计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2024,33(4):1−12 [doi: 10.15888/j.cnki.csa.009459]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于显著图的高隐蔽性模型指纹算法①

张圣尧, 潘旭东, 张　谧

(复旦大学 计算机科学技术学院, 上海 200438)
通信作者: 张圣尧, E-mail: shengyaozhang21@m.fudan.edu.cn

摘　要: 在核心任务场景下训练深度神经网络 (DNN) 需要越来越多的算力资源, 这刺激了基于云端预测 API 接口的模型的窃取与盗用, 同时也违反了模型所有者的知识产权. 为了追踪公开的非法模型副本, 深度神经网络的模型指纹技术为希望保持模型完整性的模型所有者提供了一种强大的版权验证方案. 然而, 现有的模型指纹方案主要基于输出层面的内在痕迹 (例如: 特定输入样本下的错误预测行为) , 这导致在模型指纹验证阶段缺乏隐蔽性. 本文基于模型预测时的显著图 (saliency map) 痕迹, 提出了一种全新的任意下游任务通用的模型指纹方案. 本文的方案提出了受约束的显著图操控目标, 构建标签不变和自然的指纹样本, 显著提高了模型指纹的隐蔽性. 根据对 3 种典型任务场景下全面的评估结果, 本文提出的方法被证明能够显著地增强现有方案的指纹版权验证的效果, 同时保持高度的模型指纹隐蔽性.

关键词: 模型指纹; 模型产权保护; 显著图; 隐蔽性

引用格式: 张圣尧,潘旭东,张谧.基于显著图的高隐蔽性模型指纹算法.计算机系统应用,2024,33(4):1–12. http://www.c-s-a.org.cn/1003-3254/9459.html

High-stealthiness Model Fingerprint Algorithm Based on Saliency Map

ZHANG Sheng-Yao, PAN Xu-Dong, ZHANG Mi

(School of Computer Science, Fudan University, Shanghai 200438, China)

Abstract: Training of deep neural networks (DNN) in mission-critical scenarios involves increasingly more resources, which stimulates model stealing from prediction API at the cloud and violates the intellectual property rights of the model owners. To trace public illegal model copies, DNN model fingerprint provides a promising copyright verification option for model owners who want to preserve the model integrity. However, existing fingerprinting schemes are mainly based on output-level traces (e.g., mis-prediction behavior on special inputs) to cause limited stealthiness during model fingerprint verification. This study proposes a novel task-agnostic fingerprinting scheme based on saliency map traces of model prediction. The proposed scheme puts forward a constrained manipulation objective of saliency maps to construct clean-label and natural fingerprint samples, thus significantly improving the stealthiness of model fingerprints. According to extensive evaluation results on three typical tasks, this scheme is proven to substantially enhance the fingerprint effectiveness of existing schemes and remain highly stealthy of model fingerprints.

Key words: model fingerprint; model copyright protection; saliency map; stealthiness

当今, 深度神经网络 (DNN) 依赖强大的计算算力和大量精心标注的数据来完成模型训练[1–3], 孕育了一系列经典且至今仍然被广泛使用的模型结构, 例如: ResNet[4], VGG[5] 和 MobileNet[6]等模型结构. 这些模型

① 基金项目: 国家自然科学基金 (61972099)
收稿时间: 2023-10-11; 修改时间: 2023-11-09; 采用时间: 2023-11-24; csa 在线出版时间: 2024-01-18
CNKI 网络首发时间: 2024-01-19

结构给各种关键任务提供了高精度的解决方案, 例如: 自动驾驶[7]、金融[8]和医疗诊断[9]等任务. 对于这些巨大的算力与数据量的投入, 不可避免地使得深度神经网络成为模型所有者的知识产权 (intellectual property) 中不可或缺的一部分. 在当今流行的机器学习即服务 (MLaaS) 的范式下, 深度神经网络模型常被部署在私有云上, 并为订阅的用户提供预测 API. 但通过一系列模型窃取与盗用方案[10–13], 窃取者可以对预测 API 背后的模型进行窃取, 并将窃取的模型公开发布. 这类行为往往会造成对所有者模型版权的侵犯. 因此, 我们需要一种针对深度学习模型的版权验证方案, 用于验证公开模型的版权归属情况.

模型指纹 (model fingerprint)[14–16]这一机制应运而生, 此技术基于目标模型的内在痕迹 (例如:决策边界) 来进行模型版权的验证. 通常模型指纹方案包含两部分: 指纹样本构建模块与指纹信息验证模块. 前者选择特定的模型内在痕迹作为指纹信息, 并构建指纹样本以刻画所选的模型内在痕迹. 后者通过指纹样本直接验证目标模型的指纹信息是否存在于嫌疑模型中, 以判断嫌疑模型是否是目标模型的非法副本. 相较于模型水印 (model watermarking)[17–20], 模型指纹无需修改模型参数, 可以保证模型的完整性[14], 且不影响模型在原任务上的表现, 更适用于对模型表现比较敏感的关键任务, 例如: 医疗[9]和自动驾驶[7]等.

在现有文献中, 几乎没有工作关注到模型指纹的隐蔽性要求并提出解决方案. 模型指纹隐蔽性是指模型指纹样本被检测出来的可能性. 本文认为, 指纹样本缺乏隐蔽性会导致版权验证结果的不精确. 当待验证版权的嫌疑模型, 检测到输入样本可能为指纹样本时, 其可以选择返回随机且混淆的指纹信息, 以破坏指纹信息验证过程. 因此, 几乎无法从嫌疑模型的预测 API 中收集任何真实的指纹信息, 这导致了模型版权验证结果高度不可信. 然而, 现有模型指纹方案主要利用输出级别的内在痕迹 (例如: 基于决策边界、特征空间等) 作为模型指纹信息, 导致所构建的指纹样本均缺乏隐蔽性. 一方面, 基于决策边界的指纹方案[14–16]通过对抗样本刻画模型决策边界的内在痕迹, 这些指纹样本的标签与真实标签存在不一致性, 很容易被对抗样本防御方案检测出来. 另一方面, 基于特征空间的指纹方案[21–23]通过非自然图像刻画模型特征空间的内在痕迹, 这些不自然的指纹样本很容易被异常检测算法检测出来.

本文提出了一种基于显著图 (saliency map)[24,25]的深度神经网络模型指纹方案. 显著图作为一种模型可解释性方案, 解释了模型处理样本时的重点关注位置, 是模型的内在痕迹之一. 本文首次通过显著图作为模型的内在痕迹来增强模型指纹的隐蔽性. 本文从两个角度增强了模型指纹的隐蔽性, 降低指纹样本被检测出的可能性: 一方面, 方案保持样本的标签不变性, 从而增强了基于决策边界的方案的隐蔽性; 另一方面, 方案保持指纹样本的自然性, 即确保指纹样本与原始样本之间仅存在肉眼不可见的微小扰动, 从而增强了基于特征空间的方案的隐蔽性.

基于显著图的模型指纹方案在构建指纹样本和验证模型指纹信息时, 面临以下技术挑战. 首先, 在模型指纹样本构建时, 需要同时实现准确的版权验证与指纹样本的高隐蔽性. 本文提出了一个受约束的显著图操控目标, 从显著图的角度生成指纹样本, 以描述目标模型的独特且鲁棒的内在痕迹, 同时在样本生成过程中也保持指纹样本的标签不变性和自然性. 其次, 在模型指纹信息的验证阶段, 常常仅允许以黑盒预测 API 的形式访问嫌疑模型. 然而, 现有的黑盒显著图算法[25]通常需要成千上万次的请求来计算对应样本的显著图, 这会产生巨大的验证代价. 为解决这个问题, 本文使用白盒显著图算法 Grad-CAM[24]来构建指纹样本, 并提出了一种基于遮盖的验证算法, 允许在黑盒访问下每个样本仅需两次请求便可以验证显著图.

