DOI:10.7544/issn1000-1239.2021.20200978
58(5):964-976,2021

神经网络水印技术研究进展

张颖君[1,4] 陈 恺[2,3] 周 赓[1,4] 吕培卓[2,3] 刘 勇[2] 黄 亮[5]
[1](中国科学院软件研究所可信计算与信息保障实验室 北京 100190)
[2](信息安全国家重点实验室(中国科学院信息工程研究所) 北京 100195)
[3](中国科学院大学网络空间安全学院 北京 100049)
[4](中国科学院大学计算机科学与技术学院 北京 100049)
[5](奇安信科技集团股份有限公司 北京 100015)
(yingjun2011@iscas.ac.cn)

Research Progress of Neural Networks Watermarking Technology

Zhang Yingjun[1,4], Chen Kai[2,3], Zhou Geng[1,4], Lü Peizhuo[2,3], Liu Yong[2], and Huang Liang[5]
[1](*Trusted Computing and Information Assurance Laboratory*, *Institute of Software*, *Chinese Academy of Sciences*, *Beijing* 100190)
[2](*State Key Laboratory of Information Security* (*Institute of Information Engineering*, *Chinese Academy of Sciences*), *Beijing* 100195)
[3](*School of Cyber Security*, *University of Chinese Academy of Science*, *Beijing* 100049)
[4](*College of Computer Science and Technology*, *University of Chinese Academy of Sciences*, *Beijing* 100049)
[5](*Legendsec Information Technology* (*Beijing*) *Inc*, *Beijing* 100015)

Abstract With the popularization and application of deep neural networks, the trained neural network model has become an important asset and has been provided as machine learning services (MLaaS) for users. However, as a special kind of user, attackers can extract the models when using the services. Considering the high value of the models and risks of being stolen, service providers start to pay more attention to the copyright protection of their models. The main technique is adopted from the digital watermark and applied to neural networks, called neural network watermarking. In this paper, we first analyze this kind of watermarking and show the basic requirements of the design. Then we introduce the related technologies involved in neural network watermarking. Typically, service providers embed watermarks in the neural networks. Once they suspect a model is stolen from them, they can verify the existence of the watermark in the model. Sometimes, the providers can obtain the suspected model and check the existence of watermarks from the model parameters (white-box). But sometimes, the providers cannot acquire the model. What they can only do is to check the input/output pairs of the suspected model (black-box). We discuss these watermarking methods and potential attacks against the watermarks from the viewpoint of robustness, stealthiness, and security. In the end, we discuss future directions and potential challenges.

收稿日期:2020-11-25;**修回日期**:2021-02-05

基金项目:国家自然科学基金重点项目(U1836211);国家自然科学基金项目(62072448);北京市自然科学基金项目(JQ18011);中国科学院青年创新促进会优秀会员(Y202046);大数据协同安全国家工程实验室开放课题
This work was supported by the Key Program of the National Natural Science Foundation of China (U1836211), the National Natural Science Foundation of China (62072448), the Beijing Natural Science Foundation (JQ18011), the Excellent Member of Youth Innovation Promotion Association, Chinese Academy of Sciences (Y202046), and the Open Project of National Engineering Laboratory of Big Data Collaborative Security.

通信作者:刘勇(liuyong03@qianxin.com)

Key words digital watermark; deep neural network; neural network backdoor; neural network watermark; attacks on the watermarking

摘 要 随着深度神经网络的推广应用,训练后的神经网络模型已经成为一种重要的资产并为用户提供服务.服务商在提供服务的同时,也更多地关注其模型的版权保护,神经网络水印技术应运而生.首先,分析水印及其基本需求,并对神经网络水印涉及的相关技术进行介绍;对深度神经网络水印技术进行对比,并重点对白盒和黑盒水印进行详细分析;对神经网络水印攻击技术展开对比,并按照水印攻击目标的不同,对水印鲁棒性攻击、隐蔽性攻击、安全性攻击等技术进行分类介绍;最后对未来方向与挑战进行探讨.

关键词 数字水印;深度神经网络;神经网络后门;神经网络水印;水印攻击

中图法分类号 TP391

数字水印[1]是嵌在其他数据(宿主数据)中具有可鉴别性的数字信号或模式,同时不应影响宿主数据的可用性[2].水印技术的应用主要包括版权保护、数据监控和数据跟踪等[3].随着计算机技术的发展和需要保护对象的变化和增加,数字水印技术经历了多媒体水印、软件水印到机器学习算法/模型水印的发展过程.

最早,数字水印主要针对电子广告、数字图书、网络音视频等数字产品内容进行版权保护.这些多媒体数字产品往往可以通过低廉的成本进行未授权的复制或传播.因此,通过添加水印可以保护其在非法拷贝、再次传播和盗用时的知识产权[2].多媒体水印是主要针对图像、视频、音频、文本文档等媒体的水印技术.多媒体水印可由多种模型构成,如随机数字序列、数字标识、文本以及图像等.从稳健性和安全性考虑,常常需要对水印进行随机化以及加密处理[4].多媒体水印中常用到的 2 种关键技术包括扩频水印和量化水印[5].扩频水印[6]是通过扩频通信技术,将载体信号视为宽带信号,水印信号视为窄带信号,把一个水印的能量谱扩展到很宽的频带中,从而分配到每个频率分量上的水印信号能量较小且难以检测.量化水印[7-8]根据水印信息的不同将原始载体数据量化到不同的量化区间,而检测时根据数据所属的量化区间来识别水印信息.

随着软件应用的普及,软件代码的重用问题越发突出,而软件水印是解决软件版权问题的重要手段.软件水印主要是指在代码中植入一个特殊的标识符(水印),该水印可以承载软件作者、版权等信息,事后通过特殊的提取器将其从被告软件中识别或抽取出来作为证据以达到检测目的[9].根据水印的加入位置,软件水印可以分为代码水印和数据水印[10].代码水印隐藏在程序的指令部分中,而数据水印则隐藏在包括头文件、字符串和调试信息等数据中.根据水印被加载的方式,软件水印可分为静态水印和动态水印.静态水印[11]主要是将水印植入到可执行程序的代码或数据中,其提取过程不需要运行程序.通过静态分析完成识别或提取,主要分为代码替换法[12]、静态图法[13]和抽象解释法[14]等.动态水印是将水印植入到程序的执行过程或运行状态中,即根据程序某个时刻运行的状态进行信息的编码,主要包括基于线程[15]、基于图[16]和基于路径[17]等水印技术.

近几年,机器学习(machine learning, ML),尤其是深度学习,已成为计算机领域发展最迅速的技术之一[18],越来越多的领域用到了机器学习相关模型和算法.机器学习方法主要通过统计技术构建数据模型,这些数据模型可以从数据样本(训练集)中进行学习,训练完后可以用于后续的推断.目前,鉴于有效训练这些模型需要大量的专业知识、数据和计算资源,训练后的模型可以认为是重要的资产,并作为服务 MLaaS(machine learning as a service)提供给用户使用[19].一种典型的方式是 MLaaS 平台为用户提供应用程序编程接口(application programming interface, API),该接口与位于云服务中经过训练的专有模型进行交互,用户通过 API 可以使用模型,并支付相关费用.虽然这种服务方便了开发人员,但也成为恶意用户关注的重点[20].他们企图非法使用甚至窃取相关机器学习模型及数据.此外,一些公司将机器学习模型直接进行出售,但是又担心该模型会被非授权转售或泄露.因此,水印概念被扩展到机器学习模型领域,嵌入需保护的模型,并用来保护机器学习模型,尤其是训练成本较高的神经网络

模型.一旦模型拥有者发现可能被窃取的模型,就可以通过隐藏在模型中的水印信息来验证其是否被盗用,以此来保护模型的版权.本文将重点对近年来神经网络相关水印技术展开综述.

