门攻击示意图

Figure 1　Schematic diagram of backdoor attack based on data poisoning

2.3　深度学习模型版权保护方案分类指标

除了用数字水印技术保护深度学习模型版权外, 还有部分方案基于加密算法或者利用模型自身特征保护模型版权, 如图 2 所示根据方案的实现功能、实现方式、实现时间、以及验证方式, 可以将现有方案做进一步划分:

(1) 实现功能: 目前的深度学习模型版权保护方案实现的功能可以划分为两大类: 1)所有权验证; 2)访问控制。早期的 DNN 版权保护方案只能实现所有权验证, 通过嵌入水印或用模型自身的特点标识模型版权, 在怀疑模型被盗后通过提取水印或对比模型特征的方式验证模型所有权。这种方式只能事后验证, 无法提供主动保护, 攻击者在盗取模型后仍然可以正常使用模型。之后逐渐有可以实现主动访问控制的方案提出, 在这类方案中只有授权用户才能正常使用模型, 否则无法访问模型或者所访问的模型性能很差。

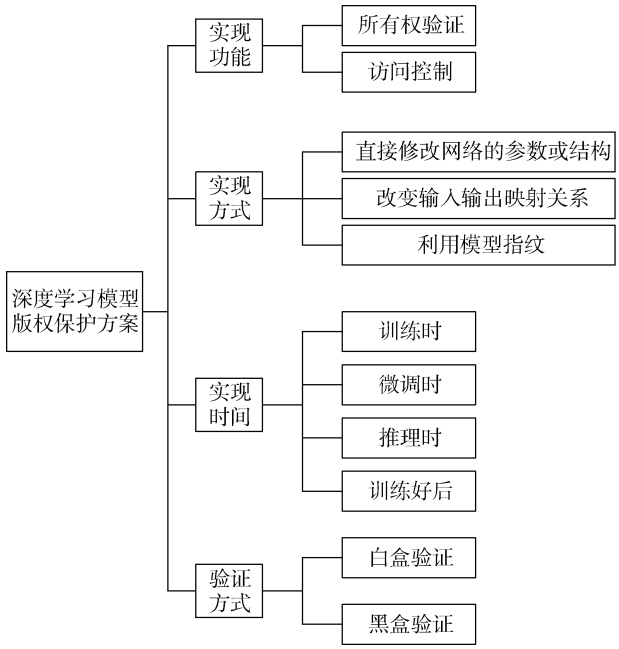


图 2　深度学习模型版权保护方案分类指标

Figure 2　Classification indexes of copyright protection scheme in deep learning model

(2) 实现方式: 根据 DNN 版权保护方案实现方式的不同可以分为三大类: 1)直接修改网络的参数或结构保护模型版权, 这类方案通过添加正则项约束在模型参数中嵌入水印, 或用水印签名直接给权重赋值, 或用加密算法直接修改权重参数等方式保护模型版权; 2)通过改变输入输出映射关系保护模型版权, 这类方案借鉴了后门技术, 通过构建特殊的输入输出对使被保护的模型对特定的输入(触发集)产生区别于正常模型的特殊输出响应, 从而标识模型实现所有权验证, 如果对所有的输入做了预处理使模型对未经处理的输入输出错误的结果则实现了访问控制, 或者在允许用户访问模型前先用特殊的数据集验证其身份从而实现访问控制; 3)利用模型自身的特点保护模型版权, 这类方案没有对模型做任何修改, 通过搜索模型自身独有的特征标识模型, 并利用这些特征保护模型版权。

(3) 实现时间: 根据水印嵌入、加密、或搜索模型自身特征时间的不同可将现有的 DNN 模型版权保护方案分为四类: 1)训练时, 有些方案需要在模型从头开始训练时就加入额外的操作以保护模型版权, 例如更改训练时的损失函数、加入辅助网络做对抗训练、增加网络输出维度以添加新的类标签、对所有输入做预处理等; 2)微调时, 例如在已有预训练模型的基础上用触发集微调模型嵌入水印, 一些在微

调时加入的方案也可以在训练时加入, 但有些方案需要用到预训练模型的信息只能在微调时加入, 例如利用预训练模型的对抗样本作为触发集[14]、在预训练模型权重参数的基础上嵌入水印[15]; 3)推理时, 这类方案没有修改模型的训练过程以及输入样本, 而是动态修改模型推理阶段的预测输出[16]; 4)训练好后, 这类方案没有改变模型的训练及推理过程, 通过修改训练好的模型参数嵌入水印, 或加密模型参数实现访问控制, 利用模型自身特征标识模型的方案也是在这一阶段搜索模型特有的特征。

(4) 验证方式: 根据验证模型所有权或验证用户访问权限时是否需要访问模型内部参数可以将 DNN 版权保护方案分为两类: 1)白盒验证, 需要访问模型内部参数, 通常对应直接修改模型参数或结构保护版权的实现方式, 因为直接在模型参数或结构中嵌入了水印, 或者对参数直接加密, 所以在提取水印或者解密模型时同样需要访问模型内部参数, 此类水印方案简称为“白盒水印”; 2)黑盒验证, 不需要访问模型内部参数, 对应于改变输入输出关系保护模型版权的实现方式, 在验证时只需要访问模型远程查询接口, 根据输入输出关系是否匹配即可验证, 此类水印方案简称“黑盒水印”。

3 深度学习模型版权保护技术

本节介绍现有的深度学习模型版权保护技术, 由于目前大多研究都以图像分类任务展开, 所以 3.1 节、3.2 节对针对图像分类任务的 DNN 版权保护方案根据实现功能和实现方式的不同做了详细的分类介绍, 3.1 节介绍了通过所有权验证方式保护模型版权的方案, 3.2 节介绍了通过访问控制方式保护模型版权的方案, 3.3 节介绍了新的应用场景, 包括其他任务及对应模型的版权保护方案以及其他验证需求。表 1 展示了深度学习模型版权保护技术的整体框架, 同时也是本节的分类结构。

表 1 深度学习模型版权保护技术整体框架

Table 1 Copyright protection technical framework of deep learning model

	图像分类任务		其他应用场景
	所有权验证	访问控制	
直接修改网络的参数或结构	(1) 无需添加辅助结构 (2) 添加额外网络嵌入水印	(1) 基于水印 (2) 基于加密	(1) 图像处理任务 (2) 生成对抗网络 (3) 图像字幕任务 (4) 文本处理任务 (5) 语音识别系统 (6) 图神经网络 (7) 多任务学习 (8) 联邦学习 (9) 强化学习
改变输入输出映射关系	(1) 整体式触发集 (2) 叠加式触发集 (3) 输出侧的研究	(1) 输入预处理 (2) 后门触发集	
其他	利用模型指纹标识所有权	—	(1) 完整性验证 (2) 所有权验证协议

(注: 表中“—”表示本单元格无内容)

3.1 通过所有权验证保护模型版权

本节介绍在DNN模型中嵌入包含所有者身份信息的水印标识模型的版权保护方案, 这是一种被动保护的方式, 不能阻止攻击者正常使用模型, 只能在怀疑模型被盗后进行事后验证。后文根据嵌入水印的方法分三大类介绍现有的研究: 1)直接修改网络参数或结构嵌入水印, 2)通过改变输入输出的映射关系嵌入水印; 3)利用模型自身的特点作为指纹标识模型。需要注意的是前两种方式都直接或间接地改变了模型参数, 前者借鉴了数字水印的方法直接将水印嵌入到参数中, 在验证时需要白盒访问模型参数提取水印; 后者利用神经网络的特性, 通过改变输入输出映射关系间接改变模型参数, 验证时通过黑盒方式访问模型, 根据模型对特定输入的响应判断所有权。第三种方式没有对模型做任何修改, 通过寻找模型独有特征表征模型。

3.1.1 直接修改网络的参数或结构嵌入水印

在这一类水印方案中大部分研究都是直接修改网络参数嵌入水印(后文简称“参数水印”), 但最近也有学者提出在模型结构中嵌入水印的方案[17]。根据嵌入水印时是否需要引入额外的神经网络, 将现有方案分为两类: 1)无需添加辅助网络: 算法只涉及到需要保护的单个网络, 通过在损失函数中添加正则项使所选参数嵌入水印[5,18-19], 或直接用水印信息修改部分参数值并在训练时不更新[20-21], 或用量化调制的方法嵌入水印[22]。2)添加额外网络嵌入水印:

为了扩大水印容量, 增强水印的鲁棒性和隐蔽性, 一些水印方案引入其他网络辅助主网络嵌入和提取水印[23-24], 或者用额外的神经网络作为水印嵌入主网络中[25]。表 2 对这一类方案进行了总结对比。

(1) 无需添加辅助结构

Uchida 等人[5]第一次提出用数字水印技术保护深度学习模型版权的方法, 定义了对模型水印的基本要求、水印嵌入方式、以及潜在的攻击方法。水印形式是多位二进制比特串, 随机生成一个满足标准正态分布的矩阵作为密钥, 水印嵌入位置是某个卷积层的权重参数。通过在损失函数中添加交叉熵损失, 使所选权重参数与密钥矩阵的乘积经过 sigmoid 函数后逼近比特签名, 由此在权重参数中嵌入水印。可以在从头训练网络的时候嵌入水印, 也可以通过微调或蒸馏的方式嵌入水印。验证所有权时需要访问模型参数才能验证。尽管这是一项开拓性的工作, 但这种方法存在很多缺点: 1)将水印嵌入权重参数, 导致模型参数统计分布发生变化, 极易被攻击者检测到, 不符合安全性要求[26]; 2)水印容量受到参数大小的限制[18]; 3)对覆写攻击鲁棒性差[5]; 4)验证时需要访问模型参数才能提取水印, 而被盗模型通常是远程部署的, 不会公开其内部参数, 使这种方法应用受限[18]; 5)水印和模型所有者没有相关性, 容易被攻击者诈称所有权。后续很多工作针对这些缺点做出了改进。