与此同时, 大多数现有的深度神经网络模型指纹方案[14,15]通常专为图像分类任务而设计, 这限制了它们在其他下游任务上的版权保护能力. 本文提出的模型指纹方案保留了 Grad-CAM 的任务通用的特性, 因此在一系列下游任务上 (例如: 分类、回归和特征相似度) 均可实现对模型的版权保护.

总体而言, 与之前的模型指纹算法相比, 本文主要有以下两方面优势.

(1) 本文在标签不变与自然图像的限制下, 提出了基于显著图的模型指纹算法, 实验表明本文所提的算法有着更强的有效性以及更强的隐蔽性, 大大提升了版权验证结果的可靠性.

(2) 本文提出的基于显著图的模型指纹算法可以用于保护各种下游任务的视觉模型, 大大提升了算法的可应用的广度以及通用性.

本文第 1 节介绍模型指纹任务的相关工作. 第

2 节给出模型指纹的威胁模型与定义. 第 3 节介绍基于显著图的模型指纹算法. 第 4 节介绍实验设置以及实验结果. 第 5 节进行总结与展望.

1 相关工作

本节首先对当前的模型指纹算法进行分类, 并分析现有工作在指纹样本隐蔽性上的局限性. 同时, 介绍了本文模型指纹方案中指纹信息所刻画的模型内在痕迹——显著图的各类算法.

1.1 模型指纹

当前的模型指纹算法主要依赖于输出级别的内在痕迹作为指纹信息, 常被分为基于决策边界的方案以及基于特征空间的方案.

(1) 基于决策边界的模型指纹方案: 这类方案通过生成对抗样本来刻画目标模型的决策边界. Cao 等人[14]首次提出生成接近于决策边界的对抗样本, 来描述模型的决策边界痕迹. 不同于前文, Lukas 等人[15]和 Zhao 等人[26]提出基于目标模型和一组本地训练的嫌疑模型的集合, 来生成更加精确刻画目标模型决策边界痕迹的对抗样本. 与此同时, Wang 等人[16]使用 DeepFool 算法[27]更加高效地生成刻画模型决策边界的对抗样本. Li 等人[28]利用特定输入的置信度向量间的相似度来判断模型版权的归属关系. Peng 等人[29]提出使用通用对抗扰动来生成指纹样本. Sun 等人[30]提出使用生成器来生成接近与决策边界的指纹样本. 但基于决策边界的模型指纹方案通常需要改变标签, 这类样本标签与真实标签的不一致性通常导致其缺乏隐蔽性.

(2) 基于特征空间的模型指纹方案: 这类方案通过生成指纹样本来刻画目标模型的特征空间. Pan 等人[23]提出生成指纹样本对来描述含有 *ReLU* 激活函数[31]的神经网络模型的线性激活区域[32–34]. Wang 等人[21,22]基于扰动的参数空间生成指纹样本, 使其输出的激活特征保持高度的一致性. 但基于特征空间的模型指纹方案, 通常需要构建高度不自然的噪声样本才能够满足一定的特征激活空间的条件, 这类高度不自然的样本和真实图像偏差较大, 导致其缺乏隐蔽性.

最近, Li 等人[35]和 Yang 等人[36]分别在模型指纹方案中讨论了指纹样本的自然性, 这些工作仅覆盖了本文讨论的指纹样本隐蔽性中的一个要求. 同时, 其提出的指纹方案分别局限于特定的生成式任务与分类任务模型, 方案通用性低. 本文提出了基于显著图的模型指纹算法, 通过构建标签不变且自然的指纹样本, 极大提高了模型指纹的隐蔽性. 同时, 先前的工作主要局限于给分类模型提供版权保护的方案, 而本文的模型指纹算法可以被广泛应用于不同类型的下游任务.

1.2 显著图

模型的可解释性系统通过解释为何做出特定的预测, 来确保模型决策过程的可靠性. 在计算机视觉领域, 研究人员提出了基于显著图的可解释性工具. 现有的显著图方案可以大致分为基于梯度、基于特征和基于遮盖的方案. 基于梯度的显著图方案[37,38]计算模型输出相对于输入图像的梯度, 以计算特定标签下的显著图. 基于特征的显著图方案[24,39]计算某个卷积层的所有通道的特征激活值的加权平均. 这两种方案是白盒场景下的显著图方案, 其需要使用模型的内部信息以计算显著图. 而针对基于预测 API 的黑盒场景, 常使用基于遮盖的显著图方案[25], 其通过使用输入图像的随机遮盖版本请求模型并获取相应的输出来经验性地计算显著图. 然而, 基于遮盖的显著图方案通常需要上千次的请求来计算显著图, 计算效率非常低.

与此同时, 一系列工作表明显著图容易受到恶意的操控. 对抗样本[40–42]不仅可以误导目标模型的预测, 还可以篡改显著图的计算结果. 同时, 后门[43]也可以被嵌入目标模型中, 通过在原始图像上添加特定触发器, 可实现操控显著图的目标. 上述工作为操纵显著图以实现模型指纹的版权保护目标奠定了基础.

2 模型指纹定义

2.1 威胁模型

模型指纹是用于保护并验证深度神经网络的知识产权归属的方案. 在模型指纹的任务中, 通常涉及模型所有者与模型窃取者的互相博弈. 借助大量的计算资源和私有数据集, 模型所有者训练模型以解决特定的视觉任务, 并会将目标模型部署在第三方平台上 (例如: Microsoft Azure) 以获取收益. 在这种情况下, 模型窃取者可能尝试通过软件/硬件漏洞[10,12]和算法攻击[11,13]等方式盗取目标模型, 然后将盗版模型发布到公共模型存储库 (例如: HuggingFace) , 或在其他第三方平台上部署盗版模型以获得同等的收益, 这不可避免地侵犯了模型所有者的知识产权. 考虑计算资源的算力成本以及私有数据集的收集成本, 模型所有者希望能够保护自己的知识产权. 模型所有者便会使用模型指纹

技术来验证嫌疑模型是否是目标模型的盗版版本.

2.2 形式化定义

本文模型指纹主要关注图像领域深度学习模型版权保护. 考虑从联合分布 $P_{X,Y}$ 中采样到数据集 $D_{\text{ori}} = \{(x_i, y_i)\}_{i=1}^{N}$, 其中 $x_i \in X \subset \mathbb{R}^{d\times d}$, y_i 为图像标签, N 为样本数量. 形式化地, 模型指纹算法主要包含以下两个阶段.

(1) 指纹样本构建阶段: $D_{\text{ver}} \leftarrow Generate(D_{\text{ori}}, m_t)$, 即: 当模型所有者欲保护目标模型版权时, 给定原数据集 D_{ori} 以及模型内在痕迹类型 t, 构建指纹数据集 D_{ver} 以描述目标模型的内在痕迹 m_t. 本文遵循先前工作的假设[14,15,23], 假设在白盒访问目标模型的前提下, 完成指纹样本的构建.

(2) 指纹信息验证阶段: $MS \leftarrow Verify(D_{\text{ver}}, m_t)$, 即: 在发现潜在的盗版嫌疑模型后, 模型所有者可以使用模型指纹样本 D_{ver}, 验证目标模型的内在痕迹 m_t 是否存在于嫌疑模型, 输出 $MS \in [0,1]$ 表示嫌疑模型的版权归属情况. 在版权验证阶段, 本文假设可通过公共模型存储库下载该模型的白盒访问形式或通过预测 API 接口的黑盒访问形式来请求嫌疑模型完成验证.