1 神经网络水印技术相关基础

当前的机器学习水印主要针对深度神经网络(deep neural network, DNN)进行研究,包括卷积网络[21]、生成对抗网络[22]以及用于自然语言处理[23]等.因此,本文将重点对神经网络水印相关技术展开介绍.

1.1 水印及其基本需求

水印用于保护某种资产.传统水印保护纸质印刷品(如书籍)、数字媒体、软件资产等.我们以书籍为例,图 1 展示了水印的植入过程.在一本正版书籍上,出版社贴上水印,从而保证了该书籍为正版书籍.通常情况下,书籍水印应该满足于 4 条基本性质,使得攻击者(盗版商)难以对其进行复制,从而保护正版书籍的版权.

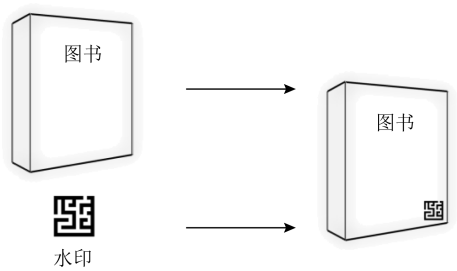


Fig. 1 Implant a watermark (for books)

图 1 水印的植入(用于书籍)

1) 保真度.即添加的水印不应影响原有书籍的阅读.

2) 安全性.即水印本身不易伪造,否则盗版书籍也能贴上相似的水印从而欺骗读者. 然而,在媒体水印安全性方面,攻击者可以在媒体上嵌入类似的水印,从而使得攻击者也可以宣称该媒体的拥有权.因此,媒体水印的高安全性很难得到保障.

3) 鲁棒性.即水印不易去除.如强行去除后会损坏水印本身.随着保护对象的改变,水印会增加不同的性质或需求,或者对原有特性进行些许改变.例如在保护多媒体水印时,如强行去除后会损坏媒体本身.注意攻击者此时不关心水印是否会损坏;但如果媒体损坏,则会影响用户观看/收听媒体的体验.

在保护软件等代码时,考虑到软件本身的代码易被去除,鲁棒性很难达到.因此多数研究人员尝试隐藏软件水印.即增加隐蔽性特性.

4) 隐蔽性.即水印隐藏在软件中,不易被发现,通过隐蔽性达到不易去除的效果.

表 1 对比了多种被保护对象以及其对水印相关特性的需求.

Table 1 The Requirement of Different Protection Objects for Watermark Characteristics

表 1 不同保护对象对水印特性的需求

基本性质	书籍	媒体	软件	神经网络模型
保真度	需要	需要	需要	需要
安全性	需要	需要	需要	需要
鲁棒性	需要	需要	需要	需要
隐蔽性	不需要	不需要	需要	需要

表 1 中第 5 列是神经网络模型水印的相关需求.典型地,加入模型的水印不应该影响原始模型的准确性(保真度).水印本身不易伪造,否则攻击者也可以宣称对模型的所有权.水印应该足够鲁棒,不易被移除或者破坏.但考虑到若模型被发现后易被移除,因此隐蔽性也常常作为模型拥有者考虑的因素.

1.2 神经网络相关算法

神经网络是一种模仿动物神经网络行为特征,进行分布式并行信息处理的算法数学模型.这种网络依靠系统的复杂程度,通过调整内部大量节点之间相互连接的关系,从而达到处理信息的目的[24].

DNN[25]与传统的单层神经网络不同,是由多个隐藏层组成.这种基于层次结构的结构使其能够处理大量的高维数据.通常认为,在 DNN 中堆叠更多层可以使用户从输入数据中识别和提取更复杂的特征.大量多样的、信息丰富的训练数据对于 DNN 的成功至关重要.此外,还需要进行大量的训练来找到合适的网络结构和参数.如图 2 所示:

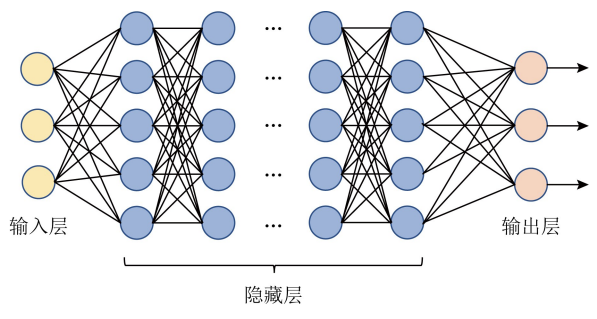


Fig. 2 Deep neural network

图 2 深度神经网络

如图 2 所示,深度神经网络一般由输入层,隐藏

层和输出层组成.输入层和输出层是单层的,而隐藏层可以根据信号处理的复杂程度扩展到多层算法.每个层包含多个节点,并且仅对相邻层施加效果.

1.3 神经网络后门

神经网络可以被嵌入后门:给定这类神经网络一个或一组特殊的实例(通常称为触发器 Trigger),神经网络执行分类任务时,会执行特殊的分类任务,将特殊的实例分类到预设的目标标签中(通常情况下会违背用户的感知)[26].后门攻击可能引发很多安全问题,例如自动驾驶模型一旦被嵌入后门[27],则当遇到某个特殊图案时,很可能发生交通意外.

神经网络后门原理如图 3 所示:

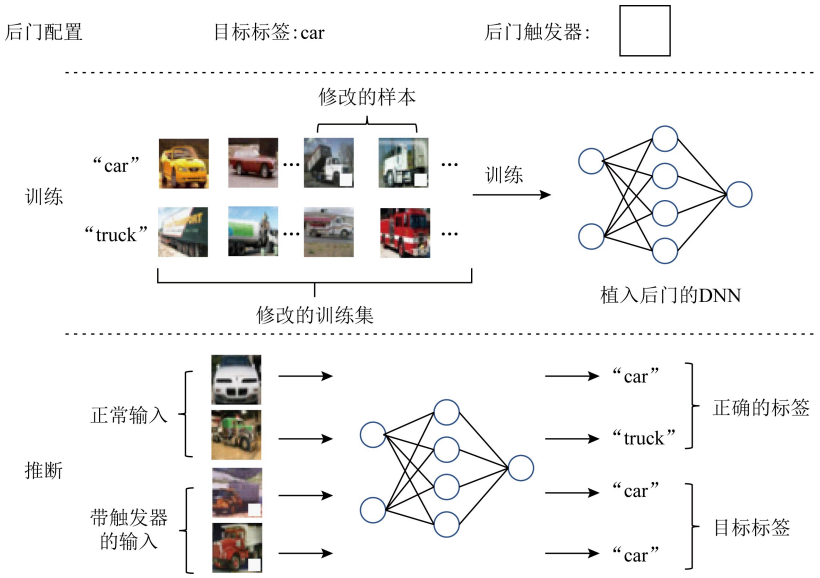


Fig. 3 Neural network backdoor

图 3 神经网络后门

图 3 显示了一个后门攻击的例子.触发器是右上角的白色正方形,目标标签为"car".在训练过程中,修改一部分训练集使其具有触发器印记,并将其对应的标签指定为目标标签.训练好的模型将会把带有触发器的样本识别为目标标签,同时仍能正确识别良性图像的标签.