DeepSigns[18]将水印嵌入到激活层的概率密度函数中。同文献[5]一样, 水印形式仍然是多位二进制比特, 随机生成一个满足标准正态分布的矩阵作为投影矩阵(密钥), 只是水印嵌入的位置由原来的权重参数变为中间层高斯分布的均值, 添加正则项约束在概率密度函数中嵌入水印。为了实现远程验证, 定义了一组触发集作为水印。在概率密度函数尾区间中选择随机样本作为触发集, 并为其分配随机标签。用触发集微调预训练模型, 使模型对特定的输入产生预定的输出, 由此证明所有权。水印载体由静态内容(模型权重)到动态内容(不同层激活函数输出的概率密度分布)的变化增强了对覆写攻击的鲁棒性。同时本方案第一次结合了白盒和黑盒两种验证方式, 便于所有权验证。

DeepMarks[19]为每个用户分配了独特的水印作为指纹, 而且能防御共谋攻击。水印形式是多位二进制比特, 但每个用户有独特的码向量, 通过添加正则项约束在卷积层的权重参数中嵌入水印。可以用正交矩阵或者平衡不完全区组设计(Balanced Incomplete Block Design, BIBD)生成独特的码向量, 但正交指纹由于具有独立性很容易遭受共谋攻击, 由 BIBD 码本生成的指纹可以更大程度地防御共谋攻击。

Tartaglione 等[20]利用模型参数的冗余性, 在模型开始训练前直接用水印信息初始化部分权重参数, 并使这些参数在训练中不更新(本方案中水印是 N 位[0,1]实数值序列)。为了增强对微调攻击的鲁棒性, 在损失函数中添加正则项约束, 惩罚对嵌入水印参数的改变。

Feng 等[21]指出添加正则项嵌入水印的方式可能影响模型正常训练, 并提出一种无需额外添加正则项约束的水印方案。在模型训练好后选择部分权重参数嵌入水印, 然后微调模型补偿嵌入水印对模型准确率的影响, 微调时不更新包含水印信息的参数。水印形式为{1, –1}序列, 并用扩频调制技术增强水印对剪枝攻击的鲁棒性。用伪随机数选取原模型的部分参数, 对所选参数做正交变换后用二值化方法使参数的符号携带水印信息。

为了减小并衡量水印嵌入导致的权重变化, DeepWatermark[22]基于量化水印方法—DM-QIM 在全连接层嵌入水印。为了增加隐蔽性以及抗噪性, 本方案在随机选择的权重参数的频域分量中嵌入水印。水印形式是多位二进制比特串, 用密钥从全连接层中选择部分权重参数, 用 DM-QIM 方法量化这些参数的频域分量后做逆 DCT 变换更新参数的时域值。使用 DM-QIM 方法引起的权重改变会随着训练过程自动收敛, 因此不需要添加额外的正则项约束。嵌入水印引起的权重变化可由嵌入的信息量和量化步长定量表示。

由于嵌入到模型参数中的水印容易被攻击者去除或检测到, Lou 等[17]提出将水印嵌入到 DNN 模型的结构框架中。基于神经网络搜索算法, 根据用户密钥产生为每个用户生成特有的网络结构。验证所有权时, 利用缓存侧信道通过 API 接口监视目标模型的推理过程得到模型的结构信息, 与原模型的结构进行对比判断所有权。

(2) 添加额外网络嵌入水印

Wang 等[23]借助一个额外的神经网络在需要保护的主网络的部分权重参数中嵌入水印, 水印形式是多位二进制比特串。附加神经网络的输入是从主网络中选择的收敛的权重参数, 损失函数是网络输出与水印签名的二进制交叉熵损失。附加的神经网络与主网络一起训练, 并通过反向传播更新主网络中被选中的权重。嵌入水印后, 附加的神经网络不分发, 只用于在验证阶段提取水印。

Wang 等[24]同样引入一个额外的神经网络(提取

器)嵌入和提取水印, 同时为了提高水印的隐蔽性, 借鉴对抗训练的思想引入一个鉴别器网络。用神经网络替代[5]中的嵌入矩阵和线性函数实现水印的嵌入和提取, 扩大了水印容量, 提高了水印对覆写攻击的鲁棒性, 同时由于神经网络出色的表示能力, 水印形式可以由比特串扩展为图像。鉴别器与主网络做对抗训练, 使嵌入水印后参数的统计分布几乎不变, 水印不易被检测到, 提高了隐蔽性。

Lv 等[25]提出用一个轻量神经网络(HufuNet)作为水印, 在公开的训练集和测试集上训练好 HufuNet, 然后将其分为两部分, 一部分嵌入需要保护的主网络的结构中, 另一部分保留用于验证所有权。用密钥计算水印嵌入的位置, 嵌入水印的参数在主网络训练时被冻结不更新。验证所有权时, 从主网络中提取出嵌入部分, 与保留部分合并得到完整的 HufuNet, 在相应的测试集上测试 HufuNet 的准确率。

表 2　通过直接修改网络的参数或结构嵌入水印实现所有权验证的 DNN 版权保护方案

Table 2　Directly modify network parameters or structure to embed watermarks to realize ownership verification and DNN copyright protection

类型	方案	方案特点	实现时间	验证方式	鲁棒性测试
无需添加辅助结构	Uchida 等[5]	用投影矩阵作为密钥, 在损失函数中添加正则项使水印嵌入到权重参数中	训练/微调时	白盒	微调、剪枝
	DeepSigns [18]	用投影矩阵作为密钥, 添加正则项约束使水印嵌入到激活函数输出的概率密度函数中	训练/微调时	白盒/黑盒	微调、剪枝、覆写攻击
	DeepMarks [19]	添加正则项在权重参数中嵌入水印, 基于抗合谋编码给每个用户分配独特水印	训练/微调时	白盒	微调、剪枝、共谋攻击
	Tartaglione 等[20]	用实值水印序列初始化部分权重参数, 并在训练时保持不变, 添加正则项约束提高对微调的鲁棒性	训练时		微调
	Feng 等[21]	模型训练好后在所选权重参数中嵌入水印, 微调其他参数补偿精度损失	训练好后(微调前)	白盒	微调、剪枝、覆写攻击
	DeepWatermark[22]	用 DM-QIM 量化方法在权重参数的频域分量中嵌入水印	训练时		剪枝、覆写攻击
添加额外网络	Lou 等[17]	利用神经网络搜索算法将水印嵌入网络结构中	训练时		微调、剪枝
	Wang 等[23]	借助附加的神经网络在主网络的部分权重参数中嵌入水印	训练时		微调、剪枝
	Wang 等[24]	用一个神经网络嵌入提取水印, 用一个鉴别器网络与主网络做对抗训练提高水印的隐蔽性	训练时	白盒	微调、剪枝、覆写攻击、属性推理攻击
	Lv 等[25]	将额外的神经网络的一部分作为水印嵌入主网络中, 另一半保留用于所有权验证	训练时		微调、剪枝、卷积核裁剪/扩充、自适应攻击

3.1.2　改变输入输出映射关系嵌入水印

由于将水印直接嵌入到网络参数或结构的水印方案在验证时需要白盒访问模型提取水印, 而被盗取的模型通常部署在远端, 且不会公开内部参数, 所以需要白盒验证的水印方案应用受限, 因此目前的研究更多聚焦于可以黑盒验证的水印方案。黑盒验证方式只能访问模型的 API 接口, 所以只能通过改变模型对输入样本的输出响应标识模型。这组具有特殊输出响应的样本称为“触发集”[6]。这种水印算法相当于在模型中添加后门, 本文用“后门水印”简称这类水印。

基于后门的水印算法设计时应该重点考虑如何选取触发集及其对应标签, 后文根据触发集及标签生成方式的不同将现有算法分为三类: 1)整体式触发集; 2)叠加式触发集; 3)输出侧的研究。前两类算法重点关注输入侧触发集的生成, 后一类研究更多关注输出侧标签的分配。前两类触发集的主要区别在于是将特定的图像(例如对抗样本[14,27], 随机抽象图像[6], 稀疏区域的样本[18]等)直接作为触发集, 还是在特定图像上叠加触发器(例如有标识的 logo[28], 由水印签名生成的图案[29-30], 或随机噪声[28]等)形成触发集。整体式触发集早期的设计[6,14,18,27]缺少与模型所有者的关联很容易被攻击者诈称所有权, 但之后提出的一些方案[31-32]解决了这一问题; 而叠加式触发集对文献[15,33-35]中提出的后门攻击鲁棒性较差。目前更多的研究聚焦于在触发集中嵌入所有者身份信息的同时提高鲁棒性和隐蔽性。表 3 比较了通过改变输入输出映射关系嵌入水印的 DNN 版权保护方案。

(1) 整体式触发集

Merrer 等[14]第一次提出可以远程验证模型所有权的零比特黑盒水印方案。利用靠近模型决策边界的对抗样本微调模型轻微改变决策边界, 这组对抗样本就是模型所有者的密钥(触发集)。验证时, 输入这组对抗样本, 判断模型的输出是否与预设相同。但

表 3 通过改变输入输出映射关系嵌入水印实现所有权验证的 DNN 版权保护方案

Table 3 Change the mapping relationship between inputs and outputs to embed watermarks to realize ownership verification and DNN copyright protection