本文提出的基于显著图的模型指纹算法基于上述两阶段设计, 具体的技术于第 3 节介绍.

2.3 后处理技术

同时, 模型窃取者对目标模型进行窃取后, 不希望真实版权的信息能够被模型指纹算法验证出来. 所以, 模型窃取者会应用一些后处理技术, 以混淆模型的参数空间, 从而逃避潜在的模型指纹版权验证[14], 同时保证模型的可用性. 根据前作的假设[14,15,23], 模型窃取者可以应用一系列后处理技术来混淆盗版目标模型的参数空间, 使得盗版目标模型的内在痕迹发生一定的改变, 以逃避模型指纹版权验证. 本文主要考虑以下后处理技术.

(1) 微调 (fine-tuning) : 为了混淆盗版目标模型, 最简单的方法是对模型的最后 K 层的参数进行微调, 同时保持前 H–K 层的参数不变.

(2) 重训 (retraining) : 为了进一步混淆盗版目标模型的内在痕迹, 重训首先会重新初始化最后 K 层的参数, 然后再对这些层进行重新的训练.

(3) 模型剪枝 (pruning) : 剪枝是一种模型压缩技术, 通常可以将模型压缩成一个体量更小的模型, 常见的两种剪枝技术是权重剪枝 (weight pruning)[44]和滤波器剪枝 (filter pruning)[45]. 权重剪枝算法[44]以单个神经元为单位, 分别将各卷积层或线性层的模型参数值较小的模型神经元按照特定的比例去除. 滤波器剪枝[45]以卷积层的通道为单位, 将模型参数均值较小的通道按照特定的比例直接去除. 在一定的剪枝比例的前提下, 此技术可以在保持模型性能稳定的前提下, 较大程度地混淆模型参数.

(4) 模型蒸馏 (distillation)[46]: 在进行模型蒸馏时, 模型窃取者使用软标签 (即: 置信度向量) 与模型预测向量, 计算均方误差损失函数或 KL 散度损失函数, 以指导模型进行通过训练学习教师模型行为. 与上述后处理技术相比, 模型蒸馏是一种更强大的参数混淆技术. 同时由于先前工作[47]指出: 在相同的模型架构上进行蒸馏可以最大可能地保留原始模型的表现, 因此本文假设模型窃取者会使用同架构模型蒸馏的后处理技术.

最后要指出的是, 模型窃取者通常不太可能同时应用多种后处理技术. 首先, 多种后处理技术会对模型的表现产生较大的负面影响. 同时, 多种后处理技术需要很大的训练成本, 若后处理技术的代价高于从头开始训练一个全新的模型的代价时, 模型窃取便失去了意义.

3 基于显著图的模型指纹算法

显著图通过揭示模型处理样本时的重点关注位置, 解释了模型的内部决策行为, 这可以被视为模型的内在痕迹. 本文基于显著图的内在痕迹来实现对深度神经网络的模型指纹版权验证.

3.1 Grad-CAM 显著图

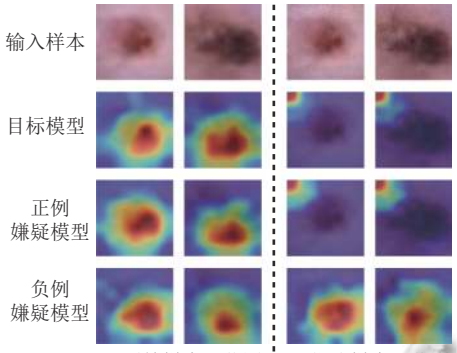
考虑到黑盒显著图方案[25]的巨大的计算代价, 本文主要将白盒显著图方案 Grad-CAM[24]作为模型指纹所刻画的模型内在痕迹. 考虑一个基于卷积神经网络 (CNN) 的图像分类模型 F. 给定输入图像 x 及其真实标签 c 的输出 $F_c(x)$, Grad-CAM 首先计算模型输出 $F_c(x)$ 相对于最后一层卷积层的每个通道特征激活值 A 的梯度信息, 来作为每个通道的重要性权重:

$$w_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial F_c(x)}{\partial A_{i,j}^k} \tag{1}$$

其中, Z 是一个归一化因子, $A_{i,j}^k$ 是位于第 k 通道的空间位置 (i,j) 上的神经元. 然后, Grad-CAM 对最后一层卷积层的每个通道特征激活值进行加权求和, 并经过 $ReLU$ 操作, 以获得真实标签 c 对应的显著图:

$$S^c = ReLU\left(\sum_k w_k^c A^k\right) \tag{2}$$

最后, Grad-CAM 通常会对显著图进行标准化, 以获得更好的可视化效果, 即: $\hat{S}^c = S^c/\|S^c\|$. 如图 1(a) 前两行所示, 如果将显著图上采样至图像相同大小时, 显著图高亮显示的区域解释了当前输入图像x被模型预测为真实标签c时最起作用的区域.



(a) 原始样本显著图 (b) 指纹样本显著图

图 1 原始样本和指纹样本的显著图对比

3.2 指纹样本生成

首先本文先进行如下定义: 当嫌疑模型确实是目标模型的盗版版本时, 本文称之为正例嫌疑模型. 相反, 当嫌疑模型与目标模型完全不相关时, 本文称之为负例嫌疑模型.

理想情况下, 如图 1(a) 前 3 行所示, 未经改动的原始样本的显著图在目标模型和正例嫌疑模型之间应该具有一定程度的一致性. 然而, 直接比较这些样本在嫌疑模型与目标模型上的显著图的一致性是不够的. 根据经验性观察, 大多数数据集 (如: ImageNet[48]) 会对图像进行预处理, 将关键物体放在图像中心. 如图 1(a) 最后一行所示, 这会导致负例嫌疑模型也会产生高亮中心区域的显著图. 所以, 直接使用未经改动的样本会使得模型指纹的版权验证结果非常不可信.

因此, 本文提出了全新的显著图操控目标, 对原始样本进行改动, 以操纵在目标模型上的显著图, 使其高亮图像的部分背景区域, 以放大正负例模型上显著图的差异. 给定一个原始训练样本x及其真实标签c, 本文通过式 (3) 高亮目标模型显著图的背景区域:

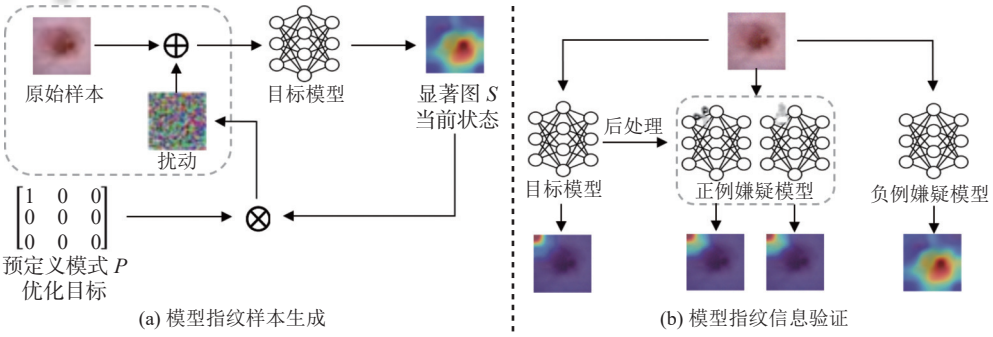
$$L_{\text{sam}} = \frac{\sum(S^c \odot (1-P))}{\sum(1-P)} - \frac{\sum(S^c \odot P)}{\sum P} \tag{3}$$

其中, S^c是未标准化的显著图, $\odot$是矩阵点乘, P是由 0 和 1 组成的目标显著图的预定义模式, 其中 0 和 1 分别标记非目标高亮以及目标高亮背景的区域. 式 (3) 第 1 项和第 2 项分别计算预定义模式P其他无关非高亮区域和目标高亮背景区域的显著图均值, 公式最终希望放大模式P的目标背景区域与其他区域间的显著图值的差异. 在本文中, 方案目标高亮左上角背景区域, 那么对于一个7×7的模式P, 本文将空间位置(0,0)设置为 1, 其他位置为 0. 经操作后的预期显著图结果如图 1(b) 所示, 由于负例嫌疑模型的显著图仍然高亮中间区域, 目标模型、正例嫌疑模型的显著图和负例嫌疑模型的显著图会存在明显不同, 从而能够实现很好的版权验证的效果.