神经网络后门攻击(或植入)主要通过 2 种方式实现:

1) 主要通过对训练集进行中毒攻击实现[27-29]:通过创建小部分后门触发数据集,满足预设后门条件——该部分数据集通过神经网络输出攻击者指定的结果;然后将触发数据集加入到正常训练集中一起进行训练;最终训练后的神经网络分类结果中包含后门.该方法主要用于在训练阶段之前,通过调整相关参数或采用替代模型,完成攻击[26].

2) 通过对训练后神经网络进行调整加入后门.文献[30]先将找到神经网络影响较大的某个模式(也称为 trojan trigger),继而通过逆向攻击获得该模型的部分训练集数据,最后使用该模式结合逆向得到的训练集进行再次训练.该方法可以不使用模型原始的训练集进行后门的植入.

在神经网络后门基础上,文献[31]提出将后门用作神经网络水印的方法,实现了对神经网络版权的保护(详见第 2 节).神经网络后门的植入相对容易,但发现和移除却非常困难.针对神经网络后门移除,文献[32]提出了一个分 2 步的过程来删除后门.该方法首先修剪网络,然后对修剪的网络进行微调,可以成功地从不同的深度神经网络实现中删除后门.文献[33]可以识别后门并重建可能的触发器,主要通过输入过滤器识别具有已知触发条件的输入、基于神经元修剪的模型修补算法和基于无学习的模型修补算法来清除后门触发器.通常情况下,对于基于后门的神经网络水印,文献[32-33]中移除方法是其重要的威胁.

神经网络后门和基于其的水印都是利用神经网络的过度参数化来学习多个任务.非法用户可以将后门用于恶意目的(例如将驾驶过程中的"停止"标志误分类为"限速"标志),但合法用户可以利用水印防止其部署模型被非法盗用.

1.4 剪枝、知识蒸馏、量化与微调

剪枝(pruning)、蒸馏(distillation)与微调(fine-tuning)是模型压缩中常用的方法[34].它们通过删除

不重要的参数和修剪神经元之间的连接来实现.这会对水印的鲁棒性造成影响,因此也常用来攻击水印(具体参见 3.1 节介绍).同时,由于微调可以对模型进行修改,因此也常用来嵌入水印(具体参见 2.4 节介绍).

模型剪枝[2]主要用来删除网络中冗余参数,降低网络复杂度,从而提高网络泛化能力,并防止过拟合[35].通常用于模型压缩,以生成更小,内存效率更高和执行效率更高的模型,且损失误差可忽略不计.通常,它通过在不更改网络结构的情况下将部分神经元归零,特别是关注于神经网络中连接较少,不重要的神经元.该方法不涉及任何重新训练,那些缺乏训练能力的攻击者可以使用模型剪枝来破坏其窃取的模型中的嵌入的水印,极有可能会影响水印提取/检测.

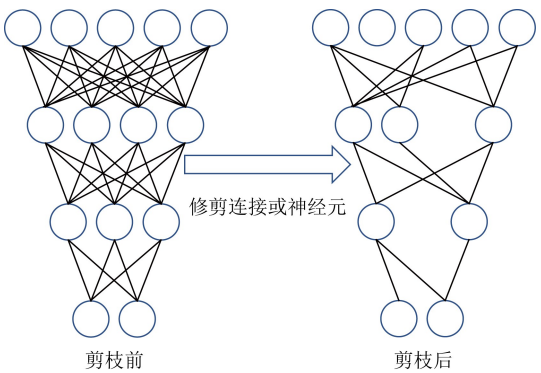


Fig. 4 Neural network pruning

图 4 神经网络剪枝

蒸馏[36],其目的是将一个大型网络的知识转移到另一个小型网络上,小型网络性能接近于大网络的性能,2 个网络可以是同构或者异构.做法是先训练一个教师网络,然后使用这个教师网络的输出和数据的真实标签去训练学生网络.通常攻击者可以不断查询大网络,得到输出结果,针对一些异常输出结果(可能是后门水印触发结果),攻击者可以剔除或者重新打标签,利用这些数据和真实标签训练小模型,最终得到清除水印的模型.然而这种攻击往往针对一些有足量训练资源的攻击者.

微调(fine-tuning)是迁移学习中常用的调参方法.迁移学习是将在解决一个问题时获得的知识应用到解决另一个不同但相关问题中,目的是在新的任务中获得更好的学习效果[37].微调主要是应用于神经网络的迁移,通过微调可以在类似任务之间进行转换,但是在无关领域之间可能会失败[38].模型微调这种攻击包括重新训练原始模型以更改模型参数并找到新的局部最小值,同时保持准确性[39].

2 神经网络水印技术

本节将针对深度神经网络水印基本概念、技术对比等方面展开介绍.

2.1 神经网络水印基本概念

神经网络水印添加过程主要是通过在模型中添加一个额外的训练目标来注入水印.如 1.1 节所述,神经网络水印加在神经网络模型上,尽可能满足表 1 中保真度、安全性、鲁棒性和隐蔽性等需求.图 5 显示了神经网络水印典型应用场景中的 3 个步骤,包括生成水印、嵌入水印和验证水印.

生成水印是指模型所有者设计特殊的水印形式,例如一个比特串或者一些经过特殊设计的训练样本,以便模型在验证水印阶段能够以某种特殊的方式验证水印的存在性.嵌入水印是指将生成的水印信息插入到神经网络模型中.在验证水印时,需要输入特定数据,然后观察模型的反馈或者输出,与预期结果进行匹配,从而验证水印的存在性.

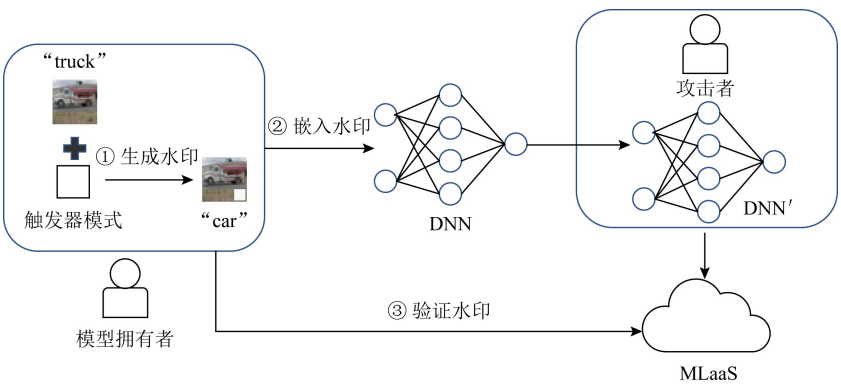


Fig. 5 Three steps to DNN watermark

图 5 DNN 水印 3 个步骤

图 5 中,首先 DNN 模型所有者为想要保护的模型生成定制的水印和预定义的水印触发条件①.在生成水印后,该框架通过训练将生成的水印嵌入到目标 DNN 中②.嵌入后,新生成的模型能够在给定水印触发条件的情况下进行水印的验证③.例如在模型被盗用并部署来提供服务时,模型所有者可以利用水印触发条件作为输入来检查服务的输出,从而进行水印的验证,向可信第三方证明该模型的所有权.

2.2 深度神经网络水印技术对比

目前市场上的服务多以深度神经网络为主,且其结构复杂,保护价值相对较高,因此,国内外研究工作多围绕深度神经网络展开,本文也主要侧重于深度神经网络水印的研究.

2017 年 Uchida 等人[21]发表基于白盒的深度神经网络水印,通过将比特串嵌入模型进行水印植入,但是由于需要知道深度神经网络模型参数等具体细节才能进行水印验证.因此,在实际应用中受限较多.此后,很多研究人员开始展开黑盒水印的研究,提高水印的实用性.