类型	方案	方案特点	实现时间	验证方式	鲁棒性测试
整体式触发集	Merrer 等[14]	用对抗样本作为触发集微调模型改变决策边界	微调时	黑盒	剪枝、奇异值分解、覆写攻击
	Adi 等[6]	用随机抽象图案作为触发集	训练/微调时		微调、覆写攻击、迁移学习
	Zhang 等[28]	选择与原任务无关的其他数据集作为触发集	训练/微调时		微调、剪枝
	DeepSigns [18]	选择特征分布在样本特征空间稀疏区域的样本作为触发集	微调时	黑盒/白盒	微调、剪枝、覆写攻击
	BlackMarks [27]	用目标对抗攻击结合编码方案嵌入多比特水印	微调时		微调、剪枝、覆写攻击
	Jia 等[36]	用 SNNL 增加触发集与训练数据集的耦合程度	训练时		剪枝-微调、迁移学习、模型窃取攻击
	Namba 等[15]	选择部分原训练样本改变其输出标签作为触发集	微调时		剪枝、查询修改攻击
	April Pyone 等[31]	用密钥对所有训练数据做像素变换, 不需要特定的触发集	训练时	黑盒	微调、剪枝、覆写攻击、歧义攻击
	Zhu 等[32]	用单向 hash 链构建触发集及对应标签	训练时		微调、重训练、伪造攻击
	Li 等[39]	用单向 hash 函数生成触发集, 用无数据蒸馏方式产生锚点, 用锚点微调和对抗微调的方式产生对微调攻击鲁棒的后触发集	微调时		微调、对抗微调、剪枝
叠加式触发集	Zhang 等[28]	在训练数据上添加特定内容或随机噪声生成触发集	训练/微调时	黑盒	微调、剪枝
	Li 等[37]	为了提高触发集的不可见性用自编码器生成触发集, 并引入鉴别器做对抗训练	训练时		微调
	Li 等[40]	在频域嵌入水印形成触发集, 并分配新的标签	训练/微调时		剪枝、欺诈攻击
	Guo 等[29]	在训练样本上叠加水印签名生成触发集	微调时		欺诈攻击
	Li 等[30]	采用空嵌入防止攻击者嵌入伪造水印, 用实嵌入降低假阳率	训练时		微调、剪枝、覆写攻击、歧义攻击、Neural Cleanse
	Guo 等[41]	用差分进化算法确定触发器位置	训练/微调时		微调、低假阳率
	Li 等[42]	用影子自编码器近似攻击者的自编码器, 生成具有穿透性质的触发集	训练时		自编码器过滤水印(查询修改攻击)
输出侧的研究	Zhong 等[43]	给触发集样本分配额外添加的类别标签降低假阳率	训练时	黑盒	微调、剪枝
	DAWN[16]	在模型推理阶段动态改变对输入样本的预测结果	推理时		模型提取攻击

是这种水印方案的性能严重依赖于对抗样本的选取, 而且对抗训练常用于提升模型的鲁棒性, 因此会造成较高的假阳率[15]。

Adi 等[6]首次提出用后门技术保护深度学习模型版权, 采用零比特水印验证方案, 不需要访问模型内部参数, 通过 API 接口判断水印是否存在从而验证所有权。随机选择一些抽象图片作为后门触发集, 并为它们分配特定标签。触发集与原数据集一起训练模型, 或用触发集微调预训练模型, 使模型对特定的输入产生预定的输出, 这种特定的输入输出映射关系就是所有者的签名。此外还提出用加密方案增加水印与所有者之间的关联。

为了扩大水印空间防止攻击者逆向恢复水印, Zhang 等[28]提出三种触发集生成方式: 1)在训练数据上添加特定内容; 2)在训练数据上添加噪声; 3)选择与原任务无关的其他数据集作为触发集。其中前两种在已有数据集上添加触发器形成触发集, 属于叠加式触发集; 第三种将无关数据集直接用作触发集, 属于整体式触发集, 其效果类似文献[6]中用抽象图片作为触发集。

Rouhani 等[18]指出 Adi 等[6]提出的水印方案会造成较高的假阳率, 为解决这一问题, 他们选择特征分布在训练数据特征空间稀疏区域的样本作为触发集, 为每个样本随机分配标签。

BlackMarks[27]是一种多比特验证的黑盒水印方案, 用自动编码器将模型的所有输出标签聚类为两类, 分别用二进制比特“0”, “1”表示。水印形式是多位二进制比特串, 利用目标对抗攻击产生与所有者签名对应的输入输出对作为触发集, 用触发集微调预训练模型嵌入水印。为了减小对抗样本的迁移性, 用多个未嵌入水印的模型辅助筛选触发集形成最终的水印密钥。验证时, 在查询接口输入触发集,

将输出结果解码为二进制比特, 计算解码结果与所有者签名的误比特率判断所有权。

Jia 等[36]用实验展示了如果触发集分布与训练数据相互独立, 模型会大致将参数分为两部分分别学习这两个任务, 使水印很容易在不影响原任务准确率的情况下被去除。他们提出在损失函数中添加软最近邻损失(soft nearest neighbor loss, SNNL)将触发集和训练数据耦合在一起, 使模型用同样的参数集合学习水印任务和原任务, 因此对水印的去除操作就会严重影响原任务的性能, 从而提升了水印的鲁棒性。

Namba 等[15]指出在原数据集上添加触发器容易被检测到然后被攻击, 于是提出不改变输入图像, 选择部分原训练样本改变其输出标签作为触发集。用触发集微调预训练模型嵌入水印。为了提高水印对微调、剪枝等攻击的鲁棒性, 在用触发集微调模型时对绝对值大的权重做指数加权, 使这部分参数显著影响模型性能。但是这样的触发集与模型所有者之间没有关联[37], 而且对权重的改变同样会被攻击者检测到[34]。

类似文献[30]中提到的空嵌入思想(在叠加式触发集中介绍), April Pyone 等[31]也提出防止攻击者嵌入伪造水印的方法, 但是本方案不需要预先定义触发集。用密钥对所有训练样本做块级像素变换, 但不改变原有的输出标签, 用变换后的样本与原训练样本一起训练模型。验证时, 对测试集的所有样本做同样的图像变换, 判断模型对变换样本的预测输出是否与原样本一致。

为了增加触发集与所有者之间的关联, 防止攻击者伪造触发集诈称所有权, Zhu 等[32]提出用不可逆的单向 hash 链构建触发集及对应标签。在验证时, 向可信第三方提供初始图像和密钥, 用相同的算法计算得到触发集, 判断模型的预测结果是否与对应的标签一致。但是这种方法生成的触发集类似噪声图像, 与原有训练样本有明显的差别, 容易遭到类似文献[38]提出的查询修改攻击。

Li 等[39]指出在现实场景中水印嵌入者未必可以访问训练数据, 并提出了无需数据先验知识的后门水印算法, 用无数据蒸馏方法产生的锚点替代训练数据集, 用于在注入后门时保持模型性能。为防止验证时公开的触发集泄露其他隐私信息, 利用单向 hash 函数逆向生成具有隐私保护性质的触发集, 并用锚点微调及对抗微调的方式产生对微调攻击鲁棒的后触发集。

(2)叠加式触发集

Li 等[37]指出之前的方案生成的触发集与原训练数据差别太大, 容易被攻击者检测到并去除, 而且因为水印缺少与模型所有者的关联, 容易被诈称所有权。为了增加水印与所有者的关联性, 用特有的 logo 作为水印图案; 为了增加水印的隐蔽性, 用自动编码器将水印嵌入到输入样本中, 同时引入一个鉴别器与编码器做对抗训练, 使触发集分布接近原样本。

为了提高水印的隐蔽性和鲁棒性, Li 等[40]用密钥在频域嵌入水印生成触发集。

为了增加触发集与模型所有者的关联, Guo 等[29]将水印签名嵌入触发集中。用 hash 函数加密可以体现身份信息的消息得到所有者签名, 将签名做伪随机置换后得到的水印图案作为触发器叠加到图像的像素上形成触发集, 用伪随机生成器为触发集分配标签。不可逆的 hash 函数可以防止攻击者逆向生成假的触发集及对应的签名诈称所有权。触发集携带了水印签名且不可伪造, 实现了多比特嵌入。

为防止攻击者嵌入伪造的水印诈称所有权, Li 等[30]同时采用两种方式生成触发集: 1)空嵌入: 在输入图像中添加触发器但不改变输出标签; 2)实嵌入: 在样本中添加触发器的同时改变输出标签。这两种方式产生的触发集与训练数据一起从头开始训练模型。用空嵌入的方式使水印显著影响模型准确率, 伪造水印会与原来的水印产生冲突导致模型准确率严重下降, 同时用实嵌入降低假阳率。

Guo 等[41]指出如果触发器扰动太微弱容易被微调攻击等去除, 而触发器太明显容易造成较高的假阳率, 并提出利用差分进化算法计算触发器嵌入位置, 在保证嵌入水印的模型保真度与鲁棒性的同时, 最大化未嵌入水印的模型对触发集样本正确分类的概率, 从而降低了假阳率。

叠加式触发集中的水印很容易被自编码器过滤, 针对这一问题 Li 等[42]提出可以穿透攻击者自编码器的后门触发集生成方法, 用一系列影子自编码器模型近似攻击者的自编码器, 通过精心设计损失函数生成具有穿透性质的触发集。同时, 在触发集中嵌入所有者的身份信息, 防止欺诈攻击。

(3) 输出侧的研究

Zhong 等人[43]指出给触发集分配训练任务原有的标签会改变模型的决策边界, 影响模型保真度和鲁棒性, 也容易产生较高的假阳率。他们提出给触发集样本分配额外添加的类别标签, 在保证模型高准确率和水印检测率的同时实现了零假阳率。文献[40]和[44]都借鉴这一思想为触发集分配了新的标签。但是这种水印容易被攻击者检测到并逃逸验证或去除水印。

之前介绍的水印方案都是在训练或微调时修改模型对触发集的预测结果嵌入水印的, Szyller 等[16]提出在模型推理阶段动态改变对输入样本的预测结果, 可以有效防止模型提取攻击。在查询接口前添加了一个 DAWN 模块, 使其对特定的输入输出不正确的预测结果, 攻击者用这些样本训练替代模型时就会嵌入水印。本方案没有对训练样本做任何修改, 只改变了模型推理时返回的预测结果。用 SHA-256 对投影到低维的输入图像做计算, 将得到的 hash 值分为两部分, 分别决定是否改变这个输入对应的输出, 以及具体的预测输出。

3.1.3 利用模型指纹标识所有权

除了人为向 DNN 模型添加水印标识模型所有权外, 还可以利用模型自身独有的特征, 如对抗样本, 作为指纹标识模型。本节简要介绍几种现有方法, 并在表 4 中做简要对比。

表 4 利用模型指纹实现所有权验证的 DNN 版权保护方案

Table 4 Use model fingerprint to realize ownership verification and DNN copyright protection

类型	方案	方案特点	实现时间	验证方式	鲁棒性测试
用对抗样本作为模型指纹	Cao 等[45]	用靠近模型决策边界的对抗样本作为模型指纹	训练好后	黑盒	微调、剪枝
	Zhao 等[46]	增强对抗样本在相同或相似模型间的迁移性并将其作为模型指纹			微调、剪枝
	Lukas 等[47]	用生成的可赠予样本作为模型指纹			微调、剪枝、模型提取攻击