正如第 1.1 节中提到的, 现有的模型指纹方案在隐蔽性方面存在缺陷. 为了实现隐蔽性, 本文进一步提出了受约束的显著图操控目标, 以生成高隐蔽性的指纹样本:

$$\begin{cases} \min_{\delta} L_{\text{sam}} + \lambda \cdot L_{\text{prd}}(x+\delta, F_c(x)) \\ \text{s.t. } \|\delta\|_p \leqslant \xi \end{cases} \tag{4}$$

其中, λ是超参数. 具体指纹算法生成流程如图 2(a) 所示. 假定有原始样本x, 预定义模式P, 待保护的目标模型以及随机初始化的图像扰动δ. 在算法迭代的每回合, 算法首先计算样本的显著图S, 以及对应的显著图S满足预定义模式P的程度L_{sam}. 紧接着, 结合式 (4) 中的约束条件, 算法迭代地对扰动δ进行梯度优化更新. 最终优化的扰动δ与原始样本x叠加, 完成最终的指纹样本生成.



(a) 模型指纹样本生成 (b) 模型指纹信息验证

图 2 基于显著图的模型指纹算法流程图

使用此优化过程有以下几点优势: 首先, L_{sam} 可以保证指纹样本在正负例模型上显著图差异较大, 有较好的版权验证效果. 其次, 优化过程中对扰动 δ 直接优化, 能更方便地将扰动 δ 约束在 ξ 范围内, 这个约束增强了指纹样本的自然性. 最后, 通过将原任务的目标函数 $L_{\text{prd}}(x+\delta, F_c(x))$ 整合到优化过程中, 指纹样本可以保持标签不变. 由于自然性和标签不变性均得到保证, 本文极大程度提高了指纹样本的隐蔽性.

3.3 指纹样本验证

如图 2(b) 所示, 经构建的指纹样本在正例、负例嫌疑模型上的显著图已经存在明显差距. 所以, 模型指纹信息的验证阶段需要设计能够区分这些差异的验证算法, 来判断模型版权归属情况. 具体而言, 本文主要判断在输入指纹样本时嫌疑模型的显著图是否会类似于指纹样本构建阶段的预定义模式 P. 给定 N 个构建的指纹样本组成的集合, 以及嫌疑模型的白盒/黑盒访问权限, 模型指纹的验证阶段会最终输出一个范围在 0–1 之间的匹配分数 (matching score) 来指示嫌疑模型是目标模型的盗版模型的可能性.

(1) 白盒验证: 在白盒场景中, 验证者可以直接计算嫌疑模型的显著图. 如图 3(a) 所示, 通过显著图在预定义模式 P 上平均值来计算匹配分数:

$$MS_{wb} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\sum \hat{S}_i^c \odot P}{\sum P} \geqslant t_{wb}\right) \tag{5}$$

其主要计算指纹样本集合中预定义模式 P 的显著图平均值大于显著阈值 t_{wb} 的比例. 其中, 显著阈值 t_{wb} 可被设置为目标模型上一组原始样本和指纹样本的显著图值的平均值.

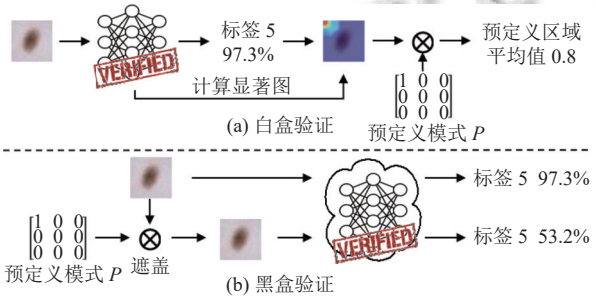


图 3 模型指纹验证阶段算法

(2) 黑盒验证: 在黑盒场景中, 验证者不被允许直接计算显著图. 受指标删除度量 (deletion metric)[25] 的启发, 本文提出了一种基于遮盖的验证算法. 如图 3(b) 所示, 该算法测量指纹样本在被预定义模式 P 掩盖前后, 模型输出 $F_c(x)$ 的概率下降的程度. 最后用于计算匹配分数:

$$MS_{bb} = \frac{1}{N}\sum_{i=1}^{N}\left(\left\|F_c(x_i^{\text{mask}}) - F_c(x_i)\right\| \geqslant t_{bb}\right) \tag{6}$$

其主要计算指纹样本集合中概率下降的幅度大于置信度降低阈值 t_{bb} 的比例, 其中, 置信度降低阈值 t_{bb} 可由在目标模型上遮盖一组原始样本来决定. 对于每个指纹样本, 本文只需分别请求遮盖样本和未遮盖样本各 1 次, 这远低于传统黑盒显著图算法的请求代价.

3.4 任务通用的模型指纹

当前的模型指纹方案[14,15,21] 仅对分类模型起到版权保护的作用, 在真实场景下存在一定局限性. 考虑到 Grad-CAM 拥有良好的任务通用的性质, 并被广泛用于给不同视觉任务提供可解释性能力, 本文进一步提出了任务通用的模型指纹算法. 为了适应不同的下游任务, 模型所有者只需要在式 (1) 中设计合理的 $F_c(x)$. 本文将第 3.2 节和第 3.3 节中提到的模型指纹方案扩展到两个全新的下游任务——线性回归和特征相似度.

(1) 线性回归: 线性回归是一种从图像中预测实数值的任务, 本文主要对基于卷积特征提取器和线性预测器的线性回归模型提供版权保护. 本文在式 (1) 中将 $F_c(x)$ 设置为 $F_c(x) = y$, 其中 y 是线性回归模型的预测输出.

(2) 特征相似度: 特征相似度是一种通过计算两个深度学习模型提取的特征之间的余弦距离来验证两个不同图像之间相似度的任务, 本文主要对特征相似度任务中的卷积特征提取器模型提供版权保护. 本文为 Grad-CAM 找到了两种合适的设计方式, 首先, 本文可以将式 (1) 中的 $F_c(x)$ 设置为特征的均方误差:

$$F_c(x) = \left\|Emb(x) - Emb(x_{\text{pair}})\right\|_2 \tag{7}$$

其中, $Emb(x)$ 是 x 的特征, x_{pair} 是 x 的配对图像. 其次, 本文也可以设置为特征间的余弦相似度:

$$F_c(x) = \frac{Emb(x)\cdot Emb(x_{\text{pair}})}{\|Emb(x)\|_2 \cdot \|Emb(x_{\text{pair}})\|_2} \tag{8}$$

本文在第 4.1 节具体介绍了线性回归与特征相似度的具体应用场景, 相关模型结构以及训练数据集.