深度神经网络水印相关技术已经得到多方面的发展,本文将对深度神经网络水印技术从多个维度进行对比,如表 2 所示:

Table 2 The Comparison of DNN Watermarking Technology

表 2 DNN 水印技术对比

参考文献	年份	水印方法	分类	水印大小	嵌入方式	方法说明
文献[21]	2017	将比特数组作为水印嵌入神经网络某层的权重	白盒	多比特	训练	适用于多种神经网络,能抵御剪枝、微调的攻击
文献[31]	2018	将抽象图片作为后门水印嵌入神经网络,后门结合了加密方法保证了所有权	黑盒	0 比特	训练/微调	能应对剪枝、微调、蒸馏攻击
文献[39]	2019	根据所有者的签名生成水印密钥对	黑盒	多比特	微调	能够抵御剪枝、微调、水印覆盖攻击
文献[40]	2019	将所有者的水印嵌入在模型的不同层中获得的数据抽象(特征输出)的概率密度函数中	白盒/黑盒	多比特	训练	能够抵御多种移除和转换攻击,包括模型压缩、模型微调和水印覆盖
文献[41]	2020	用类似 GAN 的方法,使得带水印的参数相似于不带水印的网络	白盒	多比特	训练	可以抵抗属性推断攻击和水印移除
文献[42]	2018	将指纹嵌入到训练好的模型的权重概率密度函数中	白盒	多比特	训练	能够应对指纹勾结、剪枝、微调攻击
文献[43]	2018	提供了基于文本、噪声、不相关图片作为触发器的后门水印方式	黑盒	0 比特	训练/微调	能应对剪枝、微调、蒸馏攻击
文献[44]	2018	提出了一种将由比特数组生成的特定信息作为触发器的后门水印	黑盒	多比特	微调	能够抵御模型篡改;水印具有隐蔽性,攻击者很难发现水印,并恶意声称所有权
文献[45]	2020	在训练过程中为关键样本增加新标签	黑盒	0 比特	微调	假阳性率低,能抵抗剪枝、微调攻击
文献[46]	2019	提出了一种指数加权的后门水印方法	黑盒	0 比特	训练	将后门水印对参数的影响施加在较大值的权重参数上(指数加权实现),保证水印在剪枝微调更具鲁棒性
文献[47]	2020	部署在模型的预测 API 中,通过改变客户端预测响应来动态地为部分查询添加水印	黑盒	0 比特	微调	能有效抵抗模型提取以及分布式提取攻击
文献[48]	2020	训练前根据图案将原始图像像素更改为正值或负值	黑盒	0 比特	微调	能抵抗剪枝微调、模型提取攻击,有效防止水印检测和移除,并且对迁移学习有很好的适应性
文献[49]	2019	提出了一种通过编码器生成盲后门水印方法	黑盒	0 比特	训练	由于这是一种盲水印,能够轻易的躲避人肉眼的检测,更具隐蔽性;能够抵御剪枝、微调的攻击
文献[50]	2020	利用对抗样本刻画决策边界,并以此作为水印	黑盒	0 比特	微调	能够抵御剪枝、微调、奇异值分解的攻击
文献[51]	2019	基于进化算法生成和优化触发集	黑盒	0 比特	训练	假阳性率低,能抵抗微调攻击
文献[52]	2019	在网络结构上插入一层数字护照层	白盒/黑盒	多比特	训练	使用配套护照才可以正常使用神经网络;护照被盗可以通过签名验证所有权

根据表2,我们按照DNN模型是否公开,分为白盒水印和黑盒水印2类进行介绍和对比分析.白盒水印和黑盒水印主要是根据在水印插入和验证时是否需要获取模型本身进行划分.具体地,白盒水印是指需要获取模型相关参数;黑盒水印是指水印的执行过程不需要访问模型本身,该方法主要通过机器学习服务中API对黑盒水印进行提取测试[20].

2.3 白盒水印

白盒水印是将生成的水印信息嵌入到DNN模型参数中[21,40-41],然后从模型中提取标记进行验证.

文献[21]是较早提出的水印方案,该方案通过显式的水印来保护DNN模型的知识产权.作者将水印解释为T位的字符串$\{0,1\}^T$,为了将其包含在模型中,使用了包含参数正则化器(parameter regularizer)的组合损失函数.该正则化器在某些模型参数上施加统计偏差,以表示水印.与文献[21]将水印嵌入到模型的静态内容不同,文献[40]提出将字符串嵌入到不同网络层的概率密度函数(pdf)中.由于水印信息被嵌入DNN的动态内容中.生成水印同时依赖于数据和模型,即只能通过将特定的输入数据传递给模型来触发,使得水印更加灵活且不易被检测.然而,这2种方法都会导致水印权重的分布发生变化.文献[41]基于有水印和无水印模型权重的不同,提出了一种属性推理攻击,能够检测出文献[21]和文献[40]提出2种方法的水印.在此基础上,文献[41]又提出了一种基于生成对抗网络(GAN)的白盒水印方法,使得最终的参数分布与无水印版本无法区分,因此很难被检测出来.3个方法都是针对模型而非某个用户添加水印,因此无法判断以用户为单位分发的模型是否泄漏版权.针对多用户的情况,文献[42]提出水印不应只取决于模型本身,而是由模型和用户共同确定.同时,提出了一种端到端的共谋安全水印框架.它为每个用户分配一个唯一的二进制代码矢量(也称为指纹),在保证准确性的前提下,将指纹信息嵌入DNN权重的概率分布中,能够有效地跟踪每个用户模型的使用情况.此外,尽管一些方案在抵御水印去除攻击时保持了较高的鲁棒性,但是未解决敌手的欺诈性所有权声明.即如果敌手使用被称作歧义攻击的方式,为模型伪造额外的水印,此时原始水印和新增水印都可以被验证程序检测,则无法判断模型所有权的归属.

2.4 黑盒水印

由于白盒水印在验证环节需要模型拥有者知道可疑模型的内部细节(如结构、参数等),才能提取其完整水印,并与嵌入的水印对比位错误来完成验证,因此适用性受到了很大限制.因此,有学者提出了以黑盒的方式为模型添加水印的方法,从而在无需知晓模型参数等细节的情况下进行水印的验证.

文献[31]提出了在黑盒场景下,将后门技术应用于所有权保护的水印方法.通过在模型的训练或微调的过程中,模型拥有者将抽象样本(与训练样本具有不同分布)作为后门样本加入模型训练集,如此保证模型在原始任务上的准确率,抽象样本作为后门水印进行所有权的认证.与此同时,通过commit加密方法生成的密钥进一步保证了水印在公开场景下验证所有权.相比于文献[31],文献[43]提出了基于文本触发器、噪声、不相关样本的后门样本方法,通过特定触发器来触发后门水印验证所有权.在此基础上,为了将后门技术应用于嵌入式系统,文献[44]提出了一种保护嵌入式系统神经网络模型的所有权的水印方法,将由比特数组生成的特定信息(mask)作为触发器设计后门水印.文献[31,43]所提出的方法只能实现0比特(zero-bit)水印嵌入,而白盒水印能够嵌入更多位的信息.为提升水印的容量,文献[39]提出了多比特(multi-bit)的黑盒水印方法,作者提出了一种模型相关的编码方案,将所有者的二进制签名包含在输出激活中作为模型的水印.