Cao 等[45]指出添加水印的方法会损失模型性能, 而且需要微调或重训练模型才能嵌入水印。他们提出利用模型本身的特点—不同的分类器有不同的决策边界, 利用靠近模型决策边界的对抗样本以及对应的标签作为指纹标识模型。验证时, 输入这组指纹数据, 判断模型的预测结果是否与输出标签匹配。

Zhao 等[46]提出一种对抗样本生成方式, 使对抗样本在相同或相似模型间的迁移性较高, 而在无关模型间的迁移性较弱, 判断这组精心设计的对抗样本在可疑模型上的攻击成功率判断模型所有权。

Lukas 等[47]指出 Cao 等[45]的指纹不能防御模型提取攻击, 对此提出一种只转移到替代模型(提取攻击得到的非法模型)的“可赠予对抗样本”(可迁移对抗样本的一种)生成方法。用生成的可赠予样本作为模型指纹, 查询可疑模型对它的预测输出是否与原模型输出相同判断模型所有权。

3.2 通过访问控制保护模型版权

本节介绍通过访问控制使未授权用户无法正常使用模型, 从而主动保护模型版权的方案, 类似 3.1 节分两类介绍现有研究: 1)直接修改网络的参数或结构实现访问控制, 包括基于水印的方案[48-50], 以及基于加密的方案[51-52]; 2)改变输入输出映射关系实现访问控制, 包括对所有输入做预处理[53-54], 以及通过合理设计触发集实现访问控制[44,55]。表 5 展示了可以实现访问控制的 DNN 版权保护方案。

3.2.1 直接修改网络的参数或结构实现访问控制

(1) 基于水印

Fan 等[48-49]提出在卷积层后添加一个保护层(称为 Passport 层)实现访问控制。Passport 层的尺度因子和偏移量根据 Passport 计算得到, 因此模型的推理性能取决于是否提供了正确的 Passport。Passport 有三种形式: 服从[–1,1]均匀分布的随机图案、输入一张固定图像后得到的特征映射、或者从多张图像的特征映射中随机选择一个作为该层的 Passport。因为 Passport 直接影响了中间层的输入输出映射关系, 伪造 Passport 会导致模型性能严重下降。此外, 在损失函数中添加了一个正则项约束, 使尺度因子的符号携带所有者的签名, 以防止内部成员攻击。本方案提高了水印对歧义攻击的鲁棒性, 但是显著增加了训练时间, 不分发 Passport 层所需的训练时间是正常训练的两倍以上。

Zhang 等人[50]指出 Fan 等[48]的方案采用多任务学习时需要替换原来的归一化层会影响模型性能, 并提出了不需要改变模型结构的基于 Passport 的 DNN 版权保护方案。用两个分支独立计算添加 Passport 层和未添加 Passport 层的归一化值, 使方案对不同的归一化方法都适用。

(2) 基于加密

Lin 等[51]基于混沌映射理论对模型参数加密实现访问控制。通过交换卷积层或全连接层的权重参数实现加密, 将权重置换后的模型分发给用户, 只有拥有密钥的授权用户才能使用模型。因为只对权重做了位置交换而没有改变参数值, 同时不需要用触发集进行微调、重训练, 所以模型保真度高且方案具有较高的隐蔽性。采用快速加解密算法产生的时间开销也很小。

Xue 等人[52]通过添加对抗扰动对参数加密实现访问控制。正常训练好模型后选出对模型性能有显

表 5 可实现访问控制的 DNN 版权保护方案

Table 5 DNN copyright protection schemes that can realize access control

类型		方案	方案特点	实现时间	验证方式	鲁棒性测试
直接修改网络的参数或结构实现访问控制	基于水印	Fan 等[48-49], Zhang 等[50]	在神经网络中添加 Passport 层, 模型的性能取决于是否提供正确的 Passport	训练时	白盒/黑盒	微调、剪枝、歧义攻击
	基于加密	Lin 等[51]	基于混沌映射理论对模型参数加密实现访问控制	训练好后	白盒	暴力破解、密钥泄露、侧信道攻击
		Xue 等[52]	添加对抗扰动对参数加密实现访问控制			微调、剪枝、自适应攻击
改变输入输出映射关系实现访问控制	输入预处理	Chen 等[53]	用转换模块对所有输入样本添加对抗扰动后再输入 DNN 模型	训练时	黑盒	微调
		April Pyone 等[54]	用密钥对所有样本做块级像素变换, 用变换后的样本训练模型			欺诈攻击
	后门触发集	Xue 等[55]	设计满足不同置信度要求的触发集实现访问控制、版权验证和用户指纹管理	微调时		微调、剪枝
		Sun 等[44]	用数字隐写技术结合后门触发集实现访问控制和用户管理	训练时		微调、剪枝、查询修改攻击

著影响的少量参数, 向其添加对抗扰动使模型的准确率严重下降。添加扰动的参数的位置和增加的扰动值是密钥, 解密时先找到加密参数的位置, 然后减去添加的扰动值。本方法实现了主动访问控制, 且不需要改变模型的训练过程, 时间开销也较小。

3.2.2 改变输入输出映射关系实现访问控制

(1) 输入预处理

Chen 等[53]设计了一个转换模块对输入样本添加对抗扰动后再输入 DNN 模型。他们提出三种转换模块的设计方式: 1)固定模式: 提前生成一个固定的扰动矩阵; 2)可学习模式: 通过学习找到最优通用扰动矩阵; 3)生成器模式: 用神经网络为每个输入生成不同的扰动矩阵。在 DNN 的损失函数中添加正则项约束, 使输入预处理的样本具有高准确率, 输入未处理的样本准确率很低, 且变换前后的样本相差不大。用经过预处理的样本训练模型, 使未授权用户不能使用模型。

April Pyone 等[54]也对模型的输入做了预处理。在训练模型前, 用密钥对所有训练样本做块级像素变换, 用变换后的样本训练模型。测试时, 先对测试样本做相同的置换再输入模型进行预测。如果没有正确的密钥对输入做预处理则无法使用模型。

(2) 后门触发集

Xue 等[55]基于后门技术实现了对模型的访问控制、版权验证和用户指纹管理, 攻击者无法用伪造的指纹访问模型。模型所有者设计 K 个触发器, 并为它们分配相同的标签。触发器的设计需要满足置信度要求: 1)模型对嵌入所有 K 个触发器的样本的输出具有高置信度; 2)对嵌入 k 个触发器的样本输出具有中等置信度; 3)对嵌入单个触发器的样本输出与未嵌入触发器的样本输出相同。选择部分训练样本在其中分别嵌入单个触发器形成 K 组触发集, 用触发集微调预训练模型嵌入水印。模型所有者在每个正常样本中注入所有 K 个触发器形成所有者指纹, 并随机选择 k 个触发器分别嵌入多个正常样本中形成满足条件的用户特有指纹。验证用户身份时, 判断用户提供的指纹是否满足中等置信度要求; 验证模型所有权时, 判断所有者提供的指纹是否满足高置信度要求。

Sun 等[44]为用户分配了特有的指纹图像, 在用户使用模型前进行认证和访问控制。为了防御 Namba 等[15]提出的查询修改攻击, 本方案选择与训练任务无关的数据集作为触发集, 并分配新的类标签。用图像隐写技术在触发集的最低比特位嵌入用户特有签名作为用户指纹。用水印密钥样本与训练样本一起训练模型。验证模型所有权时, 用触发集查询模型输出是否满足阈值。验证用户身份时, 从用户提供的指纹图像中提取最低比特位签名进行验证, 同时将触发集输入模型判断输出标签是否正确。

3.3 新的应用场景

现有的大多研究都关注于图像分类任务以及相应模型的版权保护方案, 目前也逐渐有学者开始探索其他任务领域以及其他不同的验证需求。本节分别介绍了: 1)其他任务及模型的版权保护方案; 2)其他验证需求, 包括用可逆水印实现完整性验证[56]及如何设计商用化验证协议[57]。表 6 对除图像分类任务外的其他任务及场景下的 DNN 版权保护方案做出总结对比。

表 6 DNN 版权保护技术在其他任务及场景中的应用

Table 6 Applications of DNN copyright protection technology in other tasks and scenarios

任务类型	方案	实现功能	方案特点	实现时间	验证方式	鲁棒性测试
图像处理任务	Zhang 等[58-59]	所有权验证	在所有输出图像中嵌入空域不可见水印(logo 图案)	训练时/训练好后	黑盒	模型提取攻击、覆写攻击
	Wu 等[60]			训练时	黑盒	图像裁剪、加噪
生成对抗网络	Ong 等[61]	所有权验证	在尺度因子的符号中嵌入签名/嵌入后门水印	训练时	白盒/黑盒	微调、覆写攻击、歧义攻击
图像字幕任务	Lim 等[62]	访问控制+所有权验证	用密钥加密 LSTM 单元的输入输出/在隐层状态的符号中嵌入签名/嵌入后门水印	训练时	白盒/黑盒	微调、剪枝、歧义攻击
文本处理任务	Yadollahi 等[63]	所有权验证	交换不同文档内的单词生成触发集, 并交换对应标签	训练时	黑盒	剪枝、暴力破解
语音识别模型	Chen 等[64]	所有权验证	在权重参数的频域分量中嵌入水印	训练好后	白盒	微调、剪枝、迁移学习
图神经网络	Zhao 等[65]	所有权验证	用随机图作为触发集, 在节点标签中嵌入水印签名	训练时	黑盒	微调、剪枝
多任务学习	Li 等[66]	所有权验证	基于多任务学习方法, 将水印嵌入任务作为一项额外的学习任务而不影响模型原本任务	训练/微调时	白盒	水印检测、覆写攻击、剪枝-微调、剪枝
联邦学习	Tekgul 等[67]	所有权验证	在噪声背景上为不同的类叠加不同的特定图案形成触发集	训练时	黑盒	微调、加剪枝、逆向工程
	Fan 等[68]		用正则项约束嵌入比特签名/嵌入后门水印	训练时	白盒/黑盒	微调、剪枝、差分隐私
	Li 等[69]		基于 Merkle 树结构提出水印方案框架, 用自编码器基于用户密钥生成不同分布的触发集	训练时	白盒/黑盒	去除攻击
强化学习	Behzadan 等[70]	所有权验证	在水印环境下对连续事件输出特殊循环状态序列	训练时	黑盒	N/A
可逆水印	Guan 等[56]	所有权验证+完整性验证	用直方图平移编码方式在主序列中嵌入水印	训练好后	白盒	具有脆弱性