4 实验分析

4.1 实验设置

(1) 任务、数据集与模型: 如表 1 所示, 本文主要

在 3 个视觉任务上进行实验. 对于分类任务, 本文在 DermaMNIST 数据集[49]上训练 ResNet-18[4], 来进行 7 类分类的皮肤病诊断. DermaMNIST 数据集包含 10 005 个皮肤病变图像. 本文将调整图像大小为 3×224×224, 以适应 ResNet-18 的标准输入形状. 对于线性回归任务, 本文在 FGNet 数据集上训练 VGG11 模型[5]进行年龄估计. FGNet 数据集由 1 002 个 3×224×224 的人脸图像组成, 人脸年龄在 0–69 岁之间. 对于特征相似度任务, 本文在 FaceScrub 数据集[50]上训练模型实现人脸匹配. 该数据集包含 530 名名人的 107 818 张 3×112×112 大小的人脸图像. 本文选择 MobileFaceNet 模型[6], 同时使用 ArcFace 损失函数[51]对模型进行训练.

表 1 任务、数据集和模型

任务标识	任务类型	数据集	模型
皮肤病	分类	DermaMNIST	ResNet-18
年龄	线性回归	FGNet	VGG11
人脸	特征相似度	FaceScrub	MobileFaceNet

(2) 嫌疑模型构建: 完成待保护的目标模型训练后, 本文构建若干嫌疑模型模拟真实验证场景. 首先, 本文采用第 2.2 节的后处理技术对目标模型进行混淆, 生成正例嫌疑模型. 对于微调和重训, 本文分别将K设置为全量参数和最后一层参数, 以训练目标模型, 共生成 4 个模型. 对于权值剪枝[44], 本文在 0.1–0.5 的剪枝比例上生成 5 个模型. 对于滤波器剪枝[45], 本文在 1/16–7/16 的剪枝比例上生成 7 个模型. 这些剪枝比例保证了模型任务表现不会大幅度下降. 对于模型蒸馏[46], 本文在 4 个不同的随机种子上对模型参数进行重新初始化, 并使用目标模型软标签进行知识蒸馏. 关于负例嫌疑模型, 本文在随机种子重新初始化模型参数, 再从头训练了 20 个模型. 对于每个任务, 本文分别都构建了 1 个目标模型, 20 个正例嫌疑模型和 20 个负例嫌疑模型.

(3) 基准方案: 现有的模型指纹均利用输出级别痕迹, 本文选择了目前最先进的基于决策边界的方案 IPGuard[14]和 ConferAE[15], 以及基于特征空间的方案 TAFA[23]和 LTRC[21]作为基准方案. 由于这些方案局限于分类任务版权保护且只支持在黑盒场景下的版权验证, 本文仅在黑盒场景下的皮肤病分类任务上比较.

(4) 实验细节: 每个任务本文均重复实验了 5 次. 指纹样本的数量设置为$N = 100$. 在指纹样本生成阶段, 作为显著图优化目标的预定义模式P设置为左上角高亮, 即$(0,0)$位置为 1; 扰动范围设置为$\xi = 0.1$; 超参数设置为$\lambda = 1$. 在指纹样本验证阶段, 显著阈值设置为$t_{wb} = 0.5$; 对于年龄估计, 置信度降低阈值设置为$t_{bb} = 1$, 其他任务设置为$t_{bb} = 0.05$.

4.2 实验指标

对于模型指纹有效性, 本文使用前作指标[14]进行评价. 首先给定一个表示为$\rho \in (0,1)$的阈值, 本文将第 3.3 节计算的匹配分数 (matching score) 高于ρ的嫌疑模型视为正例, 否则视为负例.

(1) 鲁棒性$R(\rho)$: 鲁棒性测量了真正的正例嫌疑模型被验证为正例的比例 (即: 真阳率).

(2) 独特性$U(\rho)$: 独特性测量了由真正的负例嫌疑模型被验证为负例的比例 (即: 真阴率).

(3) 鲁棒性-独特性曲线下面积 ARUC: ARUC 测量了当阈值 ρ在 (0, 1) 范围内变化时, 鲁棒性和独特性相交区域的面积, 即$\int_0^1 \min\{R(\rho), U(\rho)\}\mathrm{d}\rho$. 更高的 ARUC 值意味着可以同时获得更高的鲁棒性和更高的独特性. 实际计算时, 本文计算 ρ分别取$\{0, 1/L, \cdots, (L-1)/L, 1\}$的$\min\{R(\rho), U(\rho)\}$平均值, 其中 $L = 100$.

对于模型指纹隐蔽性, 本文测量以下指标.

(1) 准确度 Acc: 准确度测量指纹样本在模型预测上准度, 主要评估指纹样本的标签不变性.

(2) 峰值信噪比 PSNR: 峰值信噪比测量指纹样本中有效信息与噪声的比例, 主要评估与原始样本相比, 指纹样本的自然程度.

(3) 检测率 DR: 检测率测量指纹样本被潜在的检测方案判断存在版权验证行为的比例, 从检测的角度评估样本的隐蔽性.

4.3 模型指纹有效性

为了展示基于显著图的模型指纹方案在版权验证上有效性, 本文先在分类任务上进行评估. 首先, 在图 4 中报告了本文方案在白盒场景下的有效性, 白盒验证可以实现几乎完美的版权验证结果, ARUC 为 0.897±0.005, 这表示指纹样本的显著图不仅在目标模型与正例嫌疑模型中保持一致, 而且与负例嫌疑模型高度可分. 此外, 本文在黑盒场景下将基于显著图的模型指纹方案与最先进的基准方案进行了比较. 如图 5 所示, 本文将最优的基准方案 LTRC[21]的性能提高了 140%, 且本文方案 ARUC 结果的标准差相对较低. 同时, 基于显著图的模型指纹方案也是唯一可以实现 100% 鲁棒性 (即真阳率) 和 100% 独特性 (即真阴率) 的方案. 值得注意的是, 基准方案的 ARUC 存在低于原始论文结果

的现象, 这是因为本文所构建正例嫌疑模型对目标模型的参数空间混淆更大.

4.4 模型指纹隐蔽性

据调研, 特别针对模型指纹样本的检测方案还未被提出. 在本文, 本文提出使用一些潜在的检测方案来实现对指纹样本的检测. 在标签一致性方面, 本文采用对抗样本检测方案 feature squeezing[52]来检测指纹样本. 在表 2 中, 本文比较了基于显著图的模型指纹方案和基于决策边界的基准方案在准确度和检测率上的结果. 本文方案构建了标签不变的样本, 保证了 0 的检测率, 而其他方案在标签上与真实标签高度不一致, 且容易被检测出来 (DR>90%) . 此外, 根据图 6(a) 中距离分布显示, 对于本文方案的指纹样本, 检测距离均低于预定义的阈值 1. 而基准方案所生成的指纹样本, 只有极少一部分检测距离低于阈值.