上述基于模型后门的水印方案中,分类带有错误标签的关键样本,会不可避免的对模型在原始任务上的决策边界产生影响.文献[45]提出了为关键样本新增标签的方法,尝试设计不会扭曲原始决策边界的水印.考虑到部分模型水印应对剪枝微调并不具备很强的鲁棒性,文献[46]提出了一种指数加权的后门水印方法,将后门水印对参数的影响施加在较大值的权重参数上(指数加权实现),保证水印在剪枝微调更具鲁棒性.同样,为了增加应对蒸馏攻击的鲁棒性,文献[47]提出的水印部署在模型的预测API中,通过改变客户端的预测响应来动态地为部分查询添加水印.

但是,先前水印[31,43]中的后门样本的空间很大,由于神经网络能够接受增量训练和微调,攻击者可以基于此特性构造对抗样本,适当嵌入自己的水印来声明此神经网络的所有权,但这是一种伪造的所有权.文献[48]提出了一种空嵌入(null embedding)的方法,将水印包含在模型的初始训练中.由于空嵌入不依赖于增量训练,只能在初始化时期训练为模型,因此对手很难再嵌入自己盗版水印.文献[31,40,43-44]中后门水印和干净样本的数据分布具有很大的差异,隐蔽性较弱.因此,文献[49]提出了一种

通过编码器生成盲(blind)的后门水印方法,能够躲避人肉眼或部分检测器的检测,更具隐蔽性.

除了应用后门水印来做黑盒情境下所有权验证,文献[50]提出对抗样本作为水印的方法.然而由于对抗样本的迁移性[31],在没有此水印的模型上,对抗样本也可能会以较高概率被判别错误,被误识别为水印,从而发生误报.

由此可见,当前 DNN 水印方法中,白盒水印需要对模型参数进行访问和验证,应用中受限较多.黑盒水印只需要通过 API 访问相关服务来验证深度神经网络的所有权[53],目前应用较为广泛.

3 神经网络水印攻击方法

攻击者对于机器学习水印的攻击多是围绕着表 1 中的鲁棒性、隐蔽性和安全性 3 个特性展开,但同时攻击应保持原有模型的保真度(攻击者不愿意失去原有模型的准确性,因其也需要使用去除水印后的模型进行服务).分别对其展开论述,如表 3 所示.为了方便描述,形式化地,我们定义原有未含水印的模型是 M,水印为 W,模型 M 的拥有者 O 将 W 嵌入 M 得到 M_W,攻击者试图伪造 W.

Table 3 The Comparison of DNN Attack Methods

表 3 DNN 水印攻击方法对比

相关工作	年份	攻击目标	攻击方法	条件	方法说明
文献[53]	2019	鲁棒性	重训练	有限的训练数据	查询水印模型以标记来自水印模型的公共已知域的任意输入的结果
文献[54]	2020	鲁棒性	微调	有限的训练数据	通过使用弹性重量固结(EWC)方法,在非标记数据增强(AU)的基础上,FERIT 框架可以有效地去除水印
文献[55]	2019	鲁棒性	蒸馏	有限的训练数据	冗余信息(包含水印,但对蒸馏目标没有贡献)将丢失
文献[56]	2019	隐蔽性	覆盖	模型参数	通过检测水印的存在并推导其嵌入长度,通过覆盖水印来去除水印
文献[57]	2020	隐蔽性	重训练	有限的训练数据	去除 DNN 层中可能有水印的神经元或通道
文献[58]	2019	鲁棒性	微调	有限的训练数据	利用预先训练好的模型对未标记样本进行标注,并对微调后的训练数据进行扩充,去除水印
文献[59]	2019	其他	投票/检测	高质量的模型	投票机制/区分触发器实例和干净输入
文献[60]	2020	鲁棒性	微调	不需要知道水印技术和训练样本	首先对水印样本进行预处理,实现水印样本的识别;然后采用微调策略提高模型在正常样本上的可用性
文献[61]	2020	隐蔽性	微调	有限的训练数据	结合数据扩充和分布距离优化识别潜在的水印数据

3.1 对鲁棒性的攻击

鲁棒性(robustness)指水印不易去除.如强行去除后会损坏模型的保真度.攻击者的目标是在保持一定保真度的情况下,使得模型水印失去其作用,即模型拥有者 O 无法确认模型中水印的存在.一个简单的方法去除模型中的水印即重训练一个新的模型,但该方法需要大量的训练数据和计算能力.若攻击者有此能力,无需再剽窃其他人的水印.因此攻击者尝试使用少量的训练样本甚至不使用训练样本进行水印的去除.

常用的技术包括微调(fine-tuning)[32,54]、剪枝(pruning)[32,43,46]和蒸馏(distill)[47,55]等.这几类方法的基本思想是通过对含有水印的模型 M_W 进行部分调整,尝试去除水印 W;但考虑到模型本身的保真度可能下降,因此仍然需要拿出部分带有标签的数据(如训练集或者测试集中的数据)对模型进行训练,从而尽量使模型恢复到原有的准确性.具体来说:

1) 微调.微调方法不对原始模型的结构进行操作,直接采用带有标签的数据进行模型的重训练.考虑到模型本身包括 2 部分任务(水印任务 W 和原始任务 T),而重训练仅使用 T 相关的数据进行训练.因此随着训练的进行,原始 W 会在模型中逐渐遗忘[30],但 T 会尽量保持不变或增强.从而使得模型中的 W 逐渐去除.这类方法被对水印[31]具有一定效果.

2) 剪枝.剪枝方法通过改变模型结构,从而迫使原始模型“遗忘”部分已有的任务(包括水印任务 W 和原始任务 T).但同时使用训练数据进行任务 T 的增强.改变模型结构的常用方法包括剪神经元和剪边.前者直接去除神经元(通常选择任务 T 中激活值最小的神经元[32]);后者对神经元间的边进行修建[62],通常也将神经网络中的权重参数的绝对值小的参数(被认为是对任务 T 不重要的参数)值置 0,修剪连接.这类方法对部分类型的 W 会有一定效果.但事实上,由于神经网络的难以解释性,任务 W 和 T

激活的神经元很难区分,因此文献[32,62]等剪枝方法经常对任务 T 具有较大影响.

3) 蒸馏.蒸馏方法常用于将大型网络使用小型网络进行功能性替代(如选择不同的网络或者更为简化的网络)[55],在替代过程中保持原始任务 T,但不对 W 进行保存,因此较大概率能够去除 T.这类方法可以看作是剪枝的衍生,但可能需要更多地训练数据集,否则可能对原始模型 T 的准确性造成影响.

4) 其他方法.现有部分方法将微调、剪枝、蒸馏 3 种方法进行结合,如 fine-pruning[32] 方法,融合了剪枝和微调,先进行神经元的剪枝,然后进行微调.也有部分方法进行针对性的攻击,如知晓神经网络 M 中某一层含有水印[21],攻击者可对这层进行重新初始化再训练.

3.2 对隐蔽性的攻击

隐蔽性(stealthiness)指水印隐藏在模型中,不易被发现,通过隐蔽性达到不易去除的效果.从攻击者角度,可尝试发现水印,从而移除水印或者宣称对水印的所有权.隐蔽性水印的设计被用来防止此类攻击.

目前有 2 类方法可以进行隐蔽性的攻击:1)发现模型水印(尤其是后门类水印)的触发条件异常;2)发现水印中模型参数的异常.以图片分类任务为例,前者的任务是发现带有触发图案的图片与其他正常图片分布上的区别.例如文献[31]中提到可以使用抽象图案作为触发图片,易被攻击者发现与其他图片的区别,从而进行过滤,导致该触发图片无法有效对黑盒状态下的模型进行测试.文献[43]虽然将触发图案贴在正常的图片中或者使用分布不同的图案或者噪点图案等,但仍然可以被细致的分析发现其与正常图片的不同.文献[49]设计盲水印,使用编码器和判别器设计与正常图片类似且分布也类似的触发图片,使其难以被检测.