(注: 表中“N/A”表示文献中没有具体讨论)

3.3.1 其他任务版权保护方案

(1) 图像处理任务

针对输入输出都是图像的图像处理网络, 可以在模型输出的所有图像中嵌入图像形式的水印, 这样即可以标识模型, 也可以保护生成的图像。Zhang 等[58]提出在模型的所有输出图像中嵌入空域不可见水印保护模型版权, 防御模型提取攻击。引入四个独立于原任务的神经网络在训练好的图像处理模型输出的图像中嵌入水印, 一个网络用于嵌入水印, 一个网络用于提取水印, 一个鉴别器与嵌入网络做对抗训练提高水印的隐蔽性, 此外, 模拟攻击者搭建一个代理模型, 用它的输出提高提取网络的水印提取能力。通过黑盒方式验证所有权, 将可疑模型输出的图像输入提取网络, 根据是否提取出水印判断所有权。在文献[59]中补充提出嵌入多种水印图像以及在原模型训练时添加水印的方案, 并具体化了水印验证方案。

Wu 等[60]也提出可以保护输出形式为图像的 DNN 模型的水印方案, 通过将需要保护的模型和带密钥的水印提取网络一起训练, 在模型的所有输出中嵌入水印, 可以验证模型所有权以及判断某幅图像是否由特定网络生成。

(2) 生成对抗网络

同图像处理网络一样, 生成对抗网络的输出也是图像, 因此可以生成包含水印图案的输出作为标识。但生成对抗网络的输入根据任务的不同, 可能是噪声向量, 也可能是图像, 对此, Ong 等[61]提出一个对不同生成对抗网络变体通用的损失函数用于嵌入水印, 当输入触发集时, 生成包含特定触发器图案的图像作为标识, 从而标记生成对抗网络。此外, 借

鉴了文献[48]在归一化层尺度因子的符号中嵌入签名, 可以采用白盒或黑盒验证方式。

(3) 图像字幕任务

图像字幕任务根据输入的图像生成描述图像内容的文本。Lim 等[62]提出可以保护图像字幕模型(LSTM 模型)的水印方案。给正常样本添加触发器形成触发集并为其分配特殊的字幕, 用触发集与正常样本一起训练模型嵌入后门水印, 当输入叠加了触发器的图像时, 模型会输出特定的预设文本表明模型所有权。此外, 通过在 LSTM 单元中嵌入密钥和比特签名实现了访问控制, 在使用模型前必须提供正确的密钥, 否则模型性能会严重下降。

(4) 文本处理任务

在文本处理任务中, 输入数据为离散的文字, 需要设计不同于图像处理任务的触发集生成方法。Yadollahi 等[63]提出针对文本处理任务的黑盒水印方案, 在IMDB用户评论和 HamSpam两个数据集上进行了实验。通过交换文档内部分单词生成触发集, 并为其分配新的标签。用触发集和正常训练样本一起训练模型在网络中嵌入水印。验证时输入触发集样本, 根据模型输出判断所有权。

(5) 语音识别系统

Chen 等[64]针对语音识别系统提出在模型参数中嵌入水印的白盒水印方案。水印形式是二元比特串, 根据密钥值在所选层的权重参数的频域分量中嵌入水印, 选择频域系数最高的 N 的元素嵌入水印以增加对攻击的鲁棒性。本方案不需要微调或重训练模型, 直接修改预训练模型参数的频域值, 实验中展示了水印嵌入操作对参数时域值扰动很小, 不会影响模型性能。验证时对比用已有密钥从可疑模型的参数中提取的签名与真实签名的误比特率判断所有权。尽管本方案在新的任务领域进行了探索, 但是直接修改网络的参数嵌入水印的方法与任务无关。

(6) 图神经网络

图神经网络具有与循环神经网络不同的拓扑结构, 需要思考设计在此结构下如何嵌入水印。Zhao 等[65]提出了针对节点分类任务的图神经网络后门水印方案。将具有随机节点特征向量和标签的 Erdos-Renyi 随机图作为触发集, 用触发集图节点的标签携带水印信息。水印为 B 进制序列, B 为节点总类别数, 水印序列长度与节点数相同。用触发集和正常样本一起训练模型, 验证时判断模型对触发集的预测输出是否为预设标签。

(7) 多任务学习

Li 等[66]指出之前的绝大部分工作不能满足所有的水印要求, 而且只考虑了图像分类任务, 他们提出了基于多任务学习的需要白盒验证的水印方案, 将水印嵌入任务作为一项额外的学习任务而不影响模型原本任务, 从而适用于不同的模型任务。根据不同的水印要求设计对应的正则项约束, 利用密码学协议提高对水印检测攻击、歧义攻击的鲁棒性。此外, 还设计了去中心化的验证协议, 可以用时间戳有效防御覆写攻击。

(8) 联邦学习

联邦学习涉及到多方协同训练, 这对模型的版权保护提出了新的挑战。Tekgul 等[67]提出可以保护联邦学习模型的水印方案。为了防止恶意客户偷窃模型侵犯模型所有者权益, 中心服务器在每次聚合局部模型后用触发集样本重训练全局模型从而嵌入水印。由于中心服务器没有用户数据, 所以通过在噪声背景上添加与训练数据无关的特定图案形成触发集, 并为其分配对应标签, 不同的类别标签对应不同的图案。水印嵌入过程独立于聚合过程。

Fan 等[68]从保护端用户权益的角度出发, 提出了可以证明每个用户是联邦学习模型的数据所有者的水印方案框架, 并给出了每个用户嵌入不同的参数水印但又不影响全局模型性能需要满足的条件。用户在各自的局部模型上分别嵌入不同的水印, 将嵌入水印后的模型发给服务器聚合。每个用户在自己的局部模型上同时嵌入参数水印和后门水印, 对应可以采用白盒或黑盒验证方式判断所有权。

Li 等[69]针对联邦学习场景对水印算法及所有权验证提出了新的要求, 基于 Merkle 树结构提出了水印方案框架 Merkle-Sign, 并证明了这个框架可以满足所提的安全要求。此外还提出可以生成鲁棒触发集的水印方案 ATGF, 利用自编码器基于用户密钥为每个客户生成具有不同分布的触发集, 从而提高了水印的鲁棒性。

(9) 强化学习

之前的水印方案都只适用于有监督学习任务, Behzadan 等[70]提出可以保护深度强化学习策略的水印方案。定义与主任务环境不相交的水印环境, 并设计独特的循环状态序列作为标识符, 在训练主任务时交替使用原环境和水印环境。验证所有权时, 判断在水印环境下可疑策略对连续事件输出的状态转换序列是否与标识符序列一致。

3.3.2 其他验证需求

(1) 完整性验证

Guan 等[56]指出之前的水印方案都不可逆的永久的改变了嵌入水印的模型, 破坏了模型的完整性,

并提出了可逆水印方案, 可以实现完整性验证。利用剪枝理论构建主序列, 并用直方图平移方式在其中嵌入水印, 水印为二进制比特串。水印嵌入不需要改变模型的训练或推理过程, 而且提取水印后模型的参数可以复原。如果将整体模型的 hash 值作为水印嵌入, 则可以根据提取水印与嵌入水印是否一致判断模型是否被篡改。

(2) 所有权验证协议

现有用水印技术保护模型版权的研究几乎都在关注水印生成及嵌入提取算法, Li 等[57]指出要在现实场景应用这些算法需要根据不同的场景制定不同的验证协议, 并提出了所有权证明、联邦学习、知识产权转移这三种场景下的协议。

4 攻击方法

本节将现有的针对 DNN 模型版权保护方案的攻击方法分为四大类进行总结: 1)水印检测, 属于最弱的攻击类型, 常常作为攻击的第一步, 在检测到水印存在后发起后续攻击; 2)逃逸攻击, 试图在无法完全去除水印的情况下逃避所有权验证; 3)去除攻击, 旨在破坏并去除模型中的水印使模型所有者无法申明所有权, 根据已知信息、已有资源和水印类型的不同, 攻击者可以通过模型修改攻击、模型提取攻击、后门去除攻击实现水印去除; 4)欺诈攻击, 指攻击者通过在模型中新加额外的水印或者生成伪造水印或窃取模型所有者水印等方式诈称模型所有权。表 7 对比总结了不同类型的攻击方法以及适用的水印类型和攻击者能力。

4.1 水印检测

水印设计的一个要求就是要满足隐蔽性, 如果攻击者可以检测到水印存在就可能进一步发起其他攻击。

表 7 针对 DNN 模型版权保护方案的攻击方法

Table 7 Attack methods for DNN copyright protection schemes

攻击类型			攻击者能力	适用水印类型
去除攻击	水印检测	权重方差分布检测[26]	白盒访问	参数水印
		属性推理攻击[71-72]	白盒访问	通用
	逃逸攻击	集成攻击[38]	黑盒访问	后门水印
		查询修改攻击[38]	白盒访问	后门水印
		查询修改攻击[15]	黑盒访问	后门水印
		共谋攻击[19]	N/A	N/A
	模型修改攻击	剪枝[73]、微调[6] 、正则化[72]		通用
		微调[74-75]、剪枝-微调[76]	白盒访问	后门水印
		神经元重排[77]、权重移动[77]		参数水印
	模型提取攻击	重训练[72,77]、迁移学习[77]	黑盒访问	通用
		蒸馏[78]	白盒访问	通用
	后门去除	Neural Cleanse[33]、Neural Laundering[34]	白盒访问	后门水印
		Deepinspect[35]	黑盒访问	后门水印
欺诈攻击	覆写攻击	Wang 等[26]	白盒访问	通用
	歧义攻击	逆向工程生成水印[48]、伪造密钥生成水印[32,40]	白盒/黑盒	通用

(注: 表中“通用”表示对参数水印和后门水印都适用, N/A 表示文献中没有具体讨论)

Wang 等[26]指出文献[5, 18-19]这类利用密钥矩阵添加正则项嵌入水印的方法增大了参数的标准差, 通过分析权重参数的标准差分布可以检测到水印是否存在并推断出嵌入水印签名的长度。