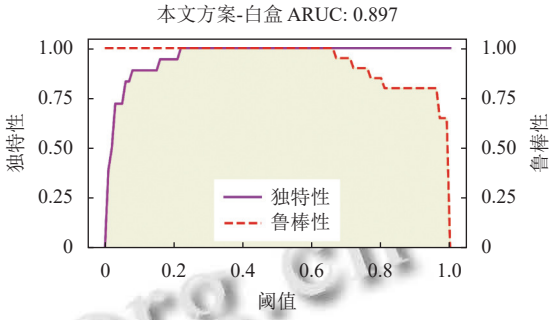
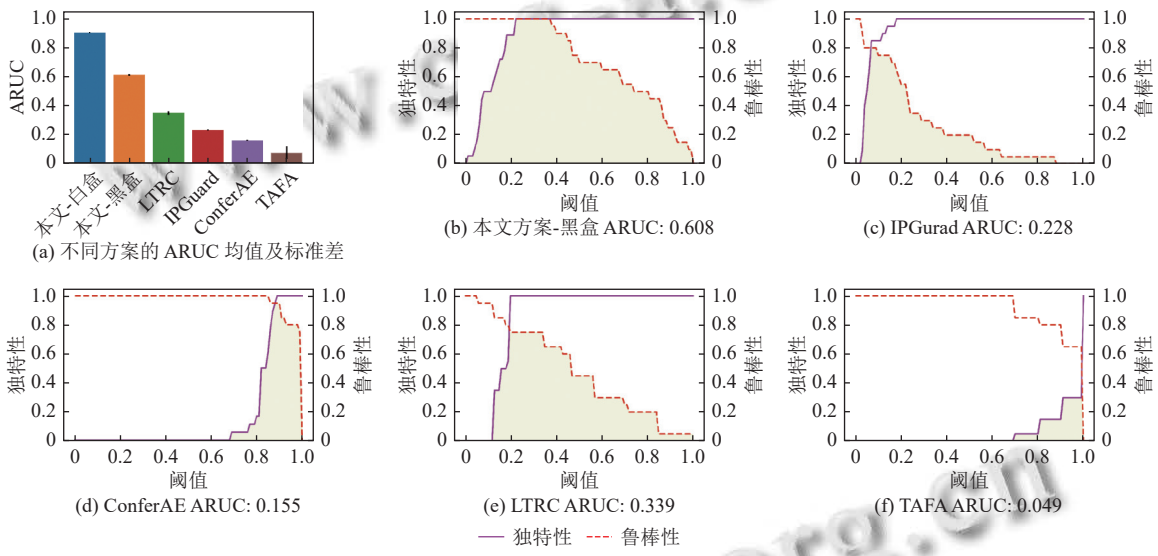


图 4 白盒场景下的鲁棒性-独特性曲线图



(a) 不同方案的 ARUC 均值及标准差

(b) 本文方案-黑盒 ARUC: 0.608

(c) IPGurad ARUC: 0.228

(d) ConferAE ARUC: 0.155

(e) LTRC ARUC: 0.339

(f) TAFA ARUC: 0.049

图 5 模型指纹方案 ARUC 比较以及本文方案和基准方案的鲁棒性-独特性曲线图

表 2 本文方案与基准方案隐蔽性对比

模型指纹方案	准确度 (%)↑	峰值信噪比 (dB)↑	检测率 (%)↓
本文方案	100	25.8253	0
IPGuard	0	19.4149	91
ConferAE	0	39.8914	93
LTRC	100	—	100
TAFA	100	10.2424	100

在自然性方面, 本文定性和定量地将基于显著图与最先进的基于特征空间的基准模型 TAFA[23]和 LTRC[21]在分类任务上进行了比较. 如图 7 所示, 基于显著图的指纹样本与原始训练样本几乎看不出区别, 而 TAFA 生成的样本非常不自然. 而因为 LTRC 直接将随机初始化的噪声作为起点生成指纹样本, 所以其生成的指纹样本也是高度不自然的. 同时, 本文采用异常检测方案 DeepSVDD[53]来检测不自然的指纹样本. 如图 6(b) 所示, 本文的指纹样本全部在正常样本的区间内, 而基于特征空间的基准方案的指纹样本均位于异常区间内. 如表 2 所示, 本文也定量地比较了峰值信噪比和检测率, 本文方案生成的指纹样本在这两个指标下都呈现更高的隐蔽性. 需要注意的是, 本文无法测量 LTRC 构建的指纹样本的峰值信噪比, 因为 LTRC 直接将随机初始化的噪声作为起点以生成指纹样本.

4.5 模型指纹通用性

不止局限于对分类任务的模型实现版权保护, 基于显著图的模型指纹算法还拥有对其他任务提供版权保护的能力. 图 8 绘制了本文方案在线性回归任务和特征相似度任务上的鲁棒性-独特性曲线. 本文的模型指纹方案在这线性任务以及特征相似度任务上均可以实现 100% 鲁棒性 (即真阳率) 和 100% 独特性 (即真阴

率). 由于特征相似度是一个更加困难的任务, 本文认为特征相似度任务上验证结果的轻微下降是可以接受的.

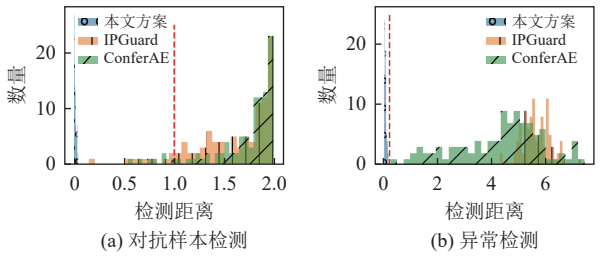


图 6 潜在检测方案的检测距离分布

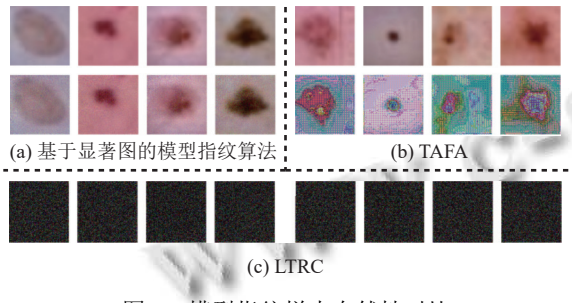


图 7 模型指纹样本自然性对比

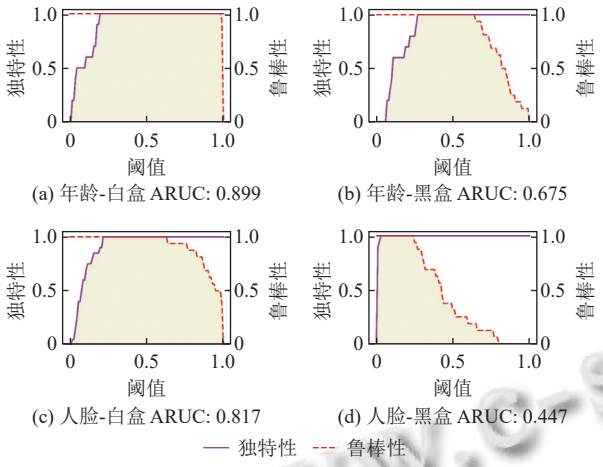


图 8 逻辑回归与特征相似度任务上模型指纹验证结果

4.6 消融实验

最后, 本文在分类任务上分析了超参数对基于显著图的模型指纹方案有效性的影响.

(1) 指纹样本数量的影响: 本文在 10–100 之间, 以 10 为步长采样指纹样本的数量, 以研究指纹样本数量的影响. 如图 9(a) 所示, 在白盒场景和黑盒场景下, ARUC 在 N 取不同取值时保持相对稳定. 即使当 $N=10$时, 依然可以提供有竞争力的结果. 这表明基于显著图的模型指纹方案是一种准确且可靠的模型指纹方案.