后者尝试在模型中发现水印部分相关的参数,例如文献[56]通过分析 DNN 模型的参数分布,发现水印及其长度,可以将水印去除并加入新的水印.文献[57]中,通过重构水印,对 DNN 模型中水印图片与非水印图片学习中各神经元及连接的进行比较,识别水印图片关键的神经元及连接并进行删除,之后对模型进行重新训练,实现对水印删除.

3.3 对安全性的攻击

根据 1.1 节定义,安全性(security)指水印本身不易伪造,否则攻击者也能伪造水印从而宣称对嵌入水印模型的所有权.这里有 2 类方法:1)攻击者尝试发现 W 的构造,从而宣称拥有 W,这类攻击我们已在 3.2 节论述,主要讲述如何发现 W;2)攻击者再次嵌入类似的水印 W' 到 M_{W} 中,得到含 2 个水印的模型 $M_{W+W'}$,从而宣称对 M 的所有权.

对于直接改变神经网络参数的水印,如文献[21]所述,攻击者可以采用同样的方法在其中植入水印.对于通过后门类植入的水印,如文献[31,40,43-44,46],攻击者仍然可以使用同样的方法加入后门水印.考虑到神经网络参数数量巨大,新植入的水印很难正好破坏掉原有水印的参数信息,导致 2 个水印会同时存在于模型上.此时攻击者 A 与原始模型拥有者 O 难以区分.例如文献[49]提出的盲水印,人的肉眼或部分检测器难以发现其水印触发样本与正常样本的区别.攻击者可以使用对抗样本作为某种触发器(与正常样本非常接近),使模型作出某种预期的行为(如分类异常),从而模仿模型的拥有者,宣称对模型的所有权.

3.4 其他方法

除上述 3 类攻击外,还有部分其他方法用于对水印的攻击.文献[59]提出 2 种方案,针对水印的查询,设计方案尝试输出正常的结果(非攻击者预期的水印结果):1)通过投票机制(ensemble attack),改变输出,躲避水印检测;2)通过区分触发水印的实例和正常数据(detector attack),躲避对水印检测.即使在水印难以删除的情况下,恶意攻击者仍可以逃避合法所有者对版权侵权的验证,从而避免了模型盗窃的发现.

4 未来方向与挑战

从目前来看,机器学习模型水印技术还处于发展前期,在理论上和实际使用过程中并不完善.攻击者仍然有多种方法能够对已有的保护方法进行攻击.在未来的研究过程中,主要有 5 个方面值得探索:

1) 更为鲁棒的神经网络模型水印.目前的水印方法,尤其是黑盒方法,还受限于模型的微调等攻击.未来的模型水印应加强该方面的保护,使得模型能够对抗传统攻击,尤其是较为简单的传统攻击.

2) 水印应减少对原始模型的影响.目前的水印对原始模型具有一定影响,例如会使得原始模型任务的准确性降低.虽然部分方法在一定的测试集上能够做到准确性降低不明显,但难免会影响模型的原始任务.由于模型的不可解释性,也难以从理论对模型的影响进行刻画.对比传统的书籍水印,其不会

降低书籍的阅读效果.因此,如何使模型水印在不降低原始任务的基础上完成水印任务,是未来的重要方向之一.

3) 公开的水印.传统的书籍水印,其水印公开,且易于被用户判断识别.但现有的神经网络模型水印,其鲁棒性依赖于水印的隐蔽性.一旦被公开,则其易被去除.未来的研究如能建立可公开(无需隐藏)的水印,则将对水印的可验证性提供较高的支持.

4) 水印的理论证明.目前的水印方法缺乏理论支持,多是在原有模型上做添加任务、特征等处理,但无法从理论上证明其各种安全特性.未来的研究如能在理论上取得进展,将会推进水印技术的鲁棒性等各类特性的安全保障能力.

5) 多样的水印.目前的水印方法相对单一,多是采用直接添加特征或者模型后门方式进行加入.如能探索更加多样的水印,并探索其组合,将会对未来的神经网络水印算法的鲁棒性和对攻击的防御能力起到帮助.

总之,更为强大的攻击总是伴随着防御方法的进步,他们相辅相成、共同成长.因此在未来的研究过程中,2 方面的研究将会同步发展.在理论上的完善,将会促进神经网络模型的进步.

5 总　　结

随着人工智能技术的广泛应用,神经网络模型的应用越来越广泛.然而,神经网络模型的训练需要大量专业知识、数据和计算资源,因此,神经网络模型已经成为一种重要的资产,并有很多厂商提供相应的服务,用户只需要远程访问 API 接口就能方便使用相关模型进行学习.与此同时,神经网络模型的恶意使用或非法传播也促使其版权急需保护,神经网络水印可以有效解决该问题.目前,神经网络水印技术主要包括白盒水印和黑盒水印,白盒水印需要对模型参数进行访问和验证,实际使用中受限较多.黑盒水印只需要远程访问 API 进行验证即可,应用更广泛.但是,当前黑盒水印大多使用后门等技术进行水印插入,当遇到神经网络压缩或迁移等变化时,容易被移除.因此,很多研究人员针对神经网络水印进行攻击,对其鲁棒性、安全性和隐蔽性进行研究,对于更好地提高水印保护能力有着重要的作用.最后,对未来方向和面临的挑战进行探讨,希望对神经网络水印未来的发展提供一些思路.

参 考 文 献

[1] Schyndel R, Tirkel A, Osborne C. A digital watermark [C] //Proc of the 1st Int Conf on Image Processing. Los Alamitos, CA: IEEE Computer Society, 1994: 86-90

[2] Chen Mingqi, Niu Xinxin, Yang Yixian. The research developments and applications of digital watermarking [J]. Journal on Communications, 2001, 22(5): 71-79 (in Chinese)
(陈明奇, 钮心忻, 杨义先. 数字水印的研究进展和应用[J]. 通信学报, 2001, 22(5): 71-79)

[3] Hartung F, Kutter M. Multimedia watermarking techniques [J]. Proceedings of the IEEE, 1999, 87(7): 1079-1107

[4] Yang Yixian, Niu Xinxin. Review of multi-media information camouflage [J]. Journal on Communications, 2002, 23(5): 32-38 (in Chinese)
(杨义先, 钮心忻. 多媒体信息伪装综论[J].通信学报, 2002, 23(5): 32-38)

[5] Chen Mingqi, Niu Xinxin, Yang Yixian. Attack methods of digital watermarking [J]. Journal of Electronics and Information Technology, 2001, 23(7): 705-711 (in Chinese)
(陈明奇, 钮心忻, 杨义先. 数字水印的攻击方法[J]. 电子与信息学报, 2001, 23(7): 705-711)

[6] Cox I J, Kilian J, Leighton F T, et al. Secure spread spectrum watermarking for multimedia [J]. IEEE Transactions on Image Processing, 1997, 6(12): 1673-1687

[7] Shun Shenghe, Lu Zheming, Niu Xiamu. Digital Watermarking Technology and Its Application [M]. Beijing, Science Press, 2004 (in Chinese)
(孙圣和, 陆哲明, 牛夏牧. 数字水印技术及应用[M]. 北京: 科学出版社, 2004)

[8] Chen B, Wornell G W. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding [J]. IEEE Transactions on Information Theory, 2001, 47(4): 1423-1443