Wang 等[71]和 Shafieinejad 等[72]分别针对参数水印和后门水印独立提出了属性推理攻击, 检测水印是否存在。攻击者训练多个嵌入水印和未嵌入水印的模型, 利用两类模型的特征差异, 选择有代表性的特征(如权重直方图, 最后一层卷积层的激活函数输出等)训练一个分类器, 判断模型是否嵌入了水印。

文献[38, 15, 33]都提出了针对后门水印的检测方法, 并提出了进一步的水印攻击方法, 将在后文具体介绍。

4.2 逃逸攻击

逃逸攻击指攻击者不能去除水印, 但是试图抑制水印效果逃避所有权验证。Hitaj 等[38]针对后门水印提出两种逃逸攻击方法: 1)集成攻击: 攻击者窃取多个模型, 对每个输入样本查询所有模型, 采用投票机制返回票数最高的作为预测结果。这样可以抑制每个模型特有的水印。2)检测攻击: 攻击者需要窃

取一个可以白盒访问的模型, 借助这个模型训练一个可以辨别输入样本是正常样本还是后门触发集的鉴别器, 如果检测到输入异常, 则拒绝查询或返回随机结果。但是当触发集样本和原样本分布相似时攻击效果受限[72]。

Namba 等[15]针对在原训练样本上叠加触发器的后门触发集提出查询修改攻击。根据样本输入自编码器前后的均方差损失和DNN模型对二者预测输出的散度大小检测输入样本是否有特殊标记。如果输入异常, 则返回 DNN 模型对编码器处理后图像的预测结果。

Chen 等在文献[19]中提到共谋攻击的概念, 并针对这种攻击提出了抗合谋编码的水印方案。共谋攻击指对同一个DNN模型拥有不同指纹的一组用户通过合作构建一个没有水印的模型, 也属于逃逸攻击的一种。

4.3 去除攻击

在去除攻击中, 攻击者通过模型修改、模型提取等方法破坏模型中的水印, 使模型无法提取正确的水印或不再对触发集特殊响应。模型修改攻击通过修改模型参数去除水印, 攻击者需要白盒访问模型; 模型提取攻击除蒸馏攻击需要白盒访问外, 重训练、迁移学习只需要黑盒访问; 后门去除攻击的方案有些需要白盒访问, 之后也有黑盒方案被提出。

4.3.1 模型修改攻击

(1) 剪枝[73]。由于深度神经网络的过参数性, 有很多神经元之间的权重参数对目标任务并没有太大的贡献, 可以通过剪枝操作剪掉这些冗余的参数。如果被剪枝的参数包含了水印信息就会破坏水印。Liu 等[76]提到的“Pruning-Aware”方法可以增强后门水印对剪枝攻击的鲁棒性。先在正常的训练样本上训练模型, 然后剪掉休眠神经元后用触发集和正常样本重训练剪枝后的模型, 然后再复原最初剪枝的参数。这样使得 DNN 模型用相同的参数同时学习了原任务和水印任务, 破坏水印就会影响原任务性能。在 3.1.2 节介绍的指数加权[15]也是对剪枝攻击的一种防御方法。

(2) 微调。由于从头训练一个深度神经网络需要大量数据和计算资源, 现在常用微调预训练模型的方式在不同的下游任务上达到很好的性能。而微调会改变内部参数以及输入输出映射关系, 因此会影响水印性能, 微调也是常见的水印攻击方法。文献[6]中定义了 4 种微调方法。Chen 等[74]指出尽管之前的工作[76]表明单独用微调很难去除水印, 但是他们发现通过精心设计学习率可以有效去除水印, 并提出一种用有限训练数据通过微调去除水印的方案。攻击者拥有有限的有标签训练数据, 以及大量具有相似分布的无标签数据, 将模型对无标签数据的输出作为其标签, 并通过数据增强技术减少对数据的需求。用这些数据以及精心设计的学习率微调水印模型, 同时用弹性权重巩固(elastic weight consolidation, EWC)在去除水印的同时减轻对原任务的影响。但是这种方案需要精心设计学习率难以得到推广, 而且搜集大量相似分布的数据费时费力[75]。Liu 等[75]只用有限的有标签训练数据实现了后门水印去除。用数据增强技术随机遮挡干净样本模拟触发集, 但不改变它们的输出标签。用干净样本和经过处理的样本微调模型, 并在损失函数中添加正则项约束使干净样本和添加扰动的样本在特征空间分布趋于一致, 从而去除水印。但是这种方案只考虑了在原训练数据上叠加触发器图案形成的触发集形式。

(3) 剪枝-微调。Liu 等[76]将剪枝和微调结合起来, 先对模型进行剪枝, 然后用干净样本微调剪枝后的模型, 在恢复模型精度的同时去除水印。

(4) 正则化。Shafieinejad 等[72]认为嵌入水印的模型对水印任务过拟合, 于是先正则化模型避免参数过拟合, 然后用无水印样本微调模型恢复预测精度。

(5) 神经元重排。Lukas 等[77]指出改变 DNN 同一隐藏层内神经元的排列顺序不会影响模型性能, 但这种方式会严重破坏嵌入到模型参数中的水印。针对这一问题 Li 等[79]提出神经元对齐框架增强参数水印算法的鲁棒性, 通过制定编译码规则, 将神经元对触发集序列的标量输出进行编码来标记神经元, 在验证时根据神经元对触发集的输出响应译码后恢复神经元原来的排列顺序, 进而恢复水印。

(6) 权重移动。Lukas 等[77]提出向神经网络卷积层的所有卷积核添加微小扰动, 然后微调模型恢复精度。并且他们通过实验指出目前没有一种水印攻击方法对所有水印方案都有效, 也没有一种水印方案可以防御所有的攻击, 但通过结合多种攻击方法可以去除所有类型的水印。

4.3.2 模型提取攻击

(1) 重训练。受模型提取[80]思想的启发, Shafieinejad 等[72]提出针对后门水印的黑盒攻击方案。用与原任务相似的公开无标签数据查询嵌入水印的模型获得对输入的预测输出作为样本的标签。然后用这组数据以及查询获得的标签重新训练一个替代模型。因为输入的数据不含后门, 因此替代模型去除了水印。Lukas 等[77]针对文献[16]中提出的水印方案提出通过平滑重训练方式去除水印。对每个查询输入

做随机仿射变换, 返回模型对这些输入的平均输出作为预测结果, 在得到的输入输出对上重新训练替代模型。

(2) 蒸馏[78]指将大型的复杂网络学习到的知识迁移到小型网络中, 属于模型压缩的一种方法。原模型中的水印信息在经过知识蒸馏后可能会被去除。针对蒸馏攻击 Yang 等[81]提出仅用嵌入水印的触发集额外训练一个与原模型具有相同输入输出维度的牢固模型(ingrainer model), 使它的预测输出包含水印内容。将牢固模型的损失函数作为一个正则项约束主网络的训练过程, 使其在学习原任务的同时, 学习到牢固模型包含的水印信息。

(3) 迁移学习[77]利用在其他相似但不同的任务上训练好的模型初始化替代模型参数, 通过黑盒方式访问试图攻击的原模型获得本任务中训练数据的标签, 利用这些数据训练替代模型。在训练时冻结较低层, 只更改靠近输出层的参数从而减少训练开销。

4.3.3 后门去除

一些原本用于缓解后门攻击的防御方法可以用于去除基于后门方法嵌入的水印。Wang 等[33]提出的 Neural Cleanse 可用于检测并去除后门水印, 他们提出了通过逆向工程重构触发集的算法, 利用重构的触发集和正常样本可以构建输入过滤器, 检测输入是否异常。此外, 提出两种去除后门的方法: 1)通过判断重构触发集和正常样本激活神经元的不同, 对模型进行剪枝直到模型不再响应触发集; 2)给重构的触发集分配正确的标签后微调模型, 使模型可以正确分类带有触发器的样本。但是这种算法需要已知部分有标签的干净样本, 而且只适用于在图像上叠加触发器的后门水印。

Aiken 等[34]基于文献[33]提出了后门水印去除的整体算法, 提出了具体的神经元剪枝方案, 并进一步考虑到用重构触发集微调模型时为与原任务无关的触发集分配标签的问题。

文献[33]和[34]都需要对模型的白盒访问, 而且需要已知部分干净训练样本才能重构触发集, Chen 等[35]提出不需要干净样本且通过黑盒访问就可以去除后门水印的方案。通过模型逆向方法得到一个包含所有类的替代训练数据集, 基于这些数据利用条件生成对抗网络为每个输出类重构触发集, 然后用异常检测最终确定触发集。为触发集中的样本分配正确的标签得到修补数据集, 用修补数据集和重构的触发集做对抗训练, 微调模型去除后门水印。

4.4 欺诈攻击

当水印与模型所有者之间没有关联时, 攻击者可以用泄露或窃取的水印诈称所有权。此外, 攻击者还可以通过主动添加新的水印或生成伪造水印的方式诈称所有权。

(1) 覆写攻击

覆写攻击是指攻击者用同样的水印嵌入算法, 试图在已嵌入水印的模型中嵌入自己的水印并去除(覆盖)之前的水印, 从而接管所有权。Wang 等[26]检测到水印存在并分析得到水印长度后, 用覆写攻击的方式去除原有水印并嵌入新的水印。同时他们在损失函数中添加了权重的 L2 范数, 防止嵌入水印增大权重的标准差。

值得注意的是如果攻击者嵌入了新的水印但是没有去除之前的水印, 则认为攻击没有意义。因为模型所有者可以出示只包含一个水印的模型, 而攻击者试图诈称所有权的模型同时包含两个水印, 无法造成所有权混淆[32]。

(2) 歧义攻击

歧义攻击指攻击者利用水印的可逆性通过逆向工程得到满足验证条件的伪造的水印[48], 或者用伪造的密钥生成满足验证条件的伪造水印[32,40], 从而造成歧义。歧义攻击与覆写攻击的不同之处在于: 覆写攻击通过改变模型参数或输入输出关系嵌入水印, 是一种白盒攻击方式; 而歧义攻击不改变模型参数, 只是通过逆向或伪造得到新的水印, 属于黑盒攻击[49,66]。Guo 等[29]指出如果用随机抽象图像作为触发集[6], 攻击者很容易生成伪造的触发集造成歧义。Zhu 等[17]指出构建不可逆的水印机制可以防御歧义攻击。