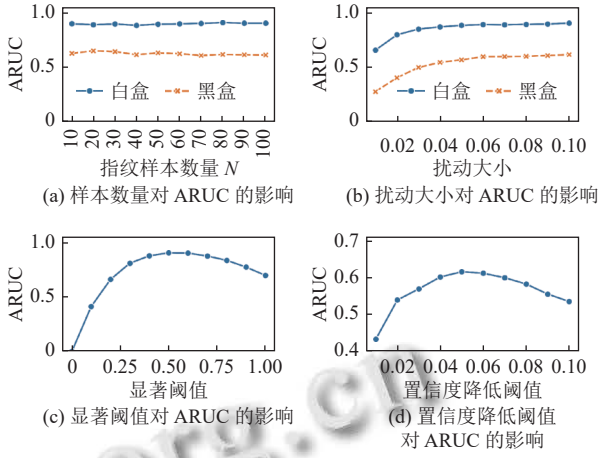


图 9 超参数对于模型指纹有效性的影响

(2) 扰动大小的影响: 扰动大小 ξ 的限制保证了指纹样本的自然性. 本文以 0.01 为步长, 将 ξ 从 0.01 变化到 0.1 来研究其对指纹有效性的影响. 如图 9 (b) 所示, ARUC 随着扰动大小增大逐渐增加, 并在 $\xi \geqslant 0.05$时变得相对稳定. 实验表明, 本文的模型指纹方案构建了高度自然的指纹样本, 以实现有竞争力的表现. 同时, 本文建议将扰动大小 ξ 设置为[0.05, 0.1].

(3) 显著阈值的影响: 本文进一步研究了白盒场景验证中显著阈值 t_{wb} 的影响. 本文将 t_{wb} 从 0 变化到 1, 并且每次增加 0.1. 如图 9 (c) 所示, 当显著阈值 t_{wb} 太小时, 模型指纹的 ARUC 受到严重影响, 因为这会导致负例嫌疑模型的假阳率增加. 同样, 当显著阈值 t_{wb} 太大时, 它会降低模型指纹的 ARUC, 因为这会增加正例嫌疑模型的假阴率. 因此, 通过将 t_{wb} 设置为[0.4,0.6], 本方案能够准确地区分正例嫌疑模型和负例嫌疑模型, 从而获得较高的 ARUC 结果.

(4) 置信度降低阈值的影响: 最后, 本文研究了黑盒场景验证中置信度降低阈值 t_{bb} 的影响. 本文将 t_{bb} 从 0.01 变化到 0.1, 每次增加 0.01. 如图 9(d) 显示, 无论是较小的还是较大的置信度降低阈值 t_{bb} 都会降低模型指纹的 ARUC. 通过将 t_{bb} 设置为[0.04, 0.06], 本文的模型指纹方案可以获得不错的结果, 能够成功地将正例嫌疑模型以及负例嫌疑模型区分开来.

5 结论与展望

本文提出了基于显著图内在痕迹的新型模型指纹方案, 它相较于传统的基于输出痕迹的模型指纹方案, 大大增强了模型指纹方案的隐秘性. 同时, 相对于当前

最先进的基准模型指纹方案, 实验结果验证了方案可以带来巨大的版权验证有效性的提升. 此外, 基于显著图的模型指纹方案还存在任务通用的特点, 能够适用于包括分类、线性回归和特征相似度等多种任务的模型产权保护. 虽然基于显著图的模型指纹方案, 不对目标模型的架构或输出做任何假设, 但考虑到不可能在实验中对所有可能的任务类型进行测试, 本文主要选择了 3 个具有代表性的任务进行评估. 在未来的工作中, 本文将在其他下游任务上 (例如: 目标检测[54]) 验证本方案的有效性, 并将方案扩展到其他模型架构上 (例如: Transformer[55]) .

参考文献

1 Devlin J, Chang MW, Lee K, *et al*. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: Association for Computational Linguistics, 2019. 4171–4186. [doi: 10.18653/v1/N19-1423]

2 Real E, Aggarwal A, Huang YP, *et al*. Regularized evolution for image classifier architecture search. Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Honolulu: AAAI, 2019. 4780–4789. [doi: 10.1609/aaai.v33i01.33014780]

3 Zhou HY, Zhang SH, Peng JQ, *et al*. Informer: Beyond efficient transformer for long sequence time-series forecasting. Proceedings of the 35th AAAI Conference on Artificial Intelligence. AAAI, 2021. 11106–11115. [doi: 10.1609/aaai.v35i12.17325]

4 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778. [doi: 10.1109/cvpr.2016.90]

5 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Proceedings of the 3rd International Conference on Learning Representations. San Diego: ICLR, 2015.

6 Chen S, Liu Y, Gao X, *et al*. MobileFaceNets: Efficient CNNs for accurate real-time face verification on mobile devices. Proceedings of the 13th Chinese Conferenceon Biometric Recognition. Urumqi: Springer, 2018. 428–438. [doi: 10.1007/978-3-319-97909-0_46]

7 Cao YL, Xiao CW, Cyr B, *et al*. Adversarial sensor attack on lidar-based perception in autonomous driving. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. London: ACM, 2019. 2267–2281. [doi: 10.1145/3319535.3339815]

8 Heaton JB, Polson NG, Witte JH. Deep learning for finance: Deep portfolios. Applied Stochastic Models in Business and Industry, 2017, 33(1): 3–12. [doi: 10.1002/asmb.2209]

9 Esteva A, Kuprel B, Novoa RA, *et al*. Dermatologist-level classification of skin cancer with deep neural networks. Nature, 2017, 542(7639): 115–118. [doi: 10.1038/nature21056]

10 Jeong H, Ryu D, Hur J. Neural network stealing via meltdown. Proceedings of the 2021 International Conference on Information Networking. Jeju Island: IEEE, 2021. 36–38. [doi: 10.1109/icoin50884.2021.9333926]

11 Tramèr F, Zhang F, Juels A, *et al*. Stealing machine learning models via prediction APIs. Proceedings of the 25th USENIX Security Symposium. Austin: USENIX Association, 2016. 601–618.

12 Yan MJ, Fletcher CW, Torrellas J. Cache telepathy: Leveraging shared resource attacks to learn DNN architectures. Proceedings of the 29th USENIX Security Symposium. USENIX Association, 2020. 2003–2020.

13 Yu HG, Yang KC, Zhang T, *et al*. CloudLeak: Large-scale deep learning models stealing through adversarial examples. Proceedings of the 27th Annual Network and Distributed System Security Symposium. San Diego: The Internet Society, 2020. [doi: 10.14722/ndss.2020.24178]

14 Cao XY, Jia JY, Gong NZ. IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary. Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security. Hong Kong: ACM, 2021. 14–25. [doi: 10.1145/3433210.3437526]

15 Lukas N, Zhang YX, Kerschbaum F. Deep neural network fingerprinting by conferrable adversarial examples. Proceedings of the 9th International Conference on Learning Representations. OpenReview.net, 2021.

16 Wang S, Chang CH. Fingerprinting deep neural networks—A DeepFool approach. Proceedings of the 2021 IEEE International Symposium on Circuits and Systems. Daegu: IEEE, 2021. 1–5. [doi: 10.1109/iscas51556.2021.9401119]

17 Rouhani BD, Chen HL, Koushanfar F. DeepSigns: A generic watermarking framework for IP protection of deep learning models. arXiv:1804.00750, 2018.