[9] Tian Zhenzhou, Liu Ting, Zheng Qinghua, et al. Software plagiarism detection: A survey [J]. Journal of Cyber Security, 2016, 1(3): 52-76 (in Chinese)
(田振洲, 刘烃, 郑庆华, 等. 软件抄袭检测研究综述[J]. 信息安全学报, 2016, 1(3): 52-76)

[10] Zhang Lihe, Yang Yixian, Niu Xinxin, et al. A survey on software watermarking [J]. Journal of Software, 2003, 4(2), 268-277 (in Chinese)
(张立和, 杨义先, 钮心忻, 等. 软件水印综述[J].软件学报, 2003, 14(2): 268-277)

[11] Hamilton J, Danicic S. A survey of static software watermarking [C] //Proc of 2011 World Congress on Internet Security. Piscataway, NJ: IEEE, 2011: 100-107

[12] Monden A, Iida H, Matsumoto K, et al. A practical method for watermarking Java programs [C] //Proc of the 24th Conf on Computer Software and Applications. Los Alamitos, CA: IEEE Computer Society, 2000: 191-197

[13] Venkatesan R, Vazirani V, Sinha S. A graph theoretic approach to software watermarking [C] //Proc of Int Workshop on Information Hiding. Berlin: Springer, 2001: 157-168

[14] Cousot P, Cousot R. An abstract interpretation-based framework for software watermarking [C] //Proc of the ACM SIGPLAN-SIGACT Conf on Principles of Programming Languages. New York: ACM, 2004: 173-185

[15] Nagra J, Thomborson C. Threading software watermarks [C] //Proc of the 6th Int Workshop on Information Hiding. Berlin: Springer, 2004: 208-223

[16] Collberg C, Huntwork A, Carter E, et al. More on graph theoretic software watermarks: Implementation, analysis, and attacks [J]. Information and Software Technology, 2009, 51(1): 56-67

[17] Collberg C, Carter E, Debray S, et al. Dynamic path-based software watermarking [J]. ACM SIGPLAN Notices, 2004, 39(6): 107-118

[18] Alpaydin E. Introduction to Machine Learning [M]. Cambridge, MA: MIT Press, 2014

[19] Ribeiro M, Grolinger K, Capretz M A. MLaaS: Machine learning as a service [C] //Proc of the 14th Int Conf on Machine Learning and Applications. Piscataway, NJ: IEEE, 2015: 896-902

[20] Boenisch F. A survey on model watermarking neural networks [EB/OL]. [2020-09-25]. https://arxiv. org/abs/2009.12153

[21] Uchida Y, Nagai Y, Sakazawa S, et al. Embedding watermarks into deep neural networks [C] //Proc of the 2017 ACM on Int Conf on Multimedia Retrieval. New York: ACM, 2017: 269-277

[22] Skripniuk V, Yu Ning, Abdelnabi S, et al. Black-box watermarking for generative adversarial networks [EB/OL]. [2020-08-03]. https://arxiv.org/abs/2007.08457v1

[23] Abdelnabi S, Fritz M. Adversarial watermarking transformer: Towards tracing text provenance with data hiding [EB/OL]. [2020-09-24]. https://arxiv. org/abs/2009.03015

[24] Baidu Wikipedia. Neural network [OL]. [2020-11-03]. https://baike.baidu.com/item/神经网络/174248? fr=aladdin (in Chinese)
(Baidu 百科. 神经网络[OL]. [2020-11-03]. https://baike.baidu.com/item/神经网络/174248? fr=aladdin)

[25] Mitchell T. Machine Learning [M]. Translated by Zeng Huajun, Zhang Yinkui, et al. Beijing: China Machine Press, 1997 (in Chinese)
(Mitchell T. 机器学习[M]. 曾华军, 张银奎, 等, 译. 北京: 机械工业出版社, 2008)

[26] He Yingzhe, Hu Xingbo, He Jingwen, et al. Privacy and security issues in machine learning systems: A survey [J]. Journal of Computer Research and Developmen, 2019, 56(10): 2049-2070 (in Chinese)
(何英哲, 胡兴波, 何锦雯, 等. 机器学习系统的隐私和安全问题综述[J]. 计算机研究与发展, 2019, 56(10): 2049-2070)

[27] Gu Tianyu, Gavitt B D, Garg S. Badnets: Identifying vulnerabilities in the machine learning model supply chain [EB/OL]. [2020-10-22]. https://arxiv.org/abs/1708.06733

[28] Chen Xinyun, Liu Chang, Li Bo, et al. Targeted backdoor attacks on deep learning systems using data poisoning [EB/OL]. [2020-10-22]. https://arxiv.org/abs/1712.05526

[29] Yao Yuanshun, Li Huiying, Zheng Haitao, et al. Latent backdoor attacks on deep neural networks [C] //Proc of the 2019 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2019: 2041-2055

[30] Liu Yingqi, Ma Shiqing, Aafer Y, et al. Trojaning attack on neural networks [C] //Proc of Network and Distributed Systems Security Symp. Reston, VA: ISOC, 2018: 1-15

[31] Adi Y, Baum C, Cisse M, et al. Turning your weakness into a strength: Watermarking deep neural networks by backdooring [C] //Proc of the 27th USENIX Security Symp. Berkeley, CA: USENIX Association, 2018: 1615-1631

[32] Liu Kang, Gavitt B D, Garg S. Fine-pruning: Defending against backdooring attacks on deep neural networks [C] //Proc of the Int Symp on Research in Attacks, Intrusions, and Defenses. Berlin: Springer, 2018: 273-294

[33] Wang Bolun, Yao Yuanshun, Shan S, et al. Neural cleanse: Identifying and mitigating backdoor attacks in neural network [C] //Proc of the 2019 IEEE Conf on Security and Privacy. Piscataway, NJ: IEEE, 2019: 707-723

[34] Lai Yejing, Hao Shanfeng, Huang Dingjiang. Methods and progress in deep neural network model compression [J]. Journal of East China Normal University: Natural Science, 2020(5): 68-82 (in Chinese)
(赖叶静, 郝珊锋, 黄定江. 深度神经网络模型压缩方法与进展[J]. 华东师范大学学报: 自然科学版, 2020(5): 68-82)

[35] Li Jiangyun, Zhao Yikai, Xue Zhuoer, et al. A survey of model compression for deep neural networks [J]. Chinese Journal of Engineering, 2019, 41(10): 1229-1239 (in Chinese)
(李江昀, 赵义凯, 薛卓尔, 等. 深度神经网络模型压缩综述[J]. 工程科学学报, 2019, 41(10): 1229-1239)

[36] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network [EB/OL]. [2020-10-26]. https://arxiv.org/abs/1503.02531v1

[37] Pan S J, Yang Qiang. A survey on transfer learning [J]. IEEE Transactions on Knowledges and Data Engineering, 2010, 22(10): 1345-1359

[38] Howard J, Ruder S. Universal language model fine-tuning for text classification [C] //Proc of the 56th Annual Meeting of the ACL. Stroudsburg, PA: ACL, 2018: 328-339

[39] Chen Huili, Rouhani B D, Koushanfar F. BlackMarks: Blackbox multibit watermarking for deep neural networks [EB/OL]. [2020-03-31]. https://arxiv.org/abs/1904.00344v1

[40] Rouhani B D, Chen Huili, Koushanfar F. Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks [C] //Proc of the 24th Int Conf on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2019: 485-497

[41] Wang Tianhao, Florian K. Robust and undetectable white-box watermarks for deep neural networks [EB/OL]. [2020-03-28]. https://arxiv.org/abs/1910.14268v1