5 未来研究方向

基于前文对现有工作的分析与总结, 本节对目前工作存在的问题及未来可以探索的研究方向做出进一步思考:

(1) 提高对攻击的鲁棒性, 增强主动防御能力。由于深度学习模型有巨大价值, 因此攻击者会尝试各种方法攻击、窃取模型。尽管从 2017 年提出保护 DNN 版权的方案至今大量研究都在探索如何更有效的保护深度学习模型版权, 但没有一种方案能防御所有的攻击[77], 因此继续提升方案的鲁棒性十分必要。此外, 应该更多的思考如何主动防御攻击者攻击, 而不仅仅是在发现模型被窃取后验证所有权。

(2) 扩展其他任务领域。目前大量研究聚焦于图像分类任务及对应模型的版权保护, 尽管最近陆续有关于其他任务领域的探索, 但仍然相对较少且处于初级阶段。未来的研究可以进一步推进 DNN 版权

保护技术在不同场景不同任务中的应用。

(3) 提高通用性。横向对比针对图像分类任务的 DNN 版权保护方案与逐渐提出的保护其他任务及对应模型的方案可以发现现有的很多方案并不具备很好的通用性。后门水印方案需要根据不同任务类型的输入输出特征及维度设计针对性的触发集; 参数水印方案与输入输出特征关联不大, 但是需要根据使用深度神经网络类型的不同设计合适的保护方案; 基于加密算法的方案[51]有较好的通用性, 但是在实际部署时可能受到硬件条件等制约。在将 DNN 版权保护技术应用到其他任务领域的同时, 探索如何设计真正通用的 DNN 版权保护方案是未来值得研究的一个问题。

(4) 完善相关理论, 增强可解释性。目前的 DNN 版权保护方案, 无论是参数水印还是后门水印都依赖实验结果检测方案性能, 而缺少理论支撑。建立完善的深度学习模型版权保护体系, 增强对方案的可解释性及理论性说明有助于未来的深入研究。

(5) 关注可落地性。大多方案都是基于小规模数据集进行实验测试, 部分方案在大规模数据集或现实工业界数据集上效果不好或者开销很大, 因此在方案设计时应该考虑它的实际应用价值, 考虑引入的成本开销是否合理、硬件需求是否可以满足。此外, 在关注底层方案设计的同时, 需要针对不同的现实场景完善上层版权验证协议。

6 结束语

本文对现有的深度学习模型版权保护方案进行了全面的梳理与总结, 以图像分类任务为例对 DNN 版权保护方案根据实现功能和实现方式的不同进行了分类介绍, 并详细介绍了 DNN 版权保护方案在其他任务及场景中的应用, 每一类绘制表格做出清晰明了的对比与总结; 分四大类总结介绍了针对 DNN 版权保护的攻击方法。对深度学习模型的版权保护十分必要, 而目前技术尚不成熟, 还有很多有待探索的方向, 希望本文可以为未来的研究提供有益的参考。

参考文献

[1] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]. *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016: 779-788.

[2] Amodei D, Ananthanarayanan S, Anubhai R, et al. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin[C]. *International conference on machine learning*, 2016: 173-182.

[3] Devlin J, Chang M W, Lee K, et al. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding[EB/OL]. 2018: 1810.04805. https://arxiv.org/abs/1810.04805v2.

[4] Strubell E, Ganesh A, McCallum A. Energy and Policy Considerations for Deep Learning in NLP[EB/OL]. 2019: 1906.02243. https://arxiv.org/abs/1906.02243v1.

[5] 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.

[6] Adi Y, Baum C, Cisse M, et al. Turning Your Weakness into a Strength: Watermarking Deep Neural Networks by Backdooring [EB/OL]. 2018: 1802.04633. https://arxiv.org/abs/1802.04633v3.

[7] van Schyndel R G, Tirkel A Z, Osborne C F. A Digital Watermark[C]. *Proceedings of 1st International Conference on Image Processing*, 1994: 86-90.

[8] Li Y, Wang H X, Barni M. A Survey of Deep Neural Network Watermarking Techniques[EB/OL]. 2021: 2103.09274. https://arxiv.org/abs/2103.09274v1.

[9] Li Y M, Jiang Y, Li Z F, et al. Backdoor Learning: A Survey [EB/OL]. 2020: 2007.08745. https://arxiv.org/abs/2007.08745v5.

[10] Gu T Y, Liu K, Dolan-Gavitt B, et al. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks[J]. *IEEE Access*, 2019, 7: 47230-47244.

[11] Liu Y F, Ma X J, Bailey J, et al. Reflection Backdoor: A Natural Backdoor Attack on Deep Neural Networks[M]. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020: 182-199.

[12] Dumford J, Scheirer W. Backdooring Convolutional Neural Networks *via* Targeted Weight Perturbations[EB/OL]. 2018: 1812.03128. https://arxiv.org/abs/1812.03128v1.

[13] Wang S, Nepal S, Rudolph C, et al. Backdoor Attacks Against Transfer Learning with Pre-Trained Deep Learning Models[J]. *IEEE Transactions on Services Computing*, 2022, 15(3): 1526-1539.

[14] Le Merrer E, Perez P, Trédan G. Adversarial Frontier Stitching for Remote Neural Network Watermarking[EB/OL]. 2017: 1711.01894. https://arxiv.org/abs/1711.01894v2.

[15] Namba R, Sakuma J. Robust Watermarking of Neural Network with Exponential Weighting[EB/OL]. 2019: 1901.06151. https://arxiv.org/abs/1901.06151v1.

[16] Szyller S, Atli B G, Marchal S, et al. DAWN: Dynamic Adversarial Watermarking of Neural Networks[EB/OL]. 2019: 1906.00830. https://arxiv.org/abs/1906.00830v5.

[17] Lou X X, Guo S W, Zhang T W, et al. When NAS Meets Watermarking: Ownership Verification of DNN Models via Cache Side Channels[EB/OL] 2021: ArXiv Preprint ArXiv:2102. 03523.

[18] Rouhani B D, Chen H L, Koushanfar F. DeepSigns: A Generic Watermarking Framework for IP Protection of Deep Learning Models[EB/OL]. 2018: 1804.00750. https://arxiv.org/abs/1804.00750v2.

[19] Chen H L, Rohani B D, Koushanfar F. DeepMarks: A Digital Fingerprinting Framework for Deep Neural Networks[EB/OL]. 2018: 1804.03648. https://arxiv.org/abs/1804.03648v1.

[20] Tartaglione E, Grangetto M, Cavagnino D, et al. Delving in the Loss Landscape to Embed Robust Watermarks into Neural Networks[C]. *2020 25th International Conference on Pattern Recog-*

nition, 2021: 1243-1250.

[21] Feng L, Zhang X P. Watermarking Neural Network with Compensation Mechanism[M]. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020: 363-375.

[22] Kuribayashi M, Tanaka T, Funabiki N. DeepWatermark: Embedding Watermark into DNN Model[C]. *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2020: 1340-1346.

[23] Wang J F, Wu H Z, Zhang X P, et al. Watermarking in Deep Neural Networks *via* Error Back-Propagation[J]. *Electronic Imaging*, 2020, 32(4): 22-1-22-9.

[24] Wang T H, Kerschbaum F. RIGA: Covert and Robust White-Box Watermarking of Deep Neural Networks[EB/OL]. 2019: 1910.14268. https://arxiv.org/abs/1910.14268v4.

[25] Lv P Z, Li P, Zhang S Z, et al. HufuNet: Embedding the Left Piece as Watermark and Keeping the Right Piece for Ownership Verification in Deep Neural Networks[EB/OL]. 2021: 2103.13628. https://arxiv.org/abs/2103.13628v1.

[26] Wang T H, Kerschbaum F. Attacks on Digital Watermarks for Deep Neural Networks[C]. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics*, *Speech and Signal Processing*, 2019: 2622-2626.

[27] Chen H L, Rouhani B D, Koushanfar F. BlackMarks: Blackbox Multibit Watermarking for Deep Neural Networks[EB/OL]. 2019: 1904.00344. https://arxiv.org/abs/1904.00344v1.

[28] Zhang J L, Gu Z S, Jang J, et al. Protecting Intellectual Property of Deep Neural Networks with Watermarking[C]. *The 2018 on Asia Conference on Computer and Communications Security*, 2018: 159-172.

[29] Guo J, Potkonjak M. Watermarking Deep Neural Networks for Embedded Systems[C]. *The International Conference on Computer-Aided Design*, 2018: 1-8.

[30] Li H Y, Wenger E, Shan S, et al. Piracy Resistant Watermarks for Deep Neural Networks[EB/OL]. 2019: 1910.01226. https://arxiv.org/abs/1910.01226v3.

[31] Maung Maung A P, Kiya H. Piracy-Resistant DNN Watermarking by Block-Wise Image Transformation with Secret Key[C]. *The 2021 ACM Workshop on Information Hiding and Multimedia Security*, 2021: 159-164.

[32] Zhu R J, Zhang X P, Shi M T, et al. Secure Neural Network Watermarking Protocol Against Forging Attack[J]. *EURASIP Journal on Image and Video Processing*, 2020, 2020(1): 37.

[33] Wang B L, Yao Y S, Shan S, et al. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks[C]. *2019 IEEE Symposium on Security and Privacy*, 2019: 707-723.

[34] Aiken W, Kim H, Woo S, et al. Neural Network Laundering: Removing Black-Box Backdoor Watermarks from Deep Neural Networks[J]. *Computers & Security*, 2021, 106: 102277.

[35] Chen H L, Fu C, Zhao J S, et al. DeepInspect: A Black-Box Trojan Detection and Mitigation Framework for Deep Neural Networks[C]. *The Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019: 4658-4664.

[36] Jia H R, Choquette-Choo C A, Chandrasekaran V, et al. Entangled watermarks as a defense against model extraction[C]. *30th USENIX Security Symposium*, 2021: 1937-1954.

[37] Li Z, Hu C Y, Zhang Y, et al. How to Prove Your Model Belongs to You: A Blind-Watermark Based Framework to Protect Intellectual Property of DNN[C]. *The 35th Annual Computer Security Applications Conference*, 2019: 126-137.