18 Uchida Y, Nagai Y, Sakazawa S, *et al*. Embedding watermarks into deep neural networks. Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval. Bucharest: ACM, 2017. 269–277. [doi: 10.1145/3078971.3078974]

19 Wang JF, Wu HZ, Zhang XP, *et al*. Watermarking in deep neural networks via error back-propagation. Proceedings of THE IS&T International Symposium on Electronic Imaging 2020: Media Watermarking, Security, and Forensics. Burlingame: Society for Imaging Science and Technology, 2020. 22-1–22-9. [doi: 10.2352/issn.2470-1173.2020.4.mwsf-022]

20 Zhang JL, Gu ZS, Jang J, *et al*. Protecting intellectual property of deep neural networks with watermarking. Proceedings of the 2018 Asia Conference on Computer and Communications Security. Incheon: ACM, 2018. 159–172. [doi: 10.1145/3196494.3196550]

21 Wang SY, Wang X, Chen PY, *et al*. Characteristic examples: High-robustness, low-transferability fingerprinting of neural networks. Proceedings of the 30th International Joint Conference on Artificial Intelligence. Montreal: IJCAI.org, 2021. 575–582. [doi: 10.24963/ijcai.2021/80]

22 Wang SY, Zhao P, Wang X, *et al*. Intrinsic examples: Robust fingerprinting of deep neural networks. Proceedings of the 32nd British Machine Vision Conference. BMVA Press, 2021. 46.

23 Pan XD, Zhang M, Lu YF, *et al*. TAFA: A task-agnostic fingerprinting algorithm for neural networks. Proceedings of the 26th European Symposium on Research in Computer Security. Darmstadt: Springer, 2021. 542–562. [doi: 10.1007/978-3-030-88418-5_26]

24 Selvaraju RR, Cogswell M, Das A, *et al*. Grad-CAM: Visual explanations from deep networks via gradient-based localization. Proceedings of the 2017 IEEE International Conference on Computer Vision. Venice: IEEE, 2017. 618–626. [doi: 10.1109/iccv.2017.74]

25 Petsiuk V, Das A, Saenko K. RISE: Randomized input sampling for explanation of black-box models. Proceedings of the 2018 British Machine Vision Conference. Newcastle: BMVA Press, 2018. 151.

26 Zhao JJ, Hu QY, Liu GY, *et al*. AFA: Adversarial fingerprinting authentication for deep neural networks. Computer Communications, 2020, 150: 488–497. [doi: 10.1016/j.comcom.2019.12.016]

27 Moosavi-Dezfooli SM, Fawzi A, Frossard P. DeepFool: A simple and accurate method to fool deep neural networks. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 2574–2582.

28 Li YC, Zhang ZQ, Liu BY, *et al*. ModelDiff: Testing-based DNN similarity comparison for model reuse detection. Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis. ACM, 2021. 139–151. [doi: 10.1145/3460319.3464816]

29 Peng ZR, Li SF, Chen GX, *et al*. Fingerprinting deep neural networks globally via universal adversarial perturbations. Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans: IEEE, 2022. 13420–13429. [doi: 10.1109/cvpr52688.2022.01307]

30 Sun Z, Yang ZW, Huang ZY, *et al*. Interesting near-boundary data: Inferring model ownership for DNNs. Proceedings of the 2023 International Joint Conference on Neural Networks. Gold Coast: IEEE, 2023. 1–8. [doi: 10.1109/ijcnn54540.2023.10191063]

31 Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. Proceedings of the 27th International Conference on Machine Learning. Haifa: Omnipress, 2010. 807–814.

32 Montufar G, Pascanu R, Cho K, *et al*. On the number of linear regions of deep neural networks. Proceedings of the 27th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2014. 2924–2932. [doi: 10.5555/2969033.2969153]

33 Serra T, Tjandraatmadja C, Ramalingam S. Bounding and counting linear regions of deep neural networks. Proceedings of the 35th International Conference on Machine Learning. Stockholm: PMLR, 2018. 4558–4566.

34 Hanin B, Rolnick D. Complexity of linear regions in deep networks. Proceedings of the 36th International Conference on Machine Learning. Long Beach: PMLR, 2019. 2596–2604.

35 Li GL, Xu GW, Qiu H, *et al*. Fingerprinting generative adversarial networks. arXiv:2106.11760, 2021.

36 Yang K, Lai KH. NaturalFinger: Generating natural fingerprint with generative adversarial networks. arXiv: 2305.17868, 2023.

37 Sundararajan M, Taly A, Yan QQ. Axiomatic attribution for deep networks. Proceedings of the 34th International Conference on Machine Learning. Sydney: JMLR.org, 2017. 3319–3328.

38 Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: Visualising image classification

models and saliency maps. Proceedings of the 2nd International Conference on Learning Representations. Banff: ICLR, 2013.

39 Zhou BL, Khosla A, Lapedriza A, *et al*. Learning deep features for discriminative localization. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 2921–2929. [doi: 10.1109/cvpr.2016.319]

40 Subramanya A, Pillai V, Pirsiavash H. Fooling network interpretation in image classification. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019. 2020–2029. [doi: 10.1109/iccv.2019.00211]

41 Zhang XY, Wang NF, Shen H, *et al*. Interpretable deep learning under fire. Proceedings of the 29th USENIX Conference on Security Symposium. USENIX Association, 2020. 94.

42 Ghorbani A, Abid A, Zou J. Interpretation of neural networks is fragile. Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Honolulu: AAAI, 2019. 3681–3688. [doi: 10.1609/aaai.v33i01.33013681]

43 Fang SH, Choromanska A. Backdoor attacks on the DNN interpretation system. Proceedings of the 36th AAAI Conference on Artificial Intelligence. AAAI, 2022. 561–570. [doi: 10.1609/aaai.v36i1.19935]

44 Han S, Pool J, Tran J, *et al*. Learning both weights and connections for efficient neural networks. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2015. 1135–1143.

45 Li H, Kadav A, Durdanovic I, *et al*. Pruning filters for efficient convnets. Proceedings of the 5th International Conference on Learning Representations. Toulon: OpenReview.net, 2017.

46 Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv:1503.02531, 2015.

47 Chen YF, Shen C, Wang C, *et al*. Teacher model fingerprinting attacks against transfer learning. Proceedings of the 31st USENIX Security Symposium, USENIX Security 2022. Boston: USENIX Association, 2022. 3593–3610.

48 Deng J, Dong W, Socher R, *et al*. ImageNet: A large-scale hierarchical image database. Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition. Miami: IEEE, 2009. 248–255. [doi: 10.1109/cvpr.2009.5206848]

49 Yang JC, Shi R, Ni BB. MedMNIST classification decathlon: A lightweight automl benchmark for medical image analysis. Proceedings of the 18th IEEE International Symposium on Biomedical Imaging. Nice: IEEE, 2021. 191–195. [doi: 10.1109/isbi48211.2021.9434062]

50 Ng HW, Winkler S. A data-driven approach to cleaning large face datasets. Proceedings of the 2014 IEEE International Conference on Image Processing. Paris: IEEE, 2014. 343–347. [doi: 10.1109/icip.2014.7025068]

51 Deng JK, Guo J, Xue NN, *et al*. ArcFace: Additive angular margin loss for deep face recognition. Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 4685–4694. [doi: 10.1109/cvpr.2019.00482]

52 Xu WL, Evans D, Qi YJ. Feature squeezing: Detecting adversarial examples in deep neural networks. Proceedings of the 25th Annual Network and Distributed System Security Symposium. San Diego: The Internet Society, 2018. [doi: 10.14722/ndss.2018.23198]

53 Ruff L, Vandermeulen R, Goernitz N, *et al*. Deep one-class classification. Proceedings of the 35th International Conference on Machine Learning. Stockholm: PMLR, 2018. 4393–4402.

54 Jarrett K, Kavukcuoglu K, Ranzato MA, *et al*. What is the best multi-stage architecture for object recognition? Proceedings of the 12th IEEE International Conference on Computer Vision. Kyoto: IEEE, 2009. 2146–2153. [doi: 10.1109/iccv.2009.5459469]

55 Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.

(校对责编: 孙君艳)