[42] Chen Huili, Rohani B D, Koushanfar F. DeepMarks: A digital fingerprinting framework for deep neural networks [EB/OL]. [2020-04-10]. https://arxiv.org/abs/1804.03648v1

[43] Zhang Jialong, Gu Zhongshu, Jang Jiyong, et al. Protecting intellectual property of deep neural networks with watermarking [C] //Proc of the 2018 on Asia Conf on Computer and Communications Security. New York: ACM, 2018: 159-172

[44] Guo Jia, Potkonjak M. Watermarking deep neural networks for embedded systems [C] //Proc of the Int Conf on Computer Aided Design. Piscataway, NJ: IEEE, 2018: 1-8

[45] Zhong Qi, Zhang Leo Yu, Zhang Jun, et al. Protecting IP of deep neural networks with watermarking: A new label helps [C] //Proc of Pacific-Asia Conf on Knowledge Discovery and Data Mining. Berlin: Springer, 2020: 462-474

[46] Namba R, Sakuma J. Robust watermarking of neural network with exponential weighting [C] //Proc of the 2019 ACM Asia Conf on Computer and Communications Security. New York: ACM, 2019: 228-240

[47] Szyller S, Atli B G, Marchal S, et al. DAWN: Dynamic adversarial watermarking of neural networks [EB/OL]. [2020-06-18]. https://arxiv.org/abs/1906.00830v4

[48] Li Huiying, Wenger E, Zhao B Y, et al. Piracy resistant watermarks for deep neural networks [EB/OL]. [2020-08-19]. https://arxiv.org/abs/1910.01226

[49] Li Zheng, Hu Chengyu, Zhang Yang, et al. How to prove your model belongs to you: A blind-watermark based framework to protect intellectual property of DNN [C] //Proc of the 35th Annual Computer Security Applications Conf. Piscataway, NJ: IEEE, 2019: 126-137

[50] Merrer E L, Perez P, Trédan G. Adversarial frontier stitching for remote neural network watermarking [J]. Neural Computing and Applications, 2020, 32: 9233-9244

[51] Jia Guo, Miodrag P. Evolutionary trigger set generation for DNN black-box watermarking [EB/OL]. [2020-06-11]. https://arxiv.org/abs/1906.04411v1

[52] Fan Lixin, Ng W K, Chan C S. Rethinking deep neural network ownership verification embedding passports to defeat ambiguity attacks [C] //Proc of the 33rd Conf on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2019: 1-10

[53] Shafeinejad M, Wang Jiaqi, Lukas N, et al. On the robustness of the backdoor-based watermarking in deep neural networks [EB/OL]. [2019-12-18]. https://arxiv.org/abs/1906.07745v1

[54] Chen Xinyun, Wang Wenxiao, Bender C, et al. REFIT: A unified watermark removal framework for deep learning systems with limited data [EB/OL]. [2020-01-22]. https://arxiv.org/abs/1911.07205v1

[55] Yang Ziqi, Dang Hung, Chang E C. Effectiveness of distillation attack and countermeasure on neural network watermarking [EB/OL]. [2019-08-14]. https://arxiv. org/abs/1906.06046v1

[56] Wang Tianhao, Kerschbaum F. Attacks on digital watermarks for deep neural networks [C] //Proc of the IEEE Int Conf on Acoustics, Speech and Signal Processing. Piscataway, NJ: IEEE, 2019: 2622-2626

[57] Aiken W, Kim H, Woo S. Neural network laundering: Removing black-box backdoor watermarks from deep neural networks [EB/OL]. [2020-10-22]. https://arxiv. org/abs/2004.11368v1

[58] Chen Xinyun, Wang Wenxiao, Ding Yiming, et al. Leveraging unlabeled data for watermark removal of deep neural networks [C/OL] //Proc of the 36th Int Conf on Machine Learning. New York: ACM, 2019 [2020-10-22]. https://ruoxijia. info/wp-content/uploads/2020/03/watermark _removal_icml19_workshop.pdf

[59] Hitaj D, Hitaj B, Mancini L. Evasion attacks against watermarking techniques found in MLaaS systems [C] //Proc of the 6th Int Conf on Software Defined Systems. Piscataway, NJ: IEEE, 2019: 55-63

[60] Guo Shangwei, Zhang Tianwei, Qiu Han, et al. The hidden vulnerability of watermarking for deep neural networks [EB/OL]. [2020-11-20]. https://arxiv.org/abs/2009.08697

[61] Liu Xuankai, Li Fengting, Wen Bihan, et al. Removing backdoor-based watermarks in neural networks with limited data [EB/OL]. [2020-09-24]. https://arxiv. org/abs/2008.00407

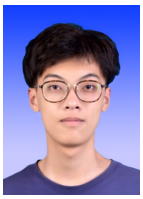
[62] Song Han, Pool J, Tran J, et al. Learning both weights and connections for efficient neural network [C] //Proc of the 28th Int Conf on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2015: 1135-1143

Zhang Yingjun, born in 1982. PhD, associate professor. Her main research interests include System security and testing.

张颖君, 1982 年生.博士,副研究员.主要研究方向为系统安全与测试.

Chen Kai, born in 1982. PhD, professor. His main research interests include software analysis and testing, smartphone and privacy.

陈　恺,1982 年生.博士,研究员.主要研究方向为软件分析与测试、智能终端与隐私.



Zhou Geng, born in 1998. Master candidate. His main research interest is cyber security.

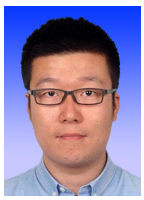
周 赓,1998 年生.硕士研究生.主要研究方向为网络空间安全.

Liu Yong, born in 1973. PhD, professor. Member of CIC. His main research interest is cyber security.

刘 勇,1973 年生.博士,研究员,CIC 会员.主要研究方向为网络空间安全.

Lü Peizhuo, born in 1998. PhD candidate. His main research interests include AI security and adversarial attack.

吕培卓,1998 年生.博士研究生.主要研究方向为人工智能安全和对抗攻击.



Huang Liang, born in 1984. PhD, assistant professor. His main research interest is cyber security.

黄 亮,1984 年生.博士,助理研究员.主要研究方向为网络空间安全.

《计算机研究与发展》征订启事

《计算机研究与发展》(Journal of Computer Research and Development)是中国科学院计算技术研究所和中国计算机学会联合主办、科学出版社出版的学术性刊物,中国计算机学会会刊.主要刊登计算机科学技术领域高水平的学术论文、最新科研成果和重大应用成果.读者对象为从事计算机研究与开发的研究人员、工程技术人员、各大专院校计算机相关专业的师生以及高新企业研发人员等.

《计算机研究与发展》于 1958 年创刊,是我国第一个计算机刊物,现已成为我国计算机领域权威性的学术期刊之一.并历次被评为我国计算机类核心期刊,多次被评为“中国百种杰出学术期刊”.此外,还被《中国学术期刊文摘》、《中国科学引文索引》、“中国科学引文数据库”、“中国科技论文统计源数据库”、美国工程索引(Ei)检索系统、日本《科学技术文献速报》、俄罗斯《文摘杂志》、英国《科学文摘》(SA)等国内外重要检索机构收录.

国内邮发代号:2-654;国外发行代号:M603

国内统一连续出版物号:CN11-1777/TP

国际标准连续出版物号:ISSN1000-1239

联系方式:

100190 北京中关村科学院南路 6 号《计算机研究与发展》编辑部

电话:+86(10)62620696(兼传真);+86(10)62600350

Email:crad@ict.ac.cn

http://crad.ict.ac.cn