[38] Hitaj D, Hitaj B, Mancini L V. Evasion Attacks Against Watermarking Techniques Found in MLaaS Systems[C]. *2019 Sixth International Conference on Software Defined Systems*, 2019: 55-63.

[39] Li F Q, Wang S L. Knowledge-Free Black-Box Watermark and Ownership Proof for Image Classification Neural Networks[EB/OL]. 2022: 2204.04522. https://arxiv.org/abs/2204.04522v1.

[40] Li M, Zhong Q, Zhang L Y, et al. Protecting the Intellectual Property of Deep Neural Networks with Watermarking: The Frequency Domain Approach[C]. *2020 IEEE 19th International Conference on Trust*, *Security and Privacy in Computing and Communications*, 2020: 402-409.

[41] Guo J, Potkonjak M. Evolutionary Trigger Set Generation for DNN Black-Box Watermarking[EB/OL]. 2019: 1906.04411. https://arxiv.org/abs/1906.04411v2.

[42] Li F Q, Wang S L. Persistent Watermark for Image Classification Neural Networks by Penetrating the Autoencoder[C]. *2021 IEEE International Conference on Image Processing*, 2021: 3063-3067.

[43] Zhong Q, Zhang L Y, Zhang J, et al. Protecting IP of Deep Neural Networks with Watermarking: A New Label Helps[M]. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020: 462-474.

[44] Sun S C, Xue M F, Wang J, et al. Protecting the Intellectual Properties of Deep Neural Networks with an Additional Class and Steganographic Images[EB/OL]. 2021: 2104.09203. https://arxiv.org/abs/2104.09203v1.

[45] Cao X Y, Jia J Y, Gong N Z. IPGuard: Protecting Intellectual Property of Deep Neural Networks *via* Fingerprinting the Classification Boundary[EB/OL]. 2019: 1910.12903. https://arxiv.org/abs/1910.12903v5.

[46] Zhao J J, Hu Q Y, Liu G Y, et al. AFA: Adversarial Fingerprinting Authentication for Deep Neural Networks[J]. *Computer Communications*, 2020, 150: 488-497.

[47] Lukas N, Zhang Y X, Kerschbaum F. Deep Neural Network Fingerprinting by Conferrable Adversarial Examples[EB/OL]. 2019: 1912.00888. https://arxiv.org/abs/1912.00888v4.

[48] Fan Lixin, Ng K W, Chan C S. Rethinking Deep Neural Network Ownership Verification: Embedding Passports to Defeat Ambiguity Attacks[J]. *Advances in Neural Information Processing Systems*, 2019, 32: 4716-4725.

[49] Fan L X, Ng K W, Chan C S, et al. DeepIPR: Deep Neural Network Ownership Verification with Passports[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, 44(10): 6122-6139.

[50] Zhang J, Chen D D, Liao J, et al. Passport-aware normalization for deep model protection[J]. *Advances in Neural Information Processing Systems*, 2020,33: 22619-22628.

[51] Lin N, Chen X M, Lu H, et al. Chaotic Weights: A Novel Approach to Protect Intellectual Property of Deep Neural Networks[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*

and Systems, 2021, 40(7): 1327-1339.

[52] Xue M F, Wu Z Y, Wang J, et al. AdvParams: An Active DNN Intellectual Property Protection Technique *via* Adversarial Perturbation Based Parameter Encryption[EB/OL]. 2021: 2105.13697. https://arxiv.org/abs/2105.13697v1.

[53] Chen M L, Wu M. Protect Your Deep Neural Networks from Piracy[C]. *2018 IEEE International Workshop on Information Forensics and Security*, 2018: 1-7.

[54] Pyone A, Maung M, Kiya H. Training DNN Model with Secret Key for Model Protection[C]. *2020 IEEE 9th Global Conference on Consumer Electronics*, 2020: 818-821.

[55] Xue M F, Wu Z Y, He C, et al. Active DNN IP Protection: A Novel User Fingerprint Management and DNN Authorization Control Technique[C]. *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications*, 2020: 975-982.

[56] Guan X Q, Feng H M, Zhang W M, et al. Reversible Watermarking in Deep Convolutional Neural Networks for Integrity Authentication[C]. *The 28th ACM International Conference on Multimedia*, 2020: 2273-2280.

[57] Li F Q, Wang S L, Liew A W C. Regulating Ownership Verification for Deep Neural Networks: Scenarios, Protocols, and Prospects[EB/OL]. 2021: 2108.09065. https://arxiv.org/abs/2108.09065v1.

[58] Zhang J, Chen D D, Liao J, et al. Model Watermarking for Image Processing Networks[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, 34(7): 12805-12812.

[59] Zhang J, Chen D, Liao J, et al. Deep Model Intellectual Property Protection *via* Deep Watermarking[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, 44(8): 4005-4020.

[60] Wu H Z, Liu G, Yao Y W, et al. Watermarking Neural Networks with Watermarked Images[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021, 31(7): 2591-2601.

[61] Ong D S, Seng Chan C E, Ng K W, et al. Protecting Intellectual Property of Generative Adversarial Networks from Ambiguity Attacks[C]. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021: 3629-3638.

[62] Lim J H, Chan C S, Ng K W, et al. Protect, Show, Attend and Tell: Empowering Image Captioning Models with Ownership Protection[J]. *Pattern Recognition*, 2022, 122: 108285.

[63] Yadollahi M M, Shoeleh F, Dadkhah S, et al. Robust Black-Box Watermarking for Deep NeuralNetwork Using Inverse Document Frequency[EB/OL]. 2021: 2103.05590. https://arxiv.org/abs/2103.05590v1.

[64] Chen H L, Darvish B, Koushanfar F. SpecMark: A Spectral Watermarking Framework for IP Protection of Speech Recognition Systems[C]. *Interspeech 2020*, 2020: 2312-2316.

[65] Zhao X Y, Wu H Z, Zhang X P. Watermarking Graph Neural Networks by Random Graphs[C]. *2021 9th International Symposium on Digital Forensics and Security*, 2021: 1-6.

[66] Li F Q, Wang S L. Secure Watermark for Deep Neural Networks with Multi-Task Learning[EB/OL]. 2021: 2103.10021. https://arxiv.org/abs/ 2103.10021v3.

[67] Tekgul B G A, Xia Y X, Marchal S, et al. WAFFLE: Watermarking in Federated Learning[C]. *2021 40th International Symposium on Reliable Distributed Systems*, 2021: 310-320.

[68] Li B W, Fan L X, Gu H L, et al. FedIPR: Ownership Verification for Federated Deep Neural Network Models[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023, 45(4): 4521-4536.

[69] Li F Q, Wang S L, Liew A W C. Towards Practical Watermark for Deep Neural Networks in Federated Learning[EB/OL]. 2021: 2105.03167. https://arxiv.org/abs/2105.03167v3.

[70] Behzadan V, Hsu W. Sequential Triggers for Watermarking of Deep Reinforcement Learning Policies[EB/OL]. 2019: 1906.01126. https://arxiv.org/abs/1906.01126v1.

[71] Wang T H, Kerschbaum F. RIGA: Covert and Robust White-Box Watermarking of Deep Neural Networks[EB/OL]. 2019: 1910.14268. https://arxiv.org/abs/1910.14268v4.

[72] Shafieinejad M, Lukas N, Wang J Q, et al. On the Robustness of Backdoor-Based Watermarking in Deep Neural Networks[C]. *The 2021 ACM Workshop on Information Hiding and Multimedia Security*, 2021: 177-188.

[73] Han S, Pool J, Tran J, et al. Learning both weights and connections for efficient neural network[J]. *Advances in neural information processing systems*, 2015, 28: 1135-1143.

[74] Chen X Y, Wang W X, Bender C, et al. REFIT: A Unified Watermark Removal Framework for Deep Learning Systems with Limited Data[C]. *The 2021 ACM Asia Conference on Computer and Communications Security*, 2021: 321-335.

[75] Liu X K, Li F T, Wen B H, et al. Removing Backdoor-Based Watermarks in Neural Networks with Limited Data[C]. *2020 25th International Conference on Pattern Recognition*, 2021: 10149-10156.

[76] Liu K, Dolan-Gavitt B, Garg S. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks[M]. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018: 273-294.

[77] Lukas N, Jiang E, Li X D, et al. SoK: How Robust Is Image Classification Deep Neural Network Watermarking? (Extended Version)[EB/OL]. 2021: 2108.04974. https://arxiv.org/abs/2108.04974v1.

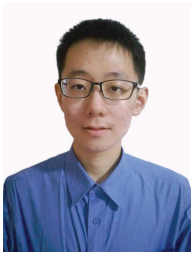
[78] Hinton G, Vinyals O, Dean J. Distilling the Knowledge in a Neural Network[EB/OL]. 2015: 1503.02531. https://arxiv.org/abs/1503.02531v1.

[79] Li F Q, Wang S L, Zhu Y. Fostering the Robustness of White-Box Deep Neural Network Watermarks by Neuron Alignment[C]. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022: 3049-3053.

[80] Tramèr F, Zhang F, Juels A, et al. Stealing Machine Learning Models *via* Prediction APIs[EB/OL]. 2016: 1609.02943. https://arxiv.org/abs/1609.02943v2.

[81] Yang Z Q, Dang H, Chang E C. Effectiveness of Distillation Attack and Countermeasure on Neural Network Watermarking[EB/OL]. 2019: 1906.06046. https://arxiv.org/abs/1906.06046v1.

李珮玄 于 2021 年在东南大学信息科学与工程专业获得学士学位。现在在上海交通大学网络空间安全专业攻读硕士学位。研究领域为人工智能安全、深度学习模型版权保护技术。Email: peixuan.li@sjtu.edu.cn



罗书卿 在上海交通大学网络空间安全专业攻读学士学位。研究领域为 AI 安全。Email: xingruyu@sjtu.edu.cn

刘功申 于上海交通大学获博士学位。现任上海交通大学网络空间安全学院教授, 博士生导师。研究领域为自然语言处理、信息安全等。Email: lgshen@sjtu.edu.cn

黄土 在上海交通大学网络空间安全专业攻读学士学位。研究领域为人工智能对抗攻击。Email: sjtuht@sjtu.edu.cn

宋家鑫 在上海交通大学计算机专业攻读学士学位。研究领域为人工智能系统和理论计算机。Email: sjtu_xiaosong@sjtu.edu.